

Р. А. Кадиркулов, Г. К. Нурмуханбетова

ИНФОРМАТИКА

Учебник для учащихся 10 класса общеобразовательной школы
естественно-математического направления















*Рекомендовано
Министерством образования и науки
Республики Казахстан*

АЛМАТЫ КІТАП БАСПАСЫ
2019

Алматы кітап баспасы

УДК 373.167.1
ББК 32.973 я 72
К 15

Условные обозначения

- | | |
|---|---|
|  – проблема урока |  – задание |
|  – подумай |  – анализ |
|  – новые знания |  – синтез |
|  – практическая работа |  – оценка |
|  – вопросы |  – работа в группе |
|  – домашнее задание |  – работа в паре |
|  – примени на практике |  – индивидуальная работа |

Кадиркулов Р. А., Нурмуханбетова Г. К.

К 15 ИНФОРМАТИКА. Учебник для учащихся 10 класса общеобразовательной школы естественно-математического направления. – Алматы: АЛМАТЫКІТАП БАСПАСЫ, 2019. – 216 с.

ISBN 978-601-01-3960-2

УДК 373.167.1
ББК 32.973 я 72

ISBN 978-601-01-3960-2

© Кадиркулов Р. А., Нурмуханбетова Г. К., 2019
© ТОО «АЛМАТЫКІТАП БАСПАСЫ», 2019

Дорогие друзья!

Изучение информатики, одного из самых интересных и динамичных предметов 10-го класса, даст вам знания, умения, навыки для эффективного использования современных технологий. Учебная программа по информатике направлена на развитие у школьников информационно-коммуникационных способностей, она научит осмысленно использовать компьютер и другие информационные средства и технологии в своей учебной и будущей профессиональной деятельности. Этот курс даст ученикам представление об алгоритмизации и программировании, веб-проектировании и базах данных.

Информатика в 10 классе требует от учащихся развития межпредметных связей, эта дисциплина тесно переплетена с алгеброй и геометрией. И потому уроки информатики укрепят ваши знания и по этим предметам. Кроме того, в течение этого года школьникам предстоит научиться применять полученную теоретическую информацию для решения практических задач, вы познакомитесь с основным функционалом операционных систем, освоите главные технологии создания, редактирования и оформления информационных объектов. По итогам этого курса десятиклассники должны будут создавать простейшие модели объектов и процессов в виде презентаций и таблиц, научатся использовать информационные ресурсы в целях самообразования.

Важная задача, которую помогает решить информатика, – предоставление учащимся возможности внести свой вклад в анализ, моделирование, программирование и разработку прикладных программ.

Кроме того, этот предмет учит школьников этичному отношению к информационным технологиям, информационной безопасности, защите информации. На уроках информатики учащиеся узнают о важности информационных технологий для развития бизнеса, социального, научного, культурного, инженерного развития общества.

Перед вами не обычный учебник. Издательство «Алматыкітап», шагая в ногу со временем, дополнительно разработало еще и его электронную версию, которая размещена на специальной учебной платформе orq.kz в сети Интернет. Работать с электронным учебником можно на любых электронных устройствах – ноутбуках, стационарных компьютерах, планшетах и даже смартфонах. Мы хотим, чтобы учебный процесс стал для вас не только познавательным, но и более комфортным.

Авторы будут постоянно обновлять и совершенствовать электронный учебник, дополняя его новыми интересными материалами. Изучение тем сопровождается интересными примерами и видеоуроками, графическими схемами и увлекательными заданиями, которые имеют функцию автоматической проверки. Теперь учебник станет вашим настоящим другом и помощником. А процесс выполнения практических заданий и проектов уже никогда не будет скучным и монотонным. Новые технологии прочно войдут в вашу учебную жизнь.

Желаем успехов!

1

Организация компьютерных сетей



Какова структура компьютерных сетей? Из чего состоят компьютерные сети?



- ✓ Каково влияние сетевых технологий на социальные изменения в обществе в настоящее время?
- ✓ Какие виды сетей вам известны?



Новые знания

В предыдущих классах вы познакомились с понятием сети, видами сетей, сетевым этикетом и угрозами, встречающимися при работе в сети. Теперь, углубляя эти знания, перейдем к изучению организации компьютерных сетей и рассмотрим, как они развиваются в настоящее время.

Передача данных через компьютерные сети является ключевым моментом для коммуникации в будущем. Сеть обеспечивает высокоскоростную передачу данных, отправление\получение сообщений на рабочем месте, обмен мгновенными сообщениями из любой точки земного шара, обмен информацией между разными компьютерами и различными типами программного обеспечения и т. д.

Компьютерные сети (Computer NetWork, net – сеть, work – работа) – совокупность компьютеров, соединенных с помощью каналов связи и средств коммутации в единую систему для обмена данными и совместного использования ресурсов.

Сегодня в мире насчитывается более 130 миллионов компьютеров, около 80% которых интегрированы в различные информационно-вычислительные сети.

Развитие и модернизация компьютерных сетей в значительной степени способствует совместному использованию данных. Персональный компьютер – эффективный инструмент для создания документов, подготовки таблиц, графиков и других видов информации. При отсутствии сети, для использования каких-либо данных другими пользователями, приходится информацию распечатывать или копировать каждый документ на диск, флеш-память, карту памяти и т.п. Кроме того, возникают сложности при повторном редактировании документов, когда каждый пользователь вносит свои изменения. Такая схема называется работой в автономной среде. Если же пользователь сможет подключить свой компьютер к другим компьютерам, то он сможет работать и с документами на других компьютерах и воспользоваться общим принтером. Объединение компьютеров и других устройств для совместной работы называется сетью. Объединение нескольких компьютерных сетей вместе для формирования более крупных сетей называется межсетевым взаимодействием.

Сеть представляет собой совокупность взаимосвязанных компьютеров для эффективного использования ресурсов (диски, файлы, принтеры, устройства связи).

На компьютерах, подсоединенных к сетям, можно совместно использовать:

- базы данных;
- информационные базы;
- принтеры;
- модемы;
- другие устройства.

Когда в нашей стране стали развиваться компьютерные сети?

19 сентября 1994-го года считается днем рождения Интернета в Казахстане. В этот день был официально зарегистрирован национальный домен верхнего уровня .kz. А широкому распространению сети Интернет в нашей стране способствовала общественная организация «Национальная телекоммуникационная ассоциация». Ее деятельность направлена на расширение телекоммуникационного рынка путем создания деловых связей предприятий, развитие телекоммуникационных сетей, использование информационных технологий, предоставления телекоммуникационных услуг. Основная задача Ассоциации заключается в том, что она оказывает практическую помощь в организации связей документов между отечественными и зарубежными операторами. Развитие альтернативных операторов связи способствовало появлению новых для РК видов связи, оборудования и технологий. К ним относятся внедрение мобильной связи, расширение сетей передачи данных и многое другое. В связи с этим развивается взаимодействие между поставщиками услуг и пользователями. Наряду с расширяющимися возможностями World Wide Web, появились множество интересных и специальных сайтов. В последнее время добавились множество различных справочников, каталогов и баз данных на всевозможные и полезные темы.

Классификация компьютерных сетей (рис. 1), как любых больших и сложных систем классифицируют по различным признакам: размер (территориальный охват) сети; принадлежность; назначение; область применения.

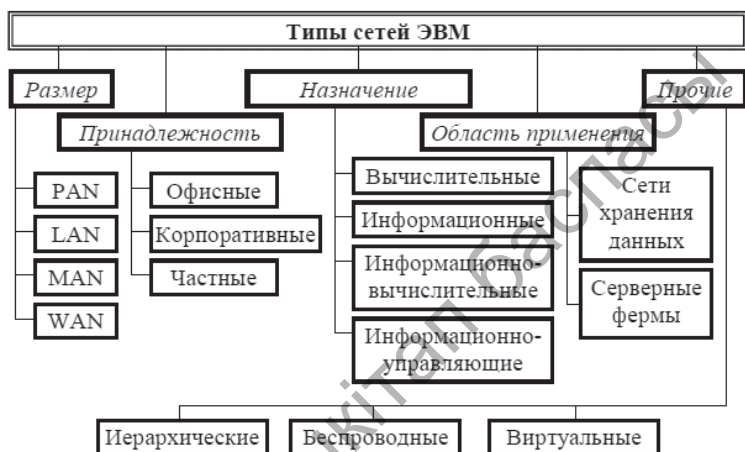


Рисунок 1. Классификация компьютерных сетей

По размеру (территориальному охвату) сети подразделяются на:

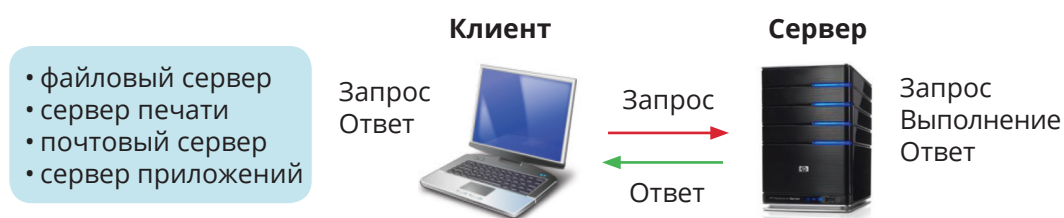
1. Персональные сети (PAN – Personal Area Network)
2. Локальные сети (LAN – Local Area Network)
3. Городская вычислительная сеть (MAN – Metropolitan Area Network)
4. Глобальная вычислительная сеть (WAN – Wide Area Network)

Клиент-сервер

В результате развития персональных компьютеров централизованную модель заменила модель клиент-сервер, предоставляющая при той же производительности возможности сетевой обработки данных.

Клиент – компьютер, который использует серверные ресурсы.

Сервер – компьютер, обеспечивающий общий доступ к ресурсам.



Персональные сети объединяют все персональные электронные устройства пользователя (телефоны, карманные персональные компьютеры, смартфоны, ноутбуки, беспроводные гарнитуры и т. п.).

Локальные сети – два и более компьютеров, расположенных в одном помещении, здании, районе города и объединенных в единую действующую совместно сеть. Локальные сети создаются для удобства пользователей в работе с общими принтерами, сканерами и другой периферийной техникой.



1. Какие есть возможности у компьютера, подсоединенного к сети?
2. Что такое сервер?
3. Как осуществляется обмен информацией?
4. В чем разница между локальной и персональной сетями?
5. Какие типы сетей вы знаете?
6. Что такое «клиент – сервер»?



Анализ



Проанализируйте развитие социальных сетей в нашей стране. Приведите примеры.



Оценка



1. Оцените преимущества компьютерной сети.
2. Оцените негативные стороны компьютерной сети.
3. Представьте 2-3 аргументированных примера оценки как преимуществ, так и недостатков использования компьютерных сетей.



Напишите эссе на тему «Технологии будущего: развитие организации сетей».

Сети. Компоненты сети



Какие компоненты вы знаете для организации работы сети?



- ✓ Какова сегодня роль сети в развитии компьютерных технологий?
- ✓ Какие виды социальных сетей ты используешь??
- ✓ В какой сети умеешь работать?



Новые знания

В связи с появлением территориальных каналов связи, не уступающих по качеству кабельным системам локальных сетей, различие между локальными и глобальными сетями уменьшается. Создан широкий спектр удобных сервисов, обеспечивающих доступ к ресурсам глобальной сети. Примером этого является широко известный широкополосный интернет-сервис. Понятие «широкополосный Интернет» напрямую связано со скоростью передачи данных и поэтому его часто еще называют высокоскоростным. Широкополосный доступ – это общее название технологий, обеспечивающих постоянное подключение к Интернету. Еще до недавнего времени одним из основных способов подключения к Интернету был коммутируемый доступ, который осуществлялся по телефонной линии, полностью занимая ее на время соединения. Широкополосный Интернет обеспечивает скорость обмена данными во много раз большую, чем при коммутируемом доступе и при этом не «захватывает» телефонную линию.

В последнее время глобальные сети также претерпели изменения. Вместо соединяющего компьютеры пассивного кабеля в них в большом количестве появилось разнообразное *коммуникационное оборудование* – коммутаторы, маршрутизаторы, шлюзы. Благодаря такому оборудованию есть возможность построения больших сетей, насчитывающих тысячи компьютеров и имеющих сложную структуру.

Условия работы сетей

Сеть должна отвечать трем основным требованиям:

1. Доступность соединений (**сетевых интерфейсов**) для связи между компьютерами и распределяющими организациями (провайдерами). Обычно подсоединение к сети осуществляется специальной съемной платой, называемой сетевой интерфейсной платой;
2. Наличие правил (**протоколов**), по которым компьютеры взаимодействуют друг с другом;
3. Присутствие списка действий (услуг) одного компьютера по отношению к другому.

Основные программно-аппаратные компоненты сети:

1. Компьютеры
2. Сетевое оборудование
3. Операционные системы
4. Сетевые приложения

К сетевому оборудованию относятся следующие устройства:

1. Сетевые узлы
2. Маршрутизаторы (router)
3. Модемы
4. Коммутаторы (мосты-bridge, switch)

Сетевые узлы

Сетевой узел (англ. *node*) – устройство, соединенное с другими устройствами как часть компьютерной сети (рис. 1). Узлами могут быть компьютеры, мобильные телефоны, карманные компьютеры, а также специальные сетевые устройства, такие как маршрутизатор, коммутатор или концентратор. Здесь термин «узел» подразумевает, что все соединительные провода устройств подключены к узлам, так же как в компьютерной сети.



Рисунок 1. Компьютерная сеть

Маршрутизатор

Маршрутизатор или роутер, рутер (англ. *router*) – устройство, которое пересылает пакеты между различными сегментами сети на основе правил и таблиц маршрутизации. Маршрутизатор (рис. 2) может связывать разнородные сети различных архитектур. В основном они используются для интеграции различных типов сетей, которые не соответствуют архитектуре и протоколам.



Рисунок 2. Роутер

Для принятия решений о пересылке пакетов используется информация о топологии сети и определенные правила, заданные администратором.

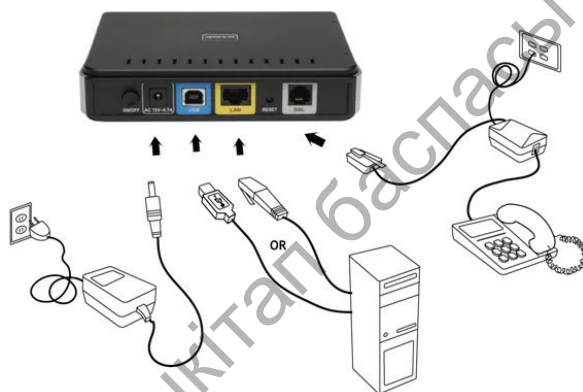


Рисунок 3. Организация сети с помощью роутера

Маршрутизатор (роутер) – это специальная подстанция, принимающая сигналы из сети и передающая эти сигналы на конкретные устройства. По сути, роутеры являются связующим звеном между пользовательскими устройствами и сервером, раздающим подключения к Интернету.

Роутеры могут предоставлять как проводное подключение по специальным кабелям, предназначенным для этого, так и беспроводное, создавая сеть Wi-Fi.

Для чего нужен маршрутизатор и какова его роль в локальной сети?

Чтобы понять работу маршрутизатора, проанализируем его участие в домашней сети. Например, у вас дома есть персональный компьютер, ноутбук, принтер, планшет, а также вместе с этими устройствами Smart телевизор и одна точка кабеля LAN, которая позволяет вам получить доступ к сети Интернет.

Вопрос: «Как можно подсоединиться к Интернету всеми имеющимися дома устройствами одновременно?».

И именно в этом случае нам нужен маршрутизатор. Если поставщик интернет-услуг использует телефонные линии, маршрутизатор подключается к Интернету через модем (рис. 3). Домашние устройства подключаются к маршрутизатору по локальной сети и через беспроводную сеть Wi-Fi. Маршрутизатор связывает гетерогенные (неоднородные) сегменты сети и направляет данные к адресам назначения на основе таблицы маршрутизации.

Модем

Модем используется для связи с удаленными компьютерами (рис. 4).

Модем – это устройство, которое выполняет модуляцию (преобразование цифровых сигналов в аналоговые сигналы) и демодуляцию (преобразование аналоговых сигналов в цифровые) сигналов.



Рисунок 4. Модем

Модемы бывают внешние и внутренние. Внутренний модем выполняется в виде платы расширения, которая вставляется в один из свободных слотов материнской платы компьютера. Внешний модем – это отдельное устройство в собственном корпусе, которое соединяется кабелем с последовательным портом компьютера.

Основной характеристикой модема являются скорость передачи информации, которая измеряется в битах в секунду.

Коммутатор (switch)

Принцип совместного использования данных позволяет создавать качественную вычислительную сеть. Из-за простоты используемого протокола стоимость установки сетей



Рисунок 5. Коммутатор

недорогая. Пропускная способность достигает от 100 Мбит/с до 1000 Мбит/с. Из-за простых топологий такие сети легко расширяются. Но расширение этих сетей сказывается на качестве. При увеличении количества подключающихся компьютеров пропускная способность сети пропорционально уменьшается. Поскольку функциональность сети снижается, количество подключаемых узлов также ограничено. Поэтому для создания большой вычислительной сети, объединяющей сотни и тысячи узлов, используются коммутаторы (мосты).

Коммутатор (switch) – устройство для соединения нескольких сегментов (рис. 5). В этом случае он поддерживает несколько процессов обмена данными одновременно для каждой пары станций разных сегментов. Технология коммутатора основана на принципе моста. Особенностью этого устройства является то, что он отправляет данные только получателю. Сеть с коммутатором характеризуется производительностью и безопасностью.

Важным отличием между маршрутизатором и коммутатором является скорость работы. Коммутатор очень быстро обрабатывает данные, так как не проверяет и не анализирует, в отличие от маршрутизатора, каждый пакет данных. Но для больших сетевых узлов лучше подходят маршрутизаторы, потому что у коммутатора размер таблицы маршрутизации мал и его трудно использовать в больших сетях.

Коммутационное устройство, в которой хранится таблица переключателей, называется **ассоциативной памятью**. Это означает, что компьютерный узел соответствует определенному порту. Когда коммутатор подключен, таблица пустая. Устройство в этом случае работает только в режиме чтения. Например, для передачи любых данных, передаваемых в порты. Информация, полученная в ходе этого процесса, анализируется, и адрес отправителя включается в таблицу.

Коммутаторы в отличие от маршрутизаторов не подключаются автоматически к Интернету.



Анализ



Проанализируйте сходства и различия между маршрутизатором и коммутатором.



1. Какие устройства необходимы для создания компьютерной сети?
2. Как вы оцениваете роль компьютерной сети?
3. Какие функции у модема?
4. Что такое узел сети?
5. Какова роль маршрутизатора в сети?
6. Какова функция коммутатора?



Напишите эссе на тему «Роль сети Интернет в нашей жизни». При написании эссе вы должны раскрыть вопрос: «Что мы приобрели и что потеряли в результате вхождения сети Интернет в нашу жизнь?».

IP-адреса



Что такое IP-адрес? Как записывается IP-адрес и в чем заключается его функция?



- ✓ Можно ли отследить компьютер, с которого вышли в сеть Интернет?
- ✓ Можно ли узнать о компьютере, работающем в сети Интернет из какой он страны и из какого региона?



Новые знания

На специальных серверах с доступом в сеть Интернет размещены новости, бизнес-данные, файлы (программы, архивы, справочные документы). Новая информация добавляется на серверы ежедневно. Типы интернет-серверов бывают различными. Информация, передаваемая одним из них, переводится на стандартный язык, который используется в сети Интернет, чтобы помочь другим понять и прочитать отправляемые данные. Правила обмена задаются специальным программным обеспечением – набором сетевых протоколов **TCP/IP** (*Transmission Control Protocol/Internet Protocol* – Протокол управления передачей/Протокол Интернета). Другими словами, с помощью набора протоколов TCP/IP осуществляется взаимодействие между компьютерами в глобальной сети Интернет. **Протокол** – это совокупность правил, используемых при передаче данных.

Виды протоколов TCP/IP

TCP (Transmission Control Protocol) обеспечивает гарантированную передачу данных по сети.

IP (Internet Protocol) – это адресный протокол, который отвечает за адресацию всей сети.

Служба транспортного протокола TCP дробит информацию на несколько частей (пакеты), присваивает каждой части номер, по которому данные впоследствии соединяются воедино, добавляет к ней «служебную» информацию (название темы, адрес сервера и адрес файла) и укладывает все это в отдельный «IP-конверт». Далее этот «конверт» отправляется по сети Интернет на главный сервер. Каждый участник сети имеет уникальный адрес, называемый IP-адресом.

IP-адрес – индивидуальный неповторимый адрес (имя) компьютера (или другого устройства), подсоединенного к сети Интернет или локальной сети.

Протоколы прикладного уровня

1. **HTTP** (Hyper Text Transfer Protocol) – передача гипертекста.
2. **FTP** (File Transfer Protocol) – передача файла.
3. **SMTP** (Simple Mail Transfer Protocol) – передача сообщений электронной почты.
4. **POP3** (Post Office Protocol Version 3) – получение электронной почты с почтовых серверов.

5. **IMAP** (Internet Message Access Protocol) – чтение сообщения электронной почты, без загрузки на компьютер. Все сообщения хранятся на сервере.

IP-адрес. 32-битное число – уникальный номер, однозначно идентифицирующий компьютер в сети Интернет. Не могут быть два идентичных IP-адреса компьютеров, но может быть несколько IP-адресов на одном и том же узле (рис. 1 а, б).

Количество всех IP-адресов:
 $N = 2^{32} = 4\,294\,967\,296$

IP-адрес состоит из двух частей: номера сети и номера компьютера (хоста) в этой сети (рис. 2). Имеется несколько классов IP-адресов. В классе А выделены 8 бит на номер сети и 24 бита на имя хоста. В классе В выделены 16 бит на номер сети и 16 бита на имя хоста. В классе С выделены 24 бита на номер сети и 8 бита на имя хоста. Есть также класс D для многоадресной рассылки и зарезервированный класс E, в настоящее время не используемый (таблица 1).

Таблица 1

| Класс | Диапазон |
|-------|-----------------------------|
| A | 0.0.0.0 – 127.255.255.255 |
| B | 128.0.0.0 – 191.255.255.255 |
| C | 192.0.0.0 – 223.255.255.255 |
| D | 224.0.0.0 – 239.255.255.255 |
| E | 240.0.0.0 – 247.255.255.255 |

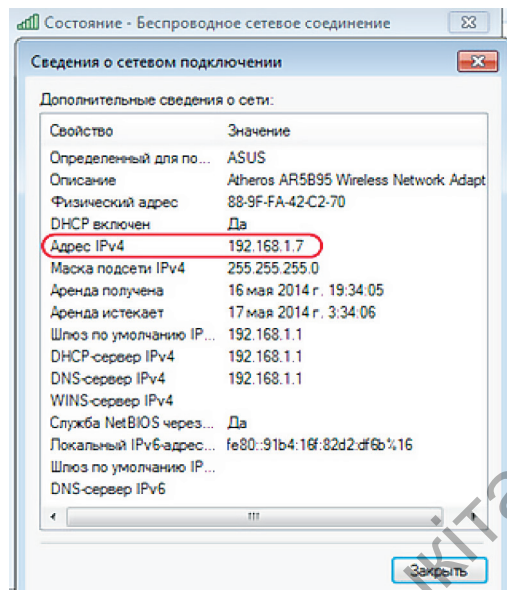


Рисунок 1 а. IP-адрес

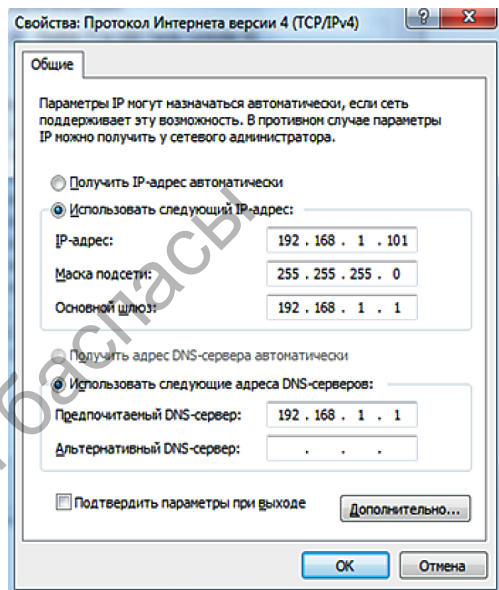


Рисунок 1 б. Использование IP-адреса

Internet Protocol (IP, «межсетевой протокол») – маршрутизируемый протокол сетевого уровня стека TCP/IP. Именно IP стал тем протоколом, который объединил отдельные компьютерные сети во в всемирную сеть Интернет. Созданная с помощью служб Интернета корпоративная IP-сеть для внутреннего пользования называется **Intranet**.

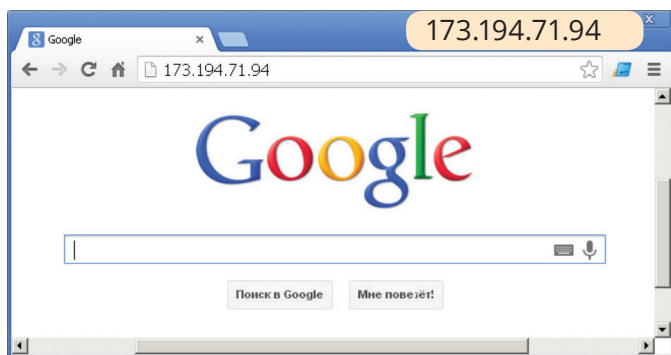
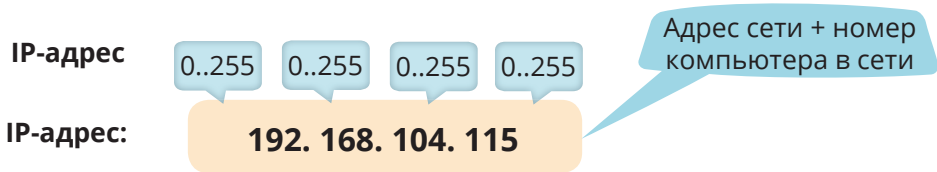


Рисунок 2. IP-адрес Google

IP-адресация

IP-адрес, записанный в 10-значной системе счисления, состоит из 4 чисел от 0 до 255.

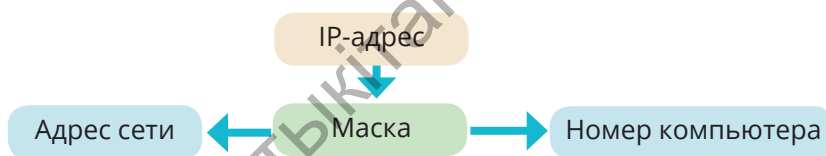
Например, из имеющихся частей (фрагментов), представленных в таблице, восстановим полный IP-адрес.

| | | | |
|-----|-------|----|------|
| .75 | 5.106 | 21 | 1.24 |
| А | Г | Б | В |

Ответ: БВГА **211.245.106. 75**

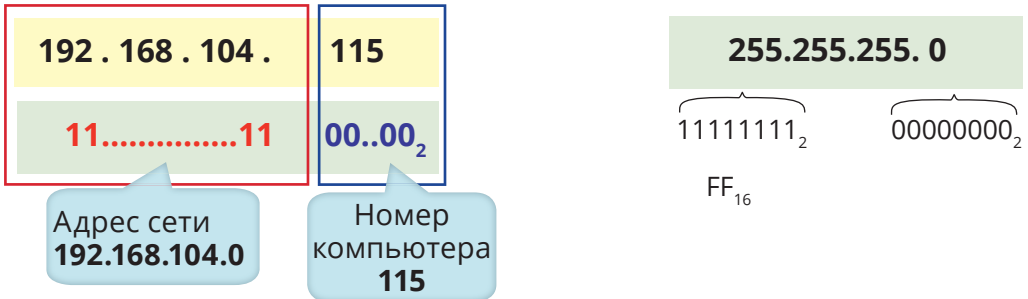
IP-адреса и маски

Маска – это шаблон, который позволяет отделить адрес сети от номера компьютера в этой сети.



Маски для разделения IP-адресов

В маске всегда сначала идут только единицы, а потом только нули:
11 ... 1100 ... 00



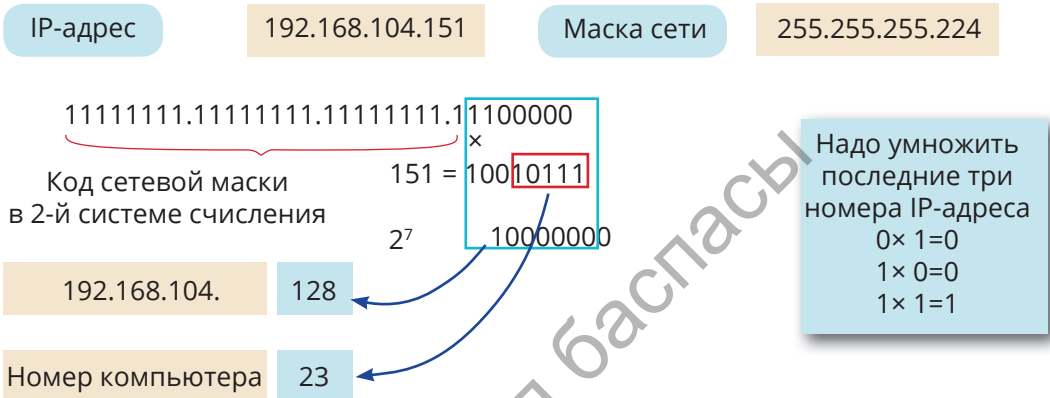
Как можно определить количество компьютеров в сети?



Формула для определения количества компьютеров в сети: $K = 2^n - 2$
Тогда адрес компьютера: $2^7 = 128$, $128 - 2 = 126$.

Ответ: **126**

Способы определения номера компьютера в сети



Примечание: Последние номера IP-адреса не могут принимать значения 0, 63, 64, 127, 128, 191, 192 и 255. То есть адреса, которые состоят только из 1 или 0, не используются для адресации сетевых узлов. Например, $63_{10} = 111111_2$



Применение

Задание 1.

Фрагменты IP-адреса компьютера представлены в столбцах от А до Д. Восстановите IP-адрес. В ответе запишите IP-адрес последовательностью чисел, которые представлены в фрагментах.

а)

| | | | |
|--------------|-----------|-------------|------------|
| 3.212 | 21 | 2.12 | .42 |
| А | Б | В | Г |

б)

| | | | |
|------------|-------------|-----------|--------------|
| .64 | 2.16 | 16 | 8.132 |
| А | Б | В | Г |

в)

| | | | |
|--------------|-------------|------------|-----------|
| 2.162 | 4.18 | .61 | 20 |
| А | Б | В | Г |

г)

| | | | | |
|------------|-----------|-----------|-------------|-----------|
| .65 | 10 | 39 | 4.28 | .2 |
| А | Б | В | Г | Д |

Задание 2.

Айгуль сообщила другу свой десятизначный IP-адрес по телефону. Друг записал следующие цифры **3717954234**. Запишите этот IP-адрес в соответствии с правилами, воспользуйтесь таблицей 1.

Задание 3: Какие из этих последовательностей могут быть маской:

- | | |
|--------------------|--------------------|
| 1) 255.255.255.122 | 6) 255.255.255.128 |
| 2) 255.255.128.255 | 7) 255.255.248.0 |
| 3) 255.255.156.0 | 8) 255.255.128.128 |
| 4) 255.255.255.192 | 9) 255.255.255.102 |
| 5) 255.255.224.192 | 10) 255.255.248.0 |

Задание 4.

Рассмотрите IP-адрес компьютера и определите его номер во внутренней сети. IP-адрес: 231.133.13.133 и маска внутренней сети: 255.255.254.0.

Задание 5.

Рассмотрите IP-адрес компьютера и определите его номер во внутренней сети. IP-адрес: 135.174.234.56 и маска внутренней сети: 255.255.248.0.



1. Какие существуют типы протоколов?
2. Какова функция протокола TCP/IP?
3. Что такое IP-адрес?
4. Какими цифрами не могут быть последние номера в IP-адресе?
5. Как определить номер компьютера в сети?



Рассмотрите IP-адрес компьютера и определите его номер во внутренней сети. IP-адрес: 251.216.108.239 и маска внутренней сети: 255.255.224.0.

Функции системы доменных имен (DNS)



Какова структура системы доменных имен (DNS)?



- ✓ Знакомы ли вы с понятием домена?
- ✓ Какие есть удобные способы запоминания числовых адресов компьютеров сети?



Новые знания

Первоначально в сети Интернет использовались IP-адреса, но когда компьютеров в сети стало больше чем 1000, был принят метод связи имен и IP-адресов, который называется **сервер доменных имен (Domain Name Server, DNS)**. DNS выдает информацию компьютерам, как найти друг друга в сети Интернет. Система DNS используется для преобразования IP-адресов в соответствующие им символьные имена узлов и обратно.

Служба доменных имен DNS является стандартной службой имен Интернета и TCP/IP. Служба DNS позволяет компьютерам-клиентам регистрировать и сопоставлять доменные имена DNS. Эти имена используются для поиска и доступа к ресурсам, предлагаемым компьютерам в вашей сети или в других сетях, таких как Интернет.

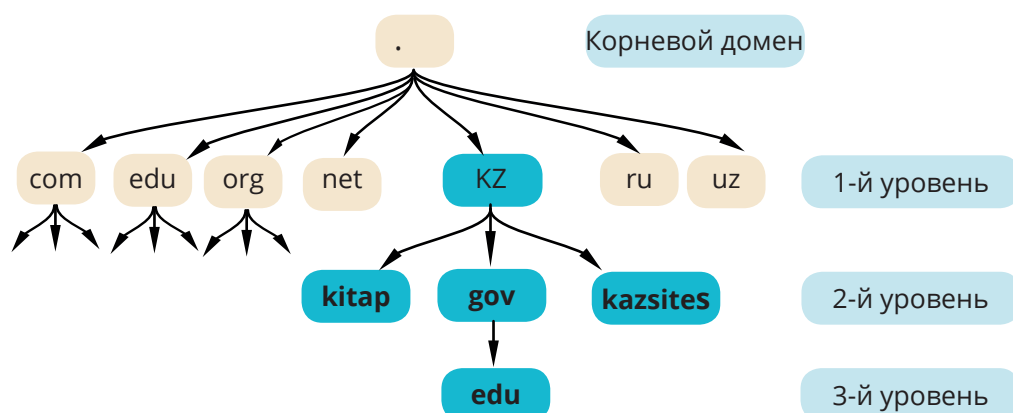
Домен (доменное имя) – это адрес сайта в сети Интернет, состоящий из набора символов и цифр.

Каждое доменное имя в Интернете является уникальным. Максимальная длина доменного имени (между точками) – 63 символа. Общая длина имени, включая поддомены, разделительные точки и имя зоны домена, может составлять до 255 знаков. Вы можете получить доступ к сайту, введя его доменное имя в строку поиска.

Не каждый компьютер, подключенный к сети Интернет, имеет доменное имя, но обязательно имеет IP-адрес. Доменные имена в основном имеют Интернет-серверы. Например, вместо IP-адреса 173.194.71.94 гораздо удобнее использовать www.google.kz. В Интернете доменная система имен использует принцип последовательных уточнений также как и в обычных почтовых адресах – страна, город, улица и дом, в который следует доставить письмо.

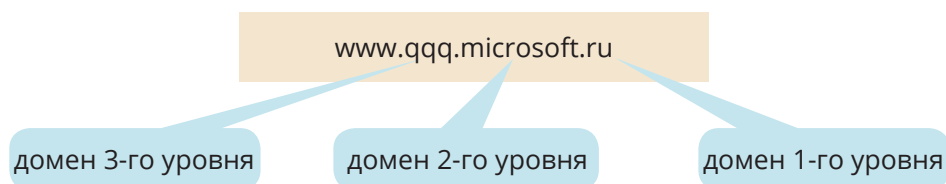
Домен верхнего (первого) уровня имеет самый высокий уровень в иерархии системы доменных имен после корневого домена и является начальной точкой отсчета (справа налево), с которой начинается доменное имя в Интернете. Домен нижнего уровня располагается левее (схема 1). Принадлежность домена к Казахстану обозначается «.kz». Но из-за большого числа

Схема 1



пользователей Интернета в Казахстане учреждение может определяться на следующем доменном уровне. Имя отдельного компьютера или сети каждая компания выбирает для себя самостоятельно, а затем регистрирует его в той организации Интернет, которая обеспечивает подключение.

Схема 2



Это имя в пределах домена верхнего уровня должно быть уникальным. В имени может быть любое число доменов, но чаще всего используются имена с количеством доменов от трех до пяти (схема 1).

Система доменной адресации гарантирует, что в Интернете нет другого компьютера с таким именем. Для доменов низкого уровня может быть использован любой адрес, а для доменов высокого уровня существуют специальные соглашения.

Система Интернет – адресов содержит домены, определяющие географическую зону. Они содержат только две буквы. Например: Казахстан – .kz, Украина – .ua, Франция – .fr, Канада – .ca, США – .us, Россия – .ru (схема 2). Также есть домены, определяющие тематику. Например: образовательные организации – edu, правительственные учреждения – gov, коммерческие организации – com. В последнее время добавились новые зоны, например biz, info, in, cn и др. (таблица 1).

Домен первого уровня

| Тип организации | Государство |
|--------------------------------------|-----------------------|
| .com коммерческие организации | .kz Казахстан |
| .edu образовательные | .ru Россия |
| .gov правительственные | .by Белоруссия |
| .net сетевые организации | .it Италия |
| .org разные организации | .jp Япония |
| .info сайт информации | .cn Китай |
| .biz бизнес | .ca Канада |
| .name отдельные данные | .de Германия |
| .museum музей | .ee Эстония |

Домен 2-го уровня – платный (рис. 1). Домен 3-го уровня в основном бесплатный. На рисунке 1 показан список доменных имен и их текущая стоимость предлагаемая казахстанской компанией. В зависимости от ситуации на рынке эти домены могут подорожать или подешеветь. Выбрав из этого списка нужный домен, вы можете создать свой сайт.

| | |
|--|--|
| <input type="checkbox"/> .kz 3 360 KZT | <input type="checkbox"/> .com 2 208 KZT |
| <input type="checkbox"/> .net 2 208 KZT | <input type="checkbox"/> .ru 2 500 KZT |
| <input type="checkbox"/> .com.kz 3 360 KZT | <input type="checkbox"/> .org.kz 3 360 KZT |
| <input type="checkbox"/> .org 2 208 KZT | <input type="checkbox"/> .info 2 208 KZT |
| <input type="checkbox"/> .biz 1 776 KZT | <input type="checkbox"/> .cc 5 920 KZT |
| <input type="checkbox"/> .us 1 776 KZT | <input type="checkbox"/> .su 2 500 KZT |
| <input type="checkbox"/> .ws 2 960 KZT | <input type="checkbox"/> .in 1 480 KZT |
| <input type="checkbox"/> .name 2 208 KZT | <input type="checkbox"/> .eu 3 700 KZT |
| <input type="checkbox"/> .mobi 2 220 KZT | <input type="checkbox"/> .tv 7 400 KZT |
| <input type="checkbox"/> .bz 5 920 KZT | |

Рисунок 1. Домены второго уровня

Регистрация домена в одной из компаний, дающих права собственности на домен, осуществляется вводом необходимой информации и годовой оплатой за использование домена. Если у вас есть разработанный сайт, то можете разместить его в сети Интернет.



Анализ



Сравните IP-адрес и доменный адрес. Покажите сходства и различия на диаграмме Венна.



Синтез



Как правильно выбрать доменное имя? Представьте, что все члены вашей семьи решили открыть персональные web-сайты в сети Интернет. Выберите им доменные имена. Обоснуйте выбор.



Оценка



Оцените важность системы доменов в сети.



1. Могут ли все компьютеры, подключенные к сети, иметь IP-адрес и доменный адрес? Обоснуйте.
2. Какова структура системы доменных имен?
3. Сайты с доменными именами какого уровня поисковиками в сети Интернет «ценятся» выше?



Мини-исследование

| | Темы исследования | Вопросы и направления исследования |
|---|---|--|
| 1 | Что такое Электронное Правительство? | Какое у него доменное имя? Почему Электронное Правительство так называется? Какое отношение сайт имеет к работе правительства? |
| 2 | Определите 6 сайтов с наибольшим числом посетителей в нашей стране и в мире. Почему у этих сайтов высокие рейтинги? | Проанализируйте сайты нашей страны и других стран. В каком направлении созданы известные сайты? Сайты какого направления популярны? Опираясь на свой опыт, используйте Интернет, опросите членов своей семьи, друзей, одноклассников и определите самые популярные 3 сайта. |



Какие меры предосторожности следует соблюдать при работе в сети?



- ✓ Как вы думаете, какие существуют угрозы при работе в сети?
- ✓ Какую информацию нельзя публиковать в сети Интернет?
- ✓ Каковы риски использования социальных сетей?



Новые знания

Информация и информационная безопасность

Информация (от лат. informatio – разъяснение) – сведения, передаваемые людьми устным, письменным или другим способом (с помощью условных сигналов, технических средств и т. д.). Информация – общенаучное понятие, включающее обмен сведениями между людьми, человеком и автоматом, автоматом и автоматом; обмен сигналами в животном и растительном мире; передачу признаков от организма к организму.

Под защитой информации в процессе сбора, обработки и передачи информации понимают меры безопасности, обеспечивающие ее конфиденциальность и полноту.

Информационная безопасность – это состояние защиты государственных информационных ресурсов, а также прав отдельных лиц и ответственности в области информации.

Защита информации – комплекс мер, направленных на обеспечение информационной безопасности. На практике защита информации означает целостность используемой информации и ресурсов, доступность и конфиденциальность. Таким образом, защита информации представляет собой набор мер для предотвращения утечки информации, ее кражи, потери, несанкционированного уничтожения, изменения, несанкционированного дублирования и модификации. Она включает организационные, программные и технические методы и инструменты для соблюдения мер безопасности.

Конституция Республики Казахстан гарантирует возможность свободного получения необходимой для каждого лица информации, а также возможность ее публикации в любой законной форме.

Тайны, которые охраняются законами в стране:

- Специальная информация. Тайны семейной и личной жизни;
- Тайна письма, сообщений посредством телефона, телеграфа и др. видов связи;
- Тайна усыновления;
- Медицинская тайна;
- Коммерческая и служебная тайна;
- Банковская тайна;

- Служебная информация на рынке ценных бумаг;
- Тайны следствия и предварительного следствия;
- Адвокатские и нотариальные тайны.

В современном информационном обществе вся информация такого типа хранится на компьютерах, передается по сети, по электронной почте. Потому очень важно знать способы безопасного хранения информации, чтобы информация не могла попасть в посторонние руки.

Формирование режима **информационной безопасности** – сложный вопрос. Для его решения необходимы законодательные, организационные, программные и технические меры.

Целью информационной безопасности является обеспечение трех наиболее важных аспектов безопасности: доступность, целостность и конфиденциальность.

Доступность – обеспечение доступа к информации, техническим средствам и технологиям обработки субъектам, имеющим право на свободный доступ к информации.

Целостность – защита информации от «взлома» и незаконных изменений.

Целостность информации подразумевает, что если информация случайно или намеренно повреждена, то компьютеры или автоматизированные системы обеспечивают сохранность информации.

Конфиденциальность – защита от незаконного доступа или чтения частной информации (схема 1).



Схема 1. Составляющие информационной безопасности

Нарушение безопасности компьютерной системы представляет собой набор преднамеренных или непреднамеренных действий и событий, которые могут негативно повлиять на саму систему и сохраненную в ней информацию. В 2016 году выяснилось, что данные 3 миллиардов аккаунтов Yahoo попали в руки злоумышленников. До сих эту утечку по праву считают одной из самых громких и серьезных за последнее время. Также в 2016 году Uber сообщила, что киберпреступники получили доступ к информации клиентов и водителей, эти данные насчитывали более 57 миллионов записей. В 2 017 412 миллионов учетных записей пользователей сервиса для взрослых FriendFinder были также украдены в ходе кибератаки. 2017 также отметился крупнейшей компрометацией данных 147,9 миллионов американцев из-за взлома бюро кредитных историй Equifax. Злоумышленники использовали уязвимость в системе безопасности приложения на веб-сайте компании и получили доступ к номерам социального страхования, датам рождения, адресам и в ряде случаев к номерам водительских удостоверений. Вообще, согласно статистике за 2017 год, в США произошло более 130 крупномасштабных утечек в результате таргетированных кибератак. Этот показатель за год вырос на 27 %. Такие данные Cisco представила в своем отчете. Более 400 000 компьютеров по меньшей мере в 150 стра-

нах были поражены вирусом WannaCry в 2017 году. Этот вымогатель стоил миру убытков, оцениваемых в 4 миллиарда долларов. Количество атак криптоджекинга в прошлом году выросло на 8 500 %. 5,4 миллиардов атак WannaCry были заблокированы за весь 2017 год. Около 24 000 вредоносных мобильных приложений блокируются ежедневно. В 2017 году среднее число скомпрометированных данных по стране составляло 24 089. Наибольшее количество утечек происходило в Индии (более 33 тысяч файлов), второй шла Америка (28,5 тысяч). В марте 2018 года были похищены данные пользователей мобильного приложения MyFitnessPal. Американская компания Under Armour, которой принадлежит это приложение, сообщила об около 150 миллионов пострадавших. С 1 января 2005 года по 18 апреля 2018 было зарегистрировано 8854 случаев утечек.

Безопасная система –

система, которая управляет доступом к информации так, что только определенные лица или устройства, могут читать, создавать и удалять информацию.

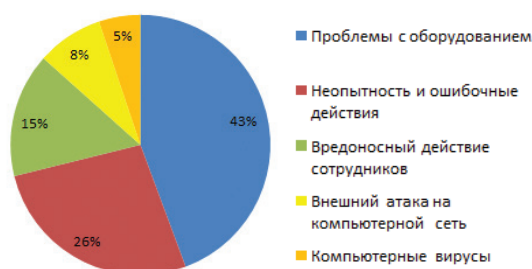
Доверительная система –

система, которая использует доступную информацию и программное обеспечение, чтобы группа пользователей одновременно могла обрабатывать информацию различной конфиденциальности, не нарушая права доступа к информации.

Надежность системы оценивается по двум основным параметрам: политика безопасности и гарантия.

Политика безопасности – набор законов, положений и правил, которые определяют методы обработки, защиты и распространения информации. Эти правила показывают, как, когда и в каком объеме пользователь может работать с информацией.

Гарантия – мера уверенности, которая возлагается на структуру и внедрение системы. Она иллюстрирует ответственность за реализацию политики безопасности.



Анализ



Какие существуют в нашей стране меры и законы обеспечения информационной безопасности? Проанализируйте.



1. Что такое информационная безопасность?
2. Что такое защита информации?
3. Какие виды информации относятся в нашей стране к тайнам?
4. Какие существуют важные аспекты защиты информации?
5. Что такое безопасность компьютерных сетей?
6. Что такое политика безопасности?

Методы защиты информации



Какие имеются способы защиты информации в компьютерной сети?



- ✓ Как вы защищаете важные для себя данные?
- ✓ Какие методы защиты информации в древние времена использовались людьми?
- ✓ Что такое криптография, шифрование?
- ✓ Насколько важна сегодня защита информации на государственном уровне и лично для человека?



Новые знания

Проблема защиты информации путем преобразования, исключающего ее прочтение посторонним лицом, волновала человеческий ум с давних времен. История криптографии развивается параллельно с историей человеческого языка. Даже сама письменность изначально было криптографической системой, потому что в древних обществах ею владели только избранные. Священные книги Древнего Египта, древней Индии могут служить этому примером. Цезарь в своих письмах использовал сложный систематизированный шифр, который сейчас называется его именем.

Когда письменность начала широко распространяться, тогда криптография стала развиваться как отдельная наука. Криптографические системы быстро развивались во время Первой и Второй мировых войн.



Рисунок 1

В послевоенное время стремительное развитие вычислительной техники существенно ускорило создание и совершенствование криптографических методов.

Термин «криптология» происходит от двух древнегреческих слов *cryptos* – скрытый и *logos* – слово. Криптология состоит из двух частей – **криптографии** и **криптоанализа**.

Криптология – это наука о математических методах обеспечения конфиденциальности, то есть невозможности прочтения информации посторонним (рис. 1).

Криптография позволяет модифицировать сообщения для защиты от изменений, кражи и использования другими третьими лицами (противниками).

Человека занимающегося криптографией называют **криптограф**. Криптограф использует методы, которые позволяют сохранить конфиденциальность и точность сообщения.

Криптоанализ обеспечивает дешифровку зашифрованной информации без ключа.

Криптосистемы на основе успешно проведенных криптоаналитических исследований могут наряду с первоначальным текстом сообщения вскрыть и его ключ.

Криптоанализ занимается поиском уязвимостей в криптосистеме, которая позволяет читать зашифрованное сообщение, ключ или их вместе одновременно. **Криптосистема** представляет собой алгоритм шифрования, всевозможные открытые тексты, шифротексты и ключи.

Современная криптография состоит из 4 основных разделов:

1. Симметричные криптосистемы.
2. Криптосистемы с открытым ключом.
3. Система электронной подписи (электронная подпись).
4. Управление ключами.

Зашифрованный текст или криптограмма – это результат применения криптографии к открытому тексту. **Шифрованием** называют процесс преобразования открытого текста с помощью шифра в зашифрованный текст. Для шифрования открытого текста криптограф всегда использует секретный ключ. Способность шифра противостоять вскрытию, всевозможным атакам на него называют криптографической стойкостью (**криптостойкостью**) шифра. В качестве информации, подлежащей шифрованию и дешифрованию, будут рассматриваться тексты, построенные на некотором алфавите. **Алфавит** – конечное множество знаков, используемых для кодирования информации. **Текст** – упорядоченный набор из элементов алфавита. В современных информационных системах (ИС) методы криптографии широко используются. В качестве примеров алфавитов.

- ✓ Алфавит Z42 – 42 символа казахского алфавита и пробел.
- ✓ Алфавит Z33 – 32 буквы русского алфавита и пробел.
- ✓ Алфавит Z256 – символы, входящие в стандартные коды ASCII и КОИ-8.
- ✓ Бинарный алфавит – $Z_2 = \{0,1\}$.
- ✓ Восьмеричный алфавит и шестнадцатеричный алфавит.

Шифры замены

При шифровании с помощью шифра замены каждый символ текста заменяется по определенному правилу на какие-либо другие буквы, числа, символы и т.д. При этом замена осуществляется так, чтобы потом по зашифрованному сообщению можно было однозначно восстановить передаваемое сообщение. Например, давайте сделаем самое простое шифрование.

Для этого мы должны заменить буквы в казахском алфавите (рис. 2). Будем считать, что А=1, Ә=2, ...Ю=41, Я=42. Тогда слово **шифр** будет выглядеть как **34 12 29 23**. Теперь давайте зашифруем слово **шифр**, используя номер пересечения строки и столбца.

шифр = 56 25 51 42

Конечно, это примитивный метод шифрования. Если нужно этот метод усложнить, то мы можем переставить буквы алфавита местами. Этот метод шифрования относится к симметричной криптографии. То есть шифровальщик и дешифровщик должны иметь на руках новую версию алфавита. Дешифрование таких шифров сложно выполнить вручную, а проверить все варианты замещения (перебор вариантов) с помощью компьютера можно в течение 1 секунды.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | А | Ә | Б | В | Г | Ғ | Д |
| 2 | Е | Ё | Ж | З | И | Й | К |
| 3 | Қ | Л | М | Н | Ң | О | Ө |
| 4 | П | Р | С | Т | У | Ұ | Ү |
| 5 | Ф | Х | Һ | Ц | Ч | Ш | Щ |
| 6 | Ъ | Ы | І | Ь | Э | Ю | Я |

Рисунок 2

Таблицы шифрования

Криптограмма появилась в эпоху Возрождения. Ее широко использовали для защиты политических, дипломатических, военных секретных сведений. Широко распространенным методом шифрования в эпоху Возрождения было табличное шифрование. В табличном шифровании ключом является размер таблицы. Этот метод шифрования похож на шифр Скитала. Например, зашифруйте сообщение:

ВСТРЕЧА С СЕКРЕТНЫМ АГЕНТОМ ЗАВТРА В ОДИННАДЦАТЬ

Запишем текст по столбцам в двумерной таблице размером 6x7 (таблица 1).

Таблица 1

| | | | | | | |
|---|---|---|---|---|---|---|
| В | А | Е | Г | З | В | А |
| С | С | Т | Е | А | О | Д |
| Т | С | Н | Н | В | Д | Ц |
| Р | Е | Ы | Т | Т | И | А |
| Е | К | М | О | Р | Н | Т |
| Ч | Р | А | М | А | Н | Ь |

Чтобы получить зашифрованный текст в таблице, прочитаем текст по строкам. Если записать зашифрованный текст, разделив его по 6 букв, то появится следующий шифрованный текст:

ВАЕГЗВ АССТЕА ОДТСНН ВДЦРЕЫ ТТИАЕК МОРНТЧ РАМАНЬ

Из современных видов криптографии наиболее известной нам является **Электронная система цифровой подписи (ЭЦП)**. Система цифровой подписи была внедрена в нашей стране 7 января 2003 года в рамках закона «Об электронном документе и электронной цифровой подписи» и сегодня успешно работает. Согласно этому закону **ЭЦП** является электронной копией подписи физического или юридического лица. Цифровая подпись защищает документ от подделки, придавая юридическую силу электронному документу. Все учреждения, предприятия, физические и юридические лица в нашей стране удостоверяют свои юридические, финансовые и другие отношения с помощью ЭЦП. В частности, физические лица могут получать более 40 видов услуг через сеть Интернет на дому с помощью электронной подписи на портале электронного правительства (<http://egov.kz>).



1. Какова цель науки криптологии?
2. Что такое шифрование?
3. Из скольких частей состоит современная криптография?
4. Что такое электронная цифровая подпись? В чем ее смысл, значимость.



Задания

Задание 1.

- а) Используйте казахский алфавит на рисунке 2, чтобы зашифровать фразу «Встреча в библиотеке в десять часов».
- б) Переставив буквы казахского алфавита на рисунке 2, зашифруйте фразу «Встреча в библиотеке в десять часов». Какие изменения вы заметили?

Задание 2.

- а) Зашифруйте фразу «Встреча в библиотеке в десять часов» с использованием таблицы шифрования (*таблица 1*).
- б) Используйте таблицы шифрования для дешифрования предложения: ЖРМТЖИ СИГЯСН ПТВАВЯ ОТЕОНЛ СЙОМЙИ ЯЛКСОО ЗЕОРИЙ



1. При изучении темы вы познакомились с двумя видами шифрования. Однако имеется много других видов шифрования. Выполните мини-исследование на тему «Виды шифрования». Напишите небольшой отчет об исследовании.
2. Проведите мини-исследование на тему «Применение криптографической системы в военной сфере». Напишите небольшой отчет об исследовании.

Методы идентификации личности



Как осуществляется защита паролей, учетных данных, биометрической аутентификации пользователей?



- ✓ Какова, на ваш взгляд, основная задача пользователей сети (например, сети Интернет)? Обоснуйте.
- ✓ Что вы понимаете под идентификацией?
- ✓ Как вы защищаете свою информацию в сети?



Новые знания

Каждый пользователь, зарегистрированный в компьютерной сети, должен быть связан с конкретной информацией, которая определяет именно его. Эта информация называется **идентификатором** пользователя.

Идентификатор состоит из числовой или символьной последовательности. Если пользователь имеет идентификатор, зарегистрированный в сети, то он считается легальным (законным) пользователем, остальные пользователи относятся к нелегальным пользователям. Прежде чем получить доступ к ресурсам компьютерной системы, пользователь должен пройти процедуру **идентификации** и **аутентификации**.



Рисунок 1. Атрибутивный способ

Идентификатор (Identification) – это уникальное имя или изображение, по которому можно распознать пользователя или объект.

Идентификация (Identification) – процедура распознавания пользователя по его идентификатору (имени). Эта функция выполняется, когда пользователь делает попытку войти в сеть. Пользователь сообщает системе по ее запросу свой идентификатор, и система проверяет в своей базе данных его наличие.

Процесс идентификации людей может осуществляться двумя способами: **атрибутивным** и **биометрическим**.

Атрибутивный способ предполагает выдачу человеку доступа либо уникального предмета: пропуска, жетона или пароля (кода), либо предмета, содержащего код (рис. 1).

Биометрический способ основан на использовании индивидуальных биологических особенностей человека. Это узоры пальцев, узоры сетчатки и радужной оболочки глаза и другие характеристики. Эксперты считают этот способ наиболее надежным (рис. 2).



Рисунок 1. Биометрический способ

Аутентификация (Authentication) – процедура проверки подлинности заявленного пользователя или устройства. Эта проверка позволяет достоверно убедиться, что пользователь или устройство является именно тем, кем (чем) себя объявляет.

При проведении аутентификации пользователь тоже активно участвует в процессе обмена информацией. Обычно пользователь подтверждает свою идентификацию. Идентификация и аутентификация являются взаимосвязанными процессами распознавания и проверки подлинности субъектов (пользователей). Именно от них зависит последующее решение системы: можно ли разрешить доступ к ресурсам системы конкретному пользователю. Для доступа ко всем сетевым ресурсам пользователю достаточно один раз пройти процедуру аутентификации (рис. 3).

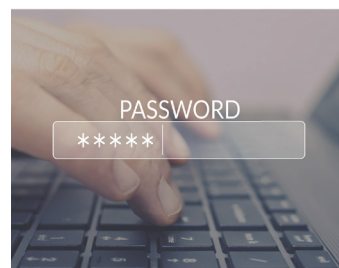


Рисунок 3.
Аутентификация с помощью пароля

Классификация биометрических характеристик человека (схема 1)

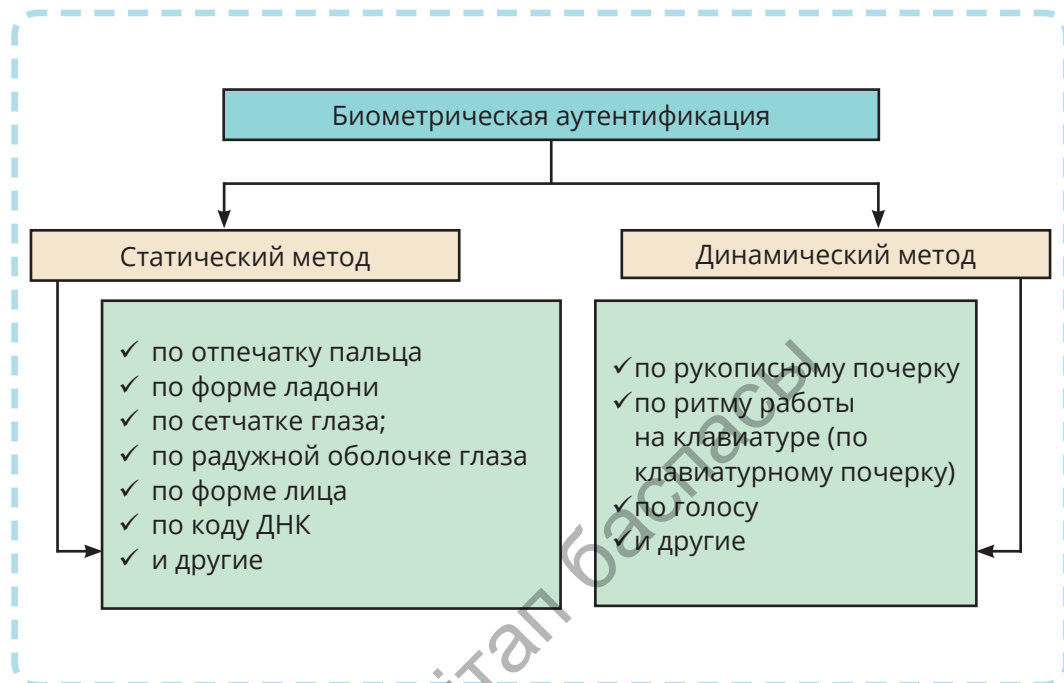


Схема 1



Анализ



В какой степени идентификация и аутентификация необходима в нашей повседневной жизни? Проанализируйте.

- ✓ В какой области наиболее часто используют идентификацию и аутентификацию?
- ✓ Где мы ежедневно встречаемся с идентификацией и аутентификацией?



Синтез



Предложите способы защиты своих идентификационных учетных данных, чтобы они не были доступны злоумышленникам.



Оценка



Насколько важны идентификация и аутентификация для защиты информации? Оцените.



1. Что такое идентификация?
2. Что такое аутентификация?
3. Какие способы идентификации имеются?
4. Что такое биометрическая аутентификация?



Напишите эссе «Современные вопросы, задачи, проблемы защиты информации в сети».

1. Создайте, используя «Правила выбора пароля» и «Три уровня сложности пароля», понравившийся логин и 3 уникальных пароля, чтобы **идентифицировать** себя в сети.
2. По рисунку 4 сформулируйте и обобщите свои мысли о проведении идентификации и аутентификации товаров в магазинах.



Рисунок 4

Урок проектных работ



Какие есть на сегодняшний день актуальные проблемы при работе в компьютерных сетях и в вопросах защиты информации?



✓ С какими проблемами сталкиваются люди из-за развития сетевых технологий, сети Интернет? Поделитесь мыслями.



Исследование

Темы для подготовки проекта

Выберите одну из предложенных тем проекта и разработайте ее индивидуально или в паре, проведите защиту. При выполнении проекта учитывайте, что кроме представления теоретического материала, можно будет представить и программный продукт проекта, что будет наиболее ценно.

Тема проекта №1:

Проблемы современного общества: Интернет-зависимость.

Тема проекта №2:

Влияние средств массовой информации на формирование общества.

Тема проекта №3:

Возможности создания сайта.

Тема проекта №4:

Интернет в нашей жизни.

Тема проекта №5:

Вирусы и опасности, которые встречаются при работе в сети Интернет.

Тема проекта №6:

Киберпреступления и пути борьбы с ними.

Тема проекта №7:

Система доменных имен.

Тема проекта №8:

Компьютерные сети и проблемы информационной безопасности в современном обществе.

Обратите внимание при защите проекта!

- ✓ Ознакомить с выбранной темой;
- ✓ Объяснить, почему была выбрана именно эта тема;
- ✓ Четко представить свои цели и задачи;
- ✓ Перечислить использованные инструменты;
- ✓ Сделать краткий обзор темы;

- ✓ Поделиться идеями, использованными в проекте;
- ✓ Обсудить новые знания и навыки, приобретенные в ходе работы над проектом;
- ✓ Ознакомить с выводами проекта;
- ✓ Выслушать мнение и оценки слушателей, уметь уважать их.

Критерии оценки проекта:

- ✓ Значимость и актуальность выдвинутых проблем, их адекватность изучаемой тематике;
- ✓ Полнота раскрытия темы;
- ✓ Самостоятельность работы над проектом;
- ✓ Оригинальность решения проблемы;
- ✓ Использование технических и наглядных средств;
- ✓ Качество выполнения продукта;
- ✓ Раскрытие содержания проекта на презентации;
- ✓ Умение отвечать на вопросы, аргументированность ответов;
- ✓ Артистизм и выразительность выступления.

В конце проекта напишите отчет!

- ✓ Почему я выбрал именно эту тему?
- ✓ Что узнал нового, чему научился?
- ✓ Как находил нужную информацию?
- ✓ Как обрабатывал информацию?
- ✓ Как совместил теорию и практику?
- ✓ Что получилось удачным?
- ✓ Что не смог выполнить?
- ✓ Что было легко?
- ✓ Что было трудно?
- ✓ Если бы пришлось выполнить проект снова, что бы я изменил?
- ✓ Мои впечатления от реализации проекта.

1

Системы счисления



Какие виды систем счисления известны?



- ✓ Как люди научились считать? Какие системы счисления вы знаете?
- ✓ Какая система счисления применяется в компьютере?
- ✓ Можно ли перевести число из одной системы счисления в другую?



Новые знания

Понятие **«число»** важно как в математике, так и в информатике. Число выражается символом или последовательностью символов некоторого алфавита. Такие символы называют *цифрами*. С понятием *числа* связано понятие *«системы счисления»*.

Система счисления – совокупность правил записи чисел с помощью определенного набора цифр.

Все системы счисления древности и настоящего времени делятся на две большие группы – **позиционные и непозиционные** (схема 1).

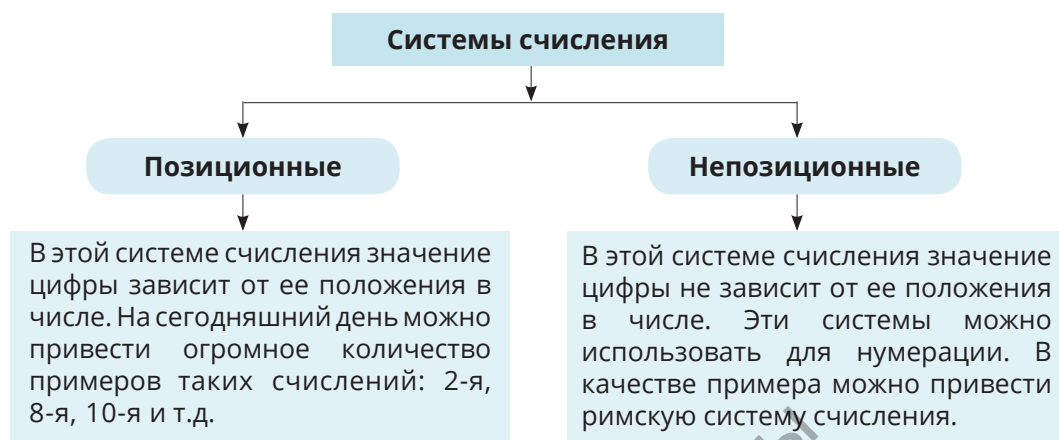


Схема 1. Системы счисления

Непозиционные системы счисления

Непозиционными системами счисления пользовались древние египтяне, греки, римляне и другие народы. Среди непозиционных систем счисления наиболее распространена римская система счисления. Например, число 36 записывается следующим образом:

$$XXXVI = 10 + 10 + 10 + 5 + 1$$

Недостатки непозиционных систем счисления:

- При записи больших чисел каждый раз приходится добавлять новые знаки.

- Невозможно записывать дробные и отрицательные числа.
- Ввиду отсутствия алгоритма вычислений трудно выполнять арифметические действия.

Позиционные системы счисления

В позиционной системе счисления значение каждой цифры зависит от позиции этой цифры в записи числа. В настоящее время наиболее распространенными системами счисления являются десятичная, двоичная, восьмеричная и шестнадцатеричная.

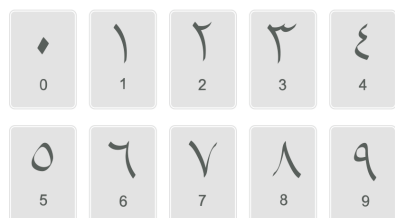


Рисунок 1. Арабские цифры

В каждой позиционной системе счисления имеется определенное количество цифр – это *основание* системы счисления.

В позиционной системе счисления значение каждой цифры изменяется в зависимости от ее положения в последовательности цифр, изображающих число.

Привычная нам система счисления называется *десятичной*, она состоит из 10 арабских цифр (рис. 1). Для записи цифры используются 10 цифр от 0 до 9-ти, основание – 10; в двоичной системе используются только цифры 0 и 1, основание – 2; восьмеричная система счисления состоит из восьми цифр, основание – 8; в шестнадцатеричной системе счисления используются десять цифр десятичной системы и для оставшихся шести чисел буквы латинского алфавита, основание – 16 (таблица 1).

Таблица 1

Системы счисления

| Наименование системы счисления | Цифры системы счисления |
|--------------------------------|--|
| Двоичная | 0,1 |
| Восьмеричная | 0, 1, 2, 3, 4, 5, 6, 7 |
| Десятичная | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 |
| Шестнадцатеричная | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A(10), B(11), C(12), D(13), E(14), F(15) |

Десятичная система счисления

Десятичная система является позиционной, так как значение цифр зависит от их положения в последовательности записи числа. Например, число 658 состоит из шести сотен, пяти десятков и восьми единиц.

$$658 = 6 \cdot 10^2 + 5 \cdot 10^1 + 8 \cdot 10^0$$

Позиция цифры при записи числа называется *разрядом*.

Двоичная система счисления

В двоичной системе счисления «2» является основанием системы. В этой системе любое число записывается цифрами 0, 1 и называется

двоичным числом. Каждый разряд (каждую цифру) двоичного числа называют битом информации.

$$101101_2 = 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 45_{10}$$

Значимость двоичной системы счисления определяется тем, что представление любой информации в компьютере является двоичным. Недостаток двоичной системы – необходимость записи большого количества знаков, а именно цифр – 0 и 1. Это затрудняет восприятие человеком двоичной системы счисления. Поэтому двоичная система используется компьютером «для внутреннего использования». Для работы же человека с компьютером выбирается система счисления с большими основаниями. Это восьмеричная и шестнадцатеричная системы счисления. Перевод и способы, облегчающие такой перевод между этими двумя системами счисления и двоичной системой, будут рассмотрены в следующих параграфах.

Восьмеричная система счисления

Восьмеричная система относится к позиционной системе счисления. В восьмеричной системе счисления количество цифр равно 8-ми. В этой системе наибольшая цифра 7.

$$714_8 = 7 * 8^2 + 1 * 8^1 + 4 * 8^0 = 460_{10}$$

Шестнадцатеричная система счисления

Для записи в шестнадцатеричной системе счисления используются десять цифр 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 десятичной системы и для оставшихся шести цифр – 10, 11, 12, 13, 14, 15 – соответствующие заглавные буквы латинского алфавита: A, B, C, D, E, F. Поэтому в шестнадцатеричной системе, например, возможна запись вида 3E5A:

$$3E5A_{16} = 3 * 16^3 + E * 16^2 + 5 * 16^1 + A * 16^0$$

Для обозначения принадлежности числа к конкретной системе счисления внизу справа у числа записывается индекс той или иной системы счисления.



Применение

1. Расположите числа 879, 3254 и 12354 по разрядам.
2. Переведите числа 1110011_2 ; 101010_2 ; 11111110_2 ; 1000111_2 в десятичную систему счисления.
3. Переведите числа из восьмеричной системы счисления в десятичную: $43_8 = ?$; $325_8 = ?$; $42_8 = ?$; $736_8 = ?$; $1235_8 = ?$
4. Переведите числа из шестнадцатеричной системы счисления в десятичную: $4A3_{16} = ?$; $3CB_{16} = ?$; $F43_{16} = ?$; $E31_{16} = ?$; $10D_{16} = ?$



Оценка



Оцените значение систем счисления в жизни человека.



1. Что такое система счисления?
2. Что такое позиционная система счисления?
3. Какие примеры непозиционной системы счисления вы можете привести?

2

Перевод чисел из одной системы счисления в другую



Как можно перевести целые числа десятичной системы счисления в двоичную и наоборот?



- ✓ Как переводить числа из одной системы счисления в другую?
- ✓ Для чего переводим числа из одной системы счисления в другую?



Новые знания

Существует несколько способов перевода чисел из одной системы счисления в другую. Из десятичной системы счисления – заданное число нужно делить на *основание* переводимой системы счисления. Например, при переводе из десятичной системы счисления в двоичную надо делить на 2.

Алгоритм перевода из десятичной системы в двоичную:

1. Целое число десятичной системы счисления делим на 2, записываем остаток.
2. Если полученное частное больше 2, то его берем вместо делимого и делим на 2 до тех пор, пока частное не станет равным 1.
3. Число в двоичной системе записывается как последовательность 1 и 0, начиная с последнего частного, и остатков от делений в обратном порядке (рис. 1, 2).
 $25_{10} = 11001_2$

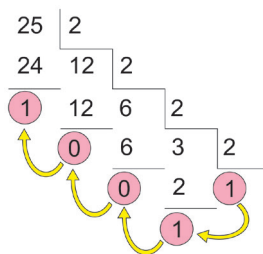
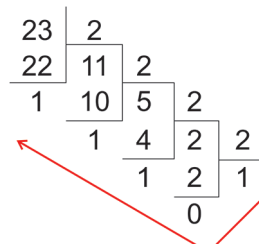


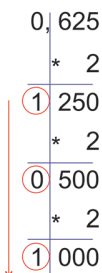
Рисунок 1



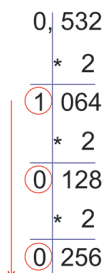
$23_{10} = 10111_2$

Рисунок 2

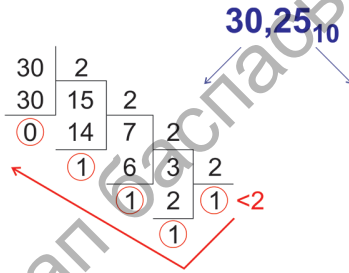
Для перевода дробных чисел десятичной системы в двоичную применяется способ, показанный ниже.



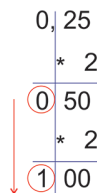
$0,625_{10} = 0,101$



$0,532_{10} = 0,100_2$



$30,25 = 11110,01_2$



Как перевести число из двоичной системы счисления в десятичную?

Перевод чисел из двоичной системы счисления в десятичную систему счисления записывается по формуле:

$$a_{n-1} p^{n-1} + a_{n-2} p^{n-2} + \dots + a_1 p^1 + a_0 p^0 + a_{-1} p^{-1} + \dots + a_{-m} p^{-m}$$

где a – это коэффициенты, p – основание СС и n – разряд (позиция) цифры в числе

Пример: Перевод числа 111100110_2 из двоичной системы счисления в десятичную.

Запишем цифры заданного числа слева направо в виде суммы произведений степеней основания двоичной системы счисления на соответствующие цифры в разряде числа. Разряды считаются, начиная с нулевого, т.е. в первом слагаемом возводим 2 в степень на единицу меньше числа разрядов. Например, если число разрядов двоичного числа равно 9, то степень числа 2 в выражении начинается с 8-ми.

$$111100110_2 = 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 256 + 128 + 64 + 0 + 0 + 32 + 4 + 2 + 0 = 486_{10}$$

Пример: Переведите дробное число $1101,11_2$ из двоичной системы в десятичную.

$$1101,11_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = 8 + 4 + 0 + 1 + \frac{1}{2} + \frac{1}{4} = 13,75_{10}$$



Применение

1. В классе $100011_2\%$ мальчиков и 1101_2 девочек. Сколько учеников в классе?
2. У меня 100 младших братьев. Одному – 1000, а второму – 1111 лет. Старший из них учится в 1001 классе. Возможно ли такое? Объясните.
3. Переведите следующие числа из десятичной системы счисления в двоичную.
1) 29; 2) 36; 3) 74; 4) 46; 5) 124; 6) 37; 7) 128; 8) 64; 9) 245; 10) 189.
4. Переведите числа, приведенные ниже, из двоичной системы счисления в десятичную.
1) 10000 ; 2) 111011; 3) 10011; 4) 111111; 5) 110011; 6) 111001.
7) 100010,01; 8) 111100,10; 9) 1010,1110; 10) 100011,111.



Синтез



Составьте две задачи, связанные с числами в двоичной системе счисления, и предложите пути их решения.



Оценка



Оцените роль систем счисления в развитии компьютерной техники.



1. Какие вы можете привести примеры позиционной системы счисления?
2. Как переводятся числа из десятичной системы в двоичную?
3. Расскажите алгоритм перевода из двоичной системы счисления в десятичную.

3

Перевод из десятичной в восьмеричную и шестнадцатеричную системы счисления



Как перевести из десятичной системы счисления в восьмеричную и шестнадцатеричную системы и наоборот?



- ✓ Почему используют именно восьмеричную и шестнадцатеричную системы счисления наряду с двоичной (а не троичную и не семеричную, например)?
- ✓ Как вы думаете, в чем причина использования восьмеричной и шестнадцатеричной систем счисления в компьютере?



Новые знания

| HTML name | Hex code | | | Decimal code | | |
|---------------------|----------|----|----|--------------|-----|-----|
| | R | G | B | R | G | B |
| Green colors | | | | | | |
| GreenYellow | AD | FF | 2F | 173 | 255 | 47 |
| Chartreuse | 7F | FF | 00 | 127 | 255 | 0 |
| LawnGreen | 7C | FC | 00 | 124 | 252 | 0 |
| Lime | 00 | FF | 00 | 0 | 255 | 0 |
| LimeGreen | 32 | CD | 32 | 50 | 205 | 50 |
| PaleGreen | 98 | FB | 98 | 152 | 251 | 152 |
| LightGreen | 90 | EE | 90 | 144 | 238 | 144 |
| MediumSpringGreen | 00 | FA | 9A | 0 | 250 | 154 |
| SpringGreen | 00 | FF | 7F | 0 | 255 | 127 |
| MediumSeaGreen | 3C | B3 | 71 | 60 | 179 | 113 |

Рисунок 1. Представления цветовой палитры в шестнадцатеричная системе

Работая с цифровыми устройствами и на компьютере, вы встречали набор символов типа 9EA. Этот набор символов похож на шифр. Но это просто число, только представленное в шестнадцатеричной системе счисления.

Существуют свои трудности в использовании двоичной системы счисления в компьютере. При использовании больших чисел в двоичной системе последовательность из цифр 0 и 1 получается слишком длинной. В таких случаях эффективно использование шестнадцатеричной системы счисления. Шестнадцатеричная система счисления используется при записи кодов ошибок. Эти ошибки возникают во время работы различных программных продуктов. Также эта система применяется для представления цветовой палитры (рис. 1).

Перевод чисел из одной системы счисления в другую осуществляется несколькими способами. Например, если переводим десятичное число в восьмеричное и шестнадцатеричное, то его делим, соответственно, на 8 и 16 (рис. 2 и 3).

Перевод числа из десятичной системы счисления в восьмеричную и шестнадцатеричную.

$$\begin{array}{r}
 315 \overline{) 8} \\
 \underline{- 24} \quad \underline{39} \overline{) 8} \\
 \quad \underline{75} \quad \underline{32} \quad 4 \\
 \quad \underline{- 72} \quad \underline{7} \\
 \quad \quad 3
 \end{array}$$

Рисунок 2

$$\begin{array}{r}
 315 \overline{) 16} \\
 \underline{- 16} \quad \underline{19} \overline{) 16} \\
 \quad \underline{155} \quad \underline{16} \quad 1 \\
 \quad \underline{- 144} \quad \underline{3} \\
 \quad \quad \mathbf{B(11)}
 \end{array}$$

Рисунок 3

$$315_{10} = 473_8 = 13B_{16} \text{ примечание: } 11_{10} = B_{16}$$

| | | | |
|---|---|-------|---|
| * | 0 | 1875 | 8 |
| | | 5000 | 8 |
| * | 1 | 0000 | 8 |
| | | 40000 | 8 |

Здесь показан прием перевода дробных чисел десятичной системы счисления в восьмеричную и шестнадцатеричную.

В итоге:
 $0,1875_{10} = 0,14_8 = 0,3_{16}$

| | | | |
|---|---|-------|----|
| * | 0 | 1875 | 16 |
| | | 30000 | 16 |



Как выполнить обратный перевод чисел 8-й и 16-й систем счисления в десятичную?

Для перевода из любой системы счисления в десятичную используется следующая формула.

$$a_{n-1} p^{n-1} + a_{n-2} p^{n-2} + \dots + a_1 p^1 + a_0 + a_{-1} p^{-1} + \dots + a_{-m} p^{-m}$$

где P – основание системы счисления, a – цифра в числе, n – количество цифр целой части числа, m – количество цифр дробной части числа.

При переводе чисел из восьмеричной и шестнадцатеричной систем счисления в десятичную систему и обратно используется тот же способ, что и при переводе из двоичной в десятичную. Разница в том, что вместо основания 2 надо использовать основания 8 и 16.

Задание: Переведите числа восьмеричной системы 315_8 и шестнадцатеричной системы 315_{16} в десятичную систему счисления.

Решение: $315_8 = 3 \cdot 8^2 + 1 \cdot 8^1 + 5 \cdot 8^0 = 205_{10}$
 $315_{16} = 3 \cdot 16^2 + 1 \cdot 16^1 + 5 \cdot 8^0 = 789_{10}$

Таблица 1

Соответствие систем счисления

| Двоичная система | Восьмеричная система |
|------------------|----------------------|
| 000 | 0 |
| 001 | 1 |
| 010 | 2 |
| 011 | 3 |
| 100 | 4 |
| 101 | 5 |
| 110 | 6 |
| 111 | 7 |

Перевод чисел двоичной системы в 8-ю и 16-ю системы счисления

Познакомимся с таблицами соответствия двоичной и восьмеричной (табл. 1) а также шестнадцатеричной (табл. 2) и двоичной систем счисления. Остановимся на особенностях использования этих таблиц. Так как $2^3 = 8$, то необходимо поделить двоичное число на группы по три цифры при переводе из двоичной системы счисления в восьмеричную систему.

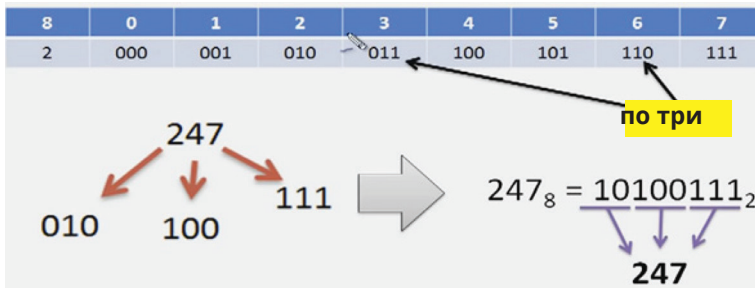


Рисунок 4

Например, рассмотрим число 247 восьмеричной системы счисления. В двоичной системе счисления это число 10100111_2 . Как видно, при разделении числа 10100111_2 на группы по три разряда, получается три группы в виде 10 100 111. В первой группе не хватает одной цифры. Вместо недостающей цифры добавляем цифру 0 (рис. 4). При переводе двоичного числа в восьмеричную систему счисления разбиваем по три цифры с конца числа, спереди можно дополнить нулями, они незначимы.

Так как $2^4 = 16$, то для перевода числа двоичной системы счисления в шестнадцатеричную данные табл. 2 используем, как показано на рис. 5, то есть разделяем в группы по четыре разряда.

Таблица 2

Соответствие значений различных систем счисления

| Двоичная система | Восьмеричная система | Шестнадцатеричная система |
|------------------|----------------------|---------------------------|
| 0000 | 0 | 0 |
| 0001 | 1 | 1 |
| 0010 | 2 | 2 |
| 0011 | 3 | 3 |
| 0100 | 4 | 4 |
| 0101 | 5 | 5 |
| 0110 | 6 | 6 |
| 0111 | 7 | 7 |
| 1000 | 10 | 8 |
| 1001 | 11 | 9 |
| 1010 | 12 | A |
| 1011 | 13 | B |
| 1100 | 14 | C |
| 1101 | 15 | D |
| 1110 | 16 | E |
| 1111 | 17 | F |

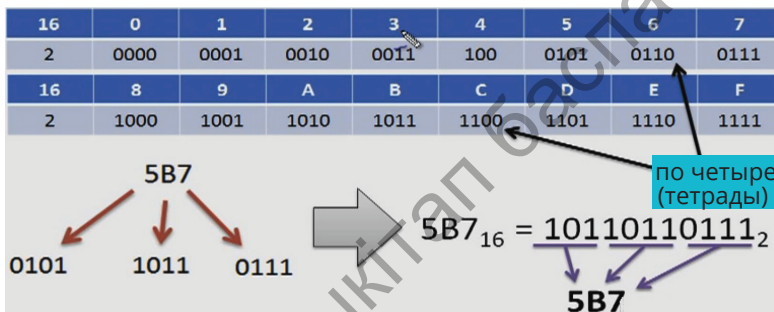


Рисунок 5



Применение

1. Переведите следующие числа из десятичной системы счисления в восьмеричную систему счисления.
1) 219 2) 136 3) 79 4) 80 5) 1024 6) 307 7) 3128 8) 64 9) 120 10) 109
2. Переведите следующие числа из десятичной системы счисления в шестнадцатеричную систему счисления.
1) 2012 2) 736 3) 729 4) 380 5) 1024 6) 237 7) 428 8) 643 9) 520 10) 21
3. Переведите числа из двоичной системы счисления в восьмеричную систему счисления.
1) 10010 2) 10011011 3) 11011 4) 110111 5) 11110011 6) 10101001
4. Переведите числа из двоичной системы счисления в шестнадцатеричную систему счисления.
1) 10111010 2) 1110011011 3) 1111011 4) 1100011 5) 11110011 6) 1011101001
5. Переведите заданные числа в двоичную систему счисления.
1) 25_8 2) 76_8 3) 172_8 4) 38_{10} 5) 10_8 6) 23_{16} 7) 128_{16} 8) $6A38_{16}$ 9) $15E_{16}$ 10) 41_{16}



1. Как выполняется перевод числа из десятичной системы счисления в восьмеричную, шестнадцатеричную и обратно?
2. Для чего применяется в компьютере шестнадцатеричная система счисления?
3. Разъясните формулы перевода двоичных чисел в десятичную систему счисления.
4. В чем заключается метод перевода чисел из одной системы счисления в другую с помощью соответствующих таблиц?

Логические операции (дизъюнкция, конъюнкция, инверсия)



Как можно использовать логические операции (дизъюнкция, конъюнкция, инверсия)?



- ✓ Что такое логика?
- ✓ Какая существует связь между двоичным кодированием и алгеброй логики?



Новые знания

Слово «**логика**» произошло от греческих слов «легос» и «логос». «Легос» означает – «говорить», «пояснять», «логос» – «мудрость».

Логикой называется наука о формах и законах правильного мышления.

Мышление – высшая форма познавательной деятельности человека. Правильное мышление играет решающую роль в деятельности людей. Чтобы человек правильно мыслил, он должен подчиняться определенным правилам и закономерностям.



Аристотель
384– 322 гг. до н.э

Хотя первые учения о формах мышления зародились на Древнем Востоке, основы логики, как науки были заложены древнегреческим мыслителем Аристотелем (384–322 гг. до н.э.). В поисках ответа на вопрос: «Как мы формируем мысль?», он исследовал законы мышления. Аристотеля считают родоначальником логики как дисциплины. Он первым исследовал логику как систему и выделил такие формы мышления, как понятие, высказывание, умозаключение. Так сформировалась формальная логика.

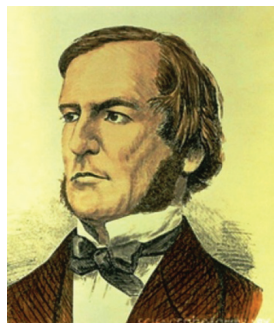
Основные формы мышления:

Понятие – это форма мышления, которая отражает существенные признаки предметов. Объем понятия может быть представлен как множество объектов. Основу современной математики составляют логика и теория множеств.

Высказывание – это форма мышления, содержащая утверждение или отрицание чего-либо. Высказывания могут быть ложными или истинными. Вопросительные и побудительные предложения не являются высказываниями.

Умозаключение – это форма мышления, при которой мысль формируется в результате синтеза и обработки нескольких высказываний и понятий.

Немецкий ученый Готфрид Лейбниц (1646–1716) первым попытался создать язык науки, который позволил бы заменить содержательные высказывания математическими отношениями.



Джордж Буль
(1815–1864)

Спустя сто лет английский математик Джордж Буль развил идею Лейбница о создании логического универсального языка, подчиненного математическим законам.

Он изобрел своеобразную алгебру – систему обозначений и правил, применимую к объектам, числам, буквам и предложениям.

Джордж Буль считается основателем логической алгебры (булева алгебра или алгебра логики).

В основе булевой алгебры лежат операции **конъюнкция (И)**, **дизъюнкция (ИЛИ)** и **отрицание (НЕ)**.

В дальнейшем стало понятно, что ее принципы могут быть положены в основу построения электронных схем переключателей. Например, протекание или отсутствие тока в цепи можно отразить как логические Истина (есть) и Ложь (нет).

В XX веке ученые объединили двоичную систему счисления с алгеброй логики Джорджа Буля, создав тем самым основу для разработки современного **цифрового электронного компьютера**.



Какая существует связь между компьютером и логикой?

Высказывание в математической логике может быть истинным или ложным, третий случай не рассматривается. Если обозначим «истину» как 1, «ложь» как 0, тогда можно выполнять операции с 1 и 0 или, другими словами, в двоичном коде.

Алгебра логики – это математический аппарат записи логических высказываний, вычислений, упрощений и преобразований.

Алгебра логики определяет правила вычислений логических условий 1 (истина) и 0 (ложь). С помощью таких правил можно составить элементы хранения в памяти и арифметические действия на компьютере.

Логические операции

Логическое сложение (дизъюнкция), обозначается символом \vee . Операция, соединяющая два или более высказываний при помощи операции **«ИЛИ»** (например, «А или В») называется логическим сложением, или **дизъюнкцией**. **Дизъюнкция** в обычном языке означает **«или (or)»** (таблица 1).

Логическое сложение двух высказываний будет ложным, только когда оба утверждения будут ложными. Например,

A = Небо ясное. B = День теплый.
Тогда $A \vee B = (\text{Небо ясное}) \vee (\text{День теплый})$

Таблица 1

Дизъюнкция

Например, рассмотрим высказывание «Буду играть во дворе, если небо будет ясным или день теплым». Это означает, что ребенок будет играть во дворе при выполнении хотя бы одного из двух условий. Ребенок будет играть, если небо ясное, а день холодный; ребенок будет во дворе, при условии, когда небо облачное, а день теплый. И лишь при условии, когда небо облачное и день холодный, ребенок останется дома.

| A | B | $A \vee B$ |
|---|---|------------|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 1 |

Логическое умножение (конъюнкция, обозначается символом \wedge). Операция, связывающая два или более высказываний при помощи операции «И», называется **логическим умножением** или **конъюнкцией**. Конъюнкция в обычном языке означает «и (and)» (таблица 2).

Значение **логического умножения** будет истинным только в случае, когда оба высказывания являются истинными. Например:

A = Небо облачное. B = день холодный.
Тогда $A \wedge B = (\text{Небо облачное}) \wedge (\text{День холодный})$

Таблица 2

Конъюнкция

Например, рассмотрим высказывание «Если будет небо облачным и день холодным, я не пойду играть во двор». Т.е., чтобы ребенок играл во дворе, небо должно быть безоблачным и день теплым. В других случаях, когда небо ясное и день холодный, либо небо облачное и день теплый, ребенок не выйдет во двор.

| A | B | $A \wedge B$ |
|---|---|--------------|
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 1 | 1 |

Логическое отрицание (инверсия). Отрицание «НЕ», которым логическое высказывание обращается в обратное, называется логическим отрицанием или инверсией (таблица 3). Инверсия в обычном языке соответствует выражениям «неверно», «обращение в обратное» и приставке «не (not)».

Алматы.ру бағасы

Инверсия обращает высказывание в противоположное, истину – в ложь, а ложь – в истину. Например,

- 1) $A = \text{Небо облачное}; \bar{A} = \text{Небо безоблачное.}$
- 2) $B = \text{Холодный день}; \bar{B} = \text{Нехолодный день.}$

Таблица 3
Инверсия

| A | \bar{A} |
|---|-----------|
| 0 | 1 |
| 1 | 0 |



Применение

Задание № 1. Обратите в противоположное следующие высказывания.

1. Сегодня в театре состоится премьера спектакля «Абай».
2. Четные числа бывают простыми.
3. Самат выполнил заданные ему пять заданий.
4. В каждой школе есть учащиеся, увлекающиеся спортом.

Задание № 2. Даны следующие простые высказывания.

1. $A = \{\text{Монитор – устройство для печати информации на бумаге}\},$
2. $B = \{\text{Принтер – устройство для ввода информации}\},$
3. $C = \{\text{Процессор – устройство обработки информации}\},$
4. $D = \{\text{Мышь – устройство хранения информации}\},$
5. $E = \{\text{Сканер – устройство, которое передает информацию в компьютер}\}.$

Используя таблицы 1, 2, 3 найдите значения следующих логических операций.

- 1) $(A \wedge B)$ 2) $(C \vee D)$ 3) $(A \wedge B)$ 4) $(B \wedge C)$; 5) $(A \vee B)$ 6) $(C \wedge D)$ 7) \bar{A} 8) \bar{B} .



1. Что такое логика?
2. Что такое формальная логика?
3. Что относится к основным формам мышления?
4. Что такое высказывание?
5. Как называются логические операции?
6. Как используют логические операции?

5

Построение таблиц истинности



Как построить таблицу истинности для логических элементов?



- ✓ Что такое таблица истинности логических элементов?
- ✓ Как можно построить таблицу истинности?



Новые знания

Таблица истинности – это таблица, в которой представлены все значения, которые может принимать высказывание при всех возможных значениях простых высказываний, входящих в нее. В таблице истинности отражаются четыре возможных исхода для двух переменных. Это пары:

$(0, 0), (0, 1), (1, 0), (1, 1)$.

Если в формуле три переменных, то количество возможных комбинаций будет равно 8-ми, так как $2^3 = 8$.

$(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)$.

Логическое выражение – это составное выражение, записанное в виде формулы.

При решении задач с логическими высказываниями необходимо построение таблицы истинности.

Алгоритм построения таблицы истинности для логических элементов

- 1 Определить количество переменных n в логическом выражении.
- 2 Подсчитать количество строк $m=2^n$, n – количество переменных.
- 3 Подсчитать количество логических операций в выражении.
- 4 Для подсчета количества столбцов в таблице сложить число переменных и число логических операций.
- 5 Установить последовательность выполнения логических операций с учетом приоритетов.
- 6 Выписать наборы входных переменных.
- 7 Заполнить таблицы.

Приоритеты при выполнении логических операций

- 1) Выражения в скобках; 2) инверсия; 3) конъюнкция; 4) дизъюнкция.

Рассмотрим практические примеры на построение таблицы истинности.

Задание 1. Для высказываний $A = \{22 - \text{четное число}\}$ и $B = \{3 - \text{четное число}\}$ вычислите значения логических операций.

Пояснение решения задачи:

1. Если применить для двух простых высказываний операцию (И) $A \wedge B$, тогда высказывание A – истинно, так как число 22 – четное. А высказывание B – ложно, так как 3 – нечетное число. Таким образом, согласно правилу конъюнкции $1 \wedge 0 = 0$ (ложь).

2. Если применить для двух простых высказываний операцию (ИЛИ) $A \vee B$, тогда высказывание A – истинно, тогда как B – ложно. Таким образом, операция $1 \vee 0 = 1$ (Истина).

3. Если высказывание A подвергнуть инверсии, то оно будет ложным, так как обратится в высказывание «число 22 нечетное». Точно так же инверсия высказывания B «число 3 нечетное» будет истинным (таблица 1).

Таблица 1

| $A \wedge B$ | $A \vee B$ | \bar{A} | \bar{B} |
|--------------|---------------|-------------|---------------|
| Ложь (0) | Истина (1) | Ложь (0) | Истина (1) |

Задание 2. Запишите таблицу истинности для заданного логического выражения.

$$(A \wedge B) \vee (B \vee C) \vee (\bar{C} \vee A)$$

Выполнение задания: Для записи таблицы истинности, воспользуемся алгоритмом ее построения. В заданном логическом выражении есть 3 переменных. Возможный набор комбинаций равен 8, то есть $2^3 = 8$. Мы запишем их в первых трех столбцах. В следующих столбцах располагаем промежуточные значения логических операций. В последнем столбце записывается значение логической функции (таблица 2).

Таблица 2

| A | B | C | $A \wedge B$ | $B \vee C$ | $(A \wedge B) \vee (B \vee C)$ | \bar{C} | $\bar{C} \vee A$ | $(A \wedge B) \vee (B \vee C) \vee (\bar{C} \vee A)$ |
|---|---|---|--------------|------------|--------------------------------|-----------|------------------|--|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |



Применение

Задание № 1. Даны высказывания – $A = \{4+19=21\}$ и $B = \{8 + 3 = 9\}$. Определите истинность следующих выражений:

$$A \wedge B, \bar{A}, \bar{B}, A \vee B$$

Задание № 2. Даны высказывания – $A = \{\text{Зимой день бывает жарким}\}$ и $B = \{\text{Тюльпан – растение}\}$. Определите истинность следующих выражений:

$$A \wedge B, \bar{A}, \bar{B}, A \vee B$$

Задание №3. Для логических функций, представленных ниже, постройте таблицу истинности. Вычислите их значения. F – логическая функция.

- $F = A \wedge (A \wedge B) \vee (A \vee B)$
- $F = \bar{A} \vee B \wedge (A \wedge A) \vee B$
- $F = (\bar{A} \wedge B) \vee (A \wedge \bar{B}) \vee A$
- $F = (A \wedge \bar{B}) \vee (A \vee B)$
- $F = A \wedge (A \vee B) \vee (\bar{A} \wedge B)$

Задание № 4. Найдите значения логических функций, приведенных ниже.

- $F = (1 \vee 1) \vee (1 \vee 0)$
- $F = (0 \wedge 0) \wedge (1 \wedge 1)$
- $F = (0 \wedge 1) \vee (0 \wedge 1)$
- $F = (0 \vee 0) \wedge (1 \wedge 1)$
- $F = (1 \vee 0) \wedge (0 \wedge 1)$

Задание № 5. Найдите значения логических функций, приведенных ниже.

- | | |
|---|---|
| 1) $F = ((0 > 5) \text{ and } (2 > 3)) \text{ or } (4 < 5)$ | 7) $F = \text{not } (5 > 6)$ |
| 2) $F = (1 < 5) \text{ and } (1 > 0)$ | 8) $F = \text{not } (6 > 5)$ |
| 3) $F = (3 > 2) \text{ or } (3 < 1)$ | 9) $F = (2 = 0) \text{ and } (2 < > 2)$ |
| 4) $F = (3 > 2) \text{ or } (2 > 0)$ | 10) $F = (2 = 0) \text{ or } (2 > 0)$ |
| 5) $F = ((1 > 2) \text{ or } (1 < 0)) \text{ and } ((-1 > 2) \text{ or } (-1 < 0))$ | 11) $F = (3 > 0) \text{ or } (2 > 0)$ |
| 6) $F = (2 > 5) \text{ and } (2 > 3)$ | 12) $F = \text{not } ((5 > 6) \text{ or } (1 = 0))$ |

Примечание. Образец выполнения задания представлен в таблице 3.

Таблица 3

| | | | | |
|---------|---------|---------|--------------------------------|--|
| $0 > 4$ | $2 > 5$ | $4 < 7$ | $(0 > 4) \text{ and } (2 > 5)$ | $F = (0 > 4) \text{ and } (2 > 5) \text{ or } (4 < 7)$ |
| 0 | 0 | 1 | 0 | 1 |



- Что такое таблица истинности?
- Приведите примеры логических выражений.
- Каков алгоритм построения таблицы истинности?
- Какие существуют приоритеты при выполнении логических операций?



Какие важные элементы существуют для выполнения логических операций в компьютере?
Как можно преобразовать логическое выражение в логическую схему или наоборот?



- ✓ Какие виды логических операций вы знаете?
- ✓ Как вы думаете, какова значимость применения логических операций на компьютере?



Новые знания

Логический элемент компьютера является частью электронной логической схемы, которая выполняет элементарную логическую функцию.

Логические элементы компьютера называются электронными схемами **и, или, не**.

Типы элементов компьютерной логики включают в себя так называемые вентили **и, или, не, и-не, или-не** и **триггеры**. Мы будем рассматривать только элементы **и, или, не**.

Логический вентиль – базовый элемент логической схемы, выполняющий элементарную логическую операцию, преобразуя множество входных логических сигналов в выходной логический сигнал.

С помощью этих схем можно выполнять любые логические функции, описывающие работу устройств компьютера. Логические выражения являются основой для создания электронных схем. Обычно вентили имеют от двух до восьми входов и один или два выхода.

Чтобы представить два логических состояния – «1» и «0» в вентилях, соответствующие им входные и выходные сигналы имеют один из двух установленных уровней напряжения. Высокий уровень обычно соответствует значению «**истина**» («1»), а низкий – значению «**ложь**» («0»).

Каждый логический элемент имеет свое **условное обозначение**, которое выражает его логическую функцию. Это упрощает запись и понимание сложных логических схем.

Работу логических элементов описывают с помощью таблиц истинности, знакомых из предыдущей темы.

Физической основой различных интегральных микросхем, составляющих основные части компьютера, является сложное **логическое выражение**.

Схема **И** (рис. 1) выполняет конъюнкцию двух или нескольких логических значений.

Схема **ИЛИ** реализует дизъюнкцию двух или более логических значений (рис. 2).

Схема **НЕ (инвертор)** реализует операцию отрицания (рис. 3).

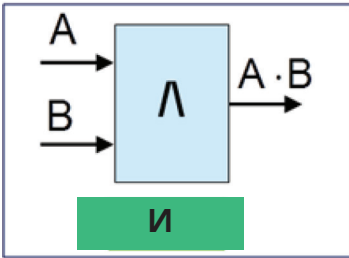


Рисунок 1

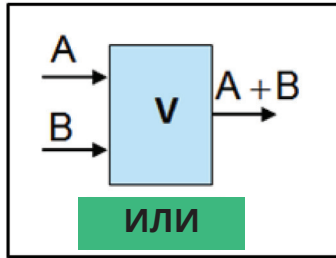


Рисунок 2

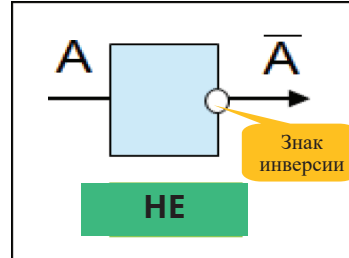


Рисунок 3

Давайте рассмотрим примеры, чтобы понять работу логических элементов компьютера.

Логические схемы являются удобным способом представления логических выражений (таблица 1).

Таблица 1

| Инвертор | Конъюнкция | Дизъюнкция |
|----------|------------|------------|
| | | |
| | | |
| | | |
| | | |

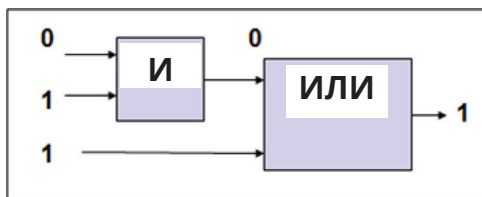


Применение

Пути решение задач

Пример 1. Нарисуйте схему, которая показывает последовательность логических операций для вычисления логического выражения **1 или (0 и 1)**. Вычислите значение логического выражения по схеме.

Решение: $1 \text{ или } (0 \text{ и } 1) = 1 \text{ или } 0 = 1$ (Мы используем 2-ую строку в столбце Конъюнктор в Таблице 1 и 2-ую строку в столбце Дизъюнктор)

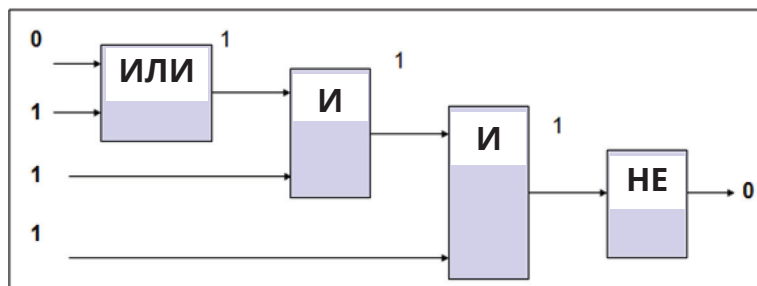


Пример 2. Вычислите значение логического выражения **НЕ (1 И (0 ИЛИ 1) И 1)**, составив схему логических элементов.

Решение: в соответствии с правилом приоритета выполнения действий сначала вычисляются действия в скобках.

Выражение НЕ(1 И (0 ИЛИ 1) И 1):

1. 0 или 1
2. 1 и 1
3. 1 и 1
4. Не (1) результат равен 0.



1. Что относится к логическим элементам компьютера?
2. Что понимается под логическим вентиляем?
3. Сколько логических вентиляей возможно составить?
4. Какова последовательность выполнения логических операций?
Вспомните!



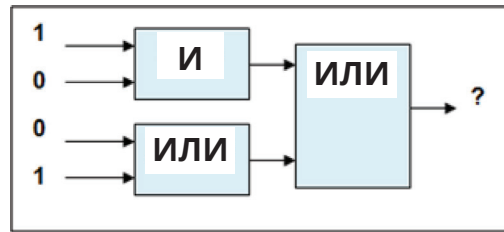
Задания

Задание 1. Вычислите значение логических выражений. Нарисуйте схему.

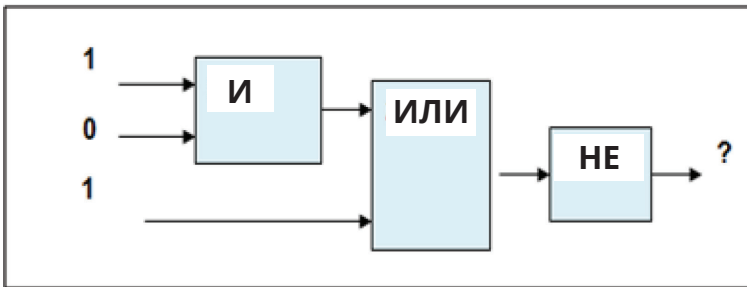
- а) не (1 и (1 или 1) и 0)
- б) 1 и (не (1 или 0) и 1)
- б) 1 и 1 или 0 и 1
- в) 1 и не (не (1) и 0)

Задание 2. Вычислите значение логического выражения по его логической схеме.

а)



б)



Задание 3. Постройте логическую схему, соответствующую логическому выражению:

$F = A \vee B \wedge C$, если $A=1, B=0, C=1$;

$F = (A \vee B) \wedge (C \vee B)$, если $A=0, B=1, C=0$;

$F = (A \wedge B \wedge C)$, если $A=0, B=0, C=1$.

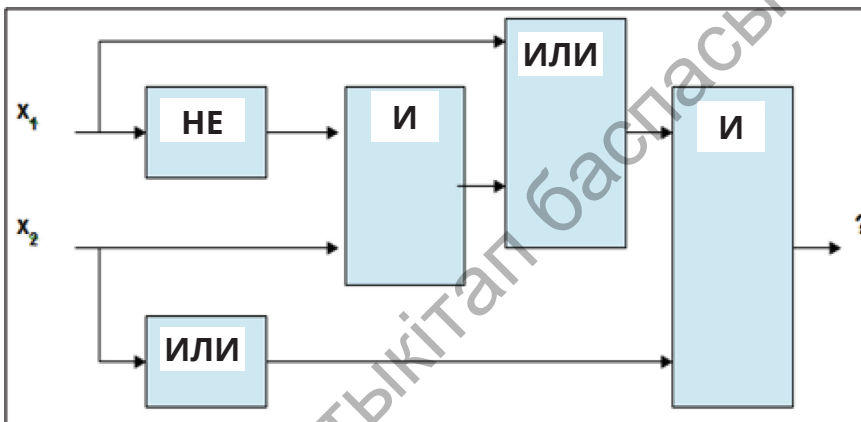
Задание 4. Составьте логическое выражение по заданной логической схеме. Вычислите значение логического выражения.

а) $x_1=0, x_2=1$;

б) $x_1=1, x_2=1$;

в) $x_1=1, x_2=0$;

г) $x_1=0, x_2=0$





Каковы характеристики устройства управления (УУ), арифметико-логического устройства (АЛУ) и регистра памяти компьютера?



- ✓ Что вы знаете о внутреннем устройстве компьютера?
- ✓ Как вы думаете, в чем сходство мышления человека и компьютера?
- ✓ Какие действия человека не может сделать компьютер?



Новые знания



Джон фон
Нейман

Наука об основных принципах построения и архитектуре вычислительных машин тесно связана с именем Джон фон Неймана. Она зародилась в 1944 году, когда Джон фон Нейман с группой ученых впервые создали ламповый компьютер ЭНИАК. В 1945 году Джон фон Нейман представил архитектуру персонального компьютера. Компьютер с такой архитектурой состоит из устройства управления, арифметико-логического устройства (АЛУ), а также устройств ввода и вывода. Эта архитектура, изобретенная более 70 лет назад, актуальна и для современных персональных компьютеров. Процессор в этой архитектуре выполняет функции АЛУ и устройства управления, считывает последовательность команд, содержащихся в памяти, исполняет их и вновь записывает в память. В таких компьютерах все составные части соединяются системными шинами.

Архитектура фон Неймана основывается на 4 принципах

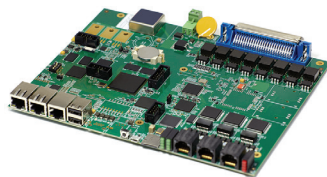
1. Принцип двоичного кодирования. Вся информация компьютера хранится в двоичном коде. Согласно этому принципу, вся информация, как данные, так и команды, кодируются двоичными цифрами 0 и 1. Каждый тип информации представляется двоичной последовательностью и имеет свой формат.



2. Принцип однородности памяти – коды команд и данных хранятся в одной и той же памяти. Иными словами, компьютер не распознает, что хранится в данной ячейке памяти: число, текст или команда. Над командой можно выполнять такие же действия, как и над данными. Распознать их можно только по способу использования, то есть одно и то же значение в ячейке памяти может использоваться и как данные, и как команда, и как адрес в зависимости от способа обращения к нему.



3. Принцип адресности – компьютеру доступны все данные из памяти. Структурно основная память состоит из пронумерованных ячеек, причем процессору в произвольный момент доступна любая ячейка. Двоичные коды команд и данных разделяются на единицы информации, называемые словами, и хранятся в ячейках памяти, а для доступа к ним используются номера соответствующих ячеек – адреса.



4. Принцип программного управления. Задачи на компьютере выполняются с помощью алгоритмов. Алгоритмы в памяти состоят из последовательных команд и выполняются в соответствующем порядке. Все вычисления, предусмотренные алгоритмом решения задачи, должны быть представлены в виде программы, состоящей из последовательности управляющих слов – команд. При необходимости, с помощью специальных команд эта последовательность может быть изменена. Решение об изменении порядка выполнения команд программы принимается на основании анализа результатов предшествующих вычислений.



Несмотря на многочисленные изменения в современных компьютерах, связанные с развитием науки и техники, они все разработаны в соответствии с основными принципами архитектуры фон Неймана (схема 1).

Схема 1

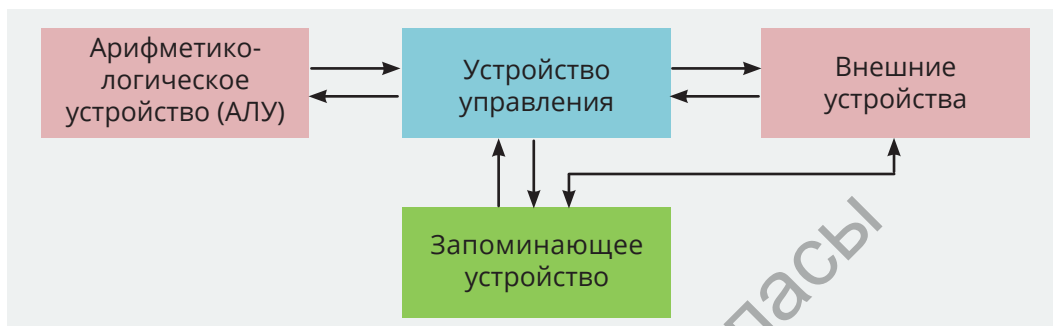


Схема 1. Архитектура ЭВМ Джона фон Неймана

Современные компьютеры отличаются количеством ядер процессора и памятью нескольких уровней (регистры, кэш-память и оперативная память).

Основные функции и внутренняя структура процессора

Процессор – это микросхема, подсоединенная к системной шине, выполняющая программы из памяти компьютера. Программа состоит из по-

следовательности команд. Каждая команда имеет различный размер и содержит информацию не только для выполнения, но также и для обработки информации. Так как все компьютеры работают с двоичными данными (единица и ноль); команды и данные вводятся в двоичном коде.

Основные функции процессора

Выбор. Каждая команда сохраняется в оперативной памяти и имеет свой адрес.

Декодирование. Все команды, предназначенные для выполнения должны быть переведены на язык Ассемблера. Код Ассемблера понятен для процессора и представляется в виде бинарных (двоичных) указаний.

Выполнение. Процессор при выполнении указаний осуществляет одно из трех действий:

- 1) дает указание АЛУ;
- 2) перемещает данные в памяти с одного места в другое;
- 3) переходит в ячейку с другим адресом.

Управление. После выполнения указаний, передает результат и записывает его в память.

Работа основных устройств процессора (рис. 1).

Арифметико-логическое устройство выполняет следующие операции:

- логические преобразования кодов с фиксированными и переменными длинами;
- арифметические операции над числами с фиксированной и плавающей запятой;

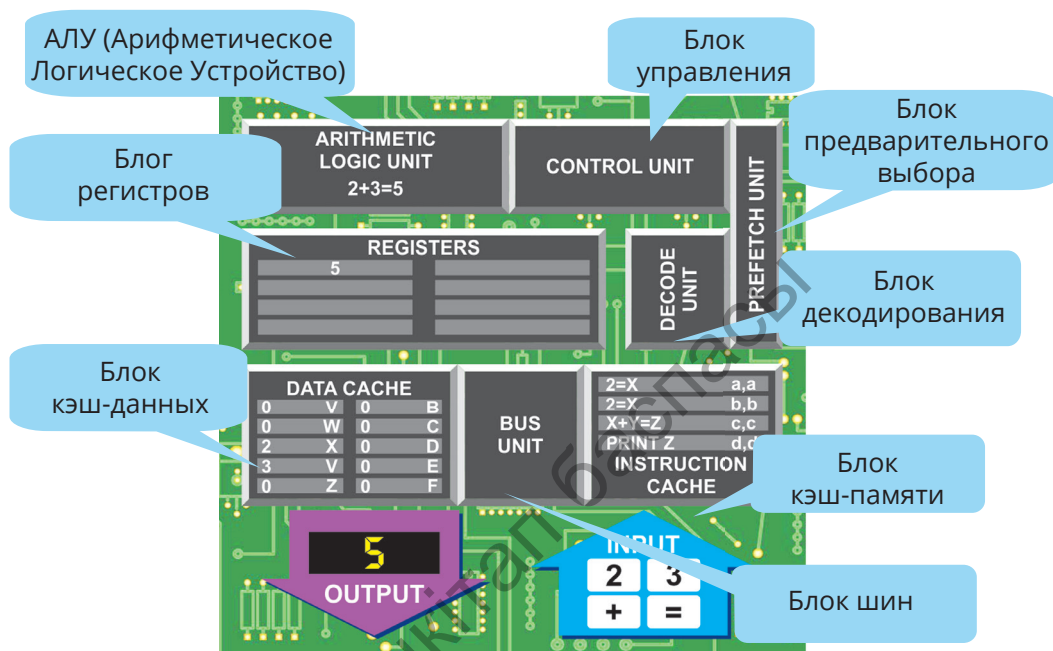


Рисунок 1. Внутренняя структура процессора

– арифметические операции с десятичными числами, представленными в двоично-десятичном коде;

– обработка алфавитно-цифровых данных переменной длины и др.

Выполнение в процессоре всех арифметико-логических операций возможно при наличии одного универсального АЛУ или нескольких специализированных АЛУ. В случае наличия специализированных АЛУ увеличивается не только количество устройств процессора, но и скорость работы программ и облегчается схема выполнения операций.

УУ (устройство управления) формирует и подает определенные сигналы управления, обусловленные спецификой выполнения различных операций.

БРУ (блок регистра управления) предназначен для временного хранения информации управления. Он содержит регистры и счетчики, участвующие в управлении вычислительным процессом:

- регистры, которые хранят информацию о состоянии процессора;
- регистр-счетчик адресов команд – счетчик команд;
- счетчик тактов;
- регистр запроса прерывания.

Блок регистров (локальная память) увеличивает логические возможности и скорость процессора. Его емкость меньше основной памяти, но быстродействие больше. Локальные адреса ячеек памяти короче адресов основных ячеек памяти. Поэтому команды для локальных ячеек будут также короче. Локальная память используется как аккумулятор, как счетчик стека.



1. Каковы принципы фон Неймана?
2. Какие преимущества имеют современные компьютеры по сравнению с первыми компьютерами?
3. Назовите внутренние устройства процессора.
4. Охарактеризуйте функции основных устройств процессора.
5. Каково назначение арифметико-логического устройства?



Задание

Используйте сеть Интернет, чтобы выбрать два процессора из современных семейств (например, Intel и AMD), и сравните их, запишите их характеристики в следующую таблицу.

| Название процессора | Тактовая частота | Количество ядер | Кэш-память | Частота системной шины | Мощность (Вт) | Стоимость |
|---------------------|------------------|-----------------|------------|------------------------|---------------|-----------|
| | | | | | | |
| | | | | | | |

По результатам сравнения напишите отчет:

- 1) Какие существуют различия между выбранными моделями процессоров?
- 2) Какой из них эффективнее использовать?
- 3) Определите причину разницы в стоимости процессора.

8

Принципы кодирования текстовой информации



Как осуществляется кодирование текстовой информации?



- ✓ вспомните, что такое код, шифр.
- ✓ Какова значимость шифрования или кодирования в жизни?
- ✓ В каких областях применяются шифры, коды?



Новые знания

Человек быстро и легко воспринимает текст. Наш мозг ежедневно обрабатывает большое количество текстовой информации. В отличие от человека, компьютерный процессор может обрабатывать только цифровую информацию. Поэтому текстовые данные преобразовываются в цифровую информацию при передаче в процессор компьютера. В предыдущих главах мы познакомились с двоичной системой счисления, с которой работает процессор компьютера.

Кодирование информации

Представление информации с помощью определенного алфавита называется **кодированием**. Код – это система условных знаков для представления информации. Хранение, получение, представление и методы обработки информации будут зависеть от типа фактически предоставленной информации. Кодирование информации иногда называют *шифрованием*, обратное же кодирование прямо связано с процессом *декодирования*. Двоичный алфавит представлен знаками 0 и 1. Чтобы отобразить информацию в двоичном коде, устройство должно различать два состояния. Например, на устройство поступает ток – это (1). Каждая цифра двоичного кода несет количество информации, равное одному *биту*.

Таблица кодов ASCII

Чтобы выразить текст числом, каждая буква сопоставляется с числовым значением. Кодировать все буквы, присваивая им свое значение. Но для всех пользователей должен быть единый стандарт. Самым распространенным видом кодирования является код ASCII. В современных компьютерах информация передается кодом ASCII (American Standard Code for Information Interchange – американский стандартный код для обмена информацией). Код ASCII был разработан в США Американским Национальным Институтом Стандартизации (ANSI) для кодирования 256 символов, что предусматривает возможность размещения в нем национальных символов с помощью специальных программ. Множество комбинаций кодирования символов называют **таблицей кодов**. Таблица кодов ASCII состоит из двух частей (*схема 1*).

Таблица кодов ASCII состоит из двух частей

Стандартный

Стандартная часть – от 0 до 127 символов. Сюда входят управляющие компьютером символы, строчные и прописные буквы латинского алфавита, десятичные цифры, знаки препинания, пробел и другие символы.

Альтернативный

Альтернативная часть – от 128 до 255 символов. Сюда входят национальные алфавиты (русский, казахский). Можно также кодировать в ней перевод десятичных, двоичных и шестнадцатеричных систем счисления по таблице соответствия.

Схема 1. Стандартная и альтернативная части таблицы кодов ASCII

Первоначально в кодах ASCII было 7 бит информации. В последующем ее расширили до 8-битной (1 байт) кодировки. Объем 7-битного кодирования по сравнению с 8-битным в 2 раза меньше. $2^7=128 < 2^8=256$. Первоначально в ASCII было возможно хранение 128 символов (рис. 1).

| ASCII control characters | | | | ASCII printable characters | | | | Extended ASCII characters | | | | | | | | | |
|--------------------------|------|-----------------------|--|----------------------------|-------|----|---|---------------------------|---|-----|---|-----|---|-----|---|-----|------|
| 00 | NULL | (Null character) | | 32 | space | 64 | @ | 96 | ` | 128 | Ç | 160 | à | 192 | Ł | 224 | Ó |
| 01 | SOH | (Start of Header) | | 33 | ! | 65 | A | 97 | a | 129 | ü | 161 | í | 193 | ± | 225 | ß |
| 02 | STX | (Start of Text) | | 34 | " | 66 | B | 98 | b | 130 | é | 162 | ó | 194 | ⌈ | 226 | Ô |
| 03 | ETX | (End of Text) | | 35 | # | 67 | C | 99 | c | 131 | â | 163 | ú | 195 | ⌋ | 227 | Õ |
| 04 | EOT | (End of Trans.) | | 36 | \$ | 68 | D | 100 | d | 132 | ä | 164 | ñ | 196 | — | 228 | ö |
| 05 | ENQ | (Enquiry) | | 37 | % | 69 | E | 101 | e | 133 | à | 165 | Ñ | 197 | † | 229 | Ï |
| 06 | ACK | (Acknowledgement) | | 38 | & | 70 | F | 102 | f | 134 | á | 166 | ª | 198 | ‡ | 230 | µ |
| 07 | BEL | (Bell) | | 39 | ' | 71 | G | 103 | g | 135 | ç | 167 | º | 199 | Ä | 231 | þ |
| 08 | BS | (Backspace) | | 40 | (| 72 | H | 104 | h | 136 | ê | 168 | ¿ | 200 | ℒ | 232 | ƒ |
| 09 | HT | (Horizontal Tab) | | 41 |) | 73 | I | 105 | i | 137 | ë | 169 | © | 201 | ℓ | 233 | ú |
| 10 | LF | (Line feed) | | 42 | * | 74 | J | 106 | j | 138 | è | 170 | ¬ | 202 | ⌌ | 234 | Û |
| 11 | VT | (Vertical Tab) | | 43 | + | 75 | K | 107 | k | 139 | í | 171 | ½ | 203 | ⌍ | 235 | Ü |
| 12 | FF | (Form feed) | | 44 | , | 76 | L | 108 | l | 140 | î | 172 | ¾ | 204 | ⌎ | 236 | ý |
| 13 | CR | (Carriage return) | | 45 | - | 77 | M | 109 | m | 141 | ï | 173 | ¸ | 205 | ≡ | 237 | ÿ |
| 14 | SO | (Shift Out) | | 46 | . | 78 | N | 110 | n | 142 | ä | 174 | « | 206 | ≠ | 238 | — |
| 15 | SI | (Shift In) | | 47 | / | 79 | O | 111 | o | 143 | Å | 175 | » | 207 | ð | 239 | · |
| 16 | DLE | (Data link escape) | | 48 | 0 | 80 | P | 112 | p | 144 | É | 176 | ¸ | 208 | ö | 240 | ≡ |
| 17 | DC1 | (Device control 1) | | 49 | 1 | 81 | Q | 113 | q | 145 | æ | 177 | ¸ | 209 | ø | 241 | ± |
| 18 | DC2 | (Device control 2) | | 50 | 2 | 82 | R | 114 | r | 146 | Æ | 178 | ¸ | 210 | È | 242 | — |
| 19 | DC3 | (Device control 3) | | 51 | 3 | 83 | S | 115 | s | 147 | ó | 179 | ¸ | 211 | É | 243 | ¼ |
| 20 | DC4 | (Device control 4) | | 52 | 4 | 84 | T | 116 | t | 148 | ô | 180 | ¸ | 212 | Ê | 244 | ½ |
| 21 | NAK | (Negative acknowl.) | | 53 | 5 | 85 | U | 117 | u | 149 | õ | 181 | À | 213 | Ë | 245 | ¾ |
| 22 | SYN | (Synchronous idle) | | 54 | 6 | 86 | V | 118 | v | 150 | ö | 182 | Á | 214 | Ì | 246 | ¸ |
| 23 | ETB | (End of trans. block) | | 55 | 7 | 87 | W | 119 | w | 151 | ï | 183 | Â | 215 | Í | 247 | ¸ |
| 24 | CAN | (Cancel) | | 56 | 8 | 88 | X | 120 | x | 152 | ÿ | 184 | Ã | 216 | Î | 248 | ¸ |
| 25 | EM | (End of medium) | | 57 | 9 | 89 | Y | 121 | y | 153 | ÿ | 185 | Ä | 217 | Ï | 249 | ¸ |
| 26 | SUB | (Substitute) | | 58 | : | 90 | Z | 122 | z | 154 | ÿ | 186 | Å | 218 | Ð | 250 | ¸ |
| 27 | ESC | (Escape) | | 59 | ; | 91 | [| 123 | { | 155 | ÿ | 187 | Ä | 219 | Ñ | 251 | ¸ |
| 28 | FS | (File separator) | | 60 | < | 92 | \ | 124 | | 156 | £ | 188 | Ä | 220 |  | 252 | ¸ |
| 29 | GS | (Group separator) | | 61 | = | 93 |] | 125 | } | 157 | ¸ | 189 | Ä | 221 | ¸ | 253 | ¸ |
| 30 | RS | (Record separator) | | 62 | > | 94 | ^ | 126 | ~ | 158 | ¸ | 190 | ¥ | 222 | ¸ | 254 | ¸ |
| 31 | US | (Unit separator) | | 63 | ? | 95 | _ | | | 159 | f | 191 | ¸ | 223 | ¸ | 255 | nbsp |

Рисунок 1. Таблица стандартной и альтернативной частей кодов ASCII

Кодирование текстовой информации

Если каждому символу алфавита ставится в соответствие определенное число (например, порядковый номер), то текстовую информацию можно закодировать с помощью десятичного кода. В Казахстане используют как

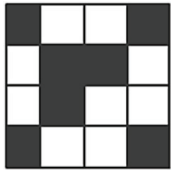
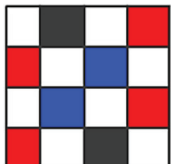
стандартную часть ASCII, включающую прописные и строчные буквы латинского алфавита, цифры, знаки препинания, арифметические знаки, так и альтернативную часть ASCII, куда входят символы кириллицы. Таким образом, если человек создаст текстовый файл и записывает его на диск, то в памяти компьютера каждый символ сохранится как комбинация восьми нулей и единиц. Когда текст выводится на экран или принтер, то будут отображаться символы в соответствии с этими кодами.

| | |
|---|-----------|
| А | 1100 0000 |
| Б | 1100 0001 |
| В | 1100 0010 |

Кодирование графической информации

ЭВМ сохраняет, обрабатывает и передает в системе двоичных кодов графические изображения так же, как и любую другую информацию. Существует достаточное количество различных программ, работающих с графическими изображениями. В них при кодировании графиков используют разные графические форматы.

Расширение сохраненного файла изображения показывает, какой формат использовался. И вы можете узнать, с помощью какого программного обеспечения возможен просмотр, редактирование и вывод на печать. Существует два способа выполнения кодирования изображений – на основе растровой и векторной графики.

| | |
|---|--|
|  | 1 0 0 1 0 1 1 0 0 1 0 0 1 0 0 1 |
|  | 00 11 00 01 01 00 10 00 00 10 00 01 01 00 11 00 |

При использовании растровой графики с помощью определенного числа бит кодируется цвет мельчайшего элемента изображения. Изображение представляется в виде большого числа мелких точек, называемых пикселями. Как в мозаике, собранной из камней или стекол, так из разноцветных точек получается рисунок. При использовании в ЭВМ растрового способа каждому пикселю выделяется определенное количество битов, которое называется битовой глубиной. Каждому цвету соответствует один двоичный код. Например, если битовая глубина равна 1, то 0 будет соответствовать белому цвету, а 1 – черному. Изображение будет черно-белым. Если битовая глубина равна 2, то каждому пикселю нужно 2 бита. В этом случае, 00 обозначает белый цвет, 01 – пурпурный, 10 – сине-зеленый и 11 – черный. Изображение будет четырехцветным. Если битовая глубина равна 3, то можно изобразить 8 цветов, а если битовая глубина 4, то, соответственно, 16 цветов. Таким образом, с помощью графических программ можно формировать изображения с 2, 4, 8, ..., 256 цветами. При увеличении количества цветов также растет объем памяти, занимаемой изображением. Это основной недостаток растровой графики.

Кодирование аудио и видеоинформации

С 1990-х годов персональные компьютеры получили возможность работать с звуковой информацией. Каждый компьютер, в котором имеются звуковая плата, микрофон и динамики, способен записывать, сохранять и редактировать звуковую информацию.

Процесс преобразования звуковой информации на компьютере



В настоящее время используется AVI (Audio-Video Interleaved) формат. Основные мультимедийные форматы AVI и WAV предъявляют высокие требования к памяти. Вот почему на практике используются различные методы сжатия звука и видео. Современный стандартный способ сжатия – MPEG (Moving Pictures Experts Group). Стандарт MPEG-1 – известный в настоящее время стандарт сжатия видео и аудио. Например, MP3 формат занимает объем в десять раз меньше, чем WAV формат, качество звука не отличается. Существует несколько программ, преобразующих WAV формат в MP3 формат. Стандарт MPEG-2 предназначен для качественного сжатия видеозаписей, телевизионных изображений и стереозвуков. Стандарты MPEG-4 дают возможность цветные полнометражные фильмы со звуковым сопровождением записать на компакт-диски с невысоким качеством и небольшим объемом.

Кодирование Юникод

Известно, что один байт состоит из 8 битов. С помощью 8 битов можем закодировать 256 возможных символов. Таким образом, в ASCII кодируются 256 символов. Но в некоторых алфавитах, например, китайском и японском, количество символов превышает 256. Как быть в таких случаях?

Для решения этой проблемы в 90-х годах прошлого века было создано кодирование Юникод. Первоначально коды Юникод вмещали 2 байта, то есть 16 битов. 16 битами можно закодировать $2^{16} = 65536$ символов. Можно отметить два самых распространенных вида Юникода. Для Интернета используется версия UTF-8, а для Windows – UTF-16 (рис. 2).

UTF-8 кодирует первые 128 символов 1 байтом, а остальные – 2 или 4 байтами. Отличительная особенность Юникода – с его помощью кодируются наряду с научно-техническими, математические, экономические символы, ноты и тому подобное.

В Windows есть два способа записи текста. Первый – с помощью кода UTF-16. Второй – с помощью дополнения, называемого уровнем Юникода, который обеспечивает поддержку Юникода на указанных старых платформах. Чтобы ознакомиться с таблицей кодирования в ASCII и UTF-16, надо –

выполнить команду в текстовом процессоре Word **Вставка – Символ – Другие символы**.

| | | | | | | | | | | | | | | | | |
|----|------|--------|--------|--------|--------|--------|--------|--------|--------|--------|------|--------|------|------|------|------|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| | D2B0 | D292 | E2809A | D293 | E2809E | E280A6 | E280A0 | E280A1 | E282AC | E280B0 | D3A8 | E280B9 | D2A2 | D29A | D2BA | D2AE |
| 80 | Ұ | Ғ | , | Ғ | „ | ... | † | ‡ | € | ‰ | Ө | ‹ | Ң | Қ | Һ | Ү |
| | D2B1 | E28098 | E28099 | E2809C | E2809D | E280A2 | E28093 | E28094 | | E284A2 | D3A9 | E280BA | D2A3 | D29B | D2BB | D2AF |
| 90 | Ү | ‘ | ’ | “ | ” | • | – | — | □ | ™ | е |) | Ң | Қ | Һ | Ү |
| | C2A0 | D08E | D19E | D296 | C2A4 | D2B2 | C2A6 | C2A7 | D081 | C2A9 | D084 | C2AB | C2AC | C2AD | C2AE | D087 |
| A0 | | У | ұ | Ж | # | Х | ! | § | Ё | © | € | « | ¬ | - | ® | İ |
| | C2B0 | C2B1 | D086 | D196 | D2B3 | C2B5 | C2B6 | C2B7 | D191 | E28496 | D194 | C2BB | D297 | D398 | D399 | D197 |
| B0 | ° | ± | ı | ı | х | μ | ¶ | · | ё | № | € | » | Ж | Ө | ә | İ |
| | D090 | D091 | D092 | D093 | D094 | D095 | D096 | D097 | D098 | D099 | D09A | D09B | D09C | D09D | D09E | D09F |
| C0 | A | B | B | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П |
| | D0A0 | D0A1 | D0A2 | D0A3 | D0A4 | D0A5 | D0A6 | D0A7 | D0A8 | D0A9 | D0AA | D0AB | D0AC | D0AD | D0AE | D0AF |
| D0 | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я |
| | D0B0 | D0B1 | D0B2 | D0B3 | D0B4 | D0B5 | D0B6 | D0B7 | D0B8 | D0B9 | D0BA | D0BB | D0BC | D0BD | D0BE | D0BF |
| E0 | a | b | B | г | д | e | ж | з | и | й | к | л | м | н | о | п |
| | D180 | D181 | D182 | D183 | D184 | D185 | D186 | D187 | D188 | D189 | D18A | D18B | D18C | D18D | D18E | D18F |
| F0 | р | с | т | у | ф | х | ц | ч | ш | щ | ъ | ы | ь | э | ю | я |

Рисунок 2. Символы казахского алфавита в коде UTF-16



Анализ



Сравните таблицы шрифтов кодирования ASCII и Юникод. Какие есть различия? Уточните.



Синтез



Выполняя команду в Word **Вставка – Символы – Шрифт**, составьте ре-бус из символов шрифтов Webdings, Wingdings.



Оценка



Какое значение имеет кодирование информации в технике и в жизни?



1. Что такое кодирование информации?
2. Что представляет собой код ASCII и для чего он используется в компью-тере? Какие бывают виды?
3. Для чего нужно кодирование Юникод? Какие существуют виды?
4. Каковы различия при кодировании текстовой, графической и звуко-вой информации?



| Тема исследования | Основные направления исследования |
|---|---|
| Основные принципы кодирования информации в компьютере | <ul style="list-style-type: none"> ✓ о криптографии и кодировании ✓ о коде ASCII ✓ о коде Юникод ✓ различия в таблицах кодирования ASCII и Юникод |

1 - 2

Пользовательские функции и процедуры



Для чего в языке программирования мы используем функции и процедуры?



- ✓ Знакомы ли вам понятия процедура и функция?
- ✓ Как вы думаете для чего их используют в программе?
- ✓ Можно ли не использовать процедуру при разработке программ?
- ✓ Какие имеются преимущества использования процедуры при написании программы на ваш взгляд?



Новые знания

Одна из распространенных ситуаций в программировании заключается в том, что во время выполнения программы определенные вычисления могут многократно применяться для расчетов с разными значениями. Но нет необходимости записывать такие вычисления много раз, а достаточно записать их в отдельную часть программы. Внутренняя программа выполняет некоторый алгоритм и доступ к ней можно получить из любой части основной программы. **Подпрограммы** – это процедуры и функции, предназначенные для выполнения специальных операций.

В языке Python, как и во всех языках программирования, имеются процедуры и функции: *стандартные*, т.е. определенные системой, и *нестандартные* (собственные), т.е. определенные самим программистом. Стандартные функции и процедуры в качестве элементов языка программирования обозначаются служебными словами, определенными в языке программирования. А нестандартные функции и процедуры записываются своими именами, заданными программистом, и включают определенный порядок действий.

Результата можно достигнуть как процедурами, так и функциями. Однако между ними существуют различия. Процедура выполняет какую-либо операцию и затем возвращает результат в список своих параметров. Функция выполняет ту же последовательность, только возвращает результат в виде некоторого своего значения. Таким образом, функция – более универсальный объект.

Независимая часть программы, которая должна быть вызвана заданным именем для выполнения определенных действий, называется **процедурой**. Процедуры делятся на процедуры с параметром и без параметров.

Процедура должна быть определена к моменту ее вызова. Определение процедуры и функции начинается со служебного слова **def. define**, (определить значение). При использовании процедуры в программе вызов осуществляется по имени с указанием соответствующих параметров. Это

место в программе называется *точкой вызова процедуры*. При вызове процедуры начинается ее исполнение. Вызов процедуры осуществляется по ее имени, за которым следуют круглые скобки, например, **SUMMA ()** или полная запись **def SUMMA ()**.

Использование процедур сокращает код и повышает удобочитаемость. Рассмотрим пример на использование процедур без параметров.

Пример 1. Дано целое число. Определите, каким оно является – положительным или отрицательным.

```
Процедура без параметров (код 1)
def Знак (): #Определение процедуры без параметров
    print ( "Отрицательное число" )
#Раздел основной программы
n = int ( input (Введите число=') )
if n < 0:
    Знак () #Вызов процедуры
else:
    print ( "Положительное число" )
```

Пример 2. Дана длина и ширина прямоугольника. Используйте процедуру с параметрами, чтобы вычислить площадь и периметр этого прямоугольника (код 2).

Рассмотрим примеры работы процедуры с параметром.

```
Процедура с параметром (код 2)
def Площадь (k,p): #Процедура вычисления площади)
    print('площадь=',k*p)
def Периметр (k,p): #Процедура вычисления периметра
    print ('периметр=',2*(k+p))
#Раздел основной программы
a = int ( input('Введите длину=') )
b = int ( input('Введите ширину=') )
Площадь (a,b)
Периметр(a,b)
```

Как вы видите в программном коде 2, процедуру `def Площадь (k, p)` вызвали в программе `Площадь (a, b)`. Процедуру можно вызвать и с конкретными значениями длины и ширины `Площадь (4, 5)`. 4 и 5 в круглых скобках считаются параметрами.

Примечание. Сколько указано переменных в вызываемой процедуре с параметром, столько переменных должна иметь процедура.

Как уже упоминалось выше, процедура может не иметь параметров, тогда они будут работать с переменными основной программы. В процедуре и функции есть понятие **локальная переменная**. Локальная переменная

(внутренняя) известна только на уровне процедуры. К этому типу переменных нельзя обратиться из основной программы и из других процедур. Они могут использоваться только при вызове подпрограммы, содержащей эту переменную. В процедуре можно использовать ключевое слово **global** для работы со всеми переменными программы (код 3).



Применение

Практическая работа

Пример 3. Задан одномерный массив $A(N)$. Найдите сумму цифр для каждого элемента массива (код 3, рисунок 1).

Код 3

```
from random import randint
#Процедура вычисления суммы чисел
def summazifr(t):
#Преобразует s в глобальную
переменную
    global s
    s=0
#Вычисляет сумму чисел
    while t != 0:
        k=t%10
        s+=k
        t=t//10
A=[]
N = int(input('Введите длину массива='))
A = [0]*N
print(Сумма цифр)
for i in range(0,N):
    A[i] = randint(1, 1000)
#Вызывает процедуру с параметром
summa zifr(A[i])
print('A[',i,']=', A[i], '=>', s, sep=' ')
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Wi
Python 3.7.2 (tags/v3.7.2
(AMD64) on win32
Type "help", "copyright",
>>>
==== RESTART: C:/Users/Us
Введите длину массива=10
Сумма цифр
A[ 0 ]= 936 => 18
A[ 1 ]= 261 => 9
A[ 2 ]= 369 => 18
A[ 3 ]= 272 => 11
A[ 4 ]= 609 => 15
A[ 5 ]= 97 => 16
A[ 6 ]= 260 => 8
A[ 7 ]= 685 => 19
A[ 8 ]= 249 => 15
A[ 9 ]= 824 => 14
>>>
```

Рисунок 1



Анализ



Запишите код 3 без процедуры. Проанализируйте код программы с процедурой и без процедуры. Во время анализа выясните:

- 1) Были ли различия в процессе выполнения или вывода результата в этих двух программах? Объясните, какие и почему.
- 2) Какой из кодов эффективнее? Почему? Приведите обоснованные примеры.



Синтез



Составьте задачу на использование процедуры. Запишите программный код вашей задачи и представьте его.



Оценка



Какова эффективность использования процедуры в процессе программирования? Какие дополнительные возможности она предоставляет вам? Оцените это!



- 1) Что такое подпрограммы?
- 2) В какой части программы используются локальные переменные?
- 3) Как ввести процедуру в программу?
- 4) Что представляют собой процедуры с параметром и без параметра?
- 5) Как бы вы объяснили работу процедуры?



Задания

Решите предложенные задачи с помощью процедуры и разработайте программный код.

Задание №1. Наименьшее число*

Даны целые числа a, b, c ($1 < a, b, c < 10^9$). Найдите наименьшее из них.

| № | Пример ввода | Результат |
|---|--------------|-----------|
| 1 | 246 809 | 246 |

Задание №2. Количество разрядов *

Даны целые числа a, b, c ($1 < a, b, c < 10^9$). Найдите количество разрядов этих чисел.

| № | Пример ввода | Результат |
|---|----------------|-----------|
| 1 | 1246 809 32589 | 4 3 5 |

Задание №3. Исследование треугольника *

Даны целые числа a, b, c ($1 < a, b, c < 10^9$). Используя эти числа, можно получить три прямоугольника. Для каждого образованного прямоугольника найдите площадь и периметр. Запишите результат в следующем порядке – стороны прямоугольника, его площадь и периметр.

| № | Пример ввода | Результат |
|---|--------------|---------------------------------------|
| 1 | 10 4 6 | 10 4 40 28 4 6 24 20 10 6 60 32 |

Задание №4. Двоичное число **

Даны целые числа a, b ($1 < a, b < 10^3$). Запишите эти числа в двоичной системе счисления.

| № | Пример ввода | Результат |
|---|--------------|---------------|
| 1 | 24 89 | 11000 1011001 |

Задание №5. Наибольшая сумма**

Даны целые числа a, b, c ($1 < a, b, c < 10^9$). Определите число с наибольшей суммой цифр в составе чисел.

| № | Пример ввода | Результат |
|---|----------------|-----------|
| 1 | 4116 809 14404 | 809 |

Задание №6. Простые числа ***

Задана одномерная таблица $A[10]$. Определите, сколько простых чисел в этой таблице. Выведите количество простых чисел.

| № | Пример ввода | Результат |
|---|-------------------------------|-----------|
| 1 | 45 68 13 31 4 15 73 97 78 100 | 4 |

Задание №7. Нечетные числа ***

Количество учащихся 10-го класса n ($n < 100$) и их дни рождения a_i даны в линейной таблице. У скольких учеников дни рождения состоят только из нечетных цифр? Определите.

| № | Пример ввода | Результат | Пояснение |
|---|--------------------|-----------|-----------|
| 1 | 5 11 25 30 15 7 | 3 | 11 15 7 |



Как используют функцию на языке программирования?



- ✓ Что такое функция? Как ты ее понимаешь?
- ✓ Каково значение функции в математике и в жизни?
- ✓ Как ты думаешь, в чем назначение функции в программе?



Новые знания

Структура функции

Функция – это часть программы, которую можно использовать без ограничений в любом месте программы. Определение функции может сформулировать следующим образом.

Независимая часть программы, которая вызывается с помощью заранее определенного имени в процессе программирования, называется **функцией**. Функция состоит из трех частей: **имени, аргумента и тела функции**.

Например, рассмотрим простейшую функцию:

```
def testfunc (myname):  
    print('Привет, %s' % myname)  
testfunc('Карина')
```

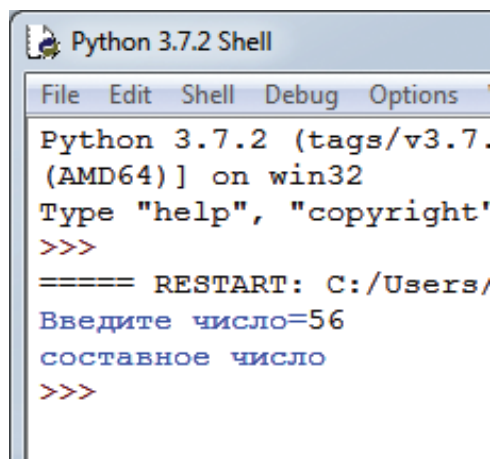
В этом примере *testfunc* – это **имя функции**, а *myname* – ее **аргумент**. Тело функции начинается с `def`. Аргумент – это переменная, которая участвует в работе функции. В большинстве случаев, после вычисления функции, ее значение необходимо вернуть в основную программу. Команда **Return** используется для завершения функции и возврата ее значения в основную программу.

Пример 1. Дано натуральное число *N*. С помощью функции определите является ли это число простым или составным. Если число простое, то выведите на печать 1, иначе 0 (код 1, рис. 1).

Если вы посмотрите на программный код примера 1, то заметите, что между программным кодом при использовании процедуры и функции раз-

Функция (код 1)

```
import math
def простое_число(K):
    p=0
    for i in range(2, round(math.sqrt(K))):
        if K%i==0:
            p=1
    if p==0:
        print('простое число')
    else:
        print('составное число')
    return
N = int ( input('Введите число=') )
if N ==4 or N ==6 :
    print ('составное число')
else:
    простое_число(N)
```



```
Python 3.7.2 Shell
File Edit Shell Debug Options
Python 3.7.2 (tags/v3.7.
(AMD64)] on win32
Type "help", "copyright"
>>>
===== RESTART: C:/Users/
Введите число=56
составное число
>>>
```

Рисунок 1

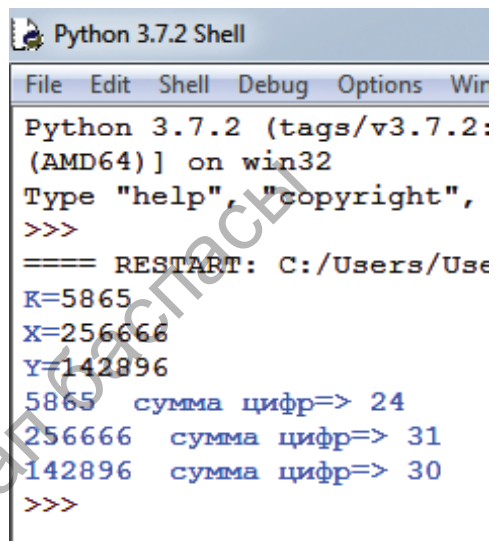
ница небольшая. Чтобы проиллюстрировать отличие функции от процедуры, рассмотрим следующий пример.

Пример 2. Заданы числа **K, X, Y**. Вычислите сумму цифр, составляющих эти числа и выведите на печать (код 2, рис. 2).

Как видно из программного кода, функция возвращает результат в переменную K в основной программе после вычисления суммы цифр. Еще одна

Код 2

```
def sumDigits(n):
    sum = 0
    while n!= 0:
        sum += n % 10
        n = n // 10
    return sum
#Код основной программы
K= int(input('K='))
X= int(input('X='))
Y= int(input('Y='))
print (K,' сумма цифр=>',sumDigits(K))
print (X,' сумма цифр=>',sumDigits(X))
print (Y,' сумма цифр=>',sumDigits(Y))
```



```
Python 3.7.2 Shell
File Edit Shell Debug Options Win
Python 3.7.2 (tags/v3.7.2:
(AMD64)] on win32
Type "help", "copyright",
>>>
===== RESTART: C:/Users/Use
K=5865
X=256666
Y=142896
5865  сумма цифр=> 24
256666  сумма цифр=> 31
142896  сумма цифр=> 30
>>>
```

Рисунок 2

особенность использования функции – то, что ее можно использовать как некоторую величину $\text{sumDigits}(K)$. Например, если вычисляется выражение $\text{sumDigits}(K) + 100$, то к сумме цифр добавляется 100.



Применение

Практическая работа

Пример 3. Используйте функцию в программе для вычисления наибольшего из чисел a, b, c (код 3, рисунок 3).

Код 3

```
#Функция нахождения наибольшего
def Max( m,k,p ):
    max=m
    if max< k:
        max=k
    if max< p:
        max=p
    return max
#Код основной программы
a= int(input('a='))
b= int(input('b='))
c= int(input('c='))
print ('Наибольшее=>',max(a,b,c))
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Win
Python 3.7.2 (tags/v3.7.2:
(AMD64) ] on win32
Type "help", "copyright",
>>>
== RESTART: C:/Users/User/
a=65
b=2669
c=233
Наибольшее=> 2669
>>>
```

Рисунок 3



Анализ



Проанализируйте код программы с использованием функции и процедуры. Во время анализа:

- 1) Определите различие и сходство в работе функции и процедуры.
- 2) Что из них эффективнее? Почему? Аргументируйте свою точку зрения.



Синтез



Используя функцию, составьте код программы, который определяет минимальное (наименьшее) число из четырех заданных чисел:

$\text{min}=\{t,p,q,l\}$



Оценка



Оцените назначение и использование функции в сравнении с процедурами в программировании.



- 1) Какое назначение функции в программировании?
- 2) Как ввести функцию в программу?
- 3) В чем назначение команды Return?
- 4) В чем основное различие функции и процедуры?



Задания

Используйте функцию при выполнении каждого задания.

Задание №1. Сумма*

Даны целые числа a, b, c, d ($1 < a, b, c, d < 10^3$). Найди наименьшее среди этих чисел. Вычислите сумму чисел между a и b между c и d .

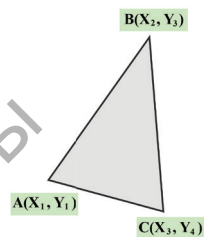
| № | Пример ввода | Результат |
|---|----------------|-------------|
| 1 | 15 45 60 80 | 930 1470 |

Задание №2. Вычислить длину сторон треугольника **

Координаты вершин треугольника заданы в целых числах $X_1, Y_1, X_2, Y_2, X_3, Y_3$. Вычислите длину сторон треугольника.

Примечание: чтобы вычислить стороны треугольника, воспользуйтесь формулой, вычисляющей расстояние между двумя точками (Геометрия, 8 класс).

| № | Пример ввода | Результат |
|---|-------------------|-------------|
| 1 | 2 1 2 5 5 1 | 4 3 5 |



Задание №3. Факториал **

Составьте программу вычисления $K!$ факториала ($0 < K < 20$).
 $K! = 1 * 2 * \dots * (K-1) * K$

| № | Пример ввода | Результат |
|---|--------------|-----------|
| 1 | 5 | 120 |

Задание №4. Восьмиричная система счисления ***

Даны целые числа a, b, c ($1 < a, b, c < 10^9$). Запишите эти числа в восьмеричной системе счисления.

| № | Пример ввода | Результат |
|---|--------------|-----------|
| 1 | 324 686 | 504 1256 |

Задание №5. НОД **

Даны целые числа a, b ($1 < a, b < 10^9$). Найдите наибольший общий делитель этих цифр (НОД).

| № | Пример ввода | Результат |
|---|--------------|-----------|
| 1 | 32 48 | 16 |

Задание №6. Простые числа ***

Задан одномерный массив $A[N]$. Среди элементов массива найдите элемент, содержащий наибольшее количество нулей ($0 < a_i < 10^9$).

| № | Пример ввода | Результат |
|---|---|-----------|
| 1 | 5 1520 3256 10003 50056 1850 | 10003 |

Задание №7. НОК **

Даны целые числа a, b ($1 < a, b < 10^9$). Найдите наибольшее общее кратное этих чисел (НОК).

| № | Пример ввода | Результат |
|---|--------------|-----------|
| 1 | 9 12 | 36 |



Как организовать рекурсию, используя функции и процедуры?



- ✓ Что такое рекурсия?
- ✓ В каких разделах математики возможно применение рекурсии?
- ✓ Могут ли функции и процедуры вызывать сами себя?



Новые знания

Рекурсия – очень распространенное явление, которое встречается не только в науке, но и в повседневной жизни. Например, вы можете включить и навести web-камеру на экран монитора компьютера, камера будет записывать изображение экрана компьютера и выводить его же на этот экран. Другими словами, получится замкнутый цикл. Чтобы определить рекурсию, нужно знать рекуррентную формулу, условия остановки рекурсии. Рекурсия очень похожа на метод математический индукции.

Рекурсия – вычислительный процесс, направленный на решение задачи с определением части функции (процедуры) через саму себя (рис. 1).

Рекурсия является одним из основных понятий в математике и информатике. Найдено решение некоторых сложных задач с использованием рекурсии в процессе программирования. Многочисленные олимпиадные задачи по предмету «Информатика» решаются с использованием рекурсии.

Рекурсия может повторяться конечное или неограниченное количество раз. Для повторения рекурсии конечное число раз должно быть некоторое условие, по которому произойдет остановка. Неограниченная рекурсия не может быть остановлена во время работы программы. Выполнение такой рекурсии или работу программы можно остановить, лишь сняв задачу через Диспетчер задач. В процессе программирования рекурсия может быть задана с помощью функции или процедуры.

Давайте рассмотрим пример, чтобы увидеть, как выполняется рекурсия.

Пример 1. Числа Фибоначчи – это ряд чисел, в котором каждое следующее число равно сумме двух предыдущих: 1, 1, 2, 3, 5, 8, 13, ...

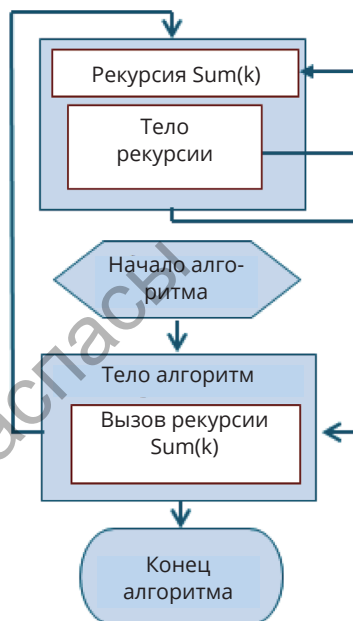


Рисунок 1

Создание программы с рекурсией для вычисления чисел Фибоначчи (код 1, рисунок 2).

Запись рекуррентных отношений последовательности Фибоначчи:

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n \geq 0$$

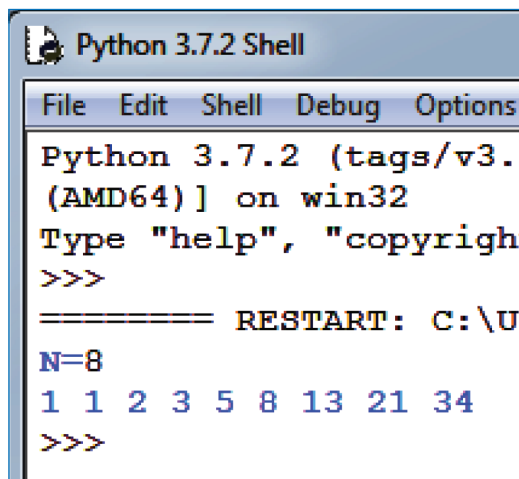
Основные особенности работы рекурсии представлены и описаны в коде 1.

Пример 2. Вычислите сумму цифр заданного числа с помощью рекурсии (код 2, рисунок 3).

Как вы видите в программном коде 2, после вызова функции **sumDig** часть программы выполняется рекурсивно (**sum += sumDig (n // 10** данная строка программы), то есть функция вызывает саму себя.

Код 1

```
#Рекурсивная функция
def Fib(p,q):
    global i,n #Глобальные переменные
    t=p+q; i+=1
    print(t, end=' ')
    if i< n:
        Fib(q,t) #Рекурсивный вызов
#Основная программа
n= int(input('N='))
x=1; y=1; i=0
print(1, 1,end=' ')
Fib(x,y); #Вызов функции
```

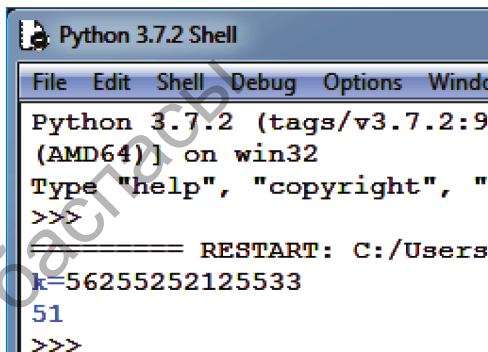


```
Python 3.7.2 Shell
File Edit Shell Debug Options
Python 3.7.2 (tags/v3.7.2:1
(AMD64)] on win32
Type "help", "copyright"
>>>
===== RESTART: C:\U
N=8
1 1 2 3 5 8 13 21 34
>>>
```

Рисунок 2

Код 2

```
def sumDig ( n ):
    sum = n % 10
    if n >= 10:
        sum += sumDig ( n // 10 )
    return sum
k= int(input('k='))
print(sumDig (k));
```



```
Python 3.7.2 Shell
File Edit Shell Debug Options Windo
Python 3.7.2 (tags/v3.7.2:9
(AMD64)] on win32
Type "help", "copyright", "
>>>
===== RESTART: C:/Users
k=56255252125533
51
>>>
```

Рисунок 3



Пример 3. Используйте функции для написания программы, которая находит наибольшее из чисел a , b , c (код 3, рисунок 3).

Код 3

```
#Функция нахождения наибольшего
def Max( m,k,p ):
    max=m
    if max< k:
        max=k
    if max< p:
        max=p
    return max
#Основная программа
a= int(input('a='))
b= int(input('b='))
c= int(input('c='))
print ('Наибольшее число=>',max-
(a,b,c))
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window
Python 3.7.2 (tags/v3.7.2:9a
(AMD64)] on win32
Type "help", "copyright", "c
>>>
==== RESTART: C:/Users/User/
a=2563
b=3523
c=25455
Наибольшее число=> 25455
>>>
```

Рисунок 4



Анализ



Выполните анализ организации рекурсии в программе получения последовательности Фибоначчи. Сформулируйте условие задачи по аналогии, чтобы можно было организовать рекурсию, а затем составьте и запустите код программы с рекурсивной функцией.



Оценка



Напишите эссе о назначении, значимости рекурсии и оценке ее роли в программировании.



- 1) Что такое рекурсия?
- 2) Приведите пример рекурсии из повседневной жизни.
- 3) Как задать рекурсию в программе?



Задания

Используйте рекурсивную функцию или процедуру для решения задач. Выполните с помощью функции.

Задание №1. Факториал *

Задано натуральное число N ($1 < N < 50$). С помощью рекурсии вычислите его факториал.

| № | Пример ввода | Результат |
|---|--------------|-----------|
| 1 | 6 | 120 |

Задание №2. Сумма**

Заданы первый член и разность арифметической прогрессии. Вычислите n -ый член прогрессии, сумму первых n -членов прогрессии. Используйте рекурсию. Первый член прогрессии $a_1=2$, разность $d=2$.

| Пример ввода | Результат | Пояснение |
|--------------|--------------------------|-----------|
| 5 | 5 член = 10 Summa= 30 | $n=5$ |

Задание №3. Время*

Используя рекурсию, напишите программу, определяющую по заданному времени h часов, m минут и s секунд, сколько времени прошло в секундах от начала дня ($0 \leq h \leq 23$, $0 \leq m \leq 59$, $0 \leq s \leq 59$).

| Пример ввода | Результат |
|--------------|-----------|
| 2 30 40 | 9040 |

Задание №4. Определение ***

Определите функцию заданных ниже программ. Что выдаст программа в результате выполнения, если в программе $K = 6808934008$.

а)

```
def Rec(n):
    global s
    p = n % 10
    if p == 0:
        s += 1
    if n != 0:
        Rec ( n // 10 )
    return s-1
k= int(input('k='))
s=0
print(Rec(k));
```

б)

```
def ZZZ(n):
    global z
    p = n % 10
    if p >= 5 and p % 2 == 0:
        z += 1
    if n != 0:
        ZZZ ( n // 10 )
    return z
k= int(input('k='))
z=0
print(ZZZ(k));
```

Работа со строками



Какие строковые функции и процедуры используются для редактирования текста в программировании?



- ✓ Что такое функция, процедура? Вспомни!
- ✓ Почему операторы для обработки строковых данных делятся на процедуры и функции?
- ✓ Как вы думаете, какие имеются между ними различия?
- ✓ С каким из методов в программировании для обработки текстовой информации вы знакомы?



Новые знания

Символы в Python состоят из цифр, букв и других символов таблицы ASCII. Место символа в этой таблице называется кодом символа. Таблица ASCII содержит 256 символов с номерами от 0 до 255. Например, в таблице ASCII число 65 соответствует букве «А», т.е. код буква «А» равен 65.

В программировании последовательность символов называется **строкой**. Строки и символы заключаются в скобки или апострофы. Максимальное количество символов в строке не ограничено, только важен объем доступной памяти.

Язык программирования Python имеет специальные функции и методы, которые работают с символами, строками. Теперь давайте рассмотрим их. Например, пусть `s = 'Казахстан.'` Тогда самая простая программа, которая показывает эту строку на экране, представлена в коде 1.

| Код 1 |
|---------------------------------------|
| <pre>s = ' Казахстан ' print(s)</pre> |
| Результат |
| Казахстан |

| Код 2 |
|--|
| <pre>s = ' Казахстан ' d = '2050' print(s + ' ' + d)</pre> |
| Результат |
| Казахстан 2050 |

Чтобы добавить строку к строке, мы будем выполнять действия, описанные в коде 2. А следующие строки программы (код 3) увеличивают количество строк на `K`. Если `K = 3`, то «УРА!», как видно из программы, повторяется три раза.

| Код 3 |
|---|
| <pre>s = "УРА!" s3 = s*3 print (s3)</pre> |
| Результат |
| УРА! УРА! УРА |

| Код 4 |
|--|
| <pre>s='информатика' print(s[4]) print(len(s)) print(len(s)-7)</pre> |
| Результат |
| <p>p 11 4</p> |

В языке программирования Python каждый символ в строке можно выделить отдельно и использовать (рис. 1). Напишите код 4, по которому символы будут отображаться на экране. Чтобы отобразить символ из любой позиции строки, поместите позицию символа в скобки []. Мы также используем функцию len () для определения длины строки.

| | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|-------|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| и | н | ф | о | р | м | а | т | и | к | а |
| s[0] | s[1] | s[2] | s[3] | s[4] | s[5] | s[6] | s[7] | s[8] | s[9] | s[10] |

Рисунок 1

В Python вы можете распечатать нужную часть строки(срез). Например, используя функцию s [i : j :k], вы можете выбрать символы от i до j с шагом k. s = 'Python', то s [0:5:2] = 'Pto'. Если k не указано, функция записывается как s [i: j] и s [1: 5] = 'ytho'.

Если функция записана как s [k:], то печатаются символы, расположенные в строке после k (код 5). Если функция задана как s [:k], то выводятся символы до k (код 5).

max(s) в строке показывает наибольший по коду в таблице символов ASCII.

min (s) определяет наименьший по коду в таблице ASCII символов в строке (код 5).

Числа могут быть преобразованы в строки с помощью функции **str ()**.

Например, набрав str (985), вы можете использовать строковые функции и методы, чтобы преобразовать

| Код 5 |
|---|
| <pre>s = 'информатика' print('s=', s) print('s[2:7]=', s[2:7]); print('s[2:7:2]=', s[2:7:2]); print('s[5:]=', s[5:]); print('s[:4]=', s[:4]) print('s[:-3]=', s[:-3]); print('s[-3:]=', s[-3:]) print('max(s)=', max(s)) print('min(s)=', min(s))</pre> |
| Результат |
| <pre>s= информатика s[2:7]= форма s[2:7:2]= фра s[5:]= матика s[:4]= инфо s[:-3]= информат s[-3:]= ика max(s)= ф min(s)= а</pre> |

ее в '895'. С помощью этого метода вы можете выделить цифры от числа. Использование функции **float ()** преобразует целое число или строку в вещественное число: например, `float ('265.4')` возвращает (265.4)

Функция **int ()** преобразует объект в целое число (`int ('253') = 253`).

Функция **ord ()** возвращает код указанного символа (`ord ('B') = 66`), а функция `chr ()` возвращает символ по коду: `chr (66) = 'B'`, который соответствует позиции в таблице ASCII.



Применение

Практическая работа

Пример 1. Напечатайте слова из словосочетания *Республика Казахстан* в отдельных строках Республика, Казахстан. Рядом с каждым словом напечатайте его длину.

| Пример ввода | Результат |
|----------------------|------------------------------|
| Республика Казахстан | Казахстан 9 Республика 10 |

Код примера 1

```
s='Республика Казахстан'  
print(s[0:9], len(s[0:9]));  
print(s[10:23], len(s[10:23]));
```

Код примера 2

```
s='Асхат 16'  
a=s[0:5]  
t=int(s[6:8])  
print(a,int(t)+1);
```

Пример 2. Дана строка, в которой записано имя и возраст юноши; 'Асхат 16'. Увеличив возраст Асхата на 1 год, выведите результат на экран (код 2).

| Пример ввода | Результат |
|--------------|-----------|
| Асхат 16 | Асхат 17 |



Синтез



Придумайте задачу, использующую функции языка программирования Python, с которыми вы ознакомились. Предоставьте решение вашей задачи и напишите программу.



- 1) Что такое строка?
- 2) Сколько символов может содержать строка в языке программирования Python?
- 3) Какая функция используется для определения положения символа в таблице ASCII?
- 4) Какие функции используют в Python, чтобы выполнить преобразование числа в строку и наоборот?



Задания

Задание №1. Поменять слова местами*

Строка, состоящая из латинских букв, содержит слова «monitor printer». Переставьте эти слова местами и выведите на печать.

| № | Пример ввода | Результат |
|---|-----------------|-----------------|
| 1 | monitor printer | printer monitor |

Задание №2. Удаление *

Запишите программный код, по которому получатся результаты, перечисленные в следующем примере.

| № | Пример ввода | Результат |
|---|--------------|---|
| 1 | Kilogramm | len=9 Kilo gramm iorm r a log |

Задание №3. Замена числа**

Задана строка "Qazaqstan-2030", состоящая из латинских букв. Замените в строке 2030 год на 2050. Слово и число должны разделяться тире.

| № | Пример ввода | Результат |
|---|----------------|----------------|
| 1 | Qazaqstan-2030 | Qazaqstan-2050 |

Строковые методы



Какие строковые методы (процедуры) используются для редактирования текста на языке программирования?



- ✓ С какими видами строковых функций вы знакомы?
- ✓ Какое различие и сходство имеется между обработкой строк программой и текста человеком? Приведите примеры.



Новые знания

В Python есть много способов для обработки символов и строк. Такие методы называются **строковым методами**. Теперь давайте ознакомимся с основными методами обработки строк (*таблица 1*).

Таблица 1

| Название метода | Выполняемая функция | Примеры использования |
|--------------------------------|--|--|
| <code>s1.count(s)</code> | Определяет, сколько раз подстрока s встречается в строке s1 | <code>s1='информатика'</code> <code>s1.count('a') =>2</code> |
| <code>s1.count(s,[i,j])</code> | Определяет, сколько раз подстрока s встречается в промежутке [i,j] строки s1 | <code>s1='информатика'</code> <code>s1.count('a',6:11) =>2</code> Если не найдена строка из поиска, то будет отображаться -1. |
| <code>s1.find(s)</code> | Определяет, с какой позиции подстрока s встречается в строке s1 в первый раз. | <code>s1='информатика'</code> <code>print(s1.find('a'))=>6</code> то есть, символ «а» в первый раз встретилось на 6 позиции. |
| <code>s1.find(s,[i,j])</code> | Определяет, с какой позиции подстрока s встречается в строке s1 в первый раз в указанном промежутке [i,j] . | <code>s1='информатика'</code> <code>print(s1.find('a',7,11))</code> равен 10 |
| <code>s1.strip()</code> | Удаляет пробелы или указанные символы слева и справа в строке s1 и переходит на следующую строку | <code>ss1='kkkkk'</code> <code>s2=s1.strip()</code> <code>print(s2)</code> |



Пример 1. Дана строка. Слова в этой строке разделены только одним пробелом. Определите, сколько слов в этой строке.

Решение. По условиям задачи вы можете использовать метод `s.count`, чтобы различать слова в строке разделенных одним пробелом. Сколько пробелов в данной строке `a`, если количество слов больше на один количества пробелов (код 1, тест 1)?

| № | Пример ввода | Результат |
|---|----------------------------|-----------|
| 1 | Dckmm mcd fcm kkkkdl dldld | 4 |
| 2 | Dckmm mcd fcm kkkkdl dldld | 2 |

Если в этом примере слова разделены одним или несколькими пробелами (тест 2), тогда для этого случая код 1 программы не будет работать. В этом случае вы должны использовать код 2. В основном алгоритме этого программного кода используется метод `s1.find(s, [i, j])`. Основная идея программы заключается в том, что если между двумя словами один пробел, то считаем слова, иначе считаем количество пробелов.

| Код 1 |
|---|
| <pre>s=input('введите строку') print(s.count("")+1)</pre> |

| Код 2 |
|---|
| <pre>s=input('Введите строку=') t=s.find(' ', 1, len(s)) #Находит первый пробел в строке r=0; k=t; #Присваивает k позицию первого пробела while t != -1: #Цикл останавливается, если не находит пробел t=s.find(' ', t+1, len(s)) #Изменяет промежуток для поиска пробела if t-k != 1: #Проверяет разницу между предыдущим и последующим количеством пробелов r+=1 #При выполнении условия считает k=t #k присваивает количество найденных на данный момент пробелов print(r+1) #Печатает количество слов</pre> |



Анализ



Сравните назначение строковых функций из таблицы 1, проведите анализ. Прокомментируйте особенности каждой из них для обработки строки. Приведите свои примеры.

Алматыкүтүп баспа



Синтез



Составьте один пример для использования строковой функции. Напишите для него программный код и опишите пути решения.



Задания

Задание №1. Буква 'k' **

Дана строка из латинских слов. Встречается ли буква 'k' в данной строке? Если встречается, то напечатайте номер позиции, иначе напечатайте 0. Если буква 'k' встречается более одного раз, тогда напечатайте общее количество встречающихся букв.

| № | Пример ввода | Результат |
|---|----------------------|-----------|
| 1 | Lo hghbhvfdkgtttt | 11 |
| 2 | Yhjllga | 0 |
| 3 | kknhjgvhghkkvthbbygk | 5 |

Задание №2. Составление предложения*

Даны слова, составленные из латинских букв "narod", "Nur-Sultan", "serdce". Составьте предложение со смыслом, разделив слова одним пробелом.

| № | Пример ввода | Результат |
|---|-------------------------------|--------------------------|
| 1 | narod Nur-Sultan serdce | Nur-Sultan serdce naroda |

Задание №3. Поиск буквы 'a' **

Задана строка из латинских букв. Узнайте, сколько букв «a» встречается в этой строке.

| № | Пример ввода | Результат |
|---|-------------------------|-----------|
| 1 | Nur-Sultan gorod mechty | 1 |

Задание №4. Определение **

Задана строка из латинских букв. Узнайте, сколько раз подстрока «rod» встречается в этой строке.

| № | Пример ввода | Результат |
|---|--------------------------------------|-----------|
| 1 | Rodina – rodimay, rodnaya storona | 3 |

Основные строковые методы



Какие строковые методы используются для редактирования текста на языке программирования?



✓ В чем различие между строковыми методами и строковыми функциями?



Новые знания

В предыдущей теме мы рассмотрели некоторые способы обработки строк в Python. Конечно, этих методов недостаточно для полноценной обработки строк при программировании. Теперь давайте продолжим знакомиться с методами, предназначенными для обработки строк, которых в Python достаточно много. Посмотреть их можно по команде **dir(str)**, получить информацию по каждому – **help(строка_метода)**. Это, в свою очередь, облегчает написание программы на языке программирования.

s1.replace(s2,s3, n)

В строке **s1** подстроку **s2** **n** раз заменяет на строку **s3**. Если не указывать **n** в скобках, то есть записать (**s2, s3**), тогда в строке **s1** все подстроки **s2** полностью заменяются на **s3** (код 1).

| Код 1 |
|--|
| <pre>s1='allaalrfal' s2='al' s3='zz' print(s1.replace(s2,s3,2)) print(s1.replace(s2,s3))</pre> |
| Результат |
| <pre>Zzlazzrfal zzlazzrfzz</pre> |

| Код 2 |
|--|
| <pre>s1= '123532'; print(s1.isdigit()) s2 = 'Arman'; print(s2.isdigit()) s1= '123532' ; print(s1.isalpha()) s2= 'Arman'; print(s2.isalpha())</pre> |
| Результат |
| <pre>True False False True</pre> |

s1.isdigit(); s2.isalpha()

Метод **s1.isdigit()** проверяет, содержит ли строка **s1** только цифры. Метод **s2.isalpha()** проверяет, содержит ли строка **s2** только буквы (код 2). Если в обоих методах результат положительный, результат будет «true», в противном случае – **false**.

S.islower(); S.isupper()

Метод **S.islower()** определяет, что данная строка содержит только строчные буквы (код 3). Если буквы в строке состоят только из нижнего регистра, результатом будет «true», в противном случае – false.

Метод **S.isupper ()** определяет, что данная строка содержит только прописные буквы (код 4). Если буквы в строке состоят только из верхнего регистра, результатом будет «true», в противном случае – false.

| Код 3 |
|--|
| <pre>S='INFORMATIKA' print(S.islower()); S='informatika' print(S.islower());</pre> |
| Результат |
| <pre>False True</pre> |

| Код 4 |
|--|
| <pre>S='INFORMATIKA' print(S.isupper()); S='informatika' print(S.isupper());</pre> |
| Результат |
| <pre>True False</pre> |



Применение

Практическая работа

Пример 1. Дана строка, состоящая из латинских букв. В этой строке замените букву «а» на «b». В результате выведите измененную строку и количество замен.

| № | Пример набора | Результат |
|---|---------------|---------------|
| 1 | abababab | babababa 4 |

Решение: Мы используем метод **replace**, чтобы изменить букву «а» на «b» по условию задачи. И мы используем функцию **count** для определения количества всех замененных «b» (код 5). Как видите в программе, надо уметь правильно использовать функции и методы, с которыми вы уже знакомы.

| Код 5 |
|--|
| <pre>s1=input() s2='a'; s3='b' s2=s1.replace(s2,s3) print(s2) print(s2.count('b'))</pre> |
| Результат |
| <pre>babababa 4</pre> |



Анализ



В зависимости от сходства выполняемых действий, сгруппируйте функции и методы для обработки строк в определенную группу.



Синтез



Составьте текст задачи для использования одного из методов строк. Решите составленную задачу.



Оценка



Оцените значение функций и методов, предназначенных для обработки строк.



1. Какой метод используется для перемещения подстроки во вторую подстроку?
2. Какие методы используются для преобразования строки в верхний и нижний регистры?
3. Какова функция метода `s1.isdigit()`?



Задания

Задание №1. Замена слова*

Первая строка содержит строку латинских букв. Замените слово, которое находится между двумя запятыми в этой строке другим словом. Слово замены находится во второй строке.

| Пример ввода | Результат |
|---|-----------------------------|
| Komputer, planshet, noutbook telefon | Komputer, telefon, noutbook |

Задание №2. Перевод в байт ***

Задано число в двоичной системе счисления. Нужно перевести это число в байт. Напишите программу, в результате выполнения которой перед заданным числом записывается нужное количество нулей до байта. Один байт содержит 8 бит. Например, $101_2 = 00000101_2$

| № | Пример ввода | Результат |
|---|--------------|-----------|
| 1 | 101 | 00000101 |
| 2 | 11 | 00000011 |

Задание №3. Сокращение слова***

Даны N слов ($0 < N < 100$). Сократите слова, в которых букв больше 6. Для этого необходимо указать первую и последнюю буквы слова и количество символов, которые должны быть удалены. Например, слово «экспортирование» сокращено до «э13е». Слова длиной 6 и меньше символов распечатать без изменений.

| № | Пример ввода | Результат |
|---|--|-----------------------|
| 1 | 3 экспортирование робототехника школа | э13е р11а школа |

Задание №4. Замена числа**

Даны N строк ($0 < N < 100$). Определите строки, состоящие только из цифр, преобразуйте их в числовой формат, прибавьте к ним число 100 и выведите на экран. Если строку не возможно преобразовать в число, выведите на экран -1.

| № | Пример ввода | Результат |
|---|--------------------|------------|
| 1 | 2 1250 Hh526 | 1350 -1 |

Задание №5. Количество букв**

Задана строка, состоящая из прописных и строчных латинских букв. Найдите количество всех букв «z» в этой строке.

| № | Пример ввода | Результат |
|---|---|-----------|
| 1 | Zebra named Zoe lives in a zoo and loves eating zucchini. | 4 |
| 2 | All animals are valid in this world. | 0 |

9 Практическая работа. Строковые методы и функции



Как можно использовать строковые функции и методы для решения задач?



✓ Как можно обрабатывать строки, используя строковые функции и методы?



Новые знания

Задание №1. Одинаковая подстрока*

Задана текстовая строка. Сколько раз подстрока «класс» встречается в этой текстовой строке? Длина текстовой строки не превышает 200 символов.

| № | Пример ввода | Результат |
|---|---|-----------|
| 1 | Я учусь в 10 классе в самом классном классе в самой классной школе. | 3 |

Задание №2. Все различные***

Задана текстовая строка. Сколько различных символов встречается в этой текстовой строке? Длина текстовой строки не превышает 255 символов.

| № | Пример ввода | Результат |
|---|--------------|-----------|
| 1 | Dreem | 4 |

Задание №3. Поиск***

Задана текстовая строка. В этой текстовой строке определите самую длинную последовательность из букв "k". Длина текстовой строки не превышает 1000 символов.

| № | Пример ввода | Результат |
|---|------------------------------------|-----------|
| 1 | aalkkaaylkkkkkkkkkasddkkkrftgkkkkk | 9 |

Задание №4. Подсчет слов**

Задан текст. Сколько в этом тексте слов начинающихся и заканчивающихся на одну и ту же букву? Длина текстовой строки не превышает 1000 символов.

| № | Пример ввода | Результат |
|---|--|-----------|
| 1 | Кок, поп, яблоко, урок, шалаш, велосипед | 3 |

Задание №5. Палиндром***

Задана текстовая строка. Определите является ли это строка палиндромом (одинаково читается слева направо и справа налево)? Если строка является палиндромом, то выведите в результате 1, иначе 0.

| № | Пример ввода | Результат |
|---|--------------|-----------|
| 1 | доход | 1 |
| 2 | топот | 1 |
| 3 | трава | 0 |

Задание №6. Слово состоящее из трех букв **

Задана текстовая строка. Посчитайте количество слов, содержащих 3 буквы в этой строке. Длина текстовой строки не превышает 1000 символов.

| № | Пример ввода | Результат |
|---|--------------------------------|-----------|
| 1 | Кок и поп пили чай с плюшками. | 3 |

Задание №7. Определение**

Задана строка, состоящая из букв и цифр. В этой строке больше букв или цифр? Если больше цифр, то на экран выведите 1, если больше букв, то на экран выведите букву «а».

| № | Пример ввода | Результат |
|---|----------------|-----------|
| 1 | Казахстан 2030 | а |
| | 2018-2019 год | 1 |



Как можно прочитать информацию из файла и записать в файл на языке программирования Python?



- ✓ Как вы думаете, для чего в процессе программирования используется чтение из файла или запись в файл?
- ✓ В чем важность и значимость чтения информации из файла или записи обработанной информации в файл в процессе программирования?

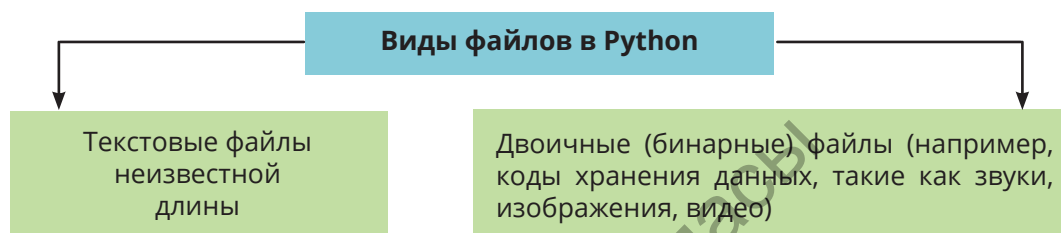


Новые знания

Информация, и введенная с клавиатуры, и полученная после выполнения программы, будет потеряна. Например, нам вместе с одноклассниками нужно написать программу, которая показывает имена друзей и соответственно номера телефонов каждого из них. И если мы не записываем эти данные в текстовый файл, то не стоит выполнять эту задачу. Самое простое решение этой проблемы – хранить большие объемы информации в заранее заданном файле, а затем читать и редактировать информацию на экране или в файле с помощью программного обеспечения. Теперь давайте рассмотрим работу функции чтения из файла и записи информации в файл в языке программирования Python.

Python имеет два типа файлов (схема 1).

Схема 1



Этапы работы с файлом:

1. Открытие файла
2. Режим чтения
3. Режим записи
4. Режим подключения данных
5. Работа с файлом
6. Закрытие файла

Чтение файла. Функция open()

Вы должны открыть файл перед чтением. Чтобы открыть файл, Python использует функцию open() (код 1). Эта функция обычно имеет два варианта.

1. Имя файла (путь к файлу), которое указывает местоположение файла.

2. Режимы открытия файлов:

«r» – открытие для чтения (reset – устанавливает связь с файлом)

«w» – открыть для записи (write – запись, запись в файл)

«a» открыть для добавления (append – добавление, добавление в файл)

Если вы не укажете второй параметр при вызове функции open (), файл откроется как текстовый файл.

В режиме чтения из файла читаем величину, загруженную для открытия файла (Код 1).

код 1

```
h=open('output.txt', «w») h.write('запись')  
f.close()# (закрытие файла )
```

код 2

```
f = open('input.txt', 'r')  
a=int(f.readline()) #(режим чтения)  
f.close()# (закрытие файла)
```

После завершения обработки данных в режиме записи в файл мы записываем их в другой текстовый или двоичный файл (код 2).

Пример 1. Во входном файле (input.txt (рис. 1)) заданы числа a и b. Считайте эти два числа из входного файла и выведите их сумму в выходной файл (output.txt (рис. 2)) (код 3).

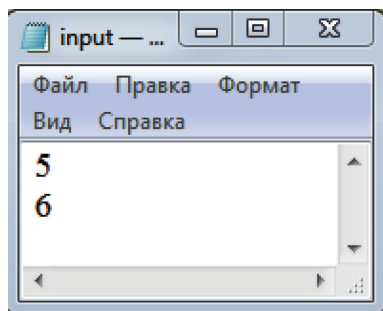


Рисунок 1

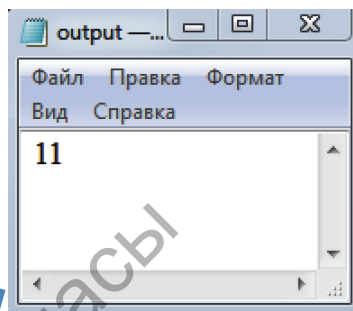


Рисунок 2

код 3

```
f = open('input.txt','r')  
a=int(f.readline())  
b=int(f.readline())  
h = open('output.txt','w')  
h.write(str(a+b))  
f.close()  
h.close()
```

Если после закрытия выходного файла (рис. 2), в файле ниже на одну строку после суммы a и b нужно записать их произведение (рис. 3), то тогда мы открываем файл для чтения, записываем код последовательности команд и перезаписываем файл.

```
код 4
f = open('input.txt','r')
a=int(f.readline())
b=int(f.readline())
t = open('output.txt', 'a')
t.write(' \n')
t.write(str(a*b))
f.close()
t.close()
```

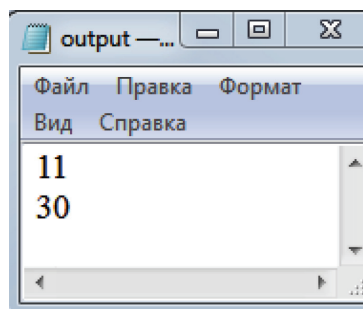


Рисунок 3



Применение

Практическая работа

Пример 2. Во входном файле (input.txt) имеется строка s (рис. 4), состоящая из латинских букв. Укажите количество различных букв в этой строке и запишите в выходной файл (output.txt) (рис. 5).

Основная идея решения примера: берем пустую строку. Давайте обозначим ее как t (код 5). Теперь ищем каждый символ из строки s в строке t . Если искомый символ не найден, то мы добавляем символ к t . В качестве ответа достаточно указать в файле длину строки t .

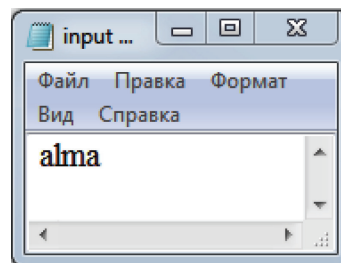


Рисунок 4

```
код 5
f = open('input.txt', 'r') #Открывает файл для
чтения
s=f.readline() #Считывает строку
h = open('output.txt', 'w') #Создает файл для
записи
t='' #Организовывает пустую строку
for i in range(0,len(s)):
    if t.count(s[i])==0 #Поиск искомой буквы в строке t
        t=t+s[i] #Если искомой буквы нет в строке t,
тогда добавляет его в эту строку
h.write(str(len(t))) #Записывает значение длины
строки t в файл
f.close() #Закрывает файл f
h.close() #Закрывает файл h
```

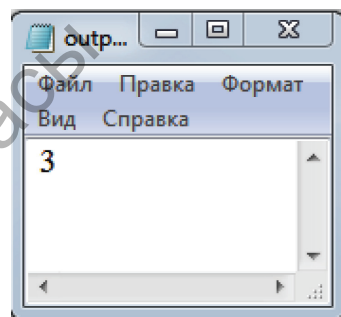


Рисунок 5



Анализ



Проанализируйте программный код 5. Если вам нужно повторно открыть файл **output.txt** и ввести строку `t` после значения количества символов, то какие вам нужно внести изменения в этот код? Определите их.



Синтез



Создайте программу, дающую возможность записать свое имя и возраст в файле **output.txt**. Представьте ее.



Оценка



Напишите небольшое эссе тему «Особенности работы с файлами в Python».



1. Какая команда используется для открытия файла?
2. Какие команды используются для чтения, записи и добавления файла?
3. Какие типы файлов есть в Python?
4. Какие шаги обработки файлов вы можете назвать?



Задания

Задание №1. Соединение слов (конкатенация)*

Во входном файле (`input.txt`) два слова из латинских букв записаны в разных строках. Прочитайте эти два слова во входном файле, объедините их и запишите во выходной файл (`output.txt`).

| input.txt | output.txt |
|---------------|------------|
| Dragon Fly | Dragonfly |
| Air Plane | Airplane |

Задание №2. Определение нечетных чисел**

Задан одномерный массив $A[N]$ ($1 \leq N \leq 99$, $a_i < 1000$). Напишите выходной файл, содержащий все нечетные числа этого массива.

| № | input.txt | output.txt |
|---|------------------------------|--------------|
| 1 | 8 14 27 53 689 1452 42 17 | 27 53 689 17 |



Как можно применять алгоритмы сортировки для решения практических задач на языке программирования Python?



- ✓ Что такое сортировка? Вспомните!
- ✓ С какими видами сортировки вы знакомы?
- ✓ Какова значимость сортировки в программировании?



Новые знания

Под сортировкой (*селекция; selection; сортировка; sorting*) – мы понимаем процесс перестановки элементов массива и упорядочивания их по определенному правилу. Например, вы можете отсортировать массив чисел по возрастанию, по убыванию, отсортировать массив по алфавиту и многое другое. Его основная цель – облегчить последующий поиск элементов, которые находятся в отсортированном множестве. Методы сортировки обычно разделяют на две категории: сортировка массивов и сортировка (последовательных) файлов. Эти два класса часто называют внутренней и внешней сортировкой. Массивы располагаются во «внутренней» (оперативной) памяти, для этой памяти характерен быстрый произвольный доступ, и сортировка называется внутренней. А файлы хранятся в более медленной, но более вместительной «внешней» памяти, т.е. на запоминающих устройствах (дисках) и сортировка называется внешней (рис. 1).

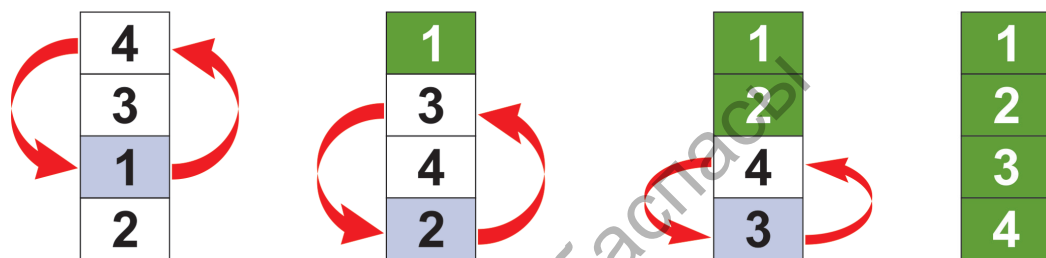


Рисунок 1. Схема сортировки

Существует несколько широко используемых алгоритмов сортировки одномерных массивов в процессе программирования. Эти алгоритмы делятся в зависимости от эффективности работы (таблица 1).

Таблица 1. Алгоритмы сортировки

| Алгоритм сортировки | Принцип работы | Количество перестановок при сортировке N=100 элементов |
|--|---|--|
| <p>Вставка - Insertion Элементы массива делятся на две части – упорядоченные и неупорядоченные. Элементы из неупорядоченной части вставляются в часть упорядоченных, сортируя сравнением</p> | <pre>def fast_insertion_sort(l): for i in range(1, len(l)): j = i - 1 value = l.pop(i) while (j >= 0) and (l[j] > value): j -= 1 l.insert(j + 1, value) return l a=[16, 25, 24, 12, 11] fast_insertion_sort(a) print (a)</pre> | <p>10000 раз</p> |
| <p>Выбор - Таңдай - Selection Находит самый наименьший (самый наибольший) элемент в массиве и заменяет его первым элементом из неотсортированных элементов.</p> | <pre>from random import randint def sel_sort(b): for i in range(len(b) - 1): m = i; j = i + 1 while j < len(b): if b[j] < b[m]: m = j j = j + 1 b[i], b[m] = b[m], b[i] a = [] for i in range(10): a.append(randint(1, 1000)) print(a) sel_sort(a) print(a)</pre> | <p>10000 раз</p> |
| <p>Quick Sort (Быстрая сортировка) считается наиболее быстрым алгоритмом сортировки среди всех типов сортировок</p> | <pre>from array import array B=array('i') #Процедура Qsort def qSort (A, p, q): if p >= q: return L = p; R = q X = A[(L+R)//2] while L <= R: while A[L] > X: L += 1 while A[R] < X: R-= 1 if L <= R:</pre> | <p>700 раз</p> |

| Алгоритм сортировки | Принцип работы | N=100 элементті үшін тексеру саны |
|---------------------|---|---|
| | <pre> A[L], A[R] = A[R], A[L] L += 1; R -= 1 qSort (A, p, R); qSort (A, L, q) #Основная программа K=10000; B=[] N=int(input()) for i in range (0,N): B[i] = input() qSort (B, 0, N-1) for i in range (0,N): print(b[i]) </pre> | |

Quick Sort (Быстрая сортировка) считается наиболее быстрым алгоритмом сортировки среди всех типов сортировок. С помощью Quick Sort числовые элементы массива сортируются по возрастанию или убыванию, текстовые элементы массива сортируются по алфавиту. Кроме того, в **Python** наряду с методами сортировки имеются функции сортировки. Вы познакомились с этими функциями в 9 классе.

Функции сортировки языка программирования Python

Используя команды из таблицы 2, можно сортировать не только числа, но и слова (таблица 2).

Таблица 2

| Вид сортировки | Программный код | Пояснение, комментарий |
|----------------|---|--|
| По возрастанию | <code>B = sorted(A)</code> | Сортирует массив A по возрастанию и загружает его в массив B |
| | <code>A.sort()</code> | Упорядочивает сам массив A |
| По убыванию | <code>B= sorted(A, reverse = True)</code> | Сортирует массив A по убыванию и загружает его в массив B |
| | <code>A.sort(reverse = True)</code> | Упорядочивает сам массив A |

Давайте рассмотрим примеры использования методов сортировки.

Пример 1. Задан одномерный массив $A [N]$ ($0 < N < 106$). Сколько разных элементов в этом массиве? Определите.

| Пример ввода | Результат | Пояснение |
|------------------------------|-----------|-----------|
| 10 12 5 4 7 12 5 4 7 7 12 | 4 | 12 5 7 4 |

Решение: Можно отсортировать массив по возрастанию или убыванию, чтобы определить, сколько разных элементов имеется в массиве. Для решения этой задачи мы используем быструю сортировку (Quick Sort). И сравним два элемента, расположенных рядом друг с другом в массиве (код 1). Для получения результата достаточно посчитать количество условий, когда два элемента, рядом стоящие, не равны между собой.

(А. Шень: Программирование: теоремы и задачи. 2004 г.)

Если учитывать, что средние компьютеры за 1 секунду приблизительно делают 1000000 операций, то в случае рассмотренном выше, потребуется 2 секунды, чтобы выяснить количество различных чисел среди 1000000 элементов. Для сортировки массива – 1 секунда, и еще 1 секунда, чтобы определить неравные элементы в упорядоченном массиве.

Как упоминалось выше, сортировка может быть применена не только к числам. Можно сортировать строки, массивы строк по алфавиту. Рассмотрим следующий пример.

```

from array import array
B=array('i')
#Процедура Qsort
def qSort ( A, p, q ):
    if p >= q: return
    L = p; R = q
    X = A[(L+R)//2]
    while L <= R:
        while A[L] > X: L += 1
        while A[R] < X: R -= 1
        if L <= R:
            A[L], A[R] = A[R], A[L]
            L += 1; R -= 1
        qSort ( A, p, R ); qSort ( A, L, q )
#Основная программа
K=1000000; B=[0]*K;z=0
N=int(input())
for i in range (0,N):
    B[i] = int(input())
qSort ( B, 0, N-1)
for i in range (0,N):
    if B[i]!=B[i+1]:
        z+=1
print(z)

```



Применение

Практическая работа

Пример 1: Задан одномерный массив строк A [N] (0 <N <100). Отсортируйте строки массива по алфавиту. Выведите первую строку упорядоченного массива.

| № | Пример ввода | Результат |
|---|---|-----------|
| 1 | 5 mala lama sala kala nala | kala |

Алматыкітап баспасы

Решение: Основная идея решения этого примера – отсортировать массив по алфавиту и напечатать первый элемент массива. Но так как Python начинает считать с нуля, мы печатаем нулевой элемент массива. Используем метод **Пузырька** (9 класс), потому что он идеально подходит для сортировки небольших списков. Сортировка массива из 100 элементов по алфавиту занимает около 1 секунды.

```
n = int(input())
a = []
for i in range(n):
    a.append(input())
for i in range(n):
    for j in range(i + 1, n):
        if a[i] > a[j]:
            a[i], a[j] = a[j], a[i]
print(a[0])
```



Анализ



Проанализируйте рассмотренные методы сортировки (*таблица 1*). Основные сходства и различия этих методов. Какой из этих алгоритмов сортировки лучше? Почему? Объясните причину.



Синтез



В примере 1 замените методы сортировки на функции сортировки языка Python и перезапишите программный код. Представьте его.



Оценка



Оцените важность методов сортировки в программировании. Напишите эссе о методах сортировки.



1. Какие методы сортировки вы знаете?
2. Какими параметрами эти методы отличаются друг от друга?
3. Как правильно использовать методы сортировки для решения задач?
4. Какие функции сортировки доступны в Python?



Задания

Задание №1. Сумма

Задан одномерный массив $A[N]$ ($1 \leq N \leq 1000$). Вычислите сумму K минимальных элементов этого массива.

| № | Пример ввода | Результат |
|---|------------------------------|-------------------------------|
| 1 | 8 10 20 30 45 31 42 12 16 | 10 12 16 20 30 31 42 45 38 |

Задание №2. Поиск

Вр входном файле (input.txt) задана строка из строчных букв английского алфавита длиной N ($1 \leq N \leq 1000$). Отсортируйте эту строку по алфавиту, найдите букву, расположенную на K позиции. В первой строке выходного файла (output.txt) записана буква, расположенная на K позиции.

| № | Пример ввода | Результат |
|---|-----------------|-----------|
| 1 | Friendship 3 | h |

Задание №3. Алфавит

Задана строка из строчных букв английского алфавита длиной N ($1 \leq N \leq 1000$). Определите, встречается ли в этой строке 26 букв английского алфавита. Если там есть все 26 букв английского алфавита, то выведите на печать слово «да», иначе «нет».

| № | Пример ввода | Результат |
|---|--------------------------------|-----------|
| 1 | Dogs are humans' best friends. | нет |

Задание №4. По двум попыткам

Спортсмену дается две попытки. В зачет идет наибольший результат. Из списка попыток получите список результатов от лучших к худшим.

Ввод натуральное число спортсменов ($N < 1000$) и далее N пар чисел – по две попытки на каждого спортсмена.

Результат: N чисел – результаты спортсменов по убыванию результатов.

| № | Пример ввода | Результат | Комментарий |
|---|---------------|-----------|--|
| 1 | 3 4 4 7 3 2 9 | 9 7 4 | Зачетный результат 1-го спортсмена 4, 2-го - 7, 3-го - 9 |

13

Определение и основные понятия графа



Что такое теория графов? В каких областях она применяется?



✓ Что такое граф? Для решения каких задач он применим?



Новые знания

Первые сведения о графах появились еще в XVIII веке, когда использовались схемы вида взаимосвязанных точек с помощью определенных прямых (линий). Первоначально эта информация часто была своего рода головоломкой. Но в конце XIX века в связи с развитием топологии, интерес к теории графов значительно вырос. Сегодня теория графов используется в различных областях, таких как социология, экономика, биология, медицина, химия, психология и программирование, дискретная математика. Во многих приложениях рассматривается связь между различными объектами. Геометрический подход к решению задач раздела дискретной математики называется теорией графов. Впервые термин «граф» ввел венгерский математик Д. Кениг.

Схема из множества точек, соединенных отрезками или дугами, называется **графом**.

Давайте установим пять разных точек на плоскости (рис. 1). Эти точки являются **вершинами** графов, а соединяющие их линии называются **ребрами**.

Если направления ребер не указываются, то граф называется **неориентированным** (рис. 1). Если ребрам графа приданы направления от одной вершины к другой, то такой граф называется **ориентированным** (рис. 2). Соответствующие вершины ориентированного графа называют началом и концом (рис. 2). **Вес ребра** – числовое значение, поставленное в соответствие данному ребру графа (рис. 1). Различные ребра могут соединять одну и ту же пару вершин. Такие ребра называют **кратными**.

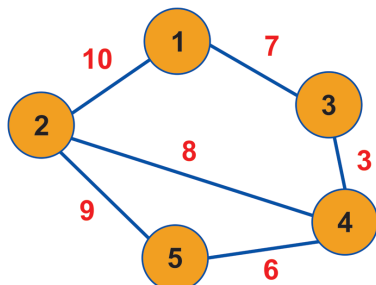


Рисунок 1

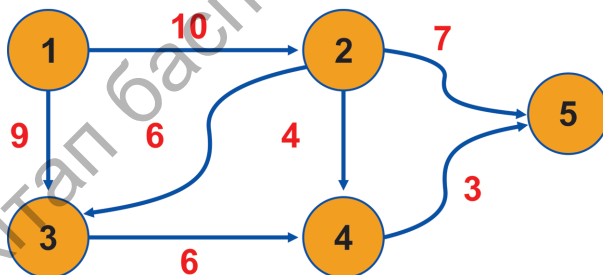


Рисунок 2

Ребро может соединять вершину саму с собой. Такое ребро называется петлей. Ребра **ориентированного графа** называются **дугами**. (рис. 2).

Граф или **неориентированный граф** обозначается буквой G. Он обозначается в следующем виде $G=(V, E)$. Здесь:

V – количество вершин (не пустое множество);

E – количество ребер;

Вершины и ребра иногда называют **элементами** графа, количество вершин графа $|V|$ называют **порядком** графа, количество ребер $|E|$ называют **размером** графа.

Например, на рисунке 1 для неориентированного графа имеет место запись $G=(5, 6)$.

Описание графов при программировании

Во время программирования должны быть описаны множества, содержащие вершины и ребра графов. Существует 2 типа описания графов. Первый – через матрицу смежности, другой – через ребра.

Представление графа через матрицу смежности

Матрица смежности графа G с конечным числом вершин n (пронумерованных числами от 1 до n) – это квадратная матрица A размера n, в которой значение элемента a_{ij} равно числу ребер из вершины i графа в вершину j.

Рассмотрим граф, показанный на рисунке 1. Для представления его через матрицу смежности описание связи через вершины и матрицу смежности будет: $G = (5, 6)$. Матрица смежности представляет собой квадратную таблицу размерами $n \times n$, где n – число вершин графа. В качестве размера матрицы указывается количество вершин графа, например $N = 5$, тогда эта матрица будет квадратной (рис. 3). Предположим, что граф представляет собой матрицу $A[5,5]$.

$A[2,5] = 9$, так как есть связь между 2-ой и 5-ой вершиной, вес связанного ребра равен 9. При равенстве строк и столбцов $i=j$, то будет $A[i, j]=0$. Ребро может соединять вершину саму с собой.

| i,j | 1 | 2 | 3 | 4 | 5 |
|-----|----|----|---|---|---|
| 1 | 0 | 10 | 7 | 0 | 0 |
| 2 | 10 | 0 | 0 | 8 | 9 |
| 3 | 7 | 0 | 0 | 3 | 0 |
| 4 | 0 | 8 | 3 | 0 | 6 |
| 5 | 0 | 9 | 0 | 6 | 0 |

Рисунок 3

Представление графа через список ребер

Список, в каждой строке которого записаны две смежные вершины и вес соединяющего их ребра, называется **списком ребер графа**. В списке ребер в каждой строке записываются две смежные вершины и вес соединяющего их ребра. Количество строк в списке ребер всегда должно быть равно величине, получающейся в результате сложения ориентированных ребер с удвоенным количеством неориентированных ребер.

Для описания графа через список ребер рассмотрим граф, представленный на рисунке 4. Тогда для описания соединения его вершин записывается так: $G=(5, 6)$, то есть учитывая, что в графе 5 вершин и 6 ребер.

| |
|--------|
| 5 6 |
| 1 2 10 |
| 1 3 7 |
| 2 4 8 |
| 2 5 9 |
| 3 4 8 |
| 4 5 6 |

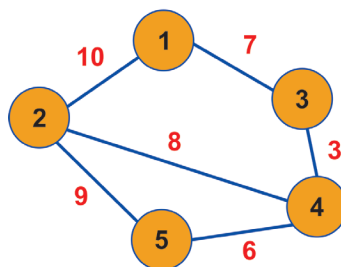


Рисунок 4

Тогда в первой строке n – число вершин, m – число ребер. В следующих m строках значение x , y и z содержат соответственно значение двух смежных вершин и вес ребра.

Графы используют в программировании для решения различных практических задач. С помощью графов изображаются схемы дорог, газопроводов, тепло- и электросети. Графы, как правило, очень удобны для описания отношений между двумя городами. Если вершины графа задают города, то вес ребер можно рассматривать как расстояние между ними. Теперь давайте рассмотрим примеры графов.

Определение количества мостов. В городе есть n мостов. Города, соединенные этими мостами, представлены в виде связной матрицы. Если в связной матрице между двумя мостами есть дорога, то вывести на печать 1, иначе 0. Определите, есть ли между мостами дорога. Входной файл **input.txt** и выходной файл **output.txt**.

| input.txt | output.txt |
|-----------|------------|
| 5 | 4 |
| 0 1 0 0 1 | |
| 1 0 1 1 0 | |
| 0 1 0 0 0 | |
| 0 1 0 0 0 | |
| 1 0 0 0 0 | |

При решении этой задачи мы вычисляем количество единиц и делим его на 2 (код 1). Причина в том, что связь между одним и тем же городом указана дважды в матрице. Если есть связь между 1-м городом и 2-м городом, существует связь между 2 и 1 (неориентированный граф).

| код 1 |
|---|
| <pre>f = open('input.txt', 'r') with open('input.txt', 'r') as f: N=int(f.readline()) A = [];z=0 for line in f: A.append(tuple(int(x) for x in line. split())) A = tuple(A)</pre> |

| продолжение |
|--|
| <pre>for i in range(0,N): for j in range(0,N): if A[i][j]==1: z+=1; h = open('output.txt', 'w') h.write(str(z//2)) h.close() f.close()</pre> |



Синтез



Придумайте пример использования графов при решении задач. Составьте условие задачи и нарисуйте схему.



Оценка



Оцените значимость графов в программировании и науке в целом.



Задания

Задание №1. Изображение схемы*

По предложенной матрице смежности составьте схему графа (рис. 1).

а) Неориентированный граф: $N = 6$ б) Ориентированный граф: $N = 5$

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |

| | | | | |
|----|----|----|----|----|
| 0 | 10 | 30 | 50 | 10 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 10 |
| 0 | 40 | 20 | 0 | 0 |
| 10 | 0 | 10 | 30 | 0 |

Задание №2. Получить матрицу смежности*

Заданные схемы графов опишите с помощью матрицы смежности (пример а (рис. 5) и через список ребер (пример б (рис. 6)).

а) Неориентированный граф $N = 5$ б) Ориентированный граф: $N = 4$

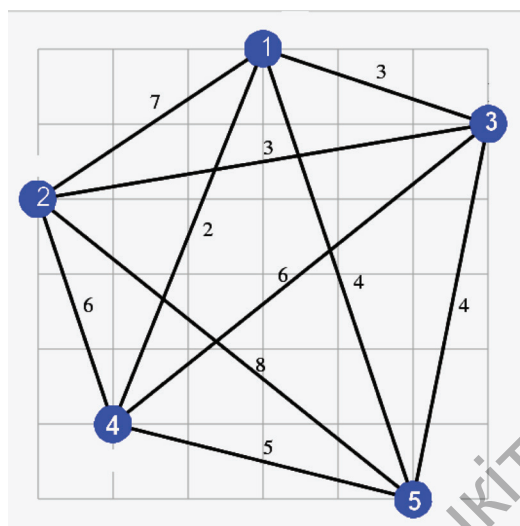


Рисунок 5. Неориентированный граф

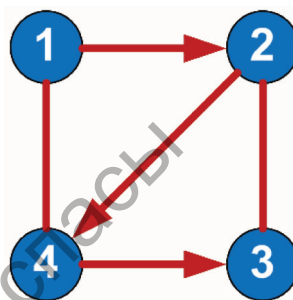


Рисунок 6. Ориентированный граф

Алгоритмы на графах



Какие типы алгоритмов рассматриваются на графах?



- ✓ Как вы думаете, в каких отраслях деятельности человека используется теория графов?
- ✓ Какие проблемы и задачи можно решить с помощью теории графов?



Новые знания

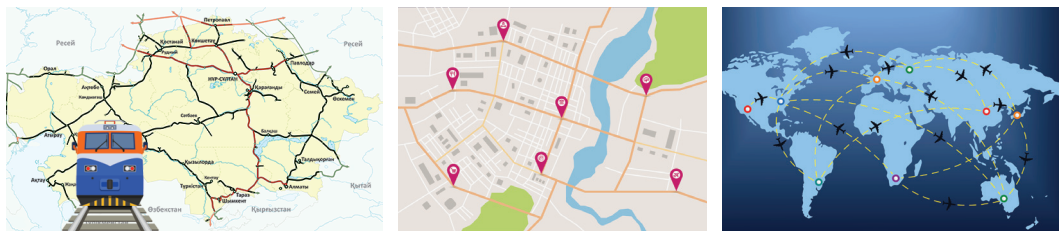


Рисунок 1

Алгоритмы методов теории графов широко используются в других областях. Примеры – использование графов для описания местоположения городских объектов, станций метро, воздушного движения между городами, железнодорожных перевозок, планирования и т. д. (рис. 1). Существует несколько известных алгоритмов для построения графов.

Алгоритм Флойда

Алгоритм Флойда – алгоритм нахождения длин кратчайших путей между всеми парами вершин в ориентированном графе.

Задан полностью ориентированный граф, описанный с помощью матрицы смежности, вес ребер указан. С помощью алгоритма Флойда можно эффективным способом построить кратчайший путь между вершинами этого графа. Ребра должны иметь вес не более 100.

Основная идея алгоритма: алгоритм Флойда показывает кратчайший путь к любым двум вершинам графов, как видно из примера. Например, самый короткий способ добраться от 1-й вершины к 3-й равен 7: $(1, 3) = 7$, а от 3-й к 1-й равен 11 $(3, 1) = 11$ (ориентированный граф) (рис. 2).

| Пример ввода | Результат |
|--------------|-----------|
| 4 | 0 5 7 13 |
| 0 5 9 100 | 12 0 2 8 |
| 100 0 2 8 | 11 16 0 7 |
| 100 100 0 7 | 4 9 11 0 |
| 4 100 100 0 | |

Фрагмент программы:

```

for k in range (0,N):
  for i in range (0,N):
    for j in range (0,N):
      if a[i][k]+a[k][j]<a[i][j]:
        a[i][j]=a[i][k]+a[k][j]

```

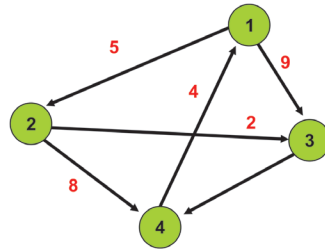


Рисунок 2

Алгоритм Дейкстры

Задан полностью ориентированный граф, описанный с помощью матрицы смежности, веса ребер записаны. Необходимо найти все кратчайшие пути из одной изначально заданной вершины графа до второй вершины. В качестве входных данных даны K, P ($1 \leq K, P \leq N$), матрица смежности. Ребра должны иметь вес не более 100.

| Пример ввода | Результат |
|-------------------------------------|-----------|
| 3 1 2 0 -1 2 3 0 -1 -1 4 0 | 6 |

Основная идея алгоритма заключается в первоначальном присваивании всем вершинам графа максимального числа. На каждом последующем шаге выбирается кратчайшее расстояние между этой заданной вершиной и другими вершинами и наибольшее число заменяется значением этого расстояния. Алгоритм на каждом шаге проверяет одну вершину. Когда все вершины будут рассмотрены, работа алгоритма завершится. Если алгоритм Флойда показывает кратчайший путь к любым двум вершинам заданного графа, то алгоритм Дейкстры находит кратчайший путь от вершины K к вершине P графа. Ниже приведен фрагмент программного кода для алгоритма Дейкстры. Здесь A, D – числовые массивы и Use – булевый (логический) массив. Булевый массив Use используется для проверки всех вершин графа, т. е. для определения, была ли проверена взятая вершина или нет.

| Фрагмент программы | продолжение |
|---|---|
| <pre> Big=101 for i in range (0,N): for j in range (0,N): A[i][j]=int(input()) if A[i][j] < 0: A[i][j]= Big D[i]= Big D[0]= Big + 1 D[K]= 0 for i in range (0,N): M = 0 </pre> | <pre> for j in range (0,N): if D[j] < D[M] and not Use[j]: M = j Use[M]= True for j in range (0,N): if D[j] > A[M][j] + D[M]: D[j] = A[M][j] + D[M] if D[P] = Big: print('-1') else print(D[P]); </pre> |



Оценка



Каково значение графов и графовых алгоритмов в науке и технике?



Задания

Задание №1. Задание на исследование

Используйте сеть Интернет для поиска информации об алгоритме Форд-Беллмана и деревьях графов. Каковы особенности этих алгоритмов? Определите области практического применения. Проанализируйте код программы.

Задание №2. Поиск наименьшего пути

По заданной схеме графа, изображенного на рисунке 3, выполните следующие задания:

Используя **алгоритм**

Дейкстры найдите кратчайший путь из вершины **К** к вершине **Р**.

а) $N = 6, K = 1, P = 6$;

ә) $N = 6, K = 2, P = 5$

б) $N = 6, K = 4, P = 6$

2) Используя **алгоритм**

Флойда запишите матрицу смежности перехода из одной вершины ко второй вершине графа. По его матрице смежности постройте матрицу кратчайших путей между его вершинами.

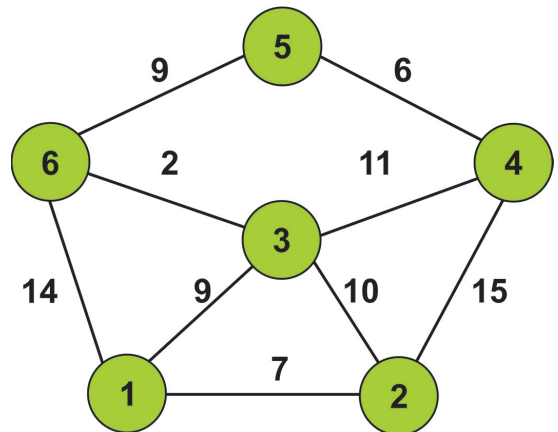


Рисунок 3. Схема графа

1

Web-проектирование. Понятие о web-сайте

Какие программы используются для создания web-страниц?



- ✓ Как вы думаете, какие программы используются при создании web-страниц?
- ✓ Для чего нужно web-программирование?
- ✓ С помощью какой программы создаются интернет-страницы?

**Новые знания****Web-программирование**

Развитие Интернета началось в начале 90-х годов, когда был создан новый протокол обмена данными между компьютерами. Этот протокол был назван HTTP (Hyper Text Transfer Protocol – протокол передачи гипертекста). Вместе с этим протоколом была создана служба World Wide Web, которая применяется для передачи файлов через сеть HTTP-серверов, названная Интернетом.

WWW (World Wide Web) называют также «Всемирной паутиной», в сокращенном виде – просто Web. Ее основная функция – организация поиска, сбора и вывода на экран необходимой информации. WWW – это мультимедийная распределенная информационная сеть на основе гипертекста.

Распределенная информационная система – информация хранится в большой системе под названием WWW-серверы. На web-серверах размещен не отдельный документ, а группа взаимосвязанных документов. Таковую группу предоставляет web-сайт. Размещение готового материала на web-сайте называется web-публикацией. Отдельный документ называется web-страницей, содержащей тексты, графические иллюстрации и разного рода мультимедийные объекты.

Пользователи, которые могут подключиться к сети, получают эту информацию через клиентские программы, называемые программами просмотра документов WWW или браузерами. В этом случае, программа сканирования отправляет через компьютерную сеть запрос HTTP-серверу, который хранит файл, содержащий необходимый документ – web-страницу. В ответ на запрос сервер отправляет востребованный файл клиенту, если файл недоступен или запрос содержит неверные параметры, то сообщается об этом пользователю. Взаимодействие клиент-сервер выполняется согласно протоколу.

Большинство файлов в сети созданы с помощью специального языка HTML (Hyper Text Markup Language). Содержание web-страниц разнообразно и представляет разные темы, но все они являются базовыми, т.е. язык написания кода страницы – это HTML. Расширение файлов во всех этих HTML-документах должно быть HTM или HTML.

Несмотря на то, что web-страницы компактно отображаются на экране, язык разметки гипертекста HTML не является языком форматирования текста, так как каждый пользователь использует компьютеры с различными параметрами.

Например, на компьютере установлен браузер, работающий в Windows, а пользователь другого компьютера использует браузер, работающий в операционной системе Linux. И так как у них разные параметры, то в них один и тот же файл будет отображаться по-разному. А на третьем компьютере можно будет текст web-страницы рассматривать в версии для слепых. И если шрифты даны по Брайлю, то результат будет совсем иным. HTML не может считаться языком для форматирования текстов, учитывая, что документы просматриваются разными пользователями на различных устройствах и с разными браузерами. Специальные программы для просмотра электронных документов, созданных по правилам языка разметки HTML, называются *браузерами*. Основная функция браузера заключается в интерпретации кода HTML и выводе визуального результата на экран монитора пользователя. Кроме этого HTML служит для функциональной разметки документа.

Понятие о web-странице

Web-сайт – обычно содержит информацию по какой-либо определенной теме, связанной соответствующими гиперссылками. Набор web-страниц обычно размещается на серверах в виде имени и адреса web-сайта. На готовые сайты несложно добавлять web-страницы. Возможна вставка нескольких web-страниц на сайт, например, электронные газеты, различные новости. Такой сайт может быть объемным, иметь сложную иерархическую структуру. Так как информация, которую вы вносите на страницу сайта может изменяться, устаревать, то необходимо систематически обновлять информацию и редактировать сайт. Чтобы ваш сайт мог быть найден по ключевым словам, нужно на главной странице размещать соответствующие гиперссылки.

Web-сайт представляет собой информационную систему, охватывающую два основных компонента:

1) **Компонент показа (front-end)**. Web-страница, которую видят посетители, называется front-end (размер страниц, графика, аудио, текст). Чтобы наглядно понять, что такое frontend, откройте страницу любого сайта – перед собой вы увидите интерфейс. Щелкнув правой кнопкой мыши, можно посмотреть код этой страницы в браузере. Код страницы описывает все, что вы видите перед собой: цвета, верстку, шрифты, расположение графических элементов и так далее.

2) **Компонент реализации (back-end)**. Набор аппаратно-программных средств, при помощи которых реализуется логика работы сайта.

Планирование сайта

Web-сайт – это маленькая модель мира. Если раньше сайт создавал один web-мастер, то в настоящее время сайты создаются целыми командами. Это web-дизайнер, программист, бизнес-консультант, маркетолог, менеджер и другие.

Этапы проектирования web-сайта взаимосвязаны.

Этапы планирования сайта:

- ✓ Изначально необходимо определить общее назначение сайта.
- ✓ Определить, какую информацию нужно разместить на сайте.
- ✓ Необходимо собрать соответствующую информацию.
- ✓ Нужно определить дизайн сайта.
- ✓ Продумать логическую структуру сайта.
- ✓ Продумать физическую структуру сайта, то есть расположить все файлы, предназначенные для размещения на сайте, по папкам.
- ✓ Проверить правильность выполнения всех действий.

Разработка web-сайта состоит из нескольких этапов:

- ✓ планирование;
- ✓ разработка элементов;
- ✓ программирование;
- ✓ тестирование;
- ✓ публикация;
- ✓ рекламирование;
- ✓ контроль.

Во время планирования необходимо ответить на некоторые вопросы и решить несколько проблем:

1. Месторасположение сайта.
2. На какую аудиторию рассчитывается сайт?
3. Какая информация будет опубликована?
4. Для чего нужно web-программирование?
5. В каком виде будет осуществлено взаимодействие с пользователями?

Во время разработки элементов необходимо учитывать, что сайт выполняется как программный продукт:

1. Создание навигационной структуры.
2. Создание дизайна страницы.
3. Подготовка текстовой и видеоинформации для заполнения страницы.

Программирование. Суть этого этапа – форматирование сайта.

Тестирование. Одним из ключевых этапов разработки сайта является тестирование. Во время тестирования проверяется правильность работы сайта, в том числе:

1. работа ссылок;
2. ошибки в текстах;
3. эффективность навигации;
4. работа почты и других форм;
5. открытие графических файлов.



Оценка

Выскажите свое мнение по следующим вопросам:



Какие из перечисленных факторов оказывают большое влияние на популярность web-сайта?

- улучшенный дизайн;
- спланированность;
- информационная ценность;
- улучшенный интерфейс;
- высокий уровень элементов программирования.



1. Что такое web-программирование?
2. Для каких целей оно используется?
3. Какова функция языка HTML?
4. Что вы понимаете под web-сайтом?
5. Каковы этапы планирования сайта?
6. Каковы способы публикации web-сайта?
7. Из каких этапов состоит разработка web-сайта?



Применение HTML-тегов при создании web-сайтов



- ✓ Какие вы знаете языки программирования?
- ✓ Как вы думаете, в чем особенность языка HTML по сравнению с другими известными вам языками?



Новые знания

HTML (HyperText Markup Language – язык разметки гипертекстов) – компьютерный язык со строгими правилами. Его назначение и функции, выполняемые им, можно определить из названия. Гипертекстом называется текст, в который вставлены специальные коды, управляющие дополнительными элементами, такими, как форматирование, иллюстрации, мультимедийные вставки и ссылки на другие документы. Под разметкой понимается вставка в текст этих кодов, определяющих то, как итоговый гипертекстовый документ должен отображаться специальной программой – браузером.

Известно, что язык разметки гипертекста HTML изначально был предложен Тимом Бернером Ли в 1989 году.

Документ на HTML языке состоит из основного текста и тегов. Каждый элемент документа отмечен тегами, которые иногда являются атрибутами и именами свойств. По сути, теги схожи со скобкой «BEGIN END» в языке программирования Pascal и определяют область текстовых элементов в документе.

Для создания файла достаточно использования простого текстового редактора Блокнот.

Все теги начинаются с символа «меньше» (<) и заканчиваются символом «больше» (>). Эту пару символов также называют треугольными скобками. После символа открытия размещаются определяющие тег ключевые слова. Теги языка HTML влияют только на ту часть документа, к которой применены, например, к абзацу. Поэтому чаще используются парные теги – открытия и закрытия. Открывающий тег влияет на часть документа, а закрывающий тег заканчивает этот эффект. Закрывающие теги начинаются с символа «/».

Структура документа HTML:

1) Весь документ начинается тегом `<HTML>` и заканчивается соответственно тегом `</HTML>`. Эти теги показывают браузеру о наличии HTML документа;

2) HTML-документ состоит из раздела заголовка и тела документа.

Раздел заголовка состоит из тегов `<head>` и `</head>`, здесь указывается информация о документе;

3) теги `<title>` и `</title>` располагаются внутри раздела заголовка и отражают название документа (то, что отображается в заголовке окна браузера);

4) основной текст отображается в теле документа, и он находится между тегами `<body>`, `</body>`.

Эти четыре тега представляют собой базовую структуру документа HTML, и эти теги обязательно отображаются во всех HTML-документах.

Примеры:

1) откройте программу Блокнот;

2) в появившемся диалоговом окне введите следующие строки (рис. 1);

3) сохраните созданный документ (запишите `index.html` в строке, где необходимо указать имя файла);

4) если сохраненный файл открыть с помощью браузера, Internet Explorer откроется web-страница (рис. 2).

Управление внутренними элементами документа на языке HTML

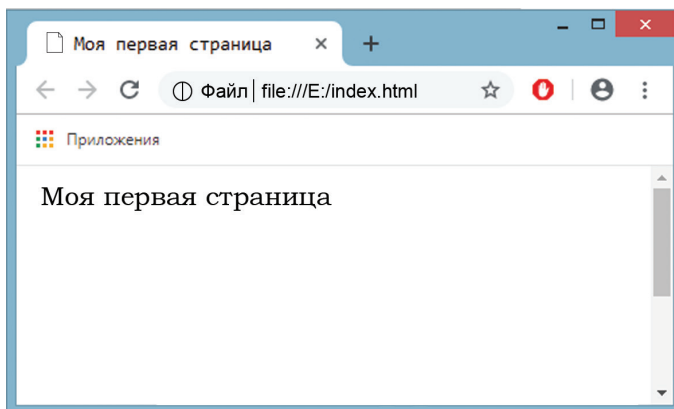


Рисунок 1. Окно программы Блокнот

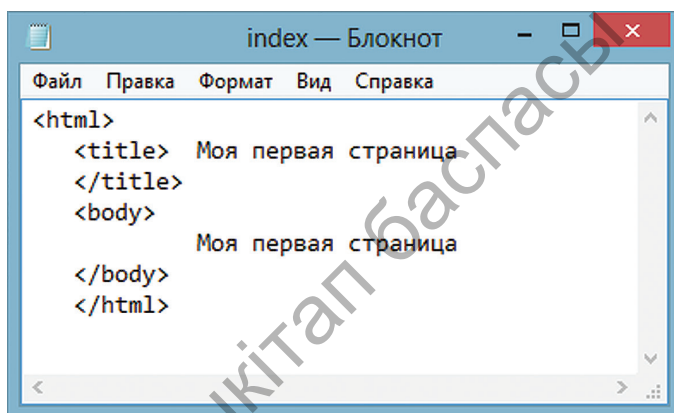


Рисунок 2. Окно web-страницы

1. На языке HTML заголовок документа может отображаться шестью различными стилями и записывается с использованием тегов **<h1>**, **</h1>**, **<h6>**, **</h6>** ;

2. **<p>** , **</p>** – обозначают начало и конец абзаца;

3. **<hr>** – добавляет в html-документ горизонтальную линию;

4. **
** – конец строки, осуществляет перевод строки;

5. **<i>**, **</i>** – текст выделяется курсивом;

6. ****, **** или ****, **** – полужирный шрифт;

7. **_{**, **}** – нижний индекс;

8. **^{**, **}** – верхний индекс;

9. Если надо перейти к новой строке без абзаца, то используется тег **
**;

10. **<dl>**, **</dl>** – Тег **<dl>** входит в тройку элементов **<dl>**, **<dt>**, **<dd>**, предназначенных для создания списка определений;

11. **** **** - создает нумерованный список

**** **** - создает маркированный список

**** **** - элемент списка

12. **<dt>**,**</dt>** – Тег **<dt>** входит в тройку элементов, предназначенных для создания списка определений. Каждый такой список начинается с контейнера **<dl>**, куда входит тег **<dt>** – создающий термин;

13. **<dd>**, **</dd>** – Тег **<dd>** входит в тройку элементов, предназначенных для создания списка определений. Каждый такой список начинается с контейнера **<dl>**, куда входит тег **<dt>** создающий термин и тег **<dd>** задающий определение этого термина;

Элементы нумерованного списка создаются тегом **** и обозначаются с помощью атрибута «*type*» (*таблица 1*).

Таблица 1

| Коды HTML | Виды списков | Пример |
|---|-------------------------------------|--|
| <code><ol type="1"> ... </code> | Арабские цифры | 1. Нур-Султан 2. Алматы 3. Атырау |
| <code><ol type="A"> ... </code> | Заглавные буквы латинского алфавита | A. Нур-Султан B. Алматы C. Атырау |
| <code><ol type="a"> ... </code> | Строчные буквы латинского алфавита | a. Нур-Султан b. Алматы c. Атырау |
| <code><ol type="I"> ... </code> | Римские цифры верхнего регистра | I. Нур-Султан II. Алматы III. Атырау |
| <code><ol type="i"> ... </code> | Римские цифры нижнего регистра | i. Нур-Султан ii. Алматы iii. Атырау |

Если нумерованный список начинается с определенного числа, то его задают атрибутом «start» тега .

Например, <ol type="1" start="3"> список начинается с 3.



Применение

Практическая работа

Запишите предложенный код в программе Блокнот, сохраните под именем «index2.html». Если шрифты отражаются неправильно на веб-страницах, вы должны выбрать «Юникод» (рис. 2) в разделе «Кодировка» и правильном наборе кода HTML откроется web-страница (рис. 3).

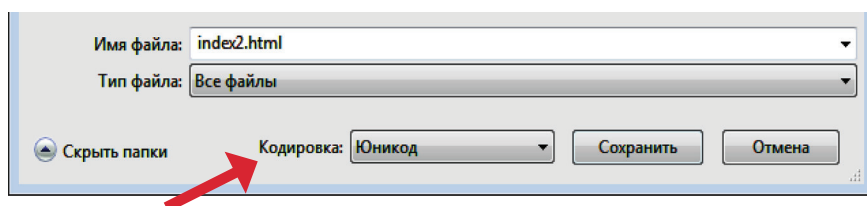


Рисунок 2. Выбор «Юникод»

```
<html>
<head>
<title> Список</title>
</head>
<body>
<b><h2> Компьютерная память</h2></b>
<ul>
<Li> Внутренняя память
<ol>
<li> Устройство оперативной памяти
<li> Устройство постоянной памяти
<li>КЭШ память </ol>
<Li> Внешняя память
<ol>
<li> Жесткий диск
<li> Флэш-память
<li>Диски </ol>
</body></html>
```

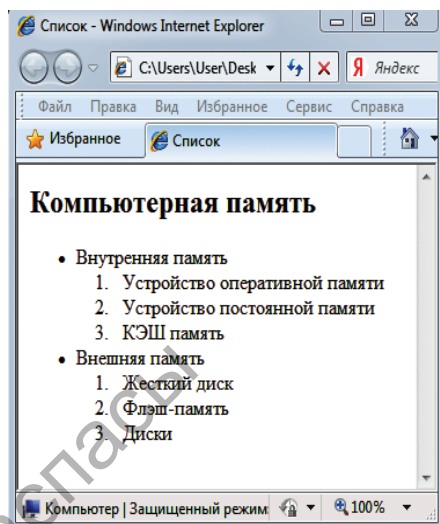


Рисунок 3. Окно web-страницы



Анализ



Проанализируйте работу внутренних элементов управления документом на языке HTML. Расположите эти команды в группы по назначениям.



Синтез



Создайте простой проект документа и напишите его с помощью HTML-кода (например, список учащихся класса).



Оценка



Оцените возможности языка HTML.



1. Язык HTML – какой это язык?
2. Из каких команд состоит основа структуры языка HTML?
3. С помощью каких тегов получается нижний индекс?
4. Как используются теги для записи полужирного и курсивного на чертания?
5. Как отобразить документы, записанные в программе Блокнот на языке HTML, в виде web-страницы?



Задания

1. Дополните фрагмент кода HTML, наберите его и получите результат.

```
<h1>Тема</h1>  
<h2> Тема </h2>  
<h3> Тема </h3>
```

2. Наберите код HTML, получите результат.

```
<html>  
<body>  
<p><b>Это полужирный шрифт</b></p>  
<p><strong>Акцентирование текста</strong></p>  
<p><big>Это заглавные буквы</big></p>  
<p><i>Это курсивный шрифт</i></p>  
</body>  
</html>
```

3. Разместите на web-страницу следующие строки.

```
Нур-Султан  
Дорогая моя столица  
Главный город наш Нур-Султан!  
С каждым годом ты больше и краше!  
И гордится тобой Казахстан!
```

3 Указание размера, цвета и типа шрифта в HTML



Какие теги языка HTML используются для задания размера, цвета и типа шрифта?



- ✓ Какие есть преимущества при представлении данных в виде таблицы?
- ✓ Как представить данные разными шрифтами на web-странице?



Новые знания

Некоторые теги могут использоваться или должны использоваться с определенными параметрами, которые записываются в открытые теги-элементы (в одном теге можно задать несколько параметров). **** – дает возможность изменять тип, размер и вид букв текста, расположенного между тегами открытия **** и закрытия ****.

Например, у тега открытия **** (**** – тег закрытия) возможны несколько атрибутов:

SIZE – определяет размер шрифта (размер стандартного шрифта – «3»). Разместив текст между тегами **** (здесь n – цифра), ему необходимо задать размер:

SIZE – определяет размер шрифта. ****, здесь размер первоначальной буквы i между 1 и 7;

```
<font size=«1»>Пример 1</font>  
<font size=«2»>Пример 2 </font>  
<font size=«3»>Пример 3 </font>  
<font size=«4»>Пример 4 </font>  
<font size=«5»>Пример 5 </font>  
<font size=«6»>Пример 6 </font>  
<font size=«7»>Пример 7 </font>
```

Color – дает возможность определить цвет текста. Задается или значением RGB шестнадцатеричной системы или одной из базовых 16 цветов ****;

```
<font color=«red»>Красный</font>
```

```
<font color=«#FF0000»>Красный</font>
```

Давайте еще раз поговорим об использовании цветов. Для установки цвет фона web-страницы необходимо записать параметр BGCOLOR внутри тега **<BODY>** и название цвета. Или задать его 6-значный код. В данных двух примерах страница окрашивается в красный цвет:

```
<body bgcolor=«red»> (использовано название цвета)
```

```
<body bgcolor=«#FF0000»> (цвет задан 6-ти значным кодом)
```

Атрибут Face служит для задания гарнитуры шрифтов, использующихся для текста. Шрифты Times New Roman, Arial, Tahoma, Courier, Courier New распознаются во всех браузерах. ****.

Тег <basefont> предназначен для задания шрифта, размера и цвета текста по умолчанию. Закрывающий тег не требуется. Указанные значения будут использоваться во всем документе за исключением тега , в котором можно переопределить параметры оформления текста. Допускается использование тега в контейнере <head> или <body>, изменять вид шрифта для части документа.

 ... – от английского emphasis – акцент, ... – от английского strong emphasis – сильный акцент.



Применение

Практическая работа

Научимся создавать таблицу, используя атрибуты языка HTML, с помощью которых редактируют таблицу. Запишите нижеследующий код в программе Блокнот и сохраните его как «index4.html». Если правильно ввести код HTML, то откроется web-страница, представленная на рисунке 1.

```
<html>
<head>
<title>Изменение шрифта</title>
</head>
<body bgcolor=808000 text=ffff33 >
<p align=center>
<font color=c0c0c0 face=«courier new»
size=+3>
Структура алгоритмов</font> </p>
<ol type =1>
<li> Линейная структура.
<li> Разветвленная структура.
<li> Циклическая структура.
</ol>
<font color=#ff66ff>
<b>Цвета(список)</b></font>
<ol type = "a" ><li> пронумеровано.</li>
</ol>
<ol type = "B" start=2 ><li> маркировано.
</li></ol>
<ol type = "i" start=3><li>уточнение. </li></ol>
</body>
</html>
```

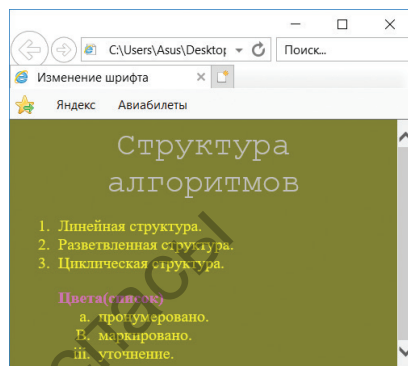


Рисунок 1. Списки на web-странице



Синтез



Для публикации в Интернете создайте на языке HTML web-страницу, содержащую данные из вашей жизни.



1. Как вы понимаете форматирование документов HTML?
2. Какие бывают физические стили? Какие атрибуты стилей вам известны?
3. Какие вы знаете логические стили?
4. Назовите названия тегов для записи верхних и нижних индексов.



Проект «Моя школа». Создайте web-страницу о своей школе, используя изученные теги и их атрибуты. Оформите web-страницу по своему усмотрению.



Задание

«Использование шрифтов в HTML»

1. Создать документ, в котором, в заголовке окна браузера должна быть надпись «Практическая работа».
2. Задать цвет текста в документе, и цвет фона.
3. Задать шрифт текста Arial и Courier New
4. С помощью атрибутов тега `<r>` `</r>` выровняйте текст по левому краю; по правому краю.
5. Выделите текст при помощи заголовков, используя шесть уровней заголовков.
6. Используя параметр `` измените размер шрифта от + до - `` для каждого абзаца.
7. Оформите текст абзацев используя различное начертание: **полужирное**, *курсив*, с подчеркиванием.



Как можно создать таблицы на web-страницах?



- ✓ Каковы преимущества представления данных в таблицах?
- ✓ Как представить данные web-страниц в виде таблицы?



Новые знания

На языке HTML таблицы строятся с помощью тегов **<table>**, **</table>**. Данные в таблице организуются специальными тегами (заголовок, количество строк и столбцов). При создании таблицы тег **<caption>**, **</caption>**, определяющий заголовок таблицы, используется один раз;

1) **<td>**, **</td>** – тег, который определяет отдельную ячейку в таблице;

2) **<th>**, **</th>** – тег, предназначенный для создания одной ячейки таблицы, которая обозначается как заголовочная. Текст в такой ячейке отображается браузером обычно жирным шрифтом и выравнивается по центру. Тег **<th>** должен размещаться внутри контейнера **<tr>**, который в свою очередь располагается внутри тега **<table>**;

3) **<tr>**, **</tr>** – тег служит контейнером для создания строки таблицы. Каждая ячейка в пределах такой строки может задаваться с помощью тега **<th>** или **<td>**.

В HTML есть атрибуты, которые позволяют данные на web-страницах представлять в виде таблиц. Список атрибутов, которые изменяют стиль таблицы, приведен в таблице 1.

Таблица 1

Список атрибутов, которые изменяют таблицу HTML

| Атрибут | По умолчанию | Использование в тегах | Значение |
|--------------------|--------------|--|---|
| align | left | Применяется во всех тегах при создании таблицы | Горизонтальное выравнивание информации в ячейке. Возможные значения: left, right, center и char (выравнивание по определенному символу) |
| bdcolor | | Применяется во всех тегах при создании таблицы | Определяет цвет фона таблицы |
| border | 0 | <table> | Толщина рамки таблицы в пикселах |
| cellpadding | 0 | <td> , <th> | Определяет расстояние между границей ячейки и ее содержимым. |
| cellspacing | 0 | <td> , <th> | Задаёт расстояние между внешними границами ячеек. |

| | | | |
|----------------|--------------------|---|--|
| colspan | 1 | <td>, <th> | Указывает количество столбцов, которые объединены в одной ячейке |
| rowspan | 1 | <td>, <th> | Указывает количество строк, которые объединены в одной ячейке |
| rules | none | <table> | Устанавливает границы между ячейками таблицы. Возможные значения: rows, cols или all |
| valign | center | <tr>, <td>, <th> | Устанавливает вертикальное выравнивание для ячеек таблицы, принимает значения: top, bottom. Этот атрибут работает только в браузере Internet Explorer, а для всех остальных применяется align. |
| width | По ширине страницы | Используется во всех тегах при создании таблицы | Устанавливает ширину таблицы в пикселах или процентах от ширины страницы. |

Определение цветов в документе

В языке HTML цвет задается одним из двух путей: с помощью шестнадцатеричного кода и по названию некоторых цветов. Преимущественно используется способ, основанный на шестнадцатеричной системе счисления, как наиболее универсальный. Каждый из трех цветов – красный, зеленый и синий – может принимать значения от 00 до FF, что в итоге образует 256 оттенков. Таким образом, общее количество цветов может быть $256 \times 256 \times 256 = 16\,777\,216$ комбинаций. Цветовая модель, основанная на красной, зеленой и синей составляющей получила название *RGB* (red, green, blue). В следующей таблице даны примеры записи цветов в RGB формате (таблица 2).

Таблица 2

Таблица цветов

| Цвет | RGB | Цвет | RGB | | |
|--------|-------------|--------|---------|--------------------------|--------|
| black | черный | 000000 | purple | фиолетовый | FF00FF |
| white | белый | FFFFFF | yellow | желтый | FFFF00 |
| red | красный | FF0000 | brown | коричневый | 996633 |
| green | зеленый | 00FF00 | orange | оранжевый | FF8000 |
| azure | бирюзовый | 00FFFF | violet | лиловый | 8000FF |
| blue | синий | 0000FF | gray | серый | A0A0A0 |
| silver | серебряный | C0C0C0 | fuchsia | ярко-фиолетовый (фуксия) | FF00FF |
| maroon | бордовый | 800000 | lime | ярко-зеленый (лайм) | 00FF00 |
| olive | оливковый | 808000 | teal | бирюзовый | 008080 |
| navy | темно-синий | 000080 | aqua | голубой (аквамарин) | 00FFFF |



Научимся создавать таблицу, используя теги языка HTML и список атрибутов, которые предназначены для редактирования таблицы. Запишите нижеприведенный код в программе **Блокнот** и сохраните файл как «index4.html». Чтобы правильно были отображены буквы на web-странице, нужно в окне сохранения в поле Кодировка выбрать «Юникод». Если правильно ввести код HTML, то откроется web-страница, как показано на рисунке 1.

Задание 1. Создайте таблицу из 3 строк и 4 столбцов.

```
<html><head><title>Таблица</title>
</head> <body> <h3>Таблица</h3>
<table cellspacing=0 cellpadding=0
width="60%" border=4>
<tr align=center> Таблица
<th bgcolor=black>
<th bgcolor=white>a12
<th bgcolor=black>
<th bgcolor=white>a14
</tr>
<tr align=center><th bgcolor=white>a21
<th bgcolor=black><th bgcolor=white>a23
<th bgcolor=black> </tr>
</table></body></html>
```

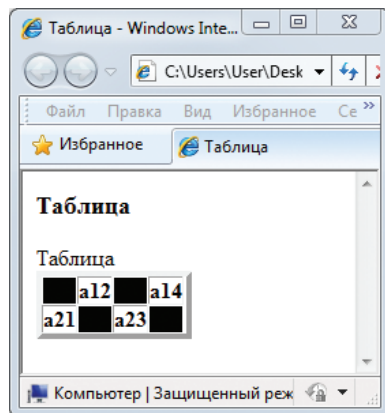


Рисунок 1. Таблица

Задание 2. Создайте цветную таблицу из 4 строк и 2 столбцов. Размер границы таблицы 1 пиксель. Запишите нижеприведенный код в программе Блокнот и сохраните их как «index5.html». Чтобы правильно были отображены буквы на web-странице, нужно в окне сохранения в поле Кодировка выбрать «Юникод» (рис. 2).

```
<html><head>
<title>Цвета таблицы</title></head>
<body bgcolor=white>
<table width="30%" border=1>
<tr>
<h2>ЦБЕТА-СМЫК</h2>
<td width="50%">cyan</td>
<td width="50%" bgcolor=cyan></td></tr>
<tr>
<td width="50%">magenta</td>
<td width="50%" bgcolor=magenta></td></tr>
<tr>
<td width="50%">yellow</td>
<td width="50%" bgcolor=yellow></td></tr>
<tr>
<td width="50%">black</td>
<td width="50%" bgcolor=black></td></tr>
</table></body></html>
```



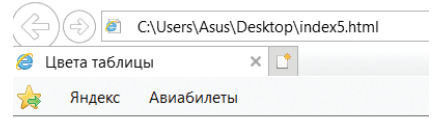
Синтез



Для публикации в Интернете создайте web-страницу на языке HTML в виде таблицы, содержащей данные, используемые в повседневной жизни.



1. Какие теги используются для создания таблицы?
2. Как можно изменять ширину и высоту таблицы?
3. Как установить внутренние и внешние границы таблицы?
4. Что вы можете сказать о цветах языка HTML?
5. С помощью какого атрибута можно объединить несколько столбцов в один столбец?
6. Как можно использовать цвета в таблице?



ЦВЕТА-СМУК




| | |
|---------|---|
| CYAN |  |
| MAGENTA |  |
| YELLOW |  |
| BLACK |  |

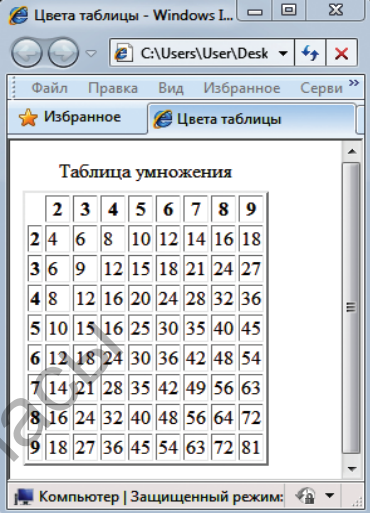
Рисунок 2. Цвета-СМУК



Задания

1. Постройте таблицу Пифагора (умножение (рис. 3)).
2. Какой результат получится при использовании следующего фрагмента кода?

```
<table border=1> <tr>
<td> текст</td>
<td> текст</td>
<td> текст</td>
</tr> <tr>
<td> текст</td>
<td> текст</td>
<td> текст</td>
</tr></table> </body></html>
```



Цвета таблицы - Windows I... C:\Users\User\Desk...
 Файл Правка Вид Избранное Серви >>
 Избранное Цвета таблицы

Таблица умножения

| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|----|----|----|----|----|
| 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 |
| 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 |
| 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 |
| 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 |
| 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 |
| 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 |

Компьютер | Защищенный режим: [lock icon]

Рисунок 3. Таблица Пифагора

5

Применение графики в HTML-документе



Как можно использовать элементы графики для web-страниц?



- ✓ Что такое графические объекты?
- ✓ Для чего применяются графические объекты в документах?
- ✓ Какова цель применения графических объектов на web-страницах?



Новые знания

Графика web-страницы

Изображения играют важную роль при оформлении web-страниц. Сами рисунки хранятся в отдельных от документов HTML-файлах, они отображаются на web-страницах с помощью браузера.

Размещение рисунка (тег ``).

Данный тег используется для размещения на web-странице графического изображения. Графическое изображение – это небольшие пиктограммы, рисунки, графические объекты и изображения карт. Установить изображение на web-странице можно с помощью тега ``, который может быть расположен в любом месте тега **body**.

Наряду с этим, адрес изображения задается значением атрибута `src` (от англ. source – источник). Этот атрибут является одним из самых необходимых атрибутов тега ``.

Например:

```

```

Для представления на экране фото Нур-Султан из файла Нур-Султан.jpg (рис. 1).

```

```

Примечание. Желательно, чтобы графические файлы, которые размещаются на web-странице, были расположены в одной папке вместе с файлом главной страницы, например, `index.html`. Графика на web-странице обычно представлена тремя форматами: `GIF`, `JPG`, `PNG`.

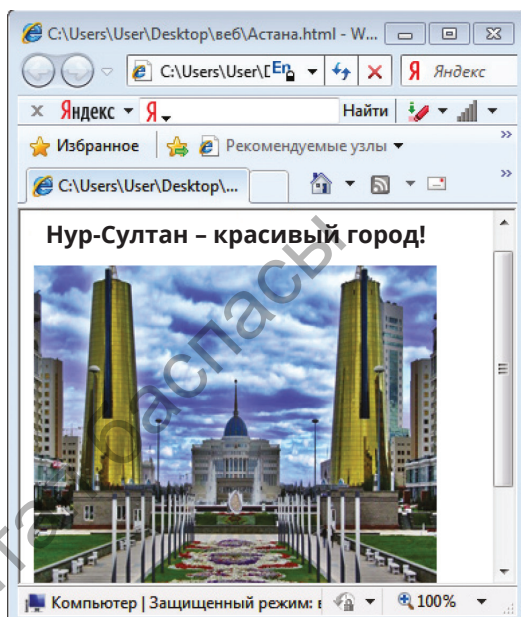


Рисунок 1. Изображение на web-странице

| Атрибуты рисунка | Форматы | Характеристика |
|------------------|--|--|
| alt | <code></code> | Если рисунок не отражается, то вместо него выводится наименование рисунка или детали про этот рисунок |
| border | <code></code> | Толщина линии (границы) рамки вокруг изображения задается пикселями. |
| align | <code></code> | показывается расположение изображения относительно текста: <ul style="list-style-type: none"> • Top – сверху • Bottom – снизу • Middle – посередине • Left – слева • Right – справа |
| height | <code></code> | задает высоту (по вертикали) изображения в окне браузера пикселями или процентами от высоты окна |
| width | <code></code> | задает размеры (по ширине) изображения в окне браузера пикселями или процентами от ширины окна |
| vspace | <code></code> | Добавляет пробелы сверху и снизу изображения в пикселях |
| hspace | <code></code> | Добавляет пробелы справа и слева от изображения в пикселях |

Слово, присваиваемое атрибуту Alt, является запоминающимся ключом, который появляется при наведении курсора. В некоторых случаях, когда изображение не отображается на web-странице, оно отразит тему изображения.

Например,

Установка фонового рисунка на web-странице

Фоновое изображение – графический файл, который заполняет окно браузера независимо от его размера, дублируя фоновый рисунок при необходимости. Графически изображение, используемое в качестве фонового изображения, появляются в результате записи атрибута **Background** и указаны в теге <body>.



Применение

Практическая работа

Научимся создавать web-страницу, используя теги языка HTML, которые имеют соответствующие атрибуты и предназначены для размещения изображений. Запишите нижеприведенный код в программе **Блокнот** и сохраните как «index.html». Создайте web-страницу «Казахстан – многонациональное государство» (рис. 2). Для этого надо выбрать рисунки из Интернета или иных источников, характеризующие культуру разных народов нашей страны.

```
<html>
<head>
<title> Казахстан –
многонациональное государство</
title>
</head>
<body bgcolor=aqua>
<font color=red><h3 align=center>
Дом дружбы</h3></font>
<br>


<br>
</body> </html>
```

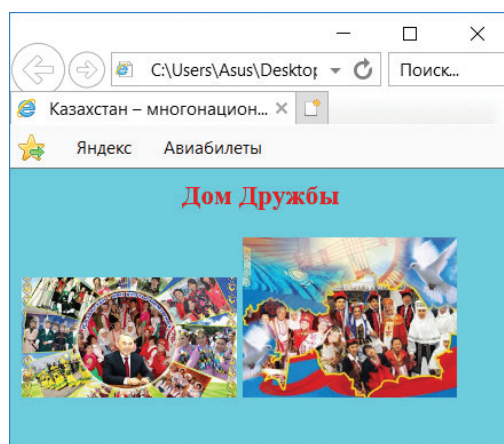


Рисунок 2. Web-страница с добавлением изображения



Синтез



Представьте на HTML языке проект web-страницы на тему «Природа Казахстана».



Оценка



Исследуйте, какие еще проекты можно создать, используя графические возможности языка HTML.



1. Какой тег применяется для размещения рисунка на web-страницу?
2. Назовите атрибуты тега IMG и опишите их назначение.
3. Что такое фоновый рисунок?
4. Какой атрибут используется для отражения фонового рисунка?



1. Создайте проект «Моя школа», используя графические возможности языка HTML. Используя изученные теги и их атрибуты, разместите текст о своей школе на web-страницу.
2. Создайте web-страницу, цель которой продвижение 4 казахстанских продуктов, используя графические возможности языка HTML.



Создайте web-страницу с текстом и фотографиями исторических памятников по истории Казахстана.

6

Выполнение практических работ на языке HTML



Как можно использовать на практике язык HTML?



- ✓ Какие языки программирования вы знаете?
- ✓ Как вы думаете, какие есть особенности у языка HTML по сравнению с языками программирования?

Выполните предложенные ниже практические работы на языке HTML. Запишите представленный ниже код в программе *Блокнот*, сохраните. Для того чтобы правильно отразились буквы на web-странице, нужно в окне сохранения в поле Кодировка выбрать «Юникод».



Практическая работа №1

Создание многоуровневого списка

Иногда приходится составлять список внутри другого списка. Такие списки называют многоуровневыми.

```
<html> <head>
<title>Многоуровневый список</title>
</head> <body>
<hr> <ul> <li>Европа
      <ol>
<li>Англия</li>
<li>Франция</li>
      <li>.....</li>
</ol>
</li> <li>Центральная Азия
      <ol>
      <li>Казахстан</li>
<li>Узбекистан</li>
      <li>.....</li>
</ol>
</body> </html>
```

Практическая работа №2

Создание таблицы

```
<html>
<head>
<title> язык html </title>
</head>
```



```
<body>
<table border=2 cols=2 bgcolor="yellow" align=center>
<tr> <th>текст 1 столбца</th>
<th>текст 2 столбца </th>
</tr>
<tr> <td>1-строка, 1-столбец</td>
<td>1-строка, 2-столбец </td>
</tr>
<tr> <td>2-строка, 1-столбец </td>
<td>2-строка, 2 -столбец </td>
</tr> </table>
</body>
</html>
```



Практическая работа №3

Ввод текста

```
<html>
<head>
<title> язык html </title>
</head>
<body bgcolor="green">
<font size= "7" face= "Times New Roman">
Запишите информацию о языке HTML из 5 предложений.
</font>
</body>
</html>
```

Практическая работа №4

Составление HTML-кода

Выберите произвольную тему, к примеру, Праздники в Казахстане. Создайте web-страницу, содержащую списки, таблицы, текст, рисунки. Наберите HTML-код и представьте его реализацию.



Как можно использовать элементы графики на web-странице?



- ✓ Что такое графические объекты?
- ✓ Для чего используются в документах графические объекты?
- ✓ Какова цель использования графических объектов на web-страницах?



Новые знания

Главным признаком HTML-документа является наличие в нём гиперссылок (или просто ссылок) на другие документы, сайты, файлы, картинки и т. д. Именно возможность вставлять в страницы ссылки на объекты вне неё и сделала Интернет столь популярным и удобным для использования. В этом уроке мы поговорим о том, как создать ссылку на сайт, на его отдельную страницу или файл.

Ссылки с одной web-страницы на другую – одно из наиболее привлекательных функций системы World Wide Web. **Гиперссылка (hyperlink)** является частью гипертекстового документа, который ссылается на другой элемент (текст, заголовок, изображение и т. д.) внутри другого документа, расположенного на локальном диске или в компьютерной сети. Для ввода ссылки нужен тег **A (anchor)**. Чтобы указать адрес ссылки нужно воспользоваться атрибутом тега **href (hypertext reference)**.

Методы использования ссылок на web-странице

Гиперссылки создаются с помощью парного тега `<a>`. Внутри тега помещается текст, который будет отображаться на web-странице. Текст ссылки отображается в браузере с подчеркиванием, цвет шрифта – синий, при наведении на ссылку курсор мыши меняет вид. Обязательным параметром тега `<a>` является атрибут `href`, который задает URL-адрес web-страницы. **` Текст ссылки `**

В этом случае гиперссылкой является выделенный «Текст ссылки», при нажатие на него откроется страница «адрес доставки».

1-й способ:

Если ссылка сделана на казахскую Википедию, значением атрибута `href` является «`http://kkk.wikipedia.org`». Запись в кавычках – это полный адрес казахской Википедии или URL-адрес сайта.

2-й способ:

Если нужно сделать переход с одной страницы на другую, то нет необходимости в полном адресе страницы. Например: `` Для перехода на вторую страницу нажмите ``

Если страница 2_str.html находится в папке Website, а первая страница, которой делаем ссылку (1_str.html) расположена в другой папке, то гиперссылку выполним по следующей команде `` Для перехода на вторую страницу нажмите ``

3-й способ:

Можно создавать ссылки не только на другие документы, но и на определенные точки внутри одной и той же страницы HTML. Для определения места ссылки в HTML-документе необходимо создать метку. Метка создается при помощи тега `<a>` с атрибутом id. Для этого используем и знак #. Обозначаем элемент ссылки через атрибут id. `<h1 id=«heading1»>` Заголовок `</h1>`

Теперь этому элементу, то есть заголовку, задаем ссылку символом #. Символ # сообщает браузеру о ссылке внутри данной страницы. После символа # записываем тег id перехода по ссылке.

Например, `` Ссылка на заголовок ``

4-й способ:

Вы можете использовать только в качестве гиперссылки не только текст, но и изображение. Для этого теги `<img...>`, определяющий ссылку, располагается между тегами `<a...>`. Если навести курсор на это изображение, то так же как если бы вы навели курсор на текстовую ссылку, графическая ссылка выполнит ссылку на любой HTML-документ или на конкретное место web-документа.

Пример:

` `

В строке `` вы можете разместить небольшое изображение компьютера в виде гиперссылки, нажав на него перейдете на файл с информацией comp.html о вашем компьютере.

Дизайн ссылки. Цвет ссылки

По умолчанию браузер устанавливает определенную цветовую схему для ссылок. Разработчик может изменять цвета по своему усмотрению. Поскольку ссылки равномерно внедрены во весь документ, атрибуты цвета ссылки размещаются в теле BODY. У тега ссылок следующие атрибуты:

link – определяет цвет еще не посещенных ссылок, то есть ссылок, которые еще не были просмотрены;

vlink – определяет цвет посещенных ссылок;

alink – определяет цвет активной ссылки, т.е. ссылки, на которую наведен курсор мыши.

Значениями каждого из этих атрибутов могут быть либо название цвета, либо шестнадцатеричное число в формате #RRGGBB. Вы можете указать один, два или все три атрибута, а также значения по умолчанию для не-известных атрибутов.

```
<body bgcolor=«#669900» text=«#333333» link=«#669900»  
vlink=«#666666» alink=«#000000»>
```



Применение

Практическая работа

Задание: Создайте две web-страницы. На первой web-странице расположите изображение часов. Сделав гиперссылку на изображение часов, создайте web-страницу, на которой будет размещена информация о часах (рис. 1 и 2).

Код 1-й web-страницы

```
<html> <head>  
<title>web-файл</title>  
</head>  
<body bgcolor=»#ffffcc»  
text=«#330066»>  
<p align=«center»>  
<font color=«#008080» size=«7»>  
<b> часы </b></font><br>  
<font size=«6»><i> Изображение часов</i></  
font>  
<br><br>  
<a href= 001.jpg > <img src=«001.jpg»  
width=«200», align=«left»> </a>  
</p>  
</body>  
</html>
```

Код 2-й web-страницы

```
<html> <head>  
<title> часы </title>  
</head>  
<body>  
Часы – это прибор для  
определения текущего  
времени суток и измерения  
продолжительности  
временных интервалов в  
единицах, меньших, чем  
одни сутки. Самые первые  
часы на земле – солнечные.  
Они были гениально  
простыми: воткнутый в  
землю шест.  
Вокруг него нарисована  
шкала времени.  
</body> </html>
```

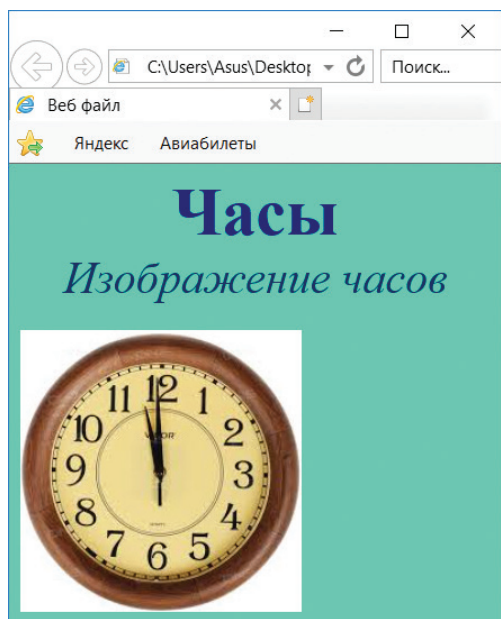


Рисунок 1. Вставка изображения часов

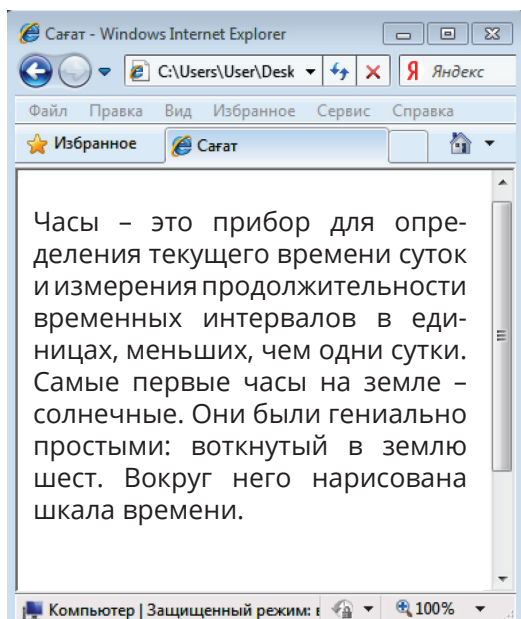


Рисунок 2



1. Какие программы используются для вывода и представления на экране гипертекста?
2. Что такое гиперссылка (hyperlink)?
3. Какие есть способы использования ссылок на web-странице?
4. Как выбираются цвета для дизайна ссылок?



Задание 1. Создайте web-страницу Index.html разместите в ней ссылку на сайт www.kitar.kz под названием «Книги на казахском языке».

Задание 2. Создайте web-страницу в HTML с гиперссылкой на 3 популярных образовательных сайта.

Задание 3. Создайте web-страницу, содержащую предложенные ниже рисунки и соответствующие им названия. Напишите HTML-код, в котором задайте ссылку на другую web-страницу, содержащую информацию об этих предметах.



Книги



Глобус



Компьютер

8 Каскадные таблицы стилей CSS (Cascading Style Sheets)



Как оформляются web-страницы?



✓ Как можно использовать каскадную таблицу стилей (CSS) при создании web-страницы?



Новые знания

Каскадная таблица стилей

CSS (Cascading Style Sheets – Каскадные таблицы стилей) – это язык, который отвечает за визуальное представление документов, написанных на языках разметки HTML, XHTML и т. д. Каскадные таблицы стилей используются создателями web-страниц для задания цветов, шрифтов, расположения и других аспектов представления web-документа. Основной целью разработки CSS являлось разделение содержимого (написанного на HTML, XHTML, XML и т. д.) и оформления документа (написанного на CSS). CSS позволяет представлять один и тот же документ в различных стилях или методах вывода. Так как, если код разделен на несколько частей, то его чтение при редактировании будет более понятным.

В 1990-е годы стандартизация web интенсивно развивалась, и веб-дизайнерам нужен был общий стандарт разработки сайтов. Таким образом, появились HTML 4.01, XHTML и стандарт CSS. Термин **“Каскадные таблицы стилей”** ввел **Хокон Виум Ли** в 1994 году. Он совместно с **Бертом Босом** стал развивать CSS и дал начало Консорциуму Всемирной Паутины (WWW), который назвал “web-стандарты”.

Преимущества:

- Четкое управление, значительно уменьшает объем кода и способствует удобному чтению.
- С помощью CSS вы можете внедрять параметры, которые нельзя передать в HTML. Например, можно убрать подчеркивание под ссылками.
- С помощью CSS легко меняется визуальное представление web-страницы. В ней можно представить визуальное представление нескольких документов одной таблицей. Например, вы написали 20-страничный web-документ полужирным зеленым шрифтом. Но через некоторое время вы хотите поменять цвет шрифта на синий или красный на всех 20 страницах. Тогда в HTML-коде вы меняете атрибуты шрифта на нужный на всех страницах. Это можно таблицей стилей – поменять все 20 страниц одновременно.
- Разделение оформления и содержания позволяет работать над дизайном и контентом параллельно. В CSS имеется понятие верстки сайта.

Создав простую web-страницу, выполним следующие действия:

Откройте новую страницу в блокноте, сохраните ее под именем *style.css* в папке, где расположена web-страница (рисунки 1, 2).

Это будет наша страница стиля. Теперь добавим страницу *style.css* к web-странице. Для этого в html есть тег `<link>` ответственный за внешние файлы. Добавляем этот тег к нашей web-странице:

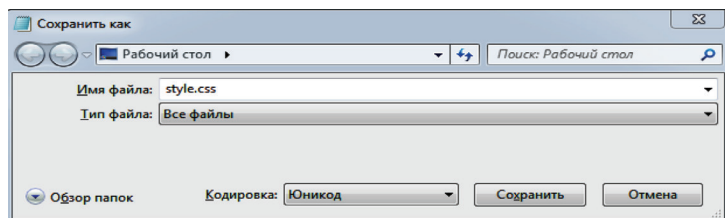


Рисунок 1. Создание файла Style.css

```
<html>
<head>
<title> Добавление CSS к HTML</title>
</head>
<body>
<h1>Это заголовок первого уровня
</h1>
Здесь располагается текст
<h2> Это заголовок второго
уровня</h2>
Здесь располагается текст
</body>
</html>
```

Код программы для 1-го варианта

```
<html>
<head>
<title> Добавление CSS к HTML</title>
<link rel="stylesheet" type="text/css"
href="style.css">
</head> <body>
<h1> Это заголовок первого уровня
</h1>
Здесь располагается текст
<h2> Это заголовок второго уровня
</h2>
Здесь располагается текст
</body> </html>
```

Код программы для 2-го варианта

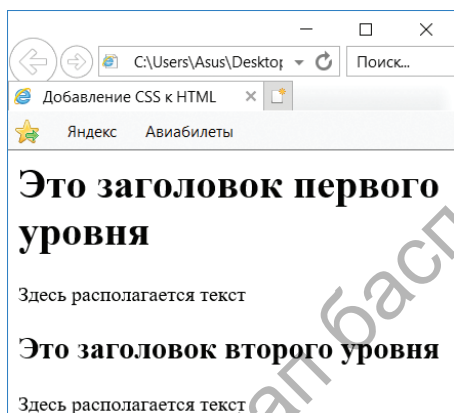
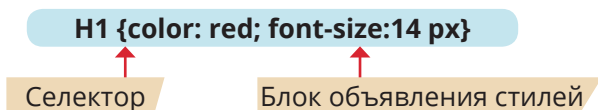


Рисунок 2. Web-страница с представлением заголовков

У CSS имеется свой синтаксис как и у других языков. В нем нет ни элементов, ни параметров, ни тегов. Этот язык имеет только правила.

Правило состоит из селектора и из блока объявления стилей, заключенных в фигурные скобки (схема 1). Внутри одного селектора может записываться несколько правил. Например, мы можем записать в одном селекторе позицию заголовка, порядок расположения, тип шрифта, цвет, стиль.

Схема 1



Блок объявления стилей состоит из свойств и их значений, разделенных точкой с запятой.



Применение

Практическая работа

Задание: Измените цвет и стиль текста web-страницы, представленной на рисунке 2, используя таблицу стилей `style.css`. Для этого добавим код, данный ниже, на страницу `style.css`. С помощью этого кода фон web-страницы, показанной на рисунке 2, изменится на синий, а текст изменится на белый (рис. 3).

```
body{background: blue; color: white; }
```

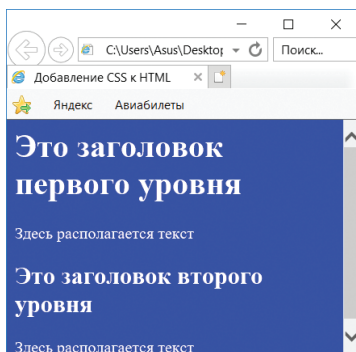


Рисунок 3. Web-страница заголовок с цветом фона и шрифта



1. Что такое CSS?
2. Когда создали CSS?
3. Как используются каскадные таблицы стилей при разработке web-страницы?
4. Что такое селектор?
5. Из чего состоит селектор?



Задание 1. Используя файл стилей `style.css`, сделаем нижеследующие изменения.

```
body{background: blue; color: white;}  
h1{color:red;}  
h2{color:yellow;}
```


9 Типы селекторов в каскадных таблицах стилей



Как использовать селекторы элемента и класса при создании web-страниц?



- ✓ Какие различия имеются между языком разметки гипертекстов HTML и каскадными таблицами стилей?
- ✓ Какие есть преимущества у каскадных таблиц стилей?



Новые знания

Селектор (англ. selector – «сортировка», «выборка») – это часть CSS-правила, которая сообщает браузеру, к какому элементу (или элементам) web-страницы будет применен стиль.

Типы селекторов

1. Селектор по элементу

2. Селектор по классу

3. Селектор по id

4. Контекстный селектор

Селектор по элементу

В языке таблиц стилей вы можете использовать в качестве селекторов любые элементы, имеющиеся в языке HTML. Например, в документе HTML, если весь текст параграфа (абзаца) должен быть представлен шрифтом Arial, размером – 14 пункта, то он будет записан следующим образом:

```
P{ font: normal 14pt Arial; }
```

здесь выбран элемент P как селектор, который указывает абзац в HTML.

Селектор по классу

Что делать если два абзаца нужно выделить одним цветом?

Код `<P style= "color:blue" >` Лишь данный абзац будет окрашен в синий цвет. Если есть несколько абзацев и каждый из них надо окрасить в HTML в один цвет, то надо использовать атрибут class.

`<P class= "blue">`. Этот и все другие абзацы будут отображены синим цветом.

Доступ к классу осуществляется через точку, показывая, что класс blue является свойством объекта `<P>`. Селектор по классу, начинающийся с точки, применяется ко всем элементам с данным классом, т. е. пустота перед точкой обозначает *любой элемент*.

```
<Style type="text/css">
```

```
P.blue{color: blue}
```

```
</Style>
```



Рассмотрим следующее практическое задание (рис. 1). В этой работе можно отметить особенности использования кодов на языках HTML и CSS.

html:

```
<p>Здесь применен селектор по элементу абзаца </p>
<p class=«green»> Здесь применен class для зеленого цвета записи абзаца </p>
<p class =«big_red »>Здесь абзац будет записан заглавными буквами и красного цвета</p>
<p>Здесь абзац будет обычным </p>
```

CSS:

```
p {
font-family: arial, verdana, sans-serif;
font-size:18px;
}
.green {color:green;}
big_red{text-transform:uppercase;
font-size:28px;
color: red;
}
```

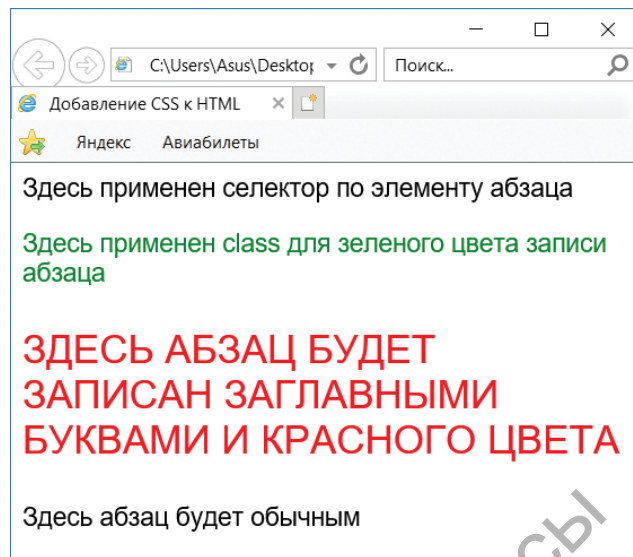


Рисунок 1. Использование различных шрифтов



1. Что такое селектор?
2. Какие бывают типы селекторов?
3. В каких случаях применяется селектор по элементу?
4. Как работает селектор по классу?
5. Каковы различия между функциональными возможностями селекторов по элементу и по классу?



- Задача 1. Создайте web-страницу о членах вашей семьи.
Задача 2. Создайте web-страницу о вашей любимой книге.

Селекторы в CSS. Идентификационные и контекстные селекторы



Как используются идентификационные и контекстные селекторы при создании web-страниц?



✓ Какова значимость селекторов в CSS?



Новые знания

Селекторы по идентификации

В нижеуказанном примере в качестве селектора рассматриваются элементы `body`, `h1`, `h2`. Если на web-странице есть несколько одинаковых элементов (например, абзац) и если все абзацы записаны черным цветом и мы хотим изменить только один из них на красный, то идентифицируем этот параграф и применяем к нему стиль. С помощью `id` можно применять стили к точно выбранному элементу, когда в HTML-коде может быть только один элемент с данным `id`.

В HTML идентификатор элемента передается через параметр `id`, а его значению присваивается отдельное (уникальное) имя. Например:

```
<p id=«pink»>Текст идентификатора параграфа(id).</p>
```

Вы можете назвать идентификатор любым словом, отличным от имен тегов, атрибутов и элементов в HTML и CSS. Например, идентификатору нельзя задать имя `body`. Теперь добавим два абзаца в web-страницу и одному из них добавим идентификатор:

```
<html>
<head>
<title>CSS id</title>
<link rel=«stylesheet» type=«text/css» href=«style.css»> </head>
<body>
<h1>Заголовок первого уровня</h1> Здесь обычный текст
<h2>Заголовок второго уровня</h2>
Здесь обычный текст <p>Обычный абзац</p>
<p id=«pink»>Идентификатор (id) абзаца</p> </body>
</html>
```

По вышеуказанному коду web-страницы видно, что в ней не использованы цвета фона, текста (рис. 1). Чтобы страница привлекала внимание наглядностью, добавляем стили для каждого абзаца в файле стилей (`style.css`).

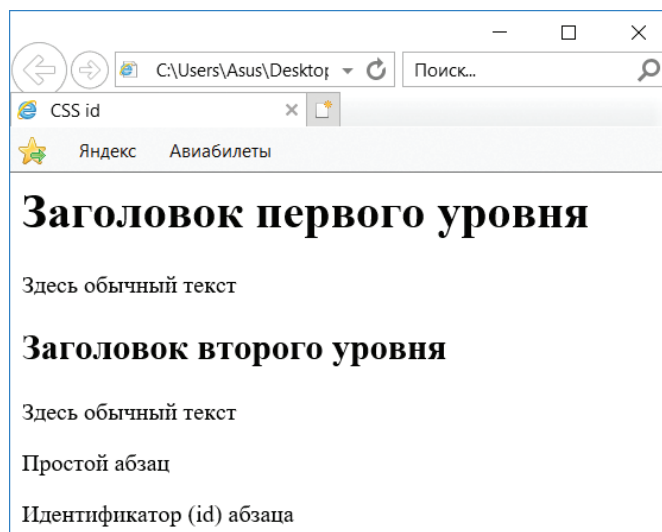


Рисунок 1. Web-страница с черно-белым изображением

Сначала весь текст по умолчанию отразится черным цветом, но записанный с помощью **id** **“pink”** текст абзаца будет розовым. В этом случае селектор состоит из элемента (**p**), разделителя (**#**) и идентификатора (**pink**). С помощью id можно применять стили к точно выбранному элементу т. е. в HTML-коде может быть только один элемент с данным id. То есть, в нашем примере мы не сможем сделать два абзаца с id «pink», т.к. он должен соответствовать только одному абзацу. Для каждого абзаца будет свой идентификатор. Создав id=«green», придаем ему стиль в таблице стилей.

8.3.3. Контекстный селектор

Зачастую при создании web-страницы вам нужно поместить один тег в другой. Чтобы обеспечить в браузере стиль этих тегов, надо применять

```

Код файла Style.css.
body{
background: aqua; color: blue;
} h1{
color:red; }
h2{ color:green;
} p{
color:black; }
p#pink{ color:maroon;
}

```

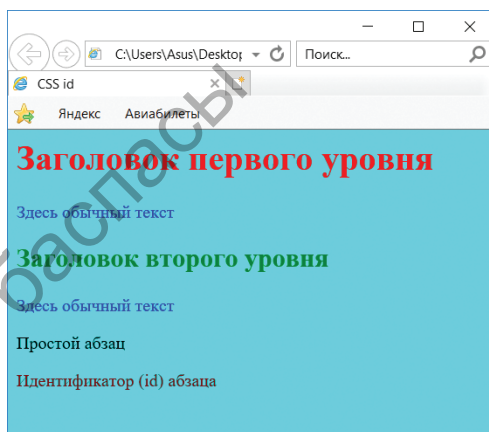


Рисунок 2. Web-страница с использованием цветов

контекстный селектор. Например, тег `` нужно поместить внутри тега абзаца `<p>`. Таким образом можно одновременно поместить стиль одного тега в другой тег.

```
<Ter1>
<Ter2> ... </Ter2>
</Ter1>
```

Контекстный селектор состоит из простых селекторов, разделенных пробелом.

Например, синтаксис для контекстного селектора выглядит следующим образом.

```
<html> <head>
<meta charset=«utf-8»>
<title>Контекстный селектор</title>
<style>
P b: {
font-family: Times New Roman, sans-serif; /* Шрифты*/
color: navy; /* Синий текст*/
}
</style> </head>
<body>
<div><b>Запись текста полужирным шрифтом</b></div>
<p><b> Запись текста цветным полужирным шрифтом </b></p>
</body> </html>
```

В этом примере показано типичное использование тега `` и случай вложения этого тега в тег абзаца `<p>`. Здесь, как показано на рисунке (рис. 2), изменяется цвет и шрифт текста.

Для контекстного селектора возможны случаи, когда вложен не один тег, а два и более последовательных тегов.

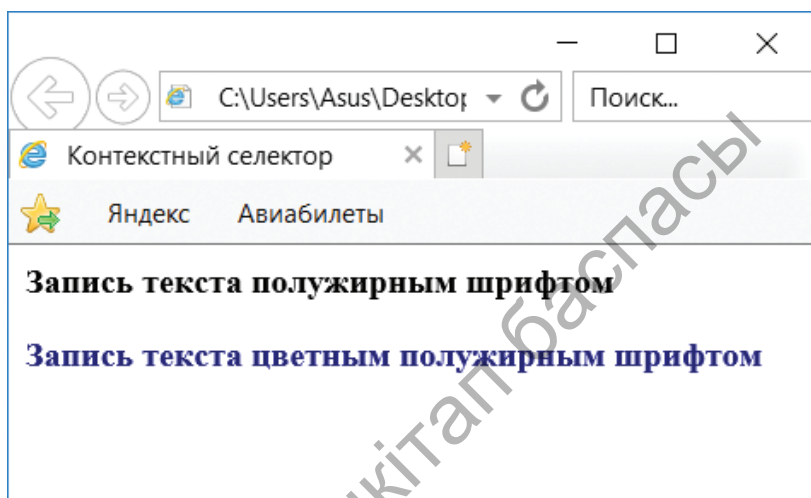


Рисунок 3. Web-страница с отображением полужирного шрифта



Применение

Практическая работа

В качестве практической работы, записав коды web-страниц по рисункам 2 и 3 в программе **Блокнот**, сохраните проект под именем «index.html». При правильном вводе кода HTML откроются web-страницы, как показаны на рисунках 2 и 3.



Анализ



Сравните и проанализируйте четыре типа селекторов. Для анализа используйте диаграмму Венна.



Синтез



Подготовьте и представьте пример, отражающий функциональные возможности и особенности одного из 4 селекторов на выбор.



Оценка



Какова роль селекторов в каскадной таблице стилей? Какие возможности реализуют селекторы при создании web-страниц в HTML? Какими достоинствами обладают селекторы?



1. Какова функция селектора уникальных идентификаторов?
2. Как записываются селекторы идентификаторов?
3. В каких случаях используются контекстные селекторы?



Задания

Задание 1. Контекстные селекторы обеспечивают большие возможности при использовании селекторов идентификатора и класса. Используя код приведенного ниже примера, создайте web-страницу, представленную на рисунке 4. Как показано в этом примере, можно задать стиль только элементу, размещенному внутри определенного класса.

```

<html> <head> <meta charset=«utf-8»>
<title>Контекстный селектор</title>
<style> A {
color: green; /* цвет текста гиперссылки – зеленый */
}
.menu {
padding: 7px; /* границы текста */
border: 1px solid #333; /* параметры границ */
background: #fc0; /* Цвет фона */
} .menu A {
color: navy; /* цвет ссылки */
}
</style> </head> <body>
<div class=«menu»>
<a href=«http://htmlbook.ru/1.html»> Традиции казахского
народа</a> |

```

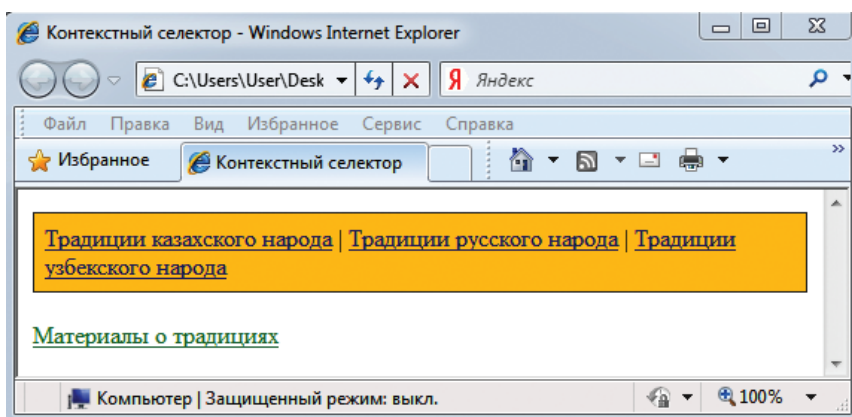


Рисунок 4. Web-страница с контекстным селектором

```

<a href=«http://htmlbook.ru/2.html»> Традиции русского
народа</a> |
<a href=«http://htmlbook.ru/3.html»> Традиции узбекского
народа</a>
</div>
<p><a href=«http://htmlbook.ru/text.html»>Материалы о
традициях
</a></p>
</body> </html>

```

Задание 2. Представьте web-страницу рисунка 4 в виде сайта. Для этого создайте три web-страницы, описывающие традиции каждого народа. Щелкнув мышью по надписи **Традиции ... народа**, вы сможете открыть web-страницу, отражающую традиции выбранного народа.



Как можно использовать HTML-теги для включения мультимедийных объектов в web-страницу?



- ✓ Для чего в web-страницу включают аудио- и видеоматериалы?
- ✓ Какие преимущества у web-страницы с аудио- и видеоматериалами?



Новые знания

Мультимедиа-ресурсы широко распространены во всем мире. Поскольку HTML не сразу имел возможность вставлять мультимедийные объекты на web-страницы, видео- и аудиофайлы рассматриваются как «внешние» объекты, которые перед размещением должны были преобразовываться через конверторы. Видео- и аудиофайлы могут использовать разные форматы. Сегодня современные форматы позволяют проигрывать аудио и видео в режиме реального времени, то есть данные напрямую извлекаются из памяти.

Звуковой фон (элемент BGSOUND)

Включить запись звука в фоновом режиме – самый простой способ оценить озвучивание страницы. Обычно этот метод используется для приветствия или музыкальной поддержки. Фоновый звук добавляется автоматически, когда пользователь загружает web-страницу. Элемент **BGSOUND** и его атрибуты используются для включения аудиофайла в фоновом режиме:

src – обозначает адрес аудио файла. Например, тег `<bgsound src="duet.mid">` определяет разовое проигрывание файла `duet.mid`.

loop – определяет количество проигрываний файла. Например, тег `<bgsound src="duet.mid" loop=5>` обозначает проигрывание файла `duet.mid` 5 раз. Если будет тег `loop="infinite"`, звук играет в течение всего пребывания пользователя на web-сайте.

Вставка аудио и видео (элемент EMBED)

Браузеры поддерживают файлы звукового формата `.AU`, `.MID/.MIDI` и `.WAV`.

` описание ` – фоновый режим не позволяет посетителю web-страницы управлять звуком. Например, он не может отключить фон. Можно управлять звуком только если установить аудиоплеер на web-странице. Он создается в любом месте страницы с тегом `<EMBED>` внутри элемента `BODY`.

Большинство браузеров дают возможность просмотра видеофайлов в часто используемых форматах. *.MPG, *.MPEG, *.QT и *.AVI Для этого, создаются специальные гиперссылки:

** описание **

Точно также наиболее просто установить видео на web-сайте можно с помощью тега <EMBED>. Его запись, атрибуты и запись атрибутов такие же, как и при вставке аудиофайлов.

Например, если вы хотите включить в документ видео с именем video.avi, вам нужно использовать тег <EMBED src = "video.avi">. Если видео- файл вставлен, то видеоклип воспроизводится в окне браузера.

Тег <EMBED> может использовать следующие атрибуты:

| Атрибут | Значение |
|-----------------------|--|
| SRC=name.avi | Наименование звукового, видео клипа |
| WIDTH=X | Ширина вводимого объекта указывается в пикселях (X) |
| HEIGHT=Y | Высота вводимого объекта указывается в пикселях (Y) |
| AUTOSTART=true | Если AUTOSTART не устанавливает значение true, видеоролик не будет воспроизводиться автоматически при загрузке документа и может быть воспроизведен с помощью кнопок управления (TRUE FALSE) |
| REPEAT=TRUE | Разрешает или не разрешает повтор звукового/видео клипа (TRUE FALSE) |
| LOOP | Указывает количество повторений, возможные значения: true, false или число |
| PLAY_LOOP=Z | Если вы установите REPEAT = TRUE, укажите количество повторений вместо Z |
| HIDDEN | Позволяет скрыть панель управления, принимает значение true или false |
| BGCOLOR | Обозначает фон устанавливаемого объекта |
| ALT | Предоставляет альтернативное содержание |
| TYPE | Показывает тип мультимедийного файла |
| QUALITY | Показывает качество мультимедийного файла |
| PLUGINSPPAGE | Отображает URL-адрес плагина для воспроизведения мультимедийных файлов |



Применение

Практическая работа

Задание 1. Откройте аудио или видеофайл с помощью ссылки на web-страницу. Для этого запишите нижеследующие строки в программу Блокнот и сохраните web-страницу под именем «index.html». При открытии файла «index.html» появляется web-страница. Вставьте пиктограммы звукового и видеофайлов (рис. 1).

Во время выполнения задания:

1. Научимся вводить в web-страницу элемент звукового фона BGSOUND.

<bg sound src=«наименование звукового файла»>

2. Научимся открывать звуковой и видеофайлы с помощью гиперссылок.

** описание **

```
<html>
<head>
<title> Ввод звукового и видеофайлов </title>
</head> <body bgcolor=lime>
<h2 align=center> Пуск звукового и видеофайлов</h2>
<p><i>Воспроизведение аудио файла при загрузке страницы:</i>
<bg sound src=«001.mp3» loop=infinite></p>
<i>Ссылка на видеофайл:</i>
<a href=«002.avi»> <img src=»005.jpg» width=30 height=20 border=2
align=middle </a>
<p><i>Ссылка на аудио файл:</i></p>
<a href=«004.mp3»><img src=«006.jpg» width=30 height=20 border=2
align=middle </a>
</body>
</html>
```

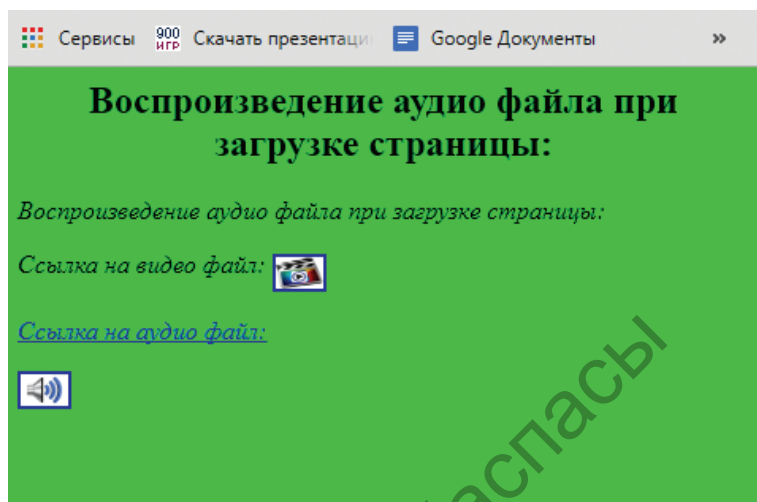


Рисунок 1. Результат выполнения задания 1

Задание 2. Создайте web-страницу, посвященную празднику Наурыз (рис. 2). Разместите на этой странице информацию о празднике и видеоклип. Набрав коды HTML, предназначенные для отражения в браузере, сохраните под именем «index2.html». Для нормального отражения шрифтов на web-странице, в поле Кодировка нужно выбрать «Юникод».

При выполнении задания:

Если видеофайл вставлен, видеоклип воспроизводится в окне браузера.

<EMBED src= "наименование видео ">

```
<html>
<head>
<title>Праздник Наурыз</title>
</head>
<body bgcolor=gold>
<h2 align=center> Праздник Наурыз </h2>
<p> Наурыз – праздник начала года, имеющий древнее происхождение. Празднуется в день весеннего равноденствия (22 марта). (Википедия – информация из открытой энциклопедии)</p>
<p><I>Видеоклип о Наурызе</I></p>
<p align=center> <embed src= " 003.mp4 " width=300 height=200> </p>
</body>
</html>
```

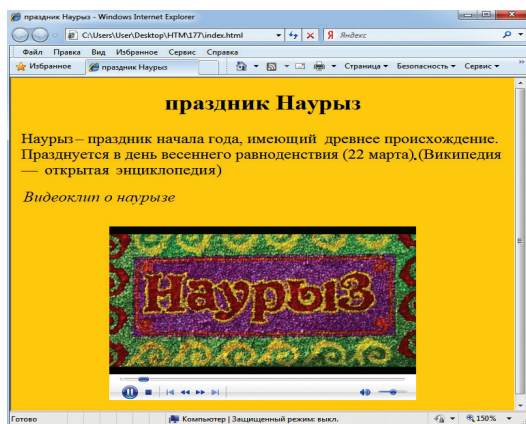


Рисунок 2. Результат выполнения 2 задания



1. Какой элемент используется для ввода звукового фона?
2. Какой тег используется для ввода бегущей строки в документе?
3. Можно ли использовать изображение в бегущей строке?
4. Какие атрибуты есть в бегущей строке?
5. Как называется элемент, используемый для ввода аудио и видео?



Задания

Задание 1. Проект «С ДНЕМ РОЖДЕНИЯ!». Создайте праздничную web-страницу для своего дня рождения, используя рассмотренные теги и атрибуты. Вставьте соответствующую песню и видеоклип. Сделайте дизайн по своему усмотрению.



Задание 2. Создайте web-страницу, содержащую 10 картинок, описывающих историю Казахстана. Создайте музыкальное сопровождение при открытии web-страницы.



Как используются скрипты при создании web-страниц?



- ✓ Как вы думаете, какими установками надо руководствоваться при создании web-страниц?
- ✓ Что такое скрипт?



Новые знания

Клиентские скрипты используются при загрузке документов HTML или в других случаях, например, когда активизируется ссылка. JavaScript, CSS и DOM вместе с HTML создали концепцию DHTML.

Язык управления сценариями просмотра гипертекстовых страниц web на стороне клиента **Javascript** создал **Брендан Эйх** из компании Netscape Communications. Это сценарный язык программирования, предоставляющий доступ к объектам приложений.

Язык Javascript основан на объектах. Объект – это набор свойств, и каждое свойство состоит из имени и значения. Значением свойства может быть функция, которую можно назвать методом объекта. Объект Javascript – это неупорядоченный набор свойств, каждое из которых имеет атрибуты, определяющих, как это свойство может использоваться. Как и все переменные JavaScript, имя объекта (которое может быть переменной) и имя свойства являются чувствительными к регистру. JavaScript содержит несколько встроенных объектов, таких, как Global, Object, Error, Function, Array, String, Boolean, Number, Math, Date, RegExp. В дополнение к встроенным в браузер объектам, вы можете определить свои собственные объекты. Кроме того, JavaScript содержит набор встроенных операций, которые не обязательно являются функциями или методами, а также набор встроенных операторов, управляющих логикой выполнения программ.

JavaScript используется для создания веб-надстроек и динамических сайтов, взаимодействующих с пользователем.

Скрипты (сценарии) могут быть **внешними (js-файлы)** или **внутренними (<script> элементы)**. Элемент **<Script>** можно разместить в раздел **<head>** web-страницы или **<body>** (прямо перед закрывающим тегом **</body>**). Сценарии могут находиться в форме и обрабатывать ввод данных пользователем. Скрипты могут запускаться событиями, происходящими в документе: загрузка документа, выгрузка, передача фокуса элементу, движение мыши и т.д.

Для добавления скрипта к странице, его нужно разместить между открывающим и закрывающим тегом **<<script> </script>** или сослаться на внешний файл скрипта, используя атрибут **src**.

JavaScript – это фрагмент кода, который выполняет некоторые действия на web-странице и задания клиентов и не является автономным языком. Для ввода кода JavaScript в документ HTML:

```

<script language=«javascript»
<!--
Здесь вставляется код скрипта
// -->
</script>

```

Первая и последняя строка фрагмента кода указывают на JavaScript. “ <!--” и “// -->” “// ->” – теги-комментарии, которые используются для соответствия типам браузера, которые не могут обрабатывать коды JavaScript. Такой браузер выводит на экран весь код скрипта.

Второй способ использования скриптов в документе – назначить обработчики событий различным элементам HTML-документа. Это делается с помощью специальных свойств сценария, которые можно использовать во многих тегах HTML. Один элемент документа может быть назначен нескольким редакторам для каждого сценария. Рассмотрим эти атрибуты подробнее (таблица 1):

Таблица 1

| | |
|--------------------|--|
| OnLoad | Браузер завершит HTML-документ |
| OnClick | Пользователь щелкает мышью на элемент |
| OnDbClick | Пользователь дважды щелкает мышью на элемент |
| OnMouseDown | Пользователь щелкает мышью |
| OnMouseOver | Пользователь двигает мышь по элементу |
| OnMouseOut | Пользователь убирает мышь с элемента |
| OnFocus | Фокусировка на вводе элемента |
| OnBlur | Потеря элемента фокуса |
| OnKeyPress | Пользователь нажимает на кнопку клавиатуры и вновь отпускает |
| OnKeyDown | Пользователь нажимает на кнопку клавиатуры снизу элемента |
| OnKeyUp | Пользователь нажимает на кнопку клавиатуры поверх элемента |
| OnSubmit | Отправляет данные формы на web-сервер |
| OnReset | Очистка формы |
| OnSelect | Пользователь выделяет текст в текстовом поле |

Если в браузере не используется JavaScript, стоит применять следующий код:

```
<noscript>  
<p align="center" >Ваш браузер не поддерживает JavaScript</p>  
</noscript>
```

С помощью скриптов можно легко создать кнопку. Для этого достаточно записать нижеследующий код (рис. 1).

```
<input type = "button" value = "Кнопка" name = "simple_but" >
```

Что нужно делать, чтобы эта кнопка заработала, заработала при нажатии? Это очень просто! Вы уже знакомы со сценарием onclick. Использовать этот сценарий для кнопки несложно:

```
<input type = "button" value = "Кнопка" name = "simple_but" >  
onclick = "alert ('Нажмите на Кнопку')">>
```

Посмотрим, как работает кнопка, созданная по приведенному выше коду. Обратите внимание, при использовании события onclick запись действия представляется в двойных кавычках, параметры функции **Alert** записываются в апострофах. Мы видим, что код JavaScript начинается с **<input type = ...>**, но он также может подключаться отдельной функцией. Синтаксис любой функции выглядит следующим образом:

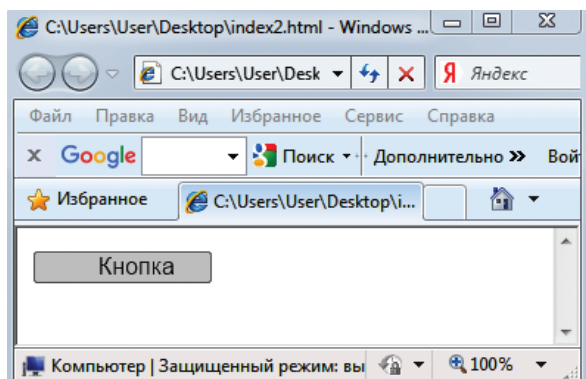


Рисунок 1. Web-страница с изображением кнопки

```
<script language="JavaScript">  
<!--  
function bring_alert()  
{  
alert('Текст');  
}  
!-->  
</script>
```

Тогда вышеописанный код изменим следующим образом:

```
<input type = "button" value = "Кнопка" name = "simple_but" >  
onclick = "bring_alert()">
```

При нажатии на кнопку (**OnClick**) вызывается функция под именем **bring_alert**.

Объекты **new** создаются вызовом свойств конструктора, например, **new String** («эта строка») создает новый объект **String**. Вызов конструктора без **new** зависит от пользователя. Этот **String** («эта строка») создает не объект, а только новую строку. Например, при выполнении нижеследующего скрипта.

```
<p>Для выполнения определенного скрипта нажми на нижеуказанную
кнопку. </p>
<input type="button" value="Нажми на кнопку" onclick="showMessage()"/>
<script type="text/javascript">
function showMessage()
{
alert('Ты нажал на кнопку.')
}
</script>
```

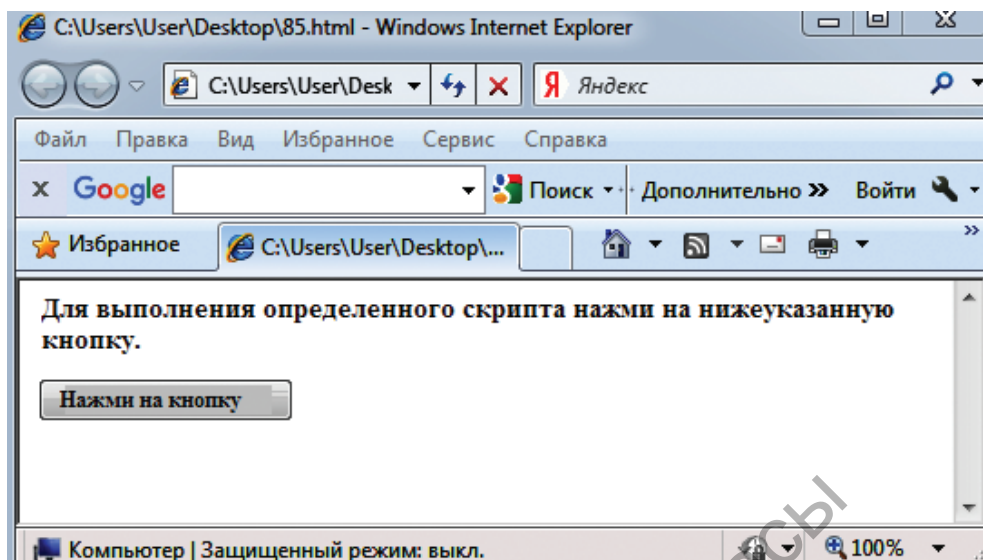


Рисунок 2. Web-страница с использованием кнопки



Применение

Практическая работа

В качестве практической работы, записав коды по рисункам 1 и 2 в программе **Блокнот**, сохраните под именем «index.html». При правильном вводе кода HTML можно увидеть результат выполнения кода в браузере. (рисунки 1 и 2)



1. Что такое скрипт?
2. Какую роль играют скрипты при создании web-страниц?
3. На каком языке записываются скрипты?
4. Каковы особенности языка JavaScript?
5. Как в JavaScript обрабатываются сценарии?
6. Как работает функция Alert?



Задание

Задание.

Текст после оператора **document.write** с определенными свойствами, отобразится в окне браузера.

Пример.

```
<html>
<body>
<script language="javascript ">
document.write("<font color= 'red'">Это текст красного цвета</font>»)
</script>
</body>
</html
```

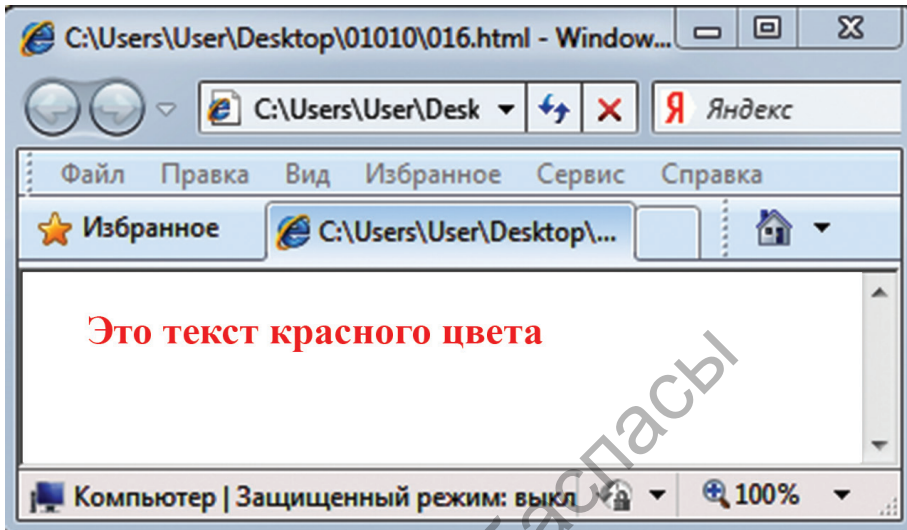


Рисунок 3. Окно выполнения задания



Как можно использовать язык HTML на практике?



- ✓ Какие вы можете назвать возможности и перспективы использования для HTML?
- ✓ Какие вы бы хотели выполнить проекты с использованием HTML?



Практическая работа №1

Разработка web-страницы на тему «Заповедники Казахстана»

Создайте проект web-страницы на тему «Заповедники Казахстана». Убедитесь, что цвет шрифта и рамки в проекте соответствуют образцам (источник данных: <https://kk.wikipedia.org/wiki/>).

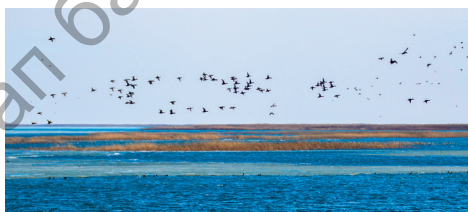
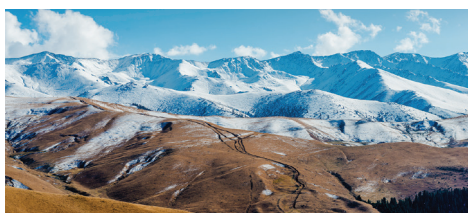
Загруженная web-страница должна иметь следующий вид.

Заповедники Казахстана

Сегодня в нашей стране имеется 10 заповедников. Общая площадь заповедников в Казахстане составляет 1 610 973 га. Этого, конечно, недостаточно, чтобы полностью отразить разнообразие природных условий в Казахстане. Так, в будущем планируется организовать 15 заповедников в стране по рекомендации ученых, представителей охраны окружающей среды.

Список заповедников Казахстана:

1. Аксу-Жабаглинский
2. Наурзумский
3. Алматинский
4. Барсакельмесский
5. Коргалжынский
6. Маркакольский
7. Устюртский
8. Западно-Алтайский
9. Алакольский
10. Каратауский





Практическая работа №2

Проект ввода чисел

Введите данный ниже код HTML. Получите результат, показанный на рисунке 1.

HTML код проекта

```
<!doctype html>
<html> <head>
<meta charset="utf-8">
<title>Ввод числа</title>
</head>
<body>
<p> Введите число из промежутка от 1 до 10:
<input type="number" max="10" min="1" value="1" style="width:33px"></p>
</body> </html>
```

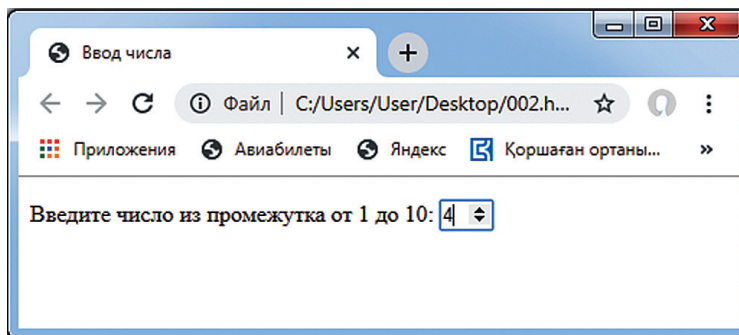


Рисунок 1



Практическая работа №3

Батарейка

Введите данный ниже код HTML. В результате получите три разные батарейки, показанные на рисунке 2.

HTML код проекта

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Батарейка</title>
<style>
.b {
```

```

width: 150px; /* Ширина */
height: 50px; /* Высота */
border: 4px solid #000; /* Параметры рамки */
margin-bottom: 10px; /* Отступ снизу */
position: relative; /* Относительное позиционирование */
}
.b::before, .b::after {
content: «»;
position: absolute;
}
.b::after { /* Добавляем контакт справа */
width: 10px;
height: 20px;
background: #000;
right: -10px; top: 15px;
}
.b::before { height: 100%; }
.low::before { /* Красная батарейка */
width: 33%;
background: #ED1C24;
}
.medium::before { /* Жёлтая батарейка */
width: 66%;
background: #FFC60B;
}

```

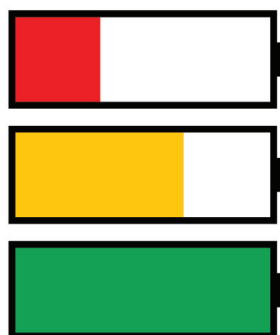


Рисунок 2



Практическая работа №4

Проект «Квадраты»

Сделайте набор квадратов, у которых меняется цвет заливки (рис. 3). При наведении на любой квадрат его цвет меняется (рис. 4).



Рисунок 3



Рисунок 4

HTML код проекта

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Квадраты</title>
<style>
div {
width: 70px; height: 70px;
display: inline-block;
margin-right: 5px;
}
div:hover { background: #ff9f19; }
.b { background: lime; }
.r { background: #4B0082; }
</style>
</head>
<body>
<div class="b"></div>
<div class="r"></div>
<div class="b"></div>
<div class="r"></div>
<div class="b"></div>
</body>
</html>
```

Проанализируйте HTML-коды всех проектов. Оцените функцию каждой команды, использованной в коде.



Как настроить соединение веб-страницы с базой данных?



- ✓ Как хранить большие объемы информации на сайте?
- ✓ Как вы думаете, зачем нужно связывать сайт с базой данных?



Новые знания

При изучении прошлых тем вы узнали, как создать сайт с помощью HTML. Чем больше объем информации, размещаемый на сайте, тем труднее управлять сайтом, тем сложнее его загрузка в сеть Интернет. Вот база данных и поможет в решении этой проблемы. Вы можете использовать базы данных для сохранения необходимой информации, которую в нужное время можно будет запросить (рис. 1). Конечно, простые сайты могут функционировать и без баз данных. А крупным сайтам, Интернет-магазинам, архивам, коммерческим сайтам, сайтам образовательных учреждений без связи с базами данных не обойтись.

Что такое база данных? База данных состоит из набора таблиц, выбранных в соответствии с содержанием сайта. Вы можете ввести в эти таблицы не только текст, но и любой код. Необходимые данные в базе данных можно получить с помощью SQL-запросов с использованием PHP-кода.

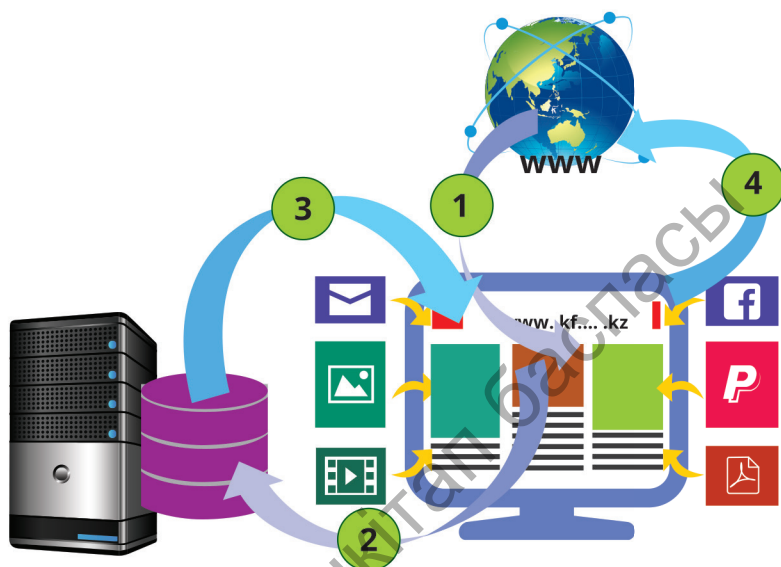


Рисунок 1. Схема использования БД на компьютере

Преимущества связи сайтов с базами данных:

- систематизация данных на сайте;
- облегчается администрирование сайта, сайт становится более гибким, можно настраивать контент по своему усмотрению и отображать его в отдельном блоке;
- есть возможность создать список данных на сайте.

Рассмотрим, как связать сайт с базой данных. Для управления базой данных используем локальные данные на компьютере. Она расположена по следующему адресу:

<http://localhost/Tools/phpMyAdmin/>

Когда вы вводите этот адрес в браузере, на компьютере появляется окно базы данных (рис. 2).

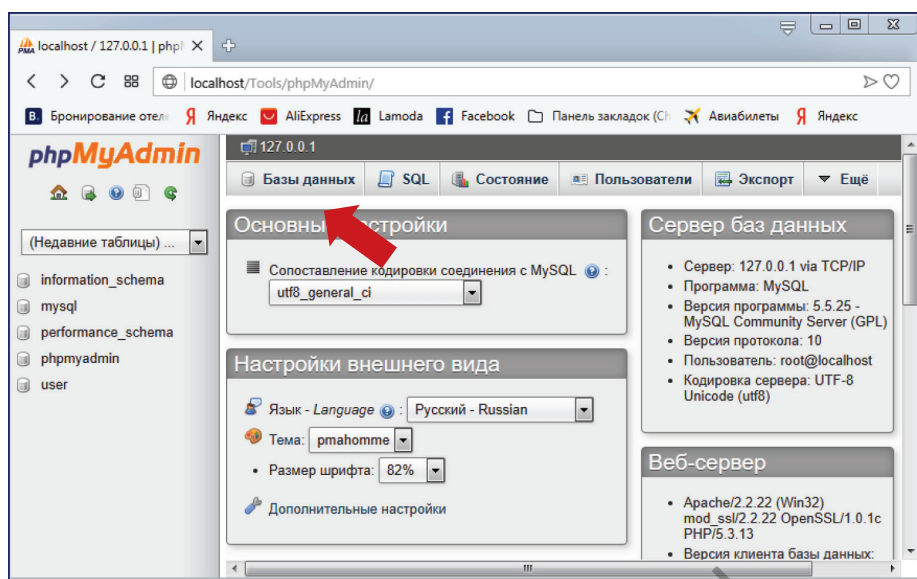


Рисунок 2

PhpMyAdmin – это интерфейс, который позволяет работать с базами данных. База данных MySQL – основной инструмент для создания динамических сайтов. Правила его использования следующие: на HTML создается сайт, по мере необходимости он отправляет запросы к базе данных с помощью PHP-скриптов и извлекает необходимые данные из базы данных или записывает в базу данных. База данных на компьютере может не работать при вводе адреса <http://localhost/Tools/phpMyAdmin/> в браузер. В этом случае вам необходимо установить **Denwer** на компьютер. **Denwer** – программное обеспечение для создания и восстановления сайтов на ПК с операционной системой Windows, без подключения к сети Интернет.

После запуска интерфейса **PhpMyAdmin**, начинаем создавать базу данных. Нажмите кнопку «База данных» (рис. 2). В поле **Создать** надо задать имя базы данных (рис. 3). На картинке мы видим, как создается база данных с именем **User**.

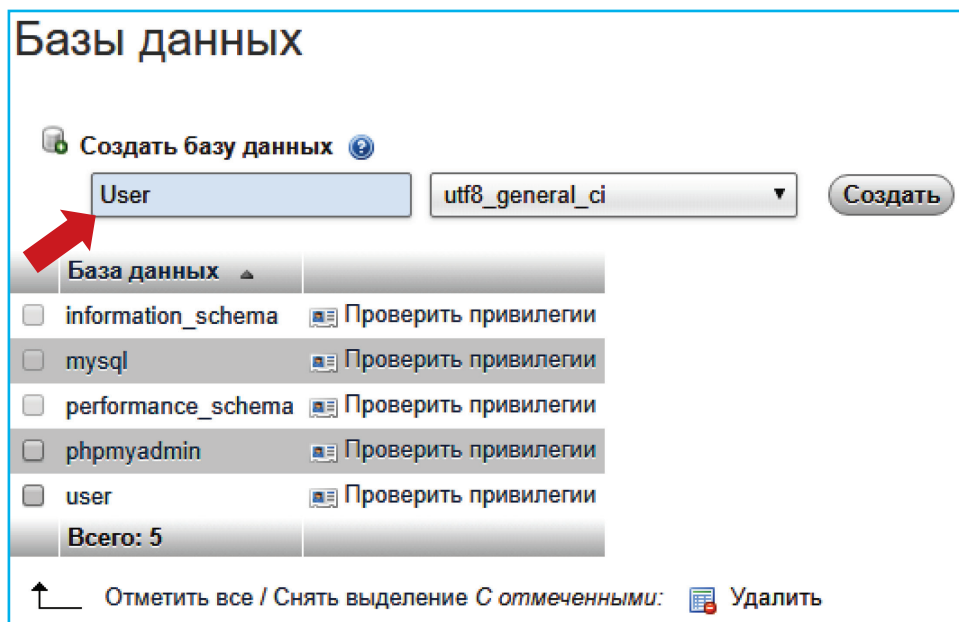


Рисунок 3

Следующий этап – надо создать таблицу для размещения соответствующих данных в базе данных **User**. Откройте базу данных **User** и нажмите кнопку «**Создать таблицу**». В появившемся поле **Поля** необходимо указать количество столбцов и строк таблицы. Следующий шаг – нужно указать типы данных, которые можно вводить в ячейки поля. Например, введем логин, пароль и адрес электронной почты для зарегистрированных пользователей на сайте, который мы организуем.

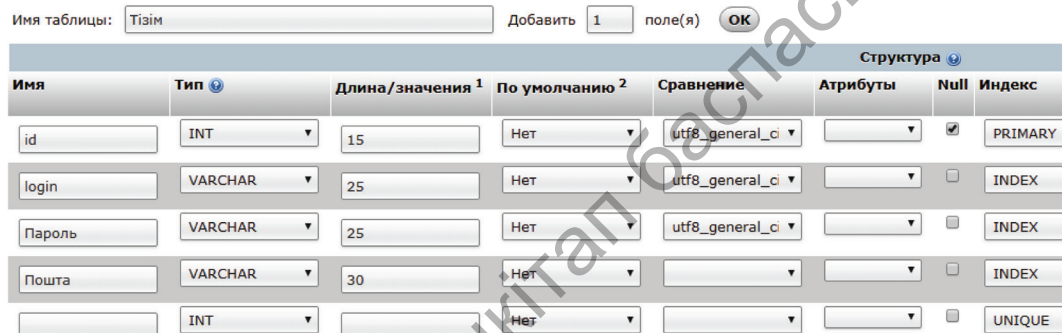


Рисунок 4

Первое поле начинается с идентификатора **id** (другими словами, с нумерации) (рис. 4). В следующем поле выбираются типы имен, написанных в указанной области. Длина значения задает длину значений типов. Например, во втором поле по названию **login** тип **VARCHAR** (не более 255 символов) является символьной величиной. Потому что **login** может состоять не только из символов, но и из цифр. Аналогичным образом, выбираются типы полей «Пароль» и «Почта». Поскольку поле 1 является ключевым, мы выбираем первичный ключ **PRIMARY** в столбце индексов.

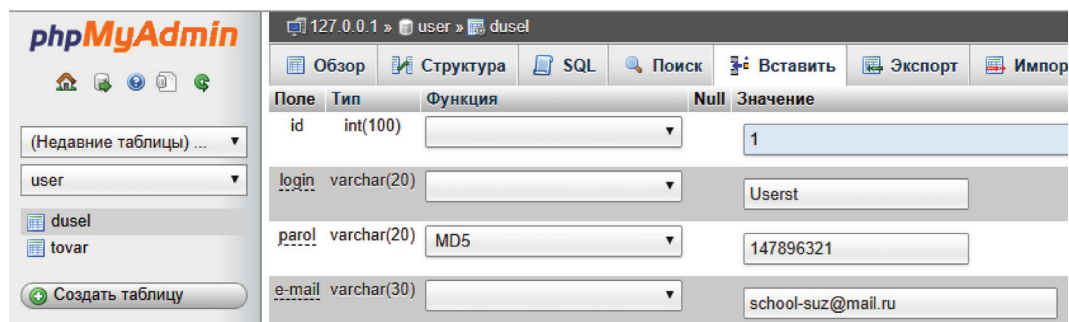


Рисунок 5

После выбора типа данных, можно заполнить поля таблицы данными, нажав кнопку «Вставить», расположенную в меню (рис. 5). И как только соответствующие поля заполнены данными, сохраняем таблицу. База данных на компьютере готова. При необходимости вы можете разместить эти данные в сети Интернет с помощью кнопки Экспорт.



Синтез



Создайте базу данных населения по городам Казахстана.



Оценка



Какова важность интеграции сайта и базы данных?



Как настроить подключение web-сайтов к базе данных?



- ✓ Какие сайты с базами данных вы можете назвать?
- ✓ Каким темам посвящены эти сайты?
- ✓ Как мы связываем базу данных и сайт?
- ✓ Вы знакомы с понятием динамический сайт?



Новые знания

Вы уже научились создавать базы данных на компьютере. Теперь давайте рассмотрим, как связать разработанный нами сайт с базой данных.

PHP является одним из самых мощных инструментов для создания web-страниц. С его помощью создаются динамические сайты.

Динамический сайт – это сайт, состоящий из динамических страниц, контента и скриптов, постоянно изменяемых и обновляемых. Эти файлы пишутся на таких языках программирования как PHP, Perl и так далее.

PHP – это язык программирования, созданный для генерации HTML-страниц на web-сервере и работы с базами данных. PHP – это очень легкий и простой для изучения язык. У него большие возможности, он универсален и встроен во все хостинги. PHP был изобретен Расмусом Лердорфом в 1994 году. Имя PHP означает Personal Home Page (персональная домашняя страница). PHP-код (скрипт) записывается внутри HTML-страницы, интерпретируется и выполняется на самом сервере.

Файл PHP состоит из текста, тегов HTML и скриптов PHP. Все скрипты выполняются на сервере, а результат отправляется в виде HTML в браузер, и уже браузер показывает его пользователю на странице.

Для записи скриптов мы будем использовать редактор **Notepad ++**. Программа PHP включает в себя класс **mysqli** и имеет методы, возможности и функции работы с базами данных **MySQL**. Эта функция и методы используются для ввода данных в базу данных и просмотра ее содержимого. Чтобы добавить информацию в базу данных с помощью web-интерфейса, пользователь должен представить данные в формате HTML, отправить их на сервер, а остальное программа выполнит сама.

MySQL – это небольшой, многопоточный сервер баз данных. MySQL стабилен, прост в использовании и чрезвычайно быстр. Его используют вместе со скриптами PHP при написании сайта.

Возможности MySQL

MySQL использует запросы стандарта ANSI 92 SQL. Имеет следующие возможности:

1. Неограниченное количество пользователей могут работать с базами данных одновременно.
2. Количество строк в таблице может достигать 50 миллионов.
3. Команды выполняются очень быстро. MySQL – самый быстрый сервер среди современных серверов.
4. Простая и эффективная безопасная система.

Установка соединения с базой данных

Доступ к серверу системы управления базами данных **MySQL** можно получить, используя ту же структуру, что и в классе **mysqli**.

```
$ mysqli = new mysqli ('host', 'username', 'passwd', 'dbname', 'port')
```

Возвращает объект, который представляет соединение с сервером MySQL после вызова конструктора класса MySQL. Или:

```
$mysqli = new mysqli();  
$mysqli->real_connect ('host', 'username', 'passwd', 'dbname', 'port')
```

В этом скрипте **host** – название хостинга к которому подсоединяешься, **username** – имя пользователя, **passwd** – пароль пользователя и **dbname** – имя базы данных, к которой посылается запрос.

Команда **\$mysqli->close()** используется для завершения соединения с сервером. Например, пользователь **student** (пароль = "123") должен набрать следующий скрипт для подключения к базе данных локального сервера.

```
<?php  
$mysqli = new mysqli("localhost", "student", "123", "db");  
if (mysqli_connect_errno())  
{ printf ("Байланыс орнатылмады: %s\n", mysqli_connect_error());  
  exit();}  
$mysqli->close();  
?>
```

Если соединение с базой данных не установлено, появится сообщение «Ошибка подключения» и номер ошибки.



Применение

Практическая работа

Соедините ваш компьютер с локальной базой данных по адресу **http://localhost/Tools/phpMyAdmin/**. Создайте базу данных названий рек и их длин в стране, используя интерфейс **PhpMyAdmin** и назовите базу данных «river». Введите и сохраните информацию о пяти основных реках нашей страны в базе данных. Установить связь с базой данных с помощью программы PHP.



Синтез



Проанализируйте один сайт с базой данных, используя сеть Интернет. Определите, какие данные содержатся в базе данных этого сайта?



Оценка



Оцените важность, значимость, ценность процессора гипертекста PHP в создании сайта и его интеграции с базой данных.



1. В чем заключается функция PHP?
2. Что такое динамический сайт?
3. Какие команды используются для подключения к базе данных?



Задание

В рамках проекта «100 книг» создайте базу данных, состоящую из 5 книг, которые вы любите читать. Напишите PHP код, чтобы установить соединение с базой данных.

| П/н | Название произведения | Автор | Год переиздания |
|-----|-----------------------|----------------|-----------------|
| 1 | Путь Абая | Мухтар Ауезов | 2006 |
| 2 | Солдат из Казахстана | Габит Мусрепов | 2007 |
| | ... | ... | ... |



Как настроить подключение веб-страницы к базе данных?



✓ Какие действия вы можете выполнить с базой данных?



Новые знания

При изучении предыдущей темы вы научились связывать компьютер с базой данных. Теперь давайте рассмотрим, какие действия мы можем с базой данных выполнить.

Любая PHP-программа, которая работает с базой данных MySQL, должна включать следующие основные действия:

1. Подключение к серверу MySQL.
2. Выбор базы данных для работы.
3. Отправка команд MySQL на сервер и получение ответа.
4. Обработка результатов.
5. Завершение соединения с сервером MySQL.

Эти операции могут быть выполнены несколькими способами:

1. Использование стандартных классических функций для доступа к базе данных MySQL.
2. Использование функций расширения `mysqli` из PHP5 и MySQL 4.1 и выше.
3. Использование классов расширения `mysqli` с версиями PHP 5 и MySQL до 4.1 и выше.

Сейчас подключение к базе данных MySQL и использование функций из расширения `Mysqli` ускоряет выполнение большинства действий баз данных, что иногда приводит к повышению производительности по сравнению с классическим методом в несколько раз. Вместе с тем повышает безопасность данных и добавляет новые возможности. Следующий код (код 1) показывает соединение с базой данных MySQL и выполнение запроса по площади пяти стран мира.

```
Код 1
?php
/* подсоединение к серверу MySQL */
$link = mysqli_connect(
    <localhost>, /* Название хоста*/
```

```

    <user>,    /* Имя пользователя */
    <password>, /* Пароль */
    <world>); /* Название базы данных */
if (!$link) {
    echo «Нет соединения с базой данных «.mysqli_connect_error();
    exit; }
/* Отправка запроса на сервер*/
if ($result = mysqli_query($link, 'SELECT Name, Area FROM Country ORDER BY
Area DESC LIMIT 5')) {
    echo «Самые крупные страны в мире <br>»;
    /* Получение результата */
    while( $row = mysqli_fetch_assoc($result) ){
        echo $row['Name'].» Площадь земли ". $row['Area'].» км<sup>2</sup>.<br>»;
    }
    /* Освобождение памяти */
    mysqli_free_result($result);
}
/* Завершение соединения с базой данных */
mysqli_close($link);
?>

```

Использование шаблонов запросов для соединения с базой данных

Использование готовых шаблонов запросов или шаблонов выражений при соединении с базой данных позволяет разработчикам программ создавать безопасный, высокопроизводительный и простой в использовании запрос.

Процесс работы с шаблонами выглядит следующим образом:

1. Подготовить шаблон параметрического запроса для сервера MySQL.
2. Связать переменные PHP с параметрами запроса.
3. Отправление команды к серверу для выполнения запроса.
4. Запрос на загрузку новых данных для указанных переменных (код 2).

```

Код 2
<?php
/* подсоединение к серверу MySQL */
$mysqli = new mysqli(<localhost>, <user>, <password>, <world>);
if (mysqli_connect_errno()) {
    echo " MySQL подсоединение к базе данных ошибочно. Код ошибки:".mysqli_connect_error();
    exit; }
/* Подготовка выражения */
if ($stmt = $mysqli->prepare("SELECT Name, Area FROM Country WHERE Area < ?
LIMIT 5")) {

```

```

$stmt->bind_param("i", $area);
$area = 100000;
$stmt->execute();
/* К подготовленному выражению представление переменных*/
$stmt->bind_result($col1, $col2);
/* Выбор значений */
while ($stmt->fetch()) {
    echo $col1." ауданы ".$col2." км<sup>2</sup>.<br>";}
/* Заккрытие выражений */
$stmt->close();
/* Завершение соединения с базой данных */
mysqli->close();
?>

```



Анализ



Проанализируйте и сравните коды программ 1 и 2. Воспользуясь сетью Интернет, определите функции команд, использованных в коде этих программ.



1. Какие действия выполняются с базой данных?
2. Какие этапы содержит процесс работы с шаблонами?



Задание

| П/н | Тема исследования | Вопросы и направления исследования |
|-----|-----------------------|---|
| 1 | Команды программы PHP | Основные команды программы PHP Основные команды, работающие с базой данных PHP |



Как оценить положительные и отрицательные последствия использования Bigdata?



- ✓ Что означает понятие Bigdata?
- ✓ Каковы преимущества и негативные последствия для человека от чрезмерной информации?



Новые знания

В настоящее время появился и используется термин Bigdata или «большие данные». **Bigdata** – это большой объем данных, появившихся в результате технологических инноваций. Он включает в себя информацию данных абонентов мобильных телефонов, которые хранились в течение многих лет, последовательность запросов поисковых систем, активность в социальных сетях и даже запись камер видеонаблюдения в общественных местах. На сегодня нет четкого определения понятия **Bigdata**. Bigdata – это различные инструменты, подходы и методы обработки как структурированных, так и неструктурированных данных для использования при решении конкретных задач и целей. И мы не можем ограничить их до 100 мегабайт или 10 терабайт. Так что общее название само по себе субъективно. **Bigdata** – это набор технологий для выполнения трех основных операций.

В них входят:

1. Объем.
2. Разнообразие (структурированная, неструктурированная информация, возможность одновременно обрабатывать различные типы данных).
3. Скорость.

Говоря понятным языком, это быстрая обработка огромных объемов данных в разных структурах.

Простейшим примером больших данных является информация, полученная с камер видеонаблюдения.

Имеются миллионы пользователей сайтов социальных сетей, таких как YouTube, Facebook, Vkontakte, а количество действий, которые они выполняют в сети, огромно. Очень важно правильно интерпретировать информацию, так как, к примеру, на YouTube действуют специальные алгоритмы, которые по-



могут зрителям быстро находить нужные ролики. Понятие больших данных и методы их анализа использовались и ранее. Например, в полицейской системе информация со всех камер позволяет найти нужный автомобиль.

Bigdata в области экономики и бизнеса

В последние несколько лет понятие Bigdata нашло свое место в слове бизнес-среды. Прогрессивные деловые люди постоянно имеют дело со все возрастающими информационными потоками. В нашей стране, которая стремится сделать бизнес движущей силой экономики, эта тенденция тоже есть. Эксперты утверждают, что правильно обрабатываемые большие объемы информации помогут понять закономерности происходящих событий и повысить эффективность бизнеса и экономики. Информация, прошедшая всесторонний анализ, ценна для любой компании. При грамотном подходе все эти сведения могут превратиться в мощный инструмент влияния и источник экономии для бизнеса. Конечно, традиционные технологии также способны обрабатывать объемную информацию, но по мере роста количества данных скорости их обработки снижаются. Если объем информации удвоился, то время на их обработку увеличится в четыре раза. А Bigdata действует эффективнее. Например, Microsoft совместно с Siemens разработал «умный» рентгеновский аппарат. Он отправляет захваченное изображение одновременно врачу и в облако (онлайн-хранилище данных). В облаке система анализирует изображения с помощью искусственного интеллекта. Он не только обрабатывает изображения, но и постоянно учится сам (machine learning). В результате диагностируют пациента и врач, и система. И в случае разногласий будет проведена повторная диагностика. Однако, ошибочно мнение, что Bigdata приносит пользу только крупным корпорациям. Сегодня малые и средние предприятия также улучшают свой сервис и расширяют свой бизнес за счет использования Bigdata. В розничной торговле, согласно информации, предоставленной Prime Source, Bigdata увеличила производительность на 49%. А объем продаж консалтингового сектора вырос на 5 миллиардов долларов, и тоже благодаря Bigdata. По мнению экспертов, бизнес имеет четыре этапа: от автоматизации отдельных процессов до использования преимуществ Big data (больших данных).

1

Первый этап – операции. То есть все данные отчетности играют важную роль в развитии бизнеса.

2

Второй этап – время изменять данные и делать прогнозы на основе быстрой аналитики.

3

Третий этап – аналитика. Анализ информации и поиск ответов на актуальные вопросы.

4

Четвертый этап – это развитие. Благодаря бизнес-аналитике оно становится более интенсивным.

Подводя итоги, отметим, что благодаря динамичному развитию цифровых технологий объем информации увеличивается с каждым днем. Эта информация собирается из различных сфер человеческой жизни через цифровые устройства, социальные сети, телефоны, видеокамеры и так далее. И если нам удастся быстро и эффективно обработать такой неструктурированный, постоянно растущий поток информации, то это, безусловно, окажет большое влияние на экономическое развитие страны и повлияет на жизнь каждого человека. Конечно, такое влияние может быть как положительным, так и иметь отрицательные последствия. Это может быть шпионаж, посягательство на личные данные людей, отрицательное влияние на сознание народа.



Анализ



Используя сеть Интернет, проанализируйте следующее:

- Основные источники Bigdata (объемной информации, больших данных).
- Влияние Bigdata на развитие бизнеса.
- Развитие Bigdata в нашей стране.



Синтез



Приведите два примера положительных и отрицательных последствий использования Bigdata. Дайте пояснение.



Оценка



Оцените пользу и вреда от использования Bigdata:

- слишком большого объема информации для человека;
- избыточной информации для общества.



Какие основные функции выполняет база данных MS Access?



- ✓ Как мы храним и обрабатываем большие объемы данных?
- ✓ Что такое база данных?
- ✓ Для чего нужно создавать базы данных?
- ✓ Какие вы знаете программы, предназначенные для создания базы данных?



Новые знания

В настоящее время любое предприятие или любая организация создает и использует базы данных.

Система управления базами данных (СУБД) представляет собой набор программ, которые предназначены для создания базы данных, дополнения ее информацией и работы с нею.

Существует несколько десятков типов систем управления базами данных, многие из которых, по сути, не являются системами управления, а являются специально разработанными языками программирования. Поэтому любой человек, освоивший эти языки, может создавать удобные для пользователя структуры базы данных и добавлять в нее элементы управления.

Требования, предъявляемые к базам данных:

- ✓ защита содержимого базы данных от копирования или уменьшение возможности копирования;
- ✓ простота обновления данных (под операцией обновления понимают добавление, удаление и изменение данных);
- ✓ обеспечение ускоренного доступа (операция поиска, чтения данных или записи их) к базе данных, отдельным элементам в соответствии с правами и требованиями группы пользователей;
- ✓ обеспечение расширения базы данных для соответствия постоянно растущим требованиям;
- ✓ поддержка целостности данных, использование только правильной, проверенной информации;
- ✓ доступ к определенным информационным элементам, к требуемым базам данных для отдельных категорий пользователей;
- ✓ доступ к базе, добавление и корректировка информации в базе данных только для обладателей авторских прав;
- ✓ обеспечение простоты создания элементов данных, редактирования, представления и вычисления в соответствии с потребностями пользователей.

Хранение информации – одна из важнейших функций компьютера. Наиболее распространенным видом хранения информации является база данных. **БД** – это файл специального формата, содержащий информацию заданной структуры. Данные представляются в таблицах.

База данных (БД) – совокупность данных из некоторой области, которые собраны с учетом общих принципов описания данных, хранения и соответствия определенным правилам.

СУБД (Система Управления Базами Данных) – набор программ для автоматизации, организации хранения и использования БД.

По способу организации связи между данными СУБД бывает:

- 1) иерархическая;
- 2) сетевая;
- 3) реляционная.

Иерархическая БД строится по принципу иерархии объектов, причем один объект считается главным, а последующие – подчиненными. Здесь объекты располагаются в виде древовидной (иерархической) структуры. В такой БД запрос направляется по иерархии. Например: файловая система Windows, набор каталогов на интернет-ресурсе.

Сетевая БД – это совокупность данных, в которой каждый объект может быть связан с любым другим объектом. Если эту модель представить в виде графического изображения, то получится схема в виде сети. Например: связанный ссылками Web-документ.

Реляционная БД – совокупность данных, представленных в виде таблицы. Это самая распространенная база данных, связь между таблицами – важная и часто используемая. В БД столбцы называются *полями*, а строки – *записями*. В таблице не должно быть повторяющихся записей. Основная особенность полей – они строятся из элементов одного типа. Это основное различие между реляционной базой данных и табличным процессором Excel.

Большинство БД организованы в виде таблицы. В таблицах адрес данных определяется пересечением строки и столбца. При проектировании БД удобно создавать структуру таблиц предварительно на бумаге. Каждое поле в таблице содержит определенную информацию, соответствующую ее заголовку. Чтобы связать данные одной таблицы с данными другой таблицы, каждая запись в таблице должна определяться полем или набором полей, которые могут однозначно идентифицировать значение каждой записи. Такое поле или набор полей называют *первичным ключом*.

Основы работы с системой управления базами данных Microsoft Access

Стандартная команда запуска MS Access: **Пуск – Программы – Microsoft Office – Microsoft Access 2010**. После загрузки программы MS Access на экране появится диалоговое окно, как показано на рисунке 1.

В этом окне MS Access предлагает пользователю выбор из трех вариантов:

- ❖ Создание новой базы данных;
- ❖ Создание web-базы данных;
- ❖ Открыть имеющуюся базу данных.

Если вам нужно создать новую БД, наведите курсор на **Создание новой БД** и щелкните кнопку «ОК». Только после этого появится Имя новой БД и указатель ее местоположения.

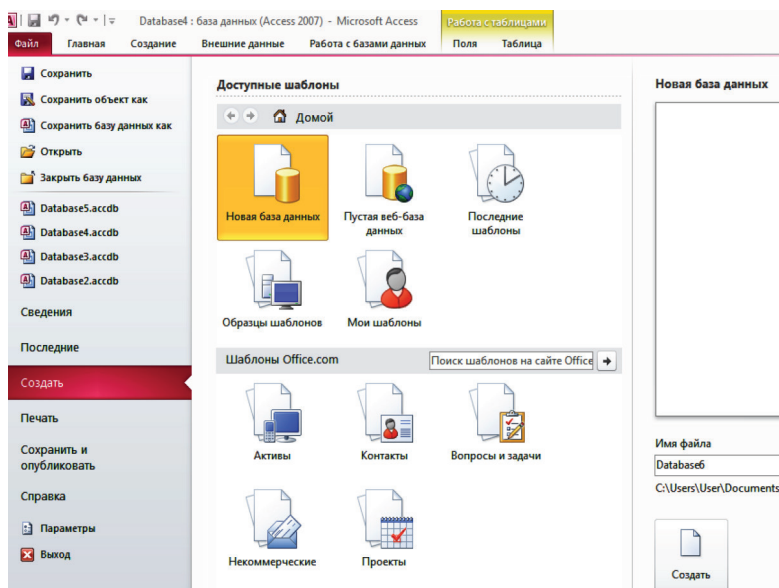


Рисунок 1. Диалоговое окно БД Access 2010

После выполнения последовательности действий: выбор папки, ввод имени файла и задание типа файла, нужно нажать кнопку «**Создать**». На экране появится окно БД (рис. 2).

В программе Access имеется семь объектов: таблица, запрос, форма, отчет, макрос, модуль и страницы.

Таблица – основной объект базы данных, состоящей из полей и записей. В таблицах хранятся данные.

Запрос – объект, который позволяет пользователю получить нужные данные из одной или нескольких таблиц. К основным относятся *запросы на*

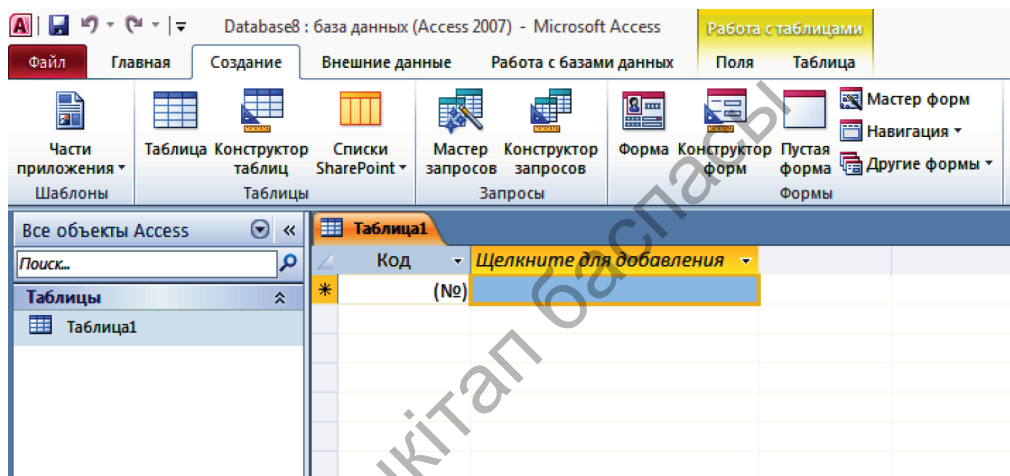


Рисунок 2. Окно «Создать» БД Access 2010

выборку, изменение, обновление, удаление, добавление данных, параметрические и перекрестные.

Форма – объект, предназначенный в основном для ввода данных в удобном виде или управления данными из таблиц или запросов.

Отчет – объект, предназначенный для выбора из базы данных требуемой информации и создания документа, который впоследствии может быть распечатан или включен в документ другого приложения. Отчет после его создания нельзя редактировать.

Макрос – набор из макрокоманд. Если некоторые операции с БД выполняются часто, то можно собрать их в одном макросе, а затем представлять их комбинацией клавиш.

Модуль – внутренние программы, написанные на языке **Visual Basic**. Если стандартные инструменты Access не дают возможности реализовать решение некоторых задач, то программист может расширить возможности системы с помощью внутренних программ.

Страницы позволяют создавать web-документы разных типов. Версия HTML – языки (шаблоны) используются для создания обычного web-документа.

Расширения файлов БД в программе MS Access – ***.mdb (2003)** и ***.accdb (2007, 2010)***.

Microsoft Access – это с точки зрения функциональности полная и реляционная СУБД. Она предоставляет пользователю все необходимые возможности: пользователь может определять и обрабатывать любую информацию, а также работать с достаточно большим объемом информации.

Используя Системы Управления Базами Данных, мы можем обрабатывать определенные данные и контролировать их совместное использование.

Система управления базами данных реализует три основные функции:

- 1) определение данных, описание структуры баз данных;
- 2) обработка данных;
- 3) управление данными.

Все эти функциональные возможности полностью поддерживаются и Microsoft Access.

Определение данных

При работе с любым документом или электронной таблицей мы можем выбрать содержимое документа или местоположение каждой ячейки в самой таблице.

В электронных таблицах возникает необходимость хранения информации, выполнения необходимых вычислений и предоставления результатов в соответствующем виде.

Если количество строк в электронной таблице больше 100 и документ многостраничный, то работать с имеющейся информацией очень сложно. Например, одна ячейка таблицы содержит дату (время), а вторая ячейка предназначена для хранения оплаты в этот день, поэтому пользователь может случайно перепутать ячейки. Можно привести множество таких примеров.

Благодаря СУБД мы можем выбрать тип информации и способы ее хранения. В то же время можем определить основные требования для удобства работы (для правильного ввода информации).

С помощью Microsoft Access можно изменять и редактировать файлы Paradox, dBASE III, dBASE IV, FoxPro и другие. В связи с этим можно импортировать эти файлы в Microsoft Access.

Обработка данных

Работа с данными в текстовом редакторе или электронной таблице значительно отличается от работы в СУБД. Вы можете в документ, подготовленный в текстовом редакторе, вставить таблицу, редактировать в ней информацию. Но документ, созданный в электронной таблице, предназначен для выполнения только одной определенной задачи и приспособить его для выполнения другой задачи сложно. СУБД применяет множество приемов для работы с информацией. С помощью только одной команды можно обновить всю информацию, размещенную в определенной части, например, в таблице базы данных.

Для выполнения запросов в таблице Microsoft Access используют язык SQL. С его помощью можно выбрать или выделить данные из одной или нескольких таблиц.

Управление данными

Текстовые документы и электронные таблицы – удобный инструмент для решения задач конкретного пользователя. Но они непригодны для совместного пользования.

Электронные таблицы предназначены для обработки информации в различной форме, но если мы осуществляем полный комплексный мониторинг информации, то нам будет недостаточно его функций.

Во время совместной работы, благодаря системе управления базами данных, мы можем защититься от несанкционированного доступа случайного пользователя к базе данных и только определенные пользователи могут использовать эти данные, добавлять новые записи. Предназначенная для совместной работы СУБД не позволяет выполнять редактирование данных случайному пользователю (эту информацию не каждый имеет право изменять по своему усмотрению).

Доступ к работе может получить как определенный пользователь, так и группа пользователей.



1. Какова цель создания базы данных?
2. Какие виды баз данных существуют?
3. Что такое иерархическая база данных?
4. Какие есть преимущества сетевой БД перед другими?
5. Что такое реляционная база данных?
6. Какую цель она выполняет?
7. Какие объекты содержит окно «Создать» базы данных MS Access 2010?
8. Охарактеризуйте функции объектов, расположенных в окне «Создать».

Принципы построение баз данных



Какие типы данных можно использовать в записях базы данных?



- ✓ Как мы создаем и храним необходимые нам данные?
- ✓ Как можно создать базы данных с помощью компьютера?



Новые знания

Вы должны указать тип данных, которые вводите в структуре таблицы базы данных. При создании таблиц с помощью команды **Создание – Конструктор таблиц** базы данных необходимо задать имя поля и его тип, описать свойства создаваемой таблицы (рис. 1).

Чтобы определить **тип поля** в режиме конструктора, выберите соответствующую ячейку в столбце «Тип поля», щелкните по символу Δ с правой стороны и выберите нужный тип из раскрывающегося списка. Свойства поля задаются в вкладке Общие. В зависимости от заданных свойств поля определяется, какая информация может вводиться. Свойство, определяющее длину поля, – это *размер поля*.

Он определяет, какой объем информации можно поместить в данное поле.

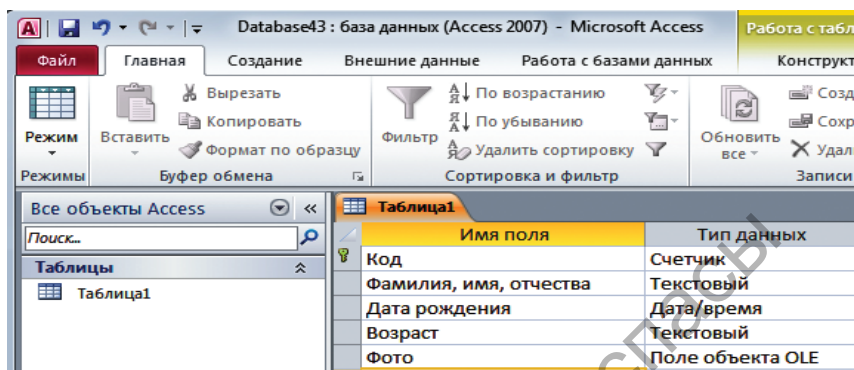


Рисунок 1

Индивидуальная характеристика любого поля – его **подпись**. Подпись – это специальное имя, отображаемое в заголовке поля в таблицах, формах, запросах, отчетах. Разным полям нельзя присвоить одну и ту же подпись. Если это свойство не заполнено, то используется имя поля.

У различных типов полей могут быть соответственно различные свойства. Давайте вкратце проясним каждый тип в списке:

| Тип поля | Характеристика |
|-------------------------------------|---|
| Текстовый тип (по умолчанию) | Используется при вводе текстовой информации. Его основной особенностью является размер. Он не превышает 256 символов |
| Числовой тип | Предназначен для ввода числовой информации. Вы можете выбрать размер этого типа поля из нескольких вариантов |
| Тип дата/ время | Используется для ввода даты и времени. В свойствах поля выбираем формат |
| Денежный тип | Необходим для ввода информации денежного типа. Можно выбрать необходимые единицы валюты из списка формата полей или задать свой шаблон |
| Тип счетчик | Предназначен для определения кода записи. Они представляют собой числовой ряд и их значения автоматически увеличиваются, не повторяясь |
| Тип MEMO | Предназначен для ввода текста, превышающего 256 символов. В полях такого типа можно хранить информацию до 65 535 символов. Особенностью этого типа является то, что информация о поле хранится в другом месте, а в самом поле хранятся указатели мест размещения информации |
| Тип OLE | Предназначен для ввода объектов в базу данных. Для этого используйте протокол OLE, помогающий вставить выбранный объект в поле с помощью буфера обмена. К таким объектам относятся графические файлы, аудио-, видеофайлы |
| Логический тип | Есть только два логических значения: да или нет, верно или неверно, 0 или 1. Тип значений выбирается из списка, данные в формате поля. Какой бы формат не был выбран, логическое поле в таблице содержит один из двух исходов |
| Гиперссылка | Ряд, выявляющий адреса букв, цифр и гиперссылок. Адрес гиперссылки состоит из следующих отделов: текст, вводимый в поле, управляющие элементы и поля адресов URL или файлов. Для задания полю гиперссылки выполняется команда Вставка – Гиперссылка и указывается ее адрес |
| Мастер подстановок | Это не специальный тип данных. Это объект, настройкой которого можно автоматизировать ввод данных в поле так, чтобы не вводить их вручную, а выбирать из раскрывающегося списка |

Создание таблицы базы данных

Предварительно создают структуру (проект) таблицы базы данных. Для выбора типа создаваемого документа:

- в Access выбираем вкладку **Создание** и затем нажимаем на кнопку **Конструктор таблиц**. Мы записываем имена полей, которые создаются в окне **Конструктор таблиц** соответственно таблице 1, рис. 1.

Для определения полей таблицы:

- в поле «Имя поля» введите название первого поля как Код учащегося.
- в строке «Типы данных» нажмите на кнопку списка и выберите тип данных **счетчик**, затем во вкладке **Размер поля** выбираем вид **Целое**.

- Назначаем **ключ** полю *Код учащегося*. Для этого на панели инструментов нажимаем на кнопку с изображением ключа или выбираем из контекстного меню.

- Чтобы не повторялись коды учащихся в поле **Код**, нужно ввести ограничение данных. Для этого: выбрать параметр **ДА** в свойстве **Индексированное поле** вкладки **Общие** (рис. 2).

Примечание. *Индекс* – это инструмент для ускорения поиска информации и запросов в таблице Access. Поле первичного ключа автоматически индексируется. Нельзя индексировать объекты в полях *МЕМО* и *Гиперссылка* или *OLE* (таблица 1).

Таблица 1

| Имя поля | Тип данных | Описание |
|---------------|------------|----------------------|
| Код-учащегося | Счетчик | |
| Фамилия | Текстовый | 20 |
| Имя | Текстовый | 15 |
| Отчество | Текстовый | 25 |
| День рождения | День/время | Формат поля: Краткий |
| Обязанность | Текстовый | 9 |
| Код предмета | Числовой | Целое |
| Предмет | Текстовый | 11 |
| Телефон | Числовой | 12 |

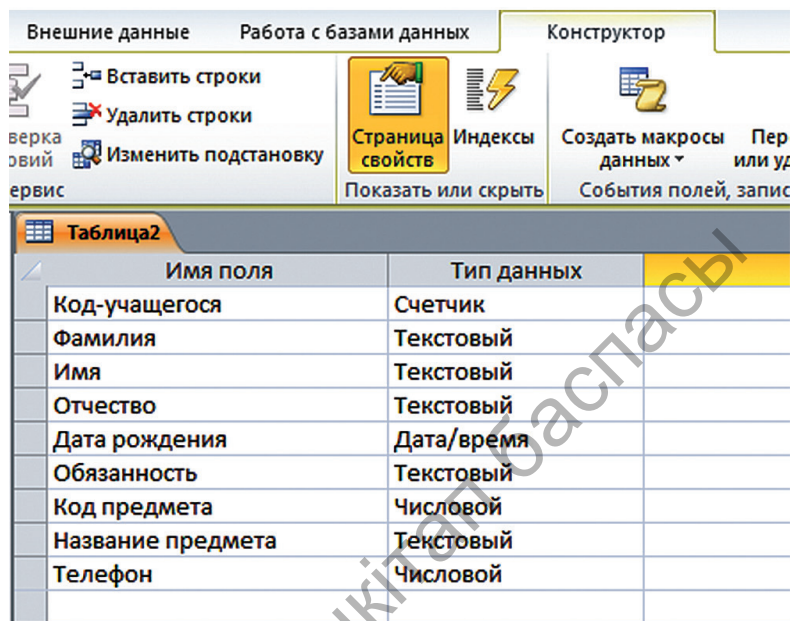


Рисунок 2



Синтез



Создайте таблицу и заполните ее данными с которыми вы встречаетесь ежедневно (например, расписание занятий, успеваемость по предметам и т.д.). Определите типы полей полученной таблицы.



1. Как запустить Конструктор таблиц?
2. Как заполнить данными таблицу, созданную в режиме Конструктора?
3. Как ввести тип поля в режиме Конструктора?
4. Каково основное свойство любого поля?
5. Какие типы полей используются для данных в базе данных?



Задание

1. Создайте базу данных областей нашей страны. Выберите типы данных полей таблицы. При заполнении используйте знания, полученные на уроке географии и информацию из сети Интернет.

| Код области | Название области | Год образования | Площадь области | Количество жителей | Название обл. центра | Процент населения | Фото области |
|-------------|------------------|-----------------|-----------------|--------------------|----------------------|-------------------|--------------|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

2. Создайте базу данных учащихся 10-го класса. Заполните базу данных в соответствии с предложенной структурой таблицы. Определите типы данных полей таблицы.

| Код учащегося | Ф.И.О. | Год рождения | E-mail | Класс обучения | ИНН | Фото учащегося |
|---------------|--------|--------------|--------|----------------|-----|----------------|
| | | | | | | |
| | | | | | | |
| | | | | | | |

3. Создайте базу данных учебных пособий, по которым обучаются учащиеся. Базу данных заполните данными в соответствии с предложенной структурой таблицы. Определите типы данных полей таблицы.

| Код учебных пособий | Наименование | Цена | Количество | Издание | Фото учебного пособия |
|---------------------|--------------|------|------------|---------|-----------------------|
| | | | | | |
| | | | | | |
| | | | | | |



Что такое реляционная база данных?



✓ Какие приемы применяли для сохранения большого объема информации в то время, когда не было компьютеров?



Новые знания

Microsoft Access – система управления реляционной базой данных. Использование **реляционных баз данных** было предложено доктором Э.Ф. Коддом из компании IBM в 1970 году. В настоящее время эта система популярнее СУБД. В основе модели **реляционной базы данных** лежит понятие **«отношение»** (*relation*, англ). Э.Ф. Кодд ввел этот термин как синоним к слову «таблица». Здесь отношение ориентировано на конкретный объект. В свою очередь, *объект* характеризуется набором *атрибутов*, а атрибуты характеризуются значениями (*доменом*). Если *столбец* таблицы соответствует *атрибуту*, *строка* таблицы называется *кортежем*. Таблицы в реляционной базе данных называют **«реляционными таблицами»**.

Простые двумерные таблицы – это реляционные базы данных. Вспомним, что в базе данных *строки* двумерной таблицы называется *записями*, а *столбцы* называются *полями*. Точнее, в таблицах базы данных каждая строка является записью, а запись содержится в нескольких полях.

Данные в базу данных можно внести несколькими способами:

- а) создавая и заполняя таблицы базы данных;
- б) изменяя содержимое созданной базы данных;
- в) создавая и заполняя формы.

Реляционная модель – самая простейшая модель, в которой данные представлены в виде таблиц.

Реляционные таблицы представляют собой двумерные таблицы и набор взаимосвязанных таблиц при создании сложных информационных процессов. В структуру реляционной базы данных должны входить следующие свойства:

В таблице не должно быть двух одинаковых записей.

Все данные в одном поле имеют один тип.

Каждый элемент таблицы является одним из элементов данных.

Каждое поле имеет определенное имя.

Порядок следования строк произвольный и характеризуется количеством полей.

В пересечении строки и столбца таблицы должно быть только одно единственное значение.

Большинство БД имеют табличную структуру. Адреса в структуре таблицы определяются пересечением столбца и строки. При проектировании удобно создавать структуру таблицы предварительно на бумаге. Каждое поле в таблице содержит определенную информацию, соответствующую ее заголовку. Чтобы связать данные одной таблицы с данными другой таблицы, каждая запись в таблице должна содержаться в поле или наборе полей, которые могут однозначно идентифицировать значение каждой записи. Такое поле или набор полей называют **первичным ключом**.

Пример:

Таблица учащихся

| id | Имя |
|----|-------|
| 1 | Арман |
| 2 | Сауле |
| 3 | Марат |

Таблица предметов

| id | Предмет | Оценка |
|----|------------|--------|
| 1 | физика | 5 |
| 1 | математика | 3 |
| 2 | география | 5 |
| 3 | литература | 5 |

Компоненты таблицы

- В каждой таблице несколько компонентов.

- **Атрибут** – наименование столбца.

- Например, **id** и **Имя** будут атрибутами.

В таблице учащихся есть два атрибута.

- **Кортеж** – информация в одной строке.

В таблице три кортежа.

(1, Арман), (2, Сауле), (3, Марат)

| id | Имя |
|----|-------|
| 1 | Арман |
| 2 | Сауле |
| 3 | Марат |

Кортеж

Атрибут

- ✓ Две таблицы могут связываться через id.
- ✓ В обеих таблицах id – первичный ключ.

Кортеж указывает на конкретный объект, а атрибут показывает характеристику данного объекта. В таблице не должно быть двух одинаковых кортежей.



Применение

Практическая работа

Создайте простую реляционную таблицу «Потребитель» в MS Access.

| Код потребителя | Потребитель | Адрес |
|-----------------|-----------------|------------------------|
| АА | 1-й потребитель | улица Желтоксан, №13 |
| АБ | 2-й потребитель | улица Достык, №25 |
| АС | 3-й потребитель | улица Республики, № 28 |
| АД | 4-й потребитель | улица Абая, № 47 |

«Код потребителя» выполняет функцию **первичного ключа**, чтобы связать данные в таблицах и присвоить индивидуальное значение каждой записи (рис. 1).

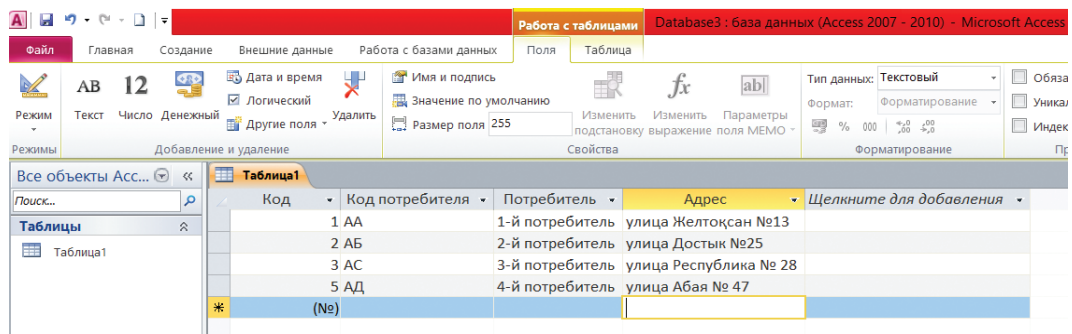


Рисунок 1. «Код потребителя»

При создании базы данных в режиме *Создание таблицы путем ввода данных* имена соответствующих полей вводятся пользователем самостоятельно. Для этого наводим курсор в **Поле1**, а затем нужно дважды щелкнуть левой кнопкой мыши. Тогда активизируется имя поля, заданное по умолчанию (рис. 1). С помощью клавиши «Delete» удаляем исходное имя поля и вводим соответствующие имена полей с клавиатуры.

Когда вы нажмете кнопку **Close (X)**, появится окно диалога – хотите ли вы сохранить изменения в структуре макета объекта или нет. После нажатия кнопки **Да** по умолчанию объект уприсваивается имя **Таблица 1** (при необходимости можно изменить имя). После нажатия кнопки **OK** всплывает диалоговое окно определения ключевого поля. В этом диалоговом окне нажимаем **Да**. Таким образом определим ключевое поле. Ключевое поле назначается коду, также сохраняются номера записей. Если вы вводите новые записи в таблицу, значение ключевого поля заполняется автоматически.

Как вы видите, данные очень удобно вставлять в атрибуты каждой таблицы при вводе данных в виде таблицы. После ввода данных сохраняем таблицу (рис. 2).

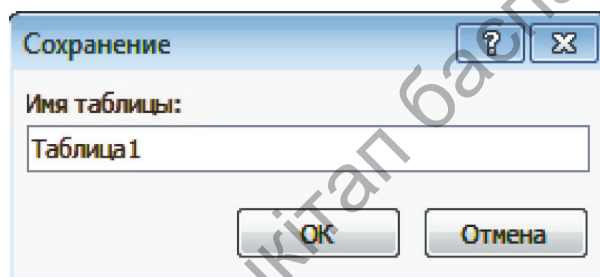


Рисунок 2



1. Какой ученый представил модель реляционной базы данных?
2. Какие два понятия в реляционной базе данных следует рассматривать как синонимы?
3. Что такое реляционная модель?
4. Какие свойства данных определяются в процессе проектирования реляционных баз данных?
5. В чем смысл понятия ключа в базе данных?
6. Какие вы можете назвать компоненты таблицы?



Задания

1. Основным компонентом реляционной базы данных является таблица. Из каких компонентов состоит таблица? Приведите примеры. Что такое атрибут и ключ?
2. Какой атрибут является первичным ключом в таблице учащихся? Может ли имя являться первичным ключом? Может ли фамилия быть первичным ключом?
 - А. Какое поле является первичным ключом в таблице предметов?
 - В. По какому предмету получил оценку Айдос Зиятбек?
 - С. Кто из учащихся получил пятерку по физике?

| Таблица учащихся | | |
|------------------|----------|-----------|
| id | Имя | Фамилия |
| 1 | Айдос | Байдаулет |
| 2 | Айдос | Зиятбек |
| 3 | Магжан | Аманбай |
| 4 | Магжан | Баймбет |
| 5 | Кенжебек | Асанов |

| Таблица предметов | | |
|-------------------|------------|--------|
| id | Предмет | Оценка |
| 1 | Математика | 5 |
| 1 | Физика | 4 |
| 2 | География | 4 |
| 2 | Химия | 3 |
| 3 | Математика | 5 |
| 5 | Физика | 5 |

3. Из таблицы выше установите:
 - 1) Какова связь в таблицах между учащимся и предметом?
 - 2) Каков средний балл у Айдоса Зиятбека?
 - 3) Кто из учащихся пока не получил оценки?
 - 4) Каков средний балл учащихся по математике?



Как можно создать однотабличную базу данных?



✓ Какие есть преимущества в представлении данных в виде таблиц?



Новые знания

Представление данных в табличном виде очень эффективно. Мы часто встречаем разнотипную информацию, которая представлена в виде таблиц. Например, журнал успеваемости, дежурства, химически список элементов, единицы физических величин, исторические даты и т.д.

Таблица (table) – структура хранения данных и представления их на пересечении столбцов и строк (*таблица 1*). Типы данных в столбцах могут быть разными, но данные в одном столбце должны быть одного типа. Таблица позволяет сортировать данные и легко находить нужную информацию.

Таблица 1

Комплекующие компьютера и их поставщики

| № | Наименование | Характеристика | Название фирмы | Адрес | Цена (тенге) |
|---|----------------|----------------|----------------|---------|--------------|
| 1 | Системный блок | Pentium | Фирма 1 | Адрес 1 | 100000 |
| 2 | Системный блок | Pentium | Фирма 2 | Адрес 2 | 90000 |
| 3 | Монитор | 15" | Фирма 1 | Адрес 1 | 50000 |
| 4 | Монитор | 15" | Фирма 2 | Адрес 2 | 60000 |
| 5 | Клавиатура | 104 клавиши | Фирма 1 | Адрес 1 | 4500 |
| 6 | Клавиатура | 104 клавиши | Фирма 2 | Адрес 2 | 4000 |
| 7 | Мышь | 3-кнопочная | Фирма 1 | Адрес 1 | 600 |
| 8 | Мышь | 3-кнопочная | Фирма 2 | Адрес 2 | 550 |

Созданная базы данных может быть как однотабличной, так и много-табличной.

Пример создания таблицы в режиме «Таблица»

Когда вы создаете новую базу данных в MS Access, то вам предлагается создать таблицу непосредственно в режиме таблицы. Здесь первому полю можно присвоить наименование «Код», и оно представляется как ключевое поле. Чтобы переименовать поле, нужно щелкнуть правой кнопкой мыши, выбрать из меню команду «Переименовать» и присвоить новое имя.

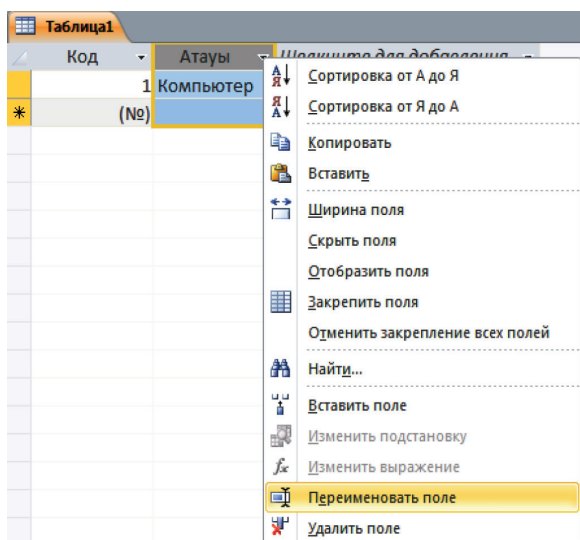


Рисунок 1

Для создания следующего поля нажмите на заголовок **Добавить поле**, выберите из списка тип данных этого поля, укажите его имя (рис. 1).

Типы и размеры для полей, которые присваиваются по умолчанию при создании таблицы в этом режиме: тип данных «Счетчик» и «Числовой» – длинное целое, тип поля «Текстовый» – 255 символов и типу данных «Дата и время» – краткий формат. Для сохранения таблицы нужно справа в диалоговом окне *сохранение* «Таблица 1» щелкнуть значок x, затем всплывет вопрос «сохранить таблицу?».

Далее выполняем следующие действия: отвечаем «Да», в окне диалога «Сохранение» заполняем поле «Имя таблицы», нажимаем «ОК». Таблица сохраняется под заданным именем, и все названия полей таблицы Access отображаются в верхнем ряду таблицы.

Чтобы экономно использовать память и время обработки объема информации базы данных, вы должны щелкнуть по имени таблицы и в режиме «Конструктор таблиц» выбрать минимально необходимые параметры для каждого типа данных заданных полей. Эти параметры представлены в нижней части окна. Конструктор таблиц представлен в разделе «Свойства поля» как «Размер поля» либо «Формат поля». Список возможных вариантов, отображаемых в свойстве, например, «Размер поля» либо

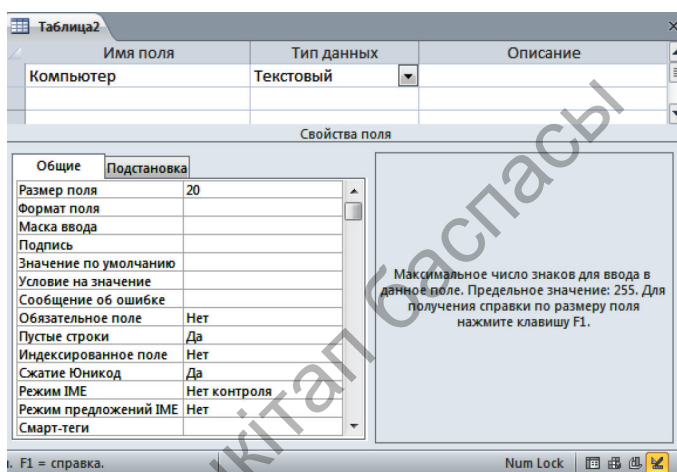


Рисунок 2. «Свойства поля»

«Формат поля», выбирается из списка свойств, которые различаются в зависимости от типа данных, указанных в поле «Тип данных» (рис. 2). Для этого в поле справа от «Размер поля» либо «Формат поля» нужно выбрать соответствующий параметр.

Свойства полей в базах данных

1. Имя поля – сообщение имени поля при выполнении автоматических операций.
2. Тип поля – определяет тип сохраняемых данных в данном поле.
3. Размер поля – указывает длину данных, хранящихся в указанном поле (по символам).
4. Формат поля – определяет формат представления данных ячеек.
5. Маска ввода – определяет способы ввода данных в ячейки.
6. Подпись – определяет заголовок данного поля в таблице.
7. Значение по умолчанию – конкретное значение автоматически вводится в ячейки.
8. Условие на значение – ограничения для проверки правильности введенной информации.
9. Сообщение об ошибке – автоматически исходящее сообщение, когда вводится неверная информации в ячейку.
10. Обязательное поле – определяет, является ли это поле обязательным при заполнении записей базы данных.

Пример создания таблицы в режиме «Конструктор таблиц»

Нажмите на значок x справа от строки «Таблица 1», чтобы выйти из предложенного режима шаблона таблиц. Затем щелкните команду «Конструктор таблиц» из командной строки «Таблица» в меню «Создание».

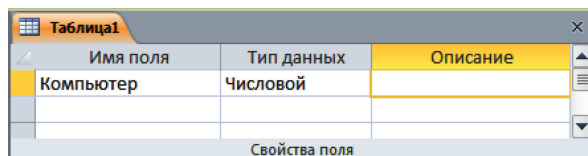


Рисунок 3. Режим «Конструктор таблиц»

В верхней рабочей области появятся столбцы для обязательного заполнения: «Имя поля» и «Тип данных». Третий столбец «Описание» необязателен для заполнения, здесь можно разместить всплывающую подсказку (рис. 3). Содержимое столбца «Имя поля» должны составлять заголовки таблицы, и поэтому имена полей набираются на клавиатуре в соответствии с назначением. Данные в столбце «Тип данных», которые вы определили, выбираются из списка, появляющегося при нажатии кнопки выбора.

Ключевое поле должно быть назначено после ввода всех имен полей, указания типа данных и их свойств. Для этого щелкните «Имя поля» или «Поле с ключевым полем» и нажмите «Ключевое поле» во вкладке «Сервис» в меню «Конструктор». В этой строке перед выбранным именем поля устанавливается знак ключа. И сохраните таблицу.

Выполните следующие действия, чтобы изменить образец таблицы:

а) щелчком правой кнопки мыши выделите имена из списка Таблицы всех объектов Access, которые нужно изменить;

б) выберите команду **Конструктор таблиц** в контекстном меню;

в) укажите строку или ячейку, где размещен знак ключа при открытии шаблона таблицы;

г) нажмите «Ключевое поле» во вкладке «Сервис» в меню «Конструктор таблиц» и удалите знак ключа;

д) после этого можно провести соответствующие изменения;

е) после внесения изменений в шаблон таблицы снова пометьте ключевое поле, а затем сохраните шаблон.



1. Каковы преимущества представления данных в таблице?
2. Какие есть способы создания однотабличных баз данных?
3. Как можно создать таблицу в режиме «Таблица»?
4. Как можно создать однотабличную базу данных в режиме «Конструктор таблиц»?
5. Какую функцию выполняет команда «Свойства поля»?
6. Какие действия надо выполнить, чтобы снова внести изменения в таблицу?



Задания

1. Постройте базу данных по предложенной таблице 1. Указав тип полей, определите «Свойства поля».

Таблица 1

| № | Наименование | Характеристика | Название фирмы | Адрес | Цена (тенге) |
|---|----------------|-------------------|----------------|---------|--------------|
| 1 | Системный блок | Процессор Pentium | Фирма 1 | Адрес 1 | 100000 |
| 2 | Системный блок | Процессор Pentium | Фирма 2 | Адрес 2 | 90000 |
| 3 | Монитор | 15" | Фирма 1 | Адрес 1 | 50000 |
| 4 | Монитор | 15" | Фирма 2 | Адрес 2 | 60000 |
| 5 | Клавиатура | 104 клавиши | Фирма 1 | Адрес 1 | 4500 |
| 6 | Клавиатура | 104 клавиши | Фирма 2 | Адрес 2 | 4000 |
| 7 | Мышь | 3-кнопочная | Фирма 1 | Адрес 1 | 600 |
| 8 | Мышь | 3-кнопочная | Фирма 2 | Адрес 2 | 550 |

Создайте базу данных по таблице 2 для дольных приставок единиц физических величин. Определите «Свойства поля», указав типы полей.

Таблица 2

| п/н | Наименование приставок к физической единице измерений | Обозначение | Множитель |
|-----|---|-------------|-----------|
| 1 | Гига | Г | 10^9 |
| 2 | Мега | М | 10^6 |
| 3 | Кило | к | 10^3 |
| 4 | Гекто | г | 10^2 |

2. В таблице 3 приведены данные о приеме пациентов стоматологической клиники. Представьте записи как базу данных. Определите имена полей и тип данных для поля.

Таблица 3

| № | Наименование специалиста | Дата рождения пациентов | Время приема | Количество вылеченных зубов | Цена лечения (тенге) |
|---|--------------------------|-------------------------|--------------|-----------------------------|----------------------|
| 1 | 1-й пациент | 31.10.1989 | 10.00 | 2 | 6000 |
| 2 | 2-й пациент | 18.04.1984 | 11.00 | 1 | 3000 |
| 3 | 3-й пациент | 11.10.1975 | 11.30 | 3 | 10000 |
| 4 | 4-й пациент | 22.05.1965 | 12.43 | 1 | 1500 |
| 5 | 5-й пациент | 05.07.1982 | 14.00 | 4 | 12000 |

6

Создание многотабличной базы данных



Что такое многотабличная база данных? Как она строится?



✓ Как вы думаете, какие есть преимущества в создании многотабличной базы данных?



Новые знания

Вы уже познакомились с созданием однотабличной базы данных. Но в некоторых случаях для размещения имеющихся данных одной таблицы недостаточно. Если эти данные имеют очень много различных характеристик, то будет большое количество сохраняемых полей. Кроме того, записи в определенных полях таблицы повторяются, база данных растет, и это увеличивает время обработки. В таких случаях лучше создать многотабличную базу данных. Рассмотрим таблицу 1, составленную при изучении прошлой темы.

Таблица 1

Комплектующие компьютера и их поставщики

| Н/п | Имя | Характеристика | Наименование фирмы | Адрес | Цена (тенге) |
|-----|----------------|-------------------|--------------------|---------|--------------|
| 1 | Системный блок | Процессор Pentium | Фирма 1 | Адрес 1 | 100000 |
| 2 | Системный блок | Процессор Pentium | Фирма 2 | Адрес 2 | 90000 |
| 3 | Монитор | 15" | Фирма 1 | Адрес 1 | 50000 |
| 4 | Монитор | 15" | Фирма 2 | Адрес 2 | 60000 |
| 5 | Клавиатура | 104 клавиши | Фирма 1 | Адрес 1 | 4500 |
| 6 | Клавиатура | 104 клавиши | Фирма 2 | Адрес 2 | 4000 |
| 7 | Мышь | 3-кнопочная | Фирма 1 | Адрес 1 | 600 |
| 8 | Мышь | 3-кнопочная | Фирма 2 | Адрес 2 | 550 |

Половина данных этой таблицы повторяется. Остановимся на причинах повторов. Неотъемлемая часть компьютерных комплектующих имеет два параметра: «Имя» и «Характеристика». А свойства комплектующих компьютера, включающие параметры «Адрес» и «Цена», относятся к характеристикам поставщиков компьютеров.

Давайте разделим таблицу 1 на две таблицы: таблица 2 «Комплектующие» и таблица 3 «Поставщик». В каждой таблице, несмотря на определенные характеристики, должен быть единый ключ. В таблице «Комплектующие» поле «Коды комплектующих» определим в качестве ключевого поля.

Таблица 2

«Комплекующие»

| Коды комплек- тующих | Наимено- вание | Характеристика |
|-------------------------|-------------------|-------------------|
| Б1 | Системный блок | Процессор Pentium |
| Б2 | Монитор | 15" |
| Б3 | Клавиатура | 104 клавиши |
| Б4 | Мышь | 3-кнопочная |

Таблица 3

«Поставщики»

| Код поставщика | Название | Адрес |
|----------------|----------|---------|
| Ж1 | Фирма 1 | Адрес 1 |
| Ж2 | Фирма 2 | Адрес 2 |

В таблице «Поставщики» в качестве ключевого поля определите поле «Код поставщика».

Соответствие таблиц

После разделения таблицы на несколько таблиц нам необходимо организовать их в единую систему. Для этого нам нужно установить связь между таблицами. Access дает возможность создать реляционную базу данных и осуществить взаимодействие этих таблиц друг с другом. Это позволяет автоматически выбирать связанные данные, из формы, отчетов и таблиц. При определении связи, ключ из одной таблицы указывает на данные во второй таблице. Ключ со ссылкой на информацию в другой таблице называется *внешним ключом*.

Для определения взаимосвязи между таблицами должны быть выполнены следующие условия:

- ✓ Связанные таблицы должны быть одного размера и иметь одно-типные поля. Но имена этих полей должны быть различными.
- ✓ Обе таблицы должны сохраняться в одной базе данных Access.

Определяя связи, важно правильно выбрать поле и указать связи. Связанные таблицы существуют как основная и подчиненная, в зависимости от того, как они определены в схеме данных. В общем случае между двумя таблицами могут быть установлены следующие основные четыре типа связей:

Отношение «один к одному». 1:1 – основано на соответствии одной записи основной таблицы только к одной записи в подчиненной и наоборот.

Отношение «многие к одному». M:1 – основано на соответствии нескольких записей основной таблицы одной записи в подчиненной таблице.

Отношение «один ко многим». 1:M – основано на соответствии одной записи в основной таблице многим записям в подчиненной таблице.

Отношение «многие ко многим». M:M – основано на соответствии одной записи основной таблицы многим записям подчиненной таблицы и, наоборот, соответствие одной записи подчиненной таблицы многим записям основной таблицы.

В примере, приведенном выше, связь между таблицами соответствует типу связи «**Многие ко многим**». Одной записи в таблице «Комплектующие» соответствуют две записи в таблице «Поставщики» (каждую комплектующую продают одновременно две фирмы), в то же время одной записи в таблице «Поставщики» соответствуют четыре записи в таблице «Комплектующие» (1 фирма продает 4 компьютерных комплектующих). Чтобы осуществить между двумя таблицами связь «**Многие ко многим**», необходимо создать третью таблицу – «Цена». Таблицы «Поставщики» и «Комплектующие» являются для таблицы «Цена» основными, а таблица «Цена» будет являться подчиненной. Между таблицами происходит обмен данными по полям, которые имеют одинаковые имена, соответствующие их значениям. Первичный ключ в основной таблице связан с внешним ключом подчиненной таблицы. Таблица «Цена» должна формироваться из следующих полей:

- «Нумерация» (ключевое поле);
- «Коды комплектующих» (внешний ключ для таблицы «Комплектующие»);
- «Код поставщика» (внешний ключ для таблицы «Поставщики»);
- «Цена» (числовое поле).

| Нумерация | Коды комплектующих | Код поставщика | Цена, тенге |
|-----------|--------------------|----------------|-------------|
| 1 | Б1 | Ж1 | 100 000 |
| 2 | Б1 | Ж2 | 90 000 |
| 3 | Б2 | Ж1 | 50 000 |
| 4 | Б2 | Ж2 | 60 000 |
| 5 | Б3 | Ж1 | 4500 |
| 6 | Б3 | Ж2 | 4000 |
| 7 | Б4 | Ж1 | 600 |
| 8 | Б4 | Ж2 | 550 |

Схема связи между таблицами показывает эти три таблицы как единую базу данных. Две взаимосвязанные двумерные таблицы можно назвать *реляционными таблицами*.



Применение



Создайте многотабличную базу данных компьютерных комплектующих и их поставщиков.



Синтез



Придумайте пример многотабличной базы данных. Запишите на бумаге проект, затем реализуйте его и получите результат. Например расписание уроков, успеваемость по предметам и т.д.



1. Каковы преимущества представления данных в многотабличном виде?
2. Какие типы связи существуют между таблицами при создании многотабличной базы данных?



Подготовьте проект базы данных «Коллекция аудиозаписей». Проект может состоять из основной таблицы «Список Аудио-CD» и подчиненной таблицы «Содержание Аудио-CD».

Создание формы для ввода данных



Как можно создать форму для ввода данных?



- ✓ Как можно преобразовать данные из табличного вида в вид формы?
- ✓ Как вы думаете, какие есть преимущества в представлении данных в виде формы?



Новые знания

Мы научились создавать однотабличную и многотабличную базы данных и создавать связь таблиц между собой. Таблицы являются основой для формирования информационной системы. В то же время создание базы данных этим не заканчивается. У информационных систем есть свои потребители. Это в основном операторы и диспетчеры. Эти пользователи не имеют глубоких знаний в области баз данных. И они должны иметь простой в использовании интерфейс для работы с базами данных. Объект базы данных, который можно использовать для создания интерфейса пользователей базы данных в программе MS Access, называется **формой**.

Создание формы с помощью *Мастера форм* в программе MS Access

Фактически с помощью формы создается графический интерфейс доступа к базе данных, который может содержать различные управляющие элементы (текстовые поля, кнопки, переключатели и так далее), а также надписи. Формы выполняют основные функции разработки БД: ввод данных, отображение, поиск, редактирование. Формы создаются с использованием инструментов **Access** в соответствии с требованиями заказчика. При создании формы решается: из какой таблицы вводятся данные в форму, определяется, в частности, какие поля таблицы, какие графические элементы управления будут использоваться: рисунок, список, флажок, элемент навигации, поле со списком и другие.

Форма, основанная на одной таблице, называется однотабличной.

Для создания формы можно использовать команду из меню **Создание формы – Мастер форм** MS Access.

В качестве примера рассмотрим, как с помощью **Мастера форм** создать форму базы данных учащихся для 10 класса (рис. 1).

| Код | Фамилия, имя | Дата рождения | Класс | Адрес | Предмет по выбору |
|-----|--------------|---------------|-------|-----------|-------------------|
| 1 | 1-й ученик | 19.10.2003 | 10 | 1-й адрес | химия |
| 2 | 2-й ученик | 10.10.2003 | 10 | 2-й адрес | физика |
| 3 | 3-й ученик | 18.04.2003 | 10 | 3-й адрес | химия |
| 4 | 4-й ученик | 01.01.2003 | 10 | 4-й адрес | биология |
| 5 | 5-й ученик | 21.05.2003 | 10 | 5-й адрес | география |
| 6 | 6-й ученик | 23.07.2003 | 10 | 6-й адрес | физика |

Рисунок 1. База данных учащихся

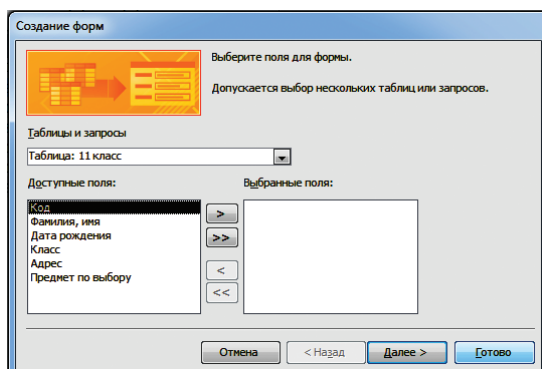


Рисунок 2. Окно Мастер форм

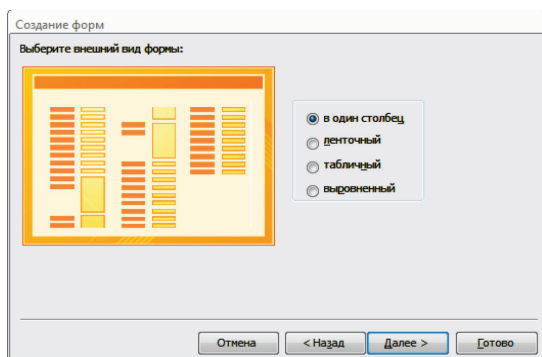


Рисунок 3. Внешний вид формы

- ✓ *один столбец* – поля размещены вертикально и подпись расположена слева от поля;
- ✓ *ленточный* – поля размещаются в одну строку и их подписи выводятся один раз в верхней части формы;
- ✓ *табличный* – поля записей расположены в виде таблицы, где каждой записи соответствует одна строка, а каждому полю – один столбец;
- ✓ *выровненный* – вид поля, аналогичный ленточному, но отличается тем, что все поля записи будут отображены на одной странице.

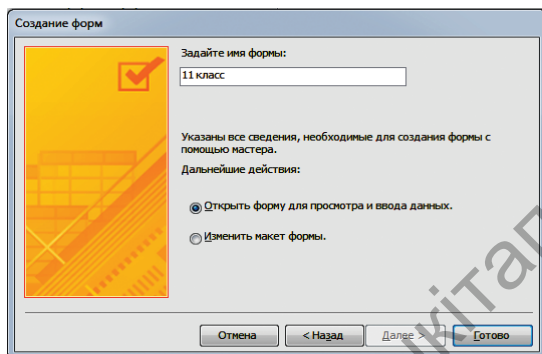


Рисунок 4. Окно задания имени формы

Для этого выполним команду **Создание – Формы – Мастер форм**. Окно **Мастер форм** представлено на рисунке 2.

При создании форм с помощью **Мастера форм** можно выбрать объект, из которого будет создаваться форма, например, таблицу или запрос, определенные поля, стиль оформления.

1-й шаг: На этом этапе режима **Мастер форм** вы должны выбрать таблицу или запрос с данными для формы и с помощью кнопки > выбрать нужное поле этой таблицы (рис. 2). Выделив названия полей, нажав на кнопку >>, выбираем и вводим все поля в новую форму, кнопки < и << осуществляют обратные действия, удаляя соответственно одно поле или все поля из новой формы.

2-й шаг: На втором этапе режима **Мастер форм** мы выбираем способы размещения данных, то есть внешний вид формы:

3-й шаг: На последнем этапе режима **Мастер форм** нужно дать форме название. СУБД MS Access выполняет это автоматически. Для начала зададим номер формы, создаваемый на основе таблицы или запроса. На этом этапе вы также можете определить следующие шаги. Для этого вам нужно выбрать переключатель «Открыть форму для просмотра и ввода данных» или «Изменить

макет формы». В любой момент в режиме Мастер форм вы можете нажать **Готово**, чтобы остановить процесс создания формы или продолжить, нажав кнопку **Далее**. Если работа режима **Мастер форм** останавливается, невыполненные шаги автоматически завершаются, форма не будет создана.

Нажимая кнопку **Готово** получим простую готовую форму, как показано на рисунке 5. Эта форма создана и показана на рисунке в виде одного столбца. В форме «10-й класс» отражены все поля таблицы. В нижней части формы для перехода от одной записи к другой помещены кнопки:

- к первой записи;
- к предыдущей записи;
- номер поля записи;
- к следующей записи;
- к последней записи;
- к новой записи.

Нажав нужную кнопку, вы можете перейти к первой записи или к последней, также можете добавить новые записи в таблицу. «Номер поля записи» показывает номер текущей записи.

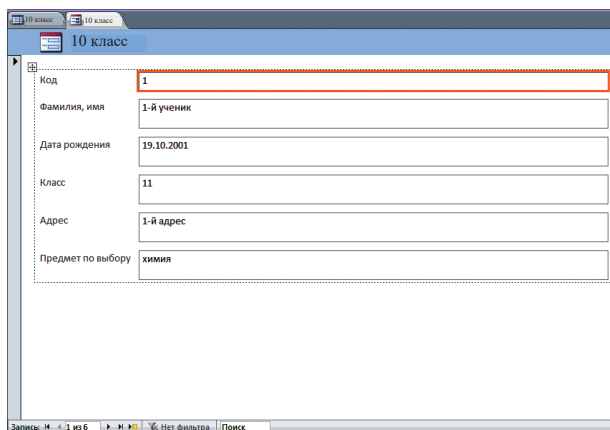


Рисунок 5. Окно в форме одного столбца



Применение

Практическая работа

В таблице 1 приведены 10 мировых рек с указанием длины и объема бассейна. Используя данные таблицы, создайте форму базы данных с помощью *Мастера формы*.

Таблица 1

| № | Название реки | Длина (км) | Объем воды бассейна (тыс. кв. км) |
|----|---------------|------------|-----------------------------------|
| 1 | Волга | 3531 | 1360 |
| 2 | Рейн | 1320 | 224 |
| 3 | Одер | 912 | 119 |
| 4 | Янцзы | 6300 | 1808 |
| 5 | Амур | 4444 | 1855 |
| 6 | Евфрат | 3065 | 673 |
| 7 | Сырдарья | 3019 | 219 |
| 8 | Амударья | 2540 | 309 |
| 9 | Нил | 6671 | 2870 |
| 10 | Конго | 4370 | 3820 |



Синтез



Создайте таблицу, содержащую какие-либо данные, встречающиеся в вашей повседневной жизни, и на ее основе создайте форму.



Оценка



Оцените значимость представления баз данных в виде формы.



1. Для чего нужна форма?
2. Какие существуют режимы создания формы?
3. Какие есть возможности выбора полей в режиме *Мастер форм*?
4. Какие элементы можно разместить в форме?
5. Какие бывают виды форм в режиме *Мастер форм*?



Задания

1. Создайте базу данных учащихся 10-го класса в виде таблицы. Создайте форму для ввода и редактирования данных.

| Код учащегося | Ф.И.О. | Год рождения | E-mail | Класс обучения | ИИН | Фото учащегося | Хобби | Успеваемость |
|---------------|--------|--------------|--------|----------------|-----|----------------|-------|--------------|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

2. Создайте однотабличную базу данных учебных пособий для учащихся 10 класса. Создайте форму для ввода и редактирования данных.

| Код учебных пособий | Наименование | Автор | Издательство | Цена | Количество | Издание | Фото учебного пособия |
|---------------------|--------------|-------|--------------|------|------------|---------|-----------------------|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Создание формы в режиме Конструктор. Практическая работа



Как можно создать форму в режиме Конструктор?



✓ Создание базы данных по таблице «Экзамен» в MS Access.

1. Создайте базу данных по таблице 1 в режиме **Конструктор**:

Таблица 1

| Порядковый номер учащегося | Фамилия | Имя | Класс | Экзамен 1 | Экзамен 2 | Экзамен 3 |
|----------------------------|---------|-----|-------|-----------|-----------|-----------|
| | | | | | | |

Свойства полей:

Порядковый номер учащегося: тип – **счетчик**, подпись – №, индексировано.

Фамилия: тип – **текстовый**, размер поля – 50, обязательное, без пустых строк, индексировано.

Имя: тип – **текстовый**, размер поля – 25, обязательное, без пустых строк, индексировано.

Класс: тип – **текстовый**, размер поля – 7, подпись – Класс, обязательное, без пустых строк, индексировано.

Экзаменационная оценка: тип – **числовой**, байтовый формат, условие на значение наименования предметов, например, история, биология, сообщение об ошибке Between 2 and 5, также используйте текстовую ошибку «неправильно ввели оценку».

Сделайте ключевым поле **Порядковый номер учащегося**. Сохраните таблицу под именем Экзамены и закройте конструктор таблиц.

2. Создайте форму для заполнения и изменения таблицы «Экзамен» с помощью *Мастер форм*. Сохраните ее под названием **Экзамены**. В меню *Создание* кнопка «Мастер форм» используется для вызова форм. Используя форму, заполните базу данных.

3. Используя конструктор, создайте форму (строится вручную).

- Выберите кнопку **Создать** на панели инструментов и в качестве основной таблицу **Экзамены**. После этого выберите Конструктор формы. Откроется область данных пустой формы. Ознакомьтесь с ними. Если есть только область данных, то добавьте **Заголовок/Примечание формы** в меню «Тип». Форма создается с помощью элементов управления. Они размещаются на панели инструментов. Элемент «Надпись» **Aa** используется для создания области текста. Элемент «Поле» **ab** используется для ввода текста, чисел, дат, вычисляемых выражений.

- Перетащите все поля, используя для переноса *Область данных* формы. В тоже время данные формы, соответствующие значениям таблицы (свойства поля «Данные»). В каждое поле в области данных добавляется запись имени поля. Удалите эти записи.

- Введите записи в столбец таблицы. Запись – **Экзаменационная ведомость**.

- Поменяйте свойство всей формы. Для этого щелкните правой кнопкой мыши по верхней левой стороне формы . По умолчанию форма будет в один столбец. Нужно перевести свойство макета «По умолчанию» в **Ленточную форму**. Тогда тип формы будет таблицей записей.

- Сохраните форму под именем **Ведомость**. Снова войдя в Конструктор форм, измените форму. Измените размеры всех элементов.

Простой метод построения отчета



Как построить отчет с помощью базы данных?



- ✓ Как преобразовать данные из табличного вида в отчет?
- ✓ Как вы думаете, какие есть преимущества представления данных в виде отчета?



Новые знания

Данные, собранные в базе данных, могут быть напечатаны на бумаге или экспортированы в другие программы в виде файла. Чтобы создавать простые отчеты в MS Access, вам нужно сделать несколько шагов.

В качестве примера давайте рассмотрим пути составления отчета на основе базы данных учащихся 10-го класса (рис. 1) с помощью **Мастера отчета**.

| Код | Фамилия, имя | Дата рождения | Класс | Адрес | Предмет по выбору |
|-----|--------------|---------------|-------|-----------|-------------------|
| 1 | 1-й ученик | 19.10.2003 | 10 | 1-й адрес | химия |
| 2 | 2-й ученик | 10.10.2003 | 10 | 2-й адрес | физика |
| 3 | 3-й ученик | 18.04.2003 | 10 | 3-й адрес | химия |
| 4 | 4-й ученик | 01.01.2003 | 10 | 4-й адрес | биология |
| 5 | 5-й ученик | 21.05.2003 | 10 | 5-й адрес | география |
| 6 | 6-й ученик | 23.07.2003 | 10 | 6-й адрес | физика |

Рисунок 1. База данных для учащихся 10 класса

Открыв базу данных, выполните команду *Создание – Отчеты – Мастер отчетов*.

Использование Мастера отчета

1-й шаг. Создание отчетов с помощью мастера не требует специальных знаний и сводится только к выбору таблиц для отчетов, списка полей отчета и их размещения.

Нажмите на кнопку, раскрывая список таблиц и запросов, и выберите из списка, таблиц базы данных таблицу на основе которой необходимо создать отчет. Здесь в списке

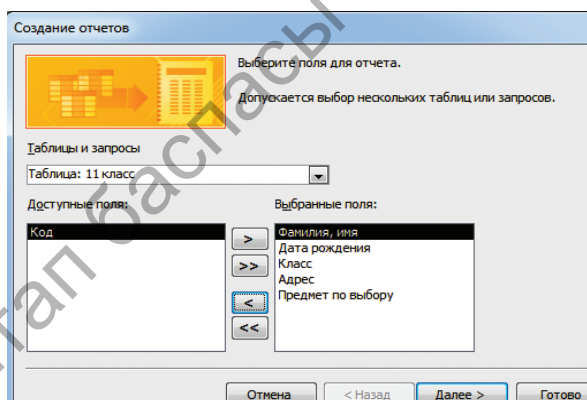


Рисунок 2. Диалоговое окно Мастера отчета

Доступные поля из выбранной таблицы представлен весь список полей (рис. 2). Из данного списка перетащите в список **Выбранные поля** поля для отчета. Закончив выбор полей, перейдите на следующий шаг нажатием кнопки **Далее**.

2-й шаг. На втором этапе создания отчета с помощью Мастера вам необходимо определить, следует ли группировать данные в одном из полей. MS Access предлагает возможность группировки данных. Мы используем предложенную версию либо свою версию с использованием кнопок в диалоговом окне. MS Access дает возможность задать группировку по 10 значениям.

Таблица 1

Диалоговое окно Мастера отчета

| Кнопка | Выполняемое действие |
|--------|---|
| | Добавляет поле в панель группировки |
| | Удаляет поле в панели группировки |
| | Увеличивает уровень группировки, выделенный на панели группировки полей |
| | Уменьшает уровень группировки, выделенный на панели группировки полей |

Можно изменить интервал группировки после того, как установлена группировка данных. Для этого нужно нажать на кнопку **Группировка** (рис. 3). В следующем диалоговом окне мы делаем сортировку записей отчета (рис. 4). Если вы хотите организовать итоговые поля в отчете, нажмите на кнопку **диалогового окна**, а затем установите соответствующие флажки возле определенных типов полей, указанных в диалоговом окне **Итоги**, для которых надо вычислить итоговые значения числовых полей. Затем, чтобы вернуться к окну сортировки, нажмите на кнопку **Назад**.

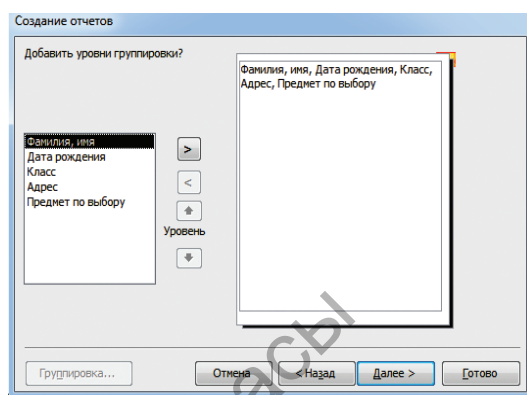


Рисунок 3. Выбор уровня группировки

Следующим шагом при создании отчета с помощью Мастера отчета является изменение сортировки записей (рис. 5). **Последний шаг** (рис. 6) – присвоение имени отчету и выбор одного из двух режимов для дальнейшей работы с отчетом:

- просмотр отчета;
- изменение сортировки записей.

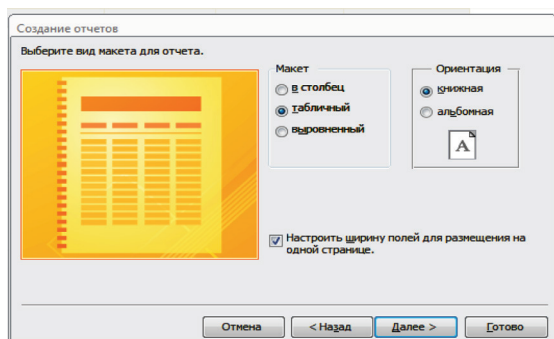


Рисунок 4. Сортировка записей отчетов

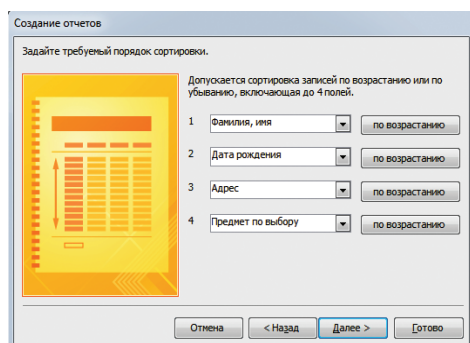


Рисунок 5. Изменение сортировки записей

Во время просмотра отчета на экране будет видна только одна часть (рис. 7). С помощью кнопки масштаба можно посмотреть полную версию страницы отчета, но данные будут видны нечетко. Тогда можно будет выбрать на панели инструментов кнопку **Две страницы** и вывести на экран двустраничный полный отчет. В режиме **Просмотр** можно ввести в меню команду **Страницы** и вывести на экран 1, 2, 4, 8 или 12 страницы.

Просмотр отчета и вывод на печать

MS Access имеет большой набор инструментов для просмотра отчетов на экране. По своему усмотрению, можно использовать любой способ:

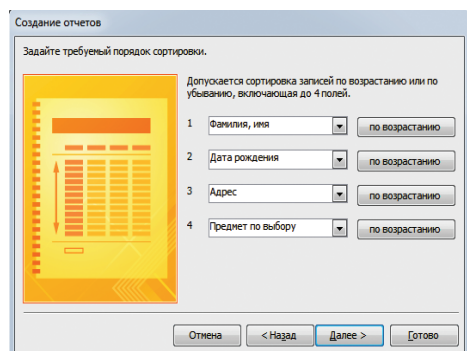


Рисунок 6. Окно диалога для задания имени отчета и выбора режима просмотра

- в главном меню – команду **Предварительный просмотр.**

- в контекстном меню – команду **Предварительный просмотр.**

В окне базы данных достаточно нажать на кнопку просмотра, чтобы выделить имя отчета, которое вы хотите увидеть.

Чтобы увидеть отчет, мы будем использовать кнопки панели инструментов и кнопки переключателя в нижней части окна просмотра.

Для отправки отчета на печать выполните команду **Файл – Печать.**

В диалоговом окне «Печать» вы можете выбрать режим печати для файла, указав количество копий, которые необходимо напечатать. Чтобы задать номер страницы, которую нужно напечатать, установите опцию страниц в области «Печать» слева в нижней части диалогового окна. Задайте С: номер первой страницы, предназначенной для печати, и введите в поле, и По: соответственно, номер последней печатаемой страницы.

Чтобы указать количество копий для печати, установите соответствующее значение в области «Копии» с помощью команды **Количество копий**.

Диалоговое окно «Печать» использует параметры «Книжный» (Portrait) и «Альбомный» (Landscape), чтобы задать опцию печати.

Объектом отчета MS Access является документ отчета, подготовленный для печати данных, необходимых пользователю БД. Объекты *форма* и *отчет* имеют сходные структуры, что означает, что большинство данных из формы подходят для создания отчета.

| 10 класс | |
|-------------------|-------------|
| Ф.И.О. | Ученик 1 |
| Год рождения | 19.11. 2006 |
| Адрес | адрес 1 |
| Предмет по выбору | химия |
| Ф.И.О. | Ученик 2 |
| Год рождения | 15.01. 2005 |
| Адрес | адрес 2 |
| Предмет по выбору | физика |

Рисунок 7. Образец отчета

При создании отчета большая часть работы может выполняться в режиме **Конструктор**. После создания отчета его можно распечатать или переместить в другие программы.

Данный отчет можно экспортировать в программы MS Excel, MS Word, web-страницы и электронную почту.

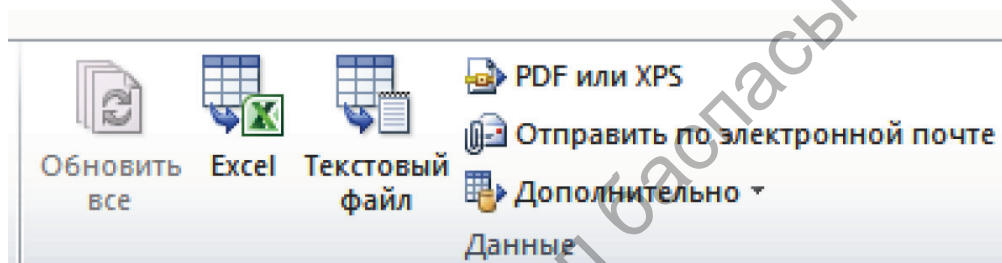


Рисунок 8. Кнопки экспорта отчета



Подготовьте отчет по таблице базы данных учащихся 10 класса, показанной на рисунке 1. По готовому отчету:



1. Отправьте на печать страницу отчета.
2. Экспортируйте отчет в программу MS Excel, т.е. переведите в файл программы MS Excel.
 1. Для чего нужен отчет базы данных?
 2. Какие есть режимы отчетов?
 3. Какие есть возможности выбора полей в режиме Мастер отчетов?
 4. Из каких шагов состоит работа в режиме Мастер отчетов?
 5. Как можно отправить на печать отчет?
 6. В какие программы и как можно экспортировать готовые отчеты?



Задания

Задание 1. Телефонная станция.

В качестве программиста телефонной станции создайте базу данных.

Поля для создания базы данных: Номер абонента, ФИО, адрес местожительства, код станции, код района проживания, оплата (долг).

По этим данным выполните следующие задания в MS Access:

1. Создание таблицы.
2. Разработка формы.
3. Подготовка отчета.
4. Вывод на печать.
5. Экспорт в программу MS Excel.

Задание 2. Книжный фонд школьной библиотеки.

В качестве школьного программиста создайте базу данных школьной библиотеки.

Поля для базы данных: автор книги, название книги, год издания, издательство, цена, количество, краткая аннотация. По этим данным выполните нижеследующие задания в MS Access:

1. Создание таблицы.
2. Разработка формы.
3. Подготовка отчета.
4. Вывод на печать.
5. Экспорт в программу MS Excel.



Как создать структурированный запрос с помощью конструктора?



✓ Как найти нужную информацию в готовой базе данных?



Новые знания

Использование запросов в базе данных является одним из самых быстрых способов получения информации. Запрос позволяет выбрать нужную информацию из одной или нескольких взаимосвязанных таблиц, форм, выполнить отчет и получить результаты в виде таблицы. Доступ также имеет QBE (Query By Example), который представляет собой набор удобных для пользователя запросов, что упрощает создание сложных запросов. Запрос также может быть выполнен с использованием исходных таблиц БД, основанных на результатах других запросов и на сформированных запросах. Чтобы создать запрос, нажмите на кнопку **Создать** на вкладке **Запрос**. Окно выбора режима Мастера запросов показано на рисунке 1.

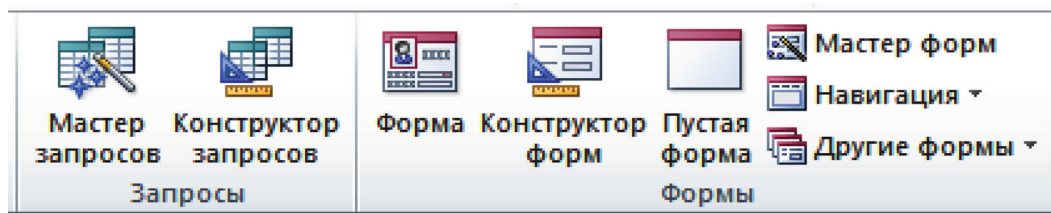


Рисунок 1. Вкладка «Запросы»

Для создания запросов будет запущен предназначенный для этого **Мастер запросов**, позволяющий выбрать необходимые поля связанных таблиц и запросов. В бланках созданных запросов нельзя добавлять и вносить изменения. Однако, объединяя записи, можно выполнить запрос на определение значений суммы, среднего, минимального, максимального и количества записей в выбранных полях для запроса. Рассмотрим пример простого запроса.

Создание простых запросов

Чтобы создать запрос в режиме **Простой запрос**, мы запустим **Мастер запросов**. Он основывается на выборе необходимых полей связанных таблиц и запросов. Рассмотрим пример Простого запроса. Для этого снова обратимся к таблице данных 10 класса, рассмотренной в предыдущем параграфе (рис. 2).

| Код | Фамилия, имя | Дата рождения | Класс | Адрес | Предмет по выбору |
|-----|--------------|---------------|-------|-----------|-------------------|
| 1 | 1-й ученик | 19.10.2003 | 10 | 1-й адрес | химия |
| 2 | 2-й ученик | 10.10.2003 | 10 | 2-й адрес | физика |
| 3 | 3-й ученик | 18.04.2003 | 10 | 3-й адрес | химия |
| 4 | 4-й ученик | 01.01.2003 | 10 | 4-й адрес | биология |
| 5 | 5-й ученик | 21.05.2003 | 10 | 5-й адрес | география |
| 6 | 6-й ученик | 23.07.2003 | 10 | 6-й адрес | физика |

Рисунок 2

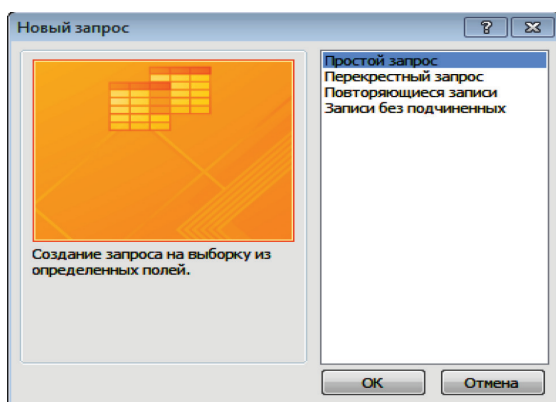


Рисунок 3. 1-я страница Мастера запросов

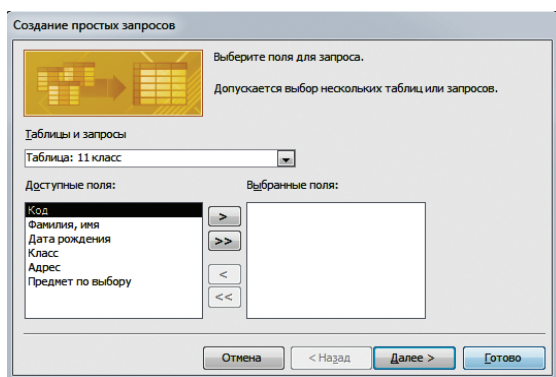


Рисунок 4. Выбор полей

Рисунок 3 показывает первый шаг в диалоге при создании Простого запроса. На этой странице вы можете создать кроме Простого запроса еще три разных вида запроса.

Режим *Перекрестного запроса* создает перекрестные запросы для представления выбранных двух таблиц в перекрестном виде.

Режим *Повторяющейся записи* позволяет выбрать таблицу, в которой вы хотите найти записи с повторяющимися полями, указать таблицу для анализа, переименовать повторяющиеся поля и выбрать другие поля для вывода на экран наряду с повторяющимися. Запрос определяет по крайней мере две идентичные записи в выбранных полях таблицы и выводит их по запросу.

Режим *Записи без подчинения* специально разработан для создания запроса с целью поиска записей в таблице, для которых

нет подчиненных записей в других таблицах. На шаге 2 **Мастера запросов** (рис. 4) выбираются поля, по которым вы собираетесь выполнить запрос.

В нашем примере для таблицы «10 класс» это Ф.И.О. учащегося, год рождения – класс, адрес, выбранный предмет. На втором этапе запрашиваемые поля выбираются из вышеуказанных полей.

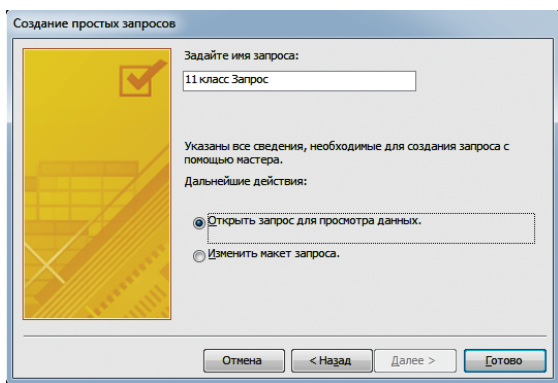


Рисунок 5. Присваивание имени запросу

На завершающем шаге создания запроса задается имя запроса (рис. 5).

После подготовки мы нажмем кнопку **Конструктор запросов** на странице **Запрос**, чтобы увидеть результаты ее создания.

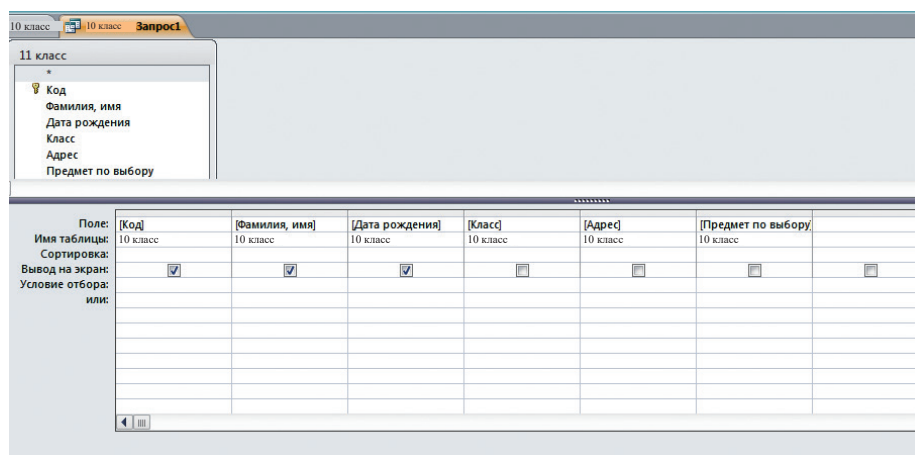


Рисунок 6. Конструктор запроса

Нажав команду **Выполнить** для запроса, состоящего из трех отмеченных полей на рисунке 6, получаем результат, представленный на рисунке 7.

| Фамилия, имя | Дата рождения | Предмет по выбору |
|--------------|---------------|-------------------|
| 1-й ученик | 19.10.2003 | химия |
| 2-й ученик | 10.10. 2003 | физика |
| 3-й ученик | 18.04.2003 | химия |
| 4-й ученик | 01.01.2003 | биология |
| 5-й ученик | 21.05.2003 | география |
| 6-й ученик | 23.07.2003 | физика |

Рисунок 7. Результат запроса



Применение

Практическая работа

В таблице представлены 10 мировых рек, их длины и объемы бассейнов. Из данных этой таблицы создайте базу данных. Сделайте запрос на именованных рек по длине с помощью Мастера запросов.

| № | Название реки | Материк расположения | Длина (км) | Объем бассейна реки (тыс. кв. км) |
|----|---------------|----------------------|------------|-----------------------------------|
| 1 | Волга | Евразия | 3531 | 1360 |
| 2 | Миссисипи | Северная Америка | 5971 | 3268 |
| 3 | Колумбия | Южная Америка | 1953 | 669 |
| 4 | Янцзы | Евразия | 6300 | 1808 |
| 5 | Амур | Евразия | 4444 | 1855 |
| 6 | Амазонка | Южная Америка | 3065 | 673 |
| 7 | Сырдарья | Евразия | 6437 | 7180 |
| 8 | Амударья | Евразия | 2540 | 309 |
| 9 | Нил | Африка | 6671 | 2870 |
| 10 | Конго | Африка | 4370 | 3820 |



Синтез



Создайте базу данных, заполнив ее данными, которые вы ежедневно обрабатываете. В созданной базе данных, сделайте запрос двух значимых полей и напечатайте их.



Оценка



Оцените важность умения создавать запросы в работе с базой данных.



1. Для чего нужен запрос?
2. Из каких шагов состоит работа Мастера запросов?
3. Какие есть возможности выбора полей в режиме Мастера запросов?
4. Как можно создать запрос с помощью Конструктора запросов?



Задания

Оптовая база

Список характеристик:

Код продукта, название продукта, количество запаса, единица измерения, цена единицы товара, описание продукта.

Задание по вышеуказанным полям:

1. Построить таблицу.
2. Создать форму.
3. Подготовить отчет.
4. Создать таблицу спроса на наименование товара и цену единицы товара.

Использование языка структурированных запросов (SQL) для данных ВЫБОР



Как можно использовать язык структурированных запросов (SQL) для данных ВЫБОР из таблицы?



✓ Можно ли использовать методы программирования при создании базы данных?



Новые знания

SQL (Structure Query Language) – язык запросов. SQL используется в каждом запросе MS Access.

SQL – это язык структурированных запросов. Он дает возможность создавать и управлять реляционной базой данных и связанной информацией, которые хранятся в таблице. SQL поддерживается лидерами в области реляционной технологии БД и является стандартным языком, поскольку он не зависит от компьютерных технологий. Поэтому каждый пользователь, желающий научиться работать с любой базой данных, должен знать SQL язык. Стандарт SQL был принят ANSI (American National Standards Institute), а в настоящее время он принят ISO (Международная организация по стандартизации).

Структура языка SQL. SQL предназначена для выполнения сложных действий в реляционных базах данных, для определения структуры базы данных и доступа к данным в общедоступных средах.

В отличие от других языков SQL предназначен только для управления базой данных. Вы можете с помощью SQL-запросов сделать запрос в базе данных. Для работы с SQL:

- 1) Откройте базу данных;
- 2) Откройте меню **Создание**, затем **Конструктор запросов**;
- 3) В появившемся диалоговом окне выберите и закройте таблицу, для которой вы хотите выполнить запрос;
- 4) Введите в действие Режим, выполняя команду **Режим SQL**;
- 5) При запуске в рабочей области появится поле SELECT FROM;
- 6) Введите нужную команду в рабочей области, а затем выполните команду **Запуск** из меню.

Язык сложных действий с данными SELECT (выбор), INSERT (вставка), UPDATE (обновление), DELETE (удаление)

Чтобы найти в списке контактов имя Мауга, выполним следующие команды на языке SQL.

```
SELECT Last_Name
FROM Contacts
WHERE First_Name = 'Mayra';
```

| Описание команд |
|--|
| SELECT – определяет поля, содержащие требуемые данные. |
| FROM – определяет таблицы, содержащие поля, указанные в SELECT. |
| WHERE – определяет условие отбора полей, которые должны соответствовать всем записям, содержащимся в результатах. |
| ORDER BY – определяет порядок сортировки результатов. |
| GROUP BY – в инструкции SQL, содержащей статистические функции, предложение SELECT определяет совокупное значение полей, которые не вычисляются. |
| HAVING – в инструкции SQL со статистическими функциями необязательный параметр SELECT определяет условия, применяемые к полям. |

Инструкция SQL состоит из нескольких частей, называемых предложениями. Каждое предложение в инструкции SQL имеет свое назначение. Некоторые предложения являются обязательными. В таблице 1 показаны наиболее часто используемые предложения SQL.

Пример из программы Access

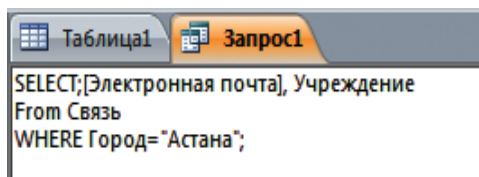


Рисунок 1. Простой запрос

В следующем примере показано, что Access может выглядеть как инструкция MS SQL для простого запроса.

Эти инструкции SQL следует читать следующим образом:

Из данных полей «Таблица контактов», «Электронная почта» и «Компания» выберите значение «Актау» в поле «Город», как показано на рис. 1.

SQL-запрос **Select** для выбора из базы данных. То есть мы можем использовать этот запрос, даже если мы не только редактируем данные, но и внесем изменения в базу данных.

Однако данные из таблицы не всегда бывают нужны полностью, иногда требуются только некоторые столбцы. В этом случае мы указываем имя поля (или полей), которые нам нужны. Например:

Если требуется выбрать несколько полей, необходимо просто перечислить их через запятую после оператора SELECT.

```
SELECT * FROM ТАБЛИЦА1;
```

* - все данные;

FROM – источник данных;

Table – название таблицы.

```
SELECT Price FROM Table
```

Price – название поле

```
SELECT price, name, model
```

```
FROM Table;
```

Название столбцов в таблице price, name, model

WHERE – условие отбора полей

В процессе отбора нам необходимо отфильтровать данные в соответствии с конкретным условием. То есть вы можете использовать оператор **WHERE** в структуре **SELECT**, чтобы выбрать только те данные, которые соответствуют условию. В данном примере **WHERE** выбирает цены выше 100.

```
SELECT price FROM Table
WHERE price > 100
```

Вы также можете использовать ключевое слово **BETWEEN** для задания условия. С помощью **BETWEEN** вы можете выбрать значения в определенном интервале.

```
SELECT table [price]
FROM table
WHERE table [price] BETWEEN 400 AND 600
```

Операторы сравнения SQL

«>» – больше; «<» – меньше; «=» – равно; «<>» не равно; «> =» – больше или равно; «< =» меньше или равно.

Чтобы проверить, что рассматриваемое поле входит в определенный промежуток, можно использовать оператор **IN**. В данном случае мы хотим выбрать цены со значениями 400 и 600.

```
SELECT table [price]
FROM table
WHERE table [price] IN (400, 600)
```

Если нам нужно получить только уникальные значения источников, мы можем указать ключевое слово **DISTINCT**.

```
SELECT DISTINCT Table.[price]
FROM Table
WHERE Table.[price] > 100
```

Оператор сортировки ORDER BY

Результаты запроса могут быть отсортированы по определенному полю (столбцу). Для выполнения сортировки мы покажем после запроса структуру **ORDER BY** и поля для сортировки (которые могут быть перечислены через запятую).

```
SELECT Table.[price]
FROM Table
ORDER BY Table.[price] DESC
```

В этом примере содержимое поля сортируется в порядке убывания, но если вместо **DESC** при сортировке написать **ASC**, то сортировка идет по возрастанию. Запись **ASC** является необязательной, поскольку сортировка по умолчанию предполагает возрастающую последовательность.



Применим на практике полученные знания. Для этого преобразуем таблицу 1 в таблицу базы данных (рис. 2).

Таблица 1

Автотранспорт

| № | Марка машины | Год выпуска | Пробег (км) | Цена (тенге) |
|---|---------------------|-------------|-------------|--------------|
| 1 | Audi 100 | 1988 | 486 000 | 15 000 |
| 2 | Volkswagen | 2007 | 111 111 | 165 000 |
| 3 | Toyota Land Cruiser | 1995 | 239 000 | 2 500 000 |
| 4 | Mercedes-Benz E 200 | 2009 | 134 000 | 4 900 000 |
| 5 | Nissan Pathfinder | 2006 | 150 000 | 450 000 |

| Код | Марка автомобиля | Год выпуска | Пробег(км) | Цена(тенге) |
|-----|---------------------|-------------|-------------|-------------|
| 1 | Audi | 1998 | 486000 | 1500000 |
| 2 | Volkswagen | 2007 | 111111 | 1650000 |
| 3 | Toyota Land Cruiser | 1995 | 239000 | 2500000 |
| 4 | Mercedes-Benz E 200 | 2009 | 134000 | 4900000 |
| 5 | Skoda Rapid | 2013 | 780000 | 4000000 |

Рисунок 2. Таблица базы данных

Сделаем запрос по таблице базы данных. Для этого нажмем на кнопку Конструктор запросов на вкладке Запрос. В диалоговом окне «Авто» выбираем Добавления таблицы. Выполнив команду **Файл-Режим SQL**, открываем окно работы **SQL** (рис. 3). Делаем в окне работы необходимые запросы с помощью команд языка **SQL**.

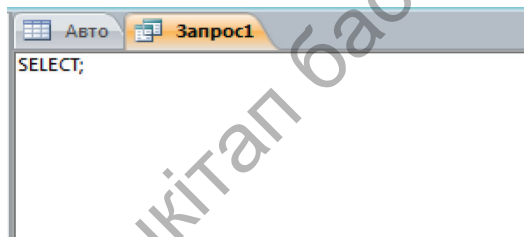


Рисунок 3. Окно создания запроса

Задание 1. Создайте запрос по маркам автомашин и по году выпуска из рис. 4.

Для этого запишите ряд ниже-следующих команд:

```
SELECT [марка автомобиля],  
[год выпуска]  
From Авто;
```

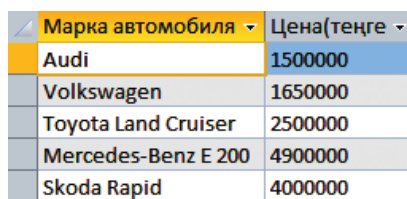


| Марка автомобиля | Год выпуска |
|---------------------|-------------|
| Audi | 1998 |
| Volkswagen | 2007 |
| Toyota Land Cruiser | 1995 |
| Mercedes-Benz E 200 | 2009 |
| Skoda Rapid | 2013 |

Рисунок 4. Запрос по году выпуска авто

Задание 2. Создайте запрос с полями Марка и Стоимость автомобилей и отсортируйте по убыванию по полю Цена (рис. 5).

```
SELECT Авто.[Марка автомашины],  
Авто.[Цена(тенге)]  
FROM Авто  
ORDER BY Авто.[Цена тенге]] DESC
```



| Марка автомобиля | Цена(тенге) |
|---------------------|-------------|
| Audi | 1500000 |
| Volkswagen | 1650000 |
| Toyota Land Cruiser | 2500000 |
| Mercedes-Benz E 200 | 4900000 |
| Skoda Rapid | 4000000 |

Рисунок 5. Запрос по стоимости авто

Анализ

Сравните с помощью диаграммы Венна запросы, созданные в базе данных с помощью языка SQL и с помощью готового конструктора.

Синтез

Подготовьте на бумаге проект создания базы данных по данным об учащихся 10-ого класса, обоснуйте его. Создайте в Access базу данных по этому проекту. В созданной базе данных создайте какой-нибудь запрос по двум полям на языке SQL.

Оценка

Оцените значение языка SQL при работе с базами данных.

1. Каким инструментом обработки данных является SQL?
2. Каким языком является SQL?
3. Что является теоретической основой языка SQL?
4. Какой оператор языка SQL управляет доступом к БД?
5. Каковы функции операторов SELECT, INSERT, DELETE, UPDATE?
6. Какую функцию выполняет оператор BETWEEN?
7. Как работают критерии отбора WHERE?
8. Какова функция оператора ORDER BY?



Задание

1) Напишите SQL-запрос в SQL для данных, приведенных в таблице 1.

Таблица 1

| Марка автомобиля | Год выпуска | Цена(тенге) |
|---------------------|-------------|-------------|
| Audi | 1998 | 1500000 |
| Volkswagen | 2007 | 1650000 |
| Toyota Land Cruiser | 1995 | 2500000 |
| Mercedes-Benz E 200 | 2009 | 4900000 |
| Skoda Rapid | 2013 | 4000000 |

2) Напишите SQL-запрос для данных, приведенных в таблице 2.

Таблица 2

| Марка автомобиля | Пробег(км) |
|---------------------|-------------|
| Audi | 486000 |
| Volkswagen | 111111 |
| Toyota Land Cruiser | 239000 |
| Mercedes-Benz E 200 | 134000 |
| Skoda Rapid | 780000 |
| * | |

3) Определите по данным таблицы 1, какие результаты дает SQL-запрос.

```
SELECT Авто. [Марка автомобиля], Авто. [Пробег (км)],  
Авто. [Цена (тенге)]  
FROM Авто
```

4) Определите по данным таблицы 2, какие результаты дает SQL запрос.

```
SELECT Авто. [Марка автомобиля], Авто. [Цена(тенге)]  
FROM Авто  
WHERE Авто. [Цена (тенге)] BETWEEN 2000000 AND 4000000
```



Как выполнить SQL-запросы в базе данных?



Применение

Практическая работа



В Access разработайте базу данных с названием «Класс», содержащую данные об учениках 10-го класса.

| № | ФИО | Дата рождения | Адрес | Вес | Рост | Вид спорта, которым занимается |
|---|-----|---------------|-------|-----|------|--------------------------------|
| 1 | ... | | | | | |
| 2 | ... | | | | | |

Выполните следующие задания с созданием SQL-запросов. Выполните запросы в заданиях 1 и 2 и определите результат.



Задания

Задание 1.

```
SELECT Класс.[ ФИО], Класс. [Адрес ] , Класс. [Рост]
FROM Класс
ORDER BY Авто.[Рост] DESC
```

Задание 2.

```
SELECT Класс.[ ФИО],Класс. [Вес]
FROM Класс
WHERE Класс[Вес] IN (50, 60)
```

Задание 3. Разработайте программу.

Выберите столбцы **ФИО** и **рост** учеников, выведите имена учеников ростом до 160 см в порядке: **ФИО, рост**.

Задание 4. Разработайте программу.

Выберите столбцы **ФИО** и **вес** учеников, выведите имена учеников весом до 45 кг в порядке: **ФИО, вес**.

Глоссарий

Арифметико-логическое устройство (АЛУ) (англ. arithmetic and logic unit, ALU) – блок процессора, который под управлением *устройства управления* (УУ) служит для выполнения арифметических и логических преобразований (начиная от элементарных) над данными, называемыми в этом случае операндами. Разрядность операндов обычно называют размером или длиной машинного слова.

База данных – совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

Веб-программирование – раздел программирования, ориентированный на разработку веб-приложений (программ, обеспечивающих функционирование динамических сайтов Всемирной паутины). Языки веб-программирования – это языки, которые в основном предназначены для работы с веб-технологиями. Языки веб-программирования можно условно разделить на две пересекающиеся группы: клиентские и серверные.

Виртуальная память (англ. Virtual memory) – технология управления памятью ЭВМ, разработанная для многозадачных операционных систем.

Двоичная система счисления – позиционная система счисления с основанием 2. Благодаря непосредственной реализации в цифровых электронных схемах на логических вентилях двоичная система используется практически во всех современных компьютерах и прочих вычислительных электронных устройствах.

Дефрагментация – перераспределение файлов и логической структуры диска для обеспечения непрерывной последовательности кластеров. Вследствие дефрагментации ускоряется чтение и запись файлов, а, следовательно, и работа программ по причине непрерывности чтения диска, то есть без дополнительных сдвигов головки жесткого диска.

Джон фон Нейман (1903–1957) – американский математик. Наиболее известен как человек, с именем которого связывают архитектуру большинства современных компьютеров (так называемая архитектура фон Неймана), применение теории операторов к квантовой механике (алгебра фон Неймана), а также как участник Манхэттенского проекта и как создатель теории игр и концепции клеточных автоматов.

Дизъюнкция (лат. disjunctio «разобшение»), логическое сложение, логическое ИЛИ, включающее ИЛИ; иногда просто ИЛИ – логическая операция, по своему применению максимально приближенная к союзу «или» в смысле «или то, или это, или оба сразу».

Информационная безопасность – это процесс обеспечения конфиденциальности, целостности и доступности информации.

Информационная система (сокр. ИС) – система, предназначенная для хранения, поиска и обработки информации и соответствующих организационных ресурсов (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Конфиденциальная информация – информация, являющаяся конфиденциальной, то есть «доверительной, не подлежащей огласке, секретной»; это понятие равнозначно с понятиями тайны или секрета.

Логическая операция – в программировании операция над выражениями логического (булевского) типа, соответствующая некоторой операции над высказываниями в алгебре логики. Как и высказывания, логические выражения могут принимать одно из двух истинностных значений – «истинно» или «ложно». Логические операции служат для получения сложных логических выражений из более простых. В свою очередь, логические выражения обычно используются как условия для управления последовательностью выполнения программы.

Материнская плата (от англ. *motherboard*, *MB* или англ. *mainboard* – главная плата) – печатная плата, являющаяся основой построения модульного устройства, например – компьютера. Материнская плата содержит основную часть устройства, дополнительные же или взаимозаменяемые платы называются *дочерними*, или *платами расширений*.

Мобильное приложение (англ. «*Mobileapp*») – программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах.

Мультимедиа (англ. **multimedia**) – контент, или содержимое, в котором одновременно представлена информация в различных формах –звук, анимированная компьютерная графика, видеоряд.

Код – взаимно однозначное отображение конечного упорядоченного множества символов, принадлежащих некоторому конечному алфавиту, на иное, необязательно упорядоченное, как правило, более обширное множество символов для кодирования, передачи, хранения или преобразования информации.

Конъюнкция (от лат. conjunctio 'союз, связь') – логическая операция, по смыслу максимально приближенная к союзу «и». Синонимы: **логическое «И», логическое умножение**, иногда просто **«И»**.

Компьютерная шина (англ. *computerbus*) в архитектуре компьютера – подсистема, служащая для передачи данных между функциональными блоками компьютера. В устройстве шины можно различить механический, электрический (физический) и логический (управляющий) уровни.

Криптография (от др.-греч. κρυπτός – скрытый и γράφω – пишу) – это область знаний о методах обеспечения конфиденциальности (невозможности прочтения информации посторонним), целостности данных (невозможности незаметного изменения информации), аутентификации (проверки подлинности авторства или иных свойств объекта), а также невозможности отказа от авторства.

Обработка изображений – любая форма обработки информации, для которой входные данные представлены изображением, например, фотографиями или видеокадрами. Обработка изображений может осуществляться как для получения изображения на выходе (например, подготовка к полиграфическому тиражированию, к телетрансляции и т. д.), так и для получения другой информации (например, распознавание текста, подсчет числа и типа клеток в поле микроскопа и т. д.). Кроме статичных двумерных изображений, обрабатывать требуется также изображения, изменяющиеся со временем, например, видео.

Операционная система, сокр. **ОС** (англ. *operatingsystem, OS*) – комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем.

Первичный ключ (англ. *primarykey*) – в реляционной модели данных один из потенциальных ключей отношения, выбранный в качестве основного ключа (или ключа по умолчанию).

Программное обеспечение – компьютерные программы, процедуры и, возможно, соответствующая документация и данные, относящиеся к функционированию компьютерной системы.

Процедура – взаимосвязанная последовательность действий в программировании.

Разрядность процессора – это величина, которая определяет размер машинного слова, то есть количество информации, которой процессор обменивается с оперативной памятью.

Резервное копирование (англ. backup) – процесс создания копии данных на носителе (жестком диске, дискете и т. д.), предназначенном для восстановления данных в оригинальном или новом месте их расположения в случае их повреждения или разрушения.

Реляционная база данных – база данных, основанная на реляционной модели данных.

Понятие «реляционный» основано на англ. *relation* («отношение», «зависимость», «связь»). Использование реляционных баз данных было предложено доктором Коддом из компании IBM в 1970 году. Для работы с реляционными БД применяют реляционные СУБД.

Скрипт – это программа или программный файл сценарий, которые автоматизируют некоторую задачу, которую пользователь делал бы вручную, используя интерфейс программы.

Таблица истинности – это таблица, описывающая логическую функцию. Логическая функция – это функция, у которой значения переменных и значение самой функции выражают истинность. Например, они принимают значения «истина» либо «ложь» (true либо false, 1 либо 0).

Тактовая частота процессора – это количество колебаний за определенный промежуток времени (в данном случае – за секунду).

Утилита (англ. *utility*) – вспомогательная компьютерная программа в составе общего программного обеспечения для выполнения специализированных типовых задач, связанных с работой оборудования и операционной системы (ОС).

Юникод (чаще всего) или **Уникод** (англ. Unicode) – стандарт кодирования символов, включающий в себя знаки почти всех письменных языков мира. В настоящее время стандарт является доминирующим в сети Интернет.

Центральный процессор (ЦП; также центральное процессорное устройство – ЦПУ; англ. *central processing unit, CPU*, дословно – *центральное обрабатывающее устройство*) – электронный блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (код программ), главная часть аппаратного обеспечения компьютера или программируемого логического контроллера. Иногда называют *микропроцессором* или просто *процессором*.

ASCII (англ. *American standard code for information interchange*) – название таблицы (кодировки, набора), в которой некоторым распространенным печатным и непечатным символам противопоставлены числовые коды.

Таблица была разработана и стандартизована в США в 1963 году. Название «ASCII» по-русски часто произносится как [аск(и)и].

SQL (англ. Structured Query Language – язык структурированных запросов) – универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных. Вопреки существующим заблуждениям SQL в его чистом (базовом) виде является информационно-логическим языком, а не языком программирования. Вместе с тем стандарт языка спецификацией SQL/PSM предусматривает возможность его процедурных расширений, с учетом которых язык уже вполне может рассматриваться в качестве языка программирования.

SQL основывается на реляционной алгебре.

HTML (от англ. *HyperText Markup Language* – «язык гипертекстовой разметки») – стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

Использованная литература

1. Агулар Р. HTML и CSS. Основа любого сайта. Изд.: Эксмо, 2010 г.
2. Богомолова О. Б. Обработка текстовой информации: практикум / А. В. Васильев, О. Б. Богомолова – 2006. – 150 с.: ил.
3. Васильева А. В. Работа в электронных таблицах: практикум / А. В. Васильев, О. Б. Богомолова – 2007. – 160 с.: ил.
4. Григорьев Д. В. Внеурочная деятельность школьников. Методический конструктор. – М.: Просвещение, 2010. – 223 с.
5. Гумерова, Г. Х. Основы компьютерной графики [Текст] : учебное пособие /Казань: Изд-во КНИТУ, 2013. – 86 с. : ил.
6. Глушаков С.В. Базы данных / С.В. Глушаков, Д.В. Ломотько. – М.: Харьков: Фолио, 2000. – 504 с.
7. Грацианова Т.Ю. Программирование в примерах и задачах. М.: Лаборатория знаний, 2016. – 373 с.
8. Ливенец М.А. Программирование мобильных приложений в MIT App Inventor. Практикум [Электронный ресурс] / М.А. Ливенец, Б.Б Ярмахов. – 2016 г.
9. Кастро Э. HTML и CSS для создания Web-страниц. Изд.: НТ Пресс, 2006 г.
10. Кошелев В. Е. Access 2007. Эффективное использование / В.Е. Кошелев. – М.: Бином-Пресс, 2013. – 590 с.
11. Муссиано Ч. HTML и XHTML. Подробное руководство 6-е издание. Изд.: Символ-Плюс, 2008 г
12. Сурядный А. С. Microsoft Access 2010. Лучший самоучитель / А.С. Сурядный. – М.: Астрель, ВКТ, 2012. – 448 с.
13. Эрик А. CSS. Каскадные таблицы стилей. Подробное руководство, 3-е издание. Изд.: Символ-Плюс, 2008 г.
14. Jason Tyler. App Inventor for Android. John Wiley and Sons, Ltd, 2011, 464 с.
15. Портал «Интерактивные онлайн-курсы». Режим доступа: <https://htmlacademy.ru/>
16. Портал «Сервер 179». Режим доступа: <https://server.179.ru/vv/653.html>
17. Портал «Интерактивные онлайн-курсы». Режим доступа: <https://old2.aikyn.kz/2018/02/17/43069.html>
18. Портал «Интерактивные онлайн-курсы». Режим доступа: <https://infourok.ru>

СОДЕРЖАНИЕ

Дорогой друг! 3

Раздел 1. КОМПЬЮТЕРНЫЕ СЕТИ И ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

| | |
|--|----|
| 1. Организация компьютерных сетей | 4 |
| 2. Сети. Компоненты сети | 7 |
| 3. IP-адреса | 11 |
| 4. Функции системы доменных имен (DNS) | 16 |
| 5. Информационная безопасность | 20 |
| 6. Методы защиты информации | 23 |
| 7. Методы идентификации личности | 27 |
| 8. Урок проектных работ | 30 |

Раздел 2. ПРЕДСТАВЛЕНИЕ ДАННЫХ

| | |
|--|----|
| 1. Системы счисления | 32 |
| 2. Перевод чисел из одной системы счисления в другую | 35 |
| 3. Перевод из десятичной в восьмеричную и шестнадцатеричную системы счисления | 37 |
| 4. Логические операции (дизъюнкция, конъюнкция, инверсия) | 41 |
| 5. Построение таблиц истинности | 45 |
| 6. Логические элементы компьютера | 48 |
| 7. Логические основы компьютера | 52 |
| 8. Принципы кодирования текстовой информации | 56 |

Раздел 3. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

| | |
|--|-----|
| 1-2. Пользовательские функции и процедуры | 61 |
| 3-4. Функция | 66 |
| 5. Рекурсия | 71 |
| 6. Работа со строками | 75 |
| 7. Строковые методы | 79 |
| 8. Основные строковые методы | 82 |
| 9. Практическая работа. Строковые методы и функции | 86 |
| 10. Работа с файлами | 88 |
| 11-12. Методы сортировки | 92 |
| 13. Определение и основные понятия графа | 98 |
| 14. Алгоритмы на графах | 102 |

Раздел 4. WEB-ПРОЕКТИРОВАНИЕ

| | |
|---|-----|
| 1. Web-проектирование. Понятие о Web-сайте | 105 |
| 2. Основы языка HTML | 109 |
| 3. Указание размера, цвета и типа шрифта в HTML | 114 |
| 4. Построение таблиц в HTML | 117 |
| 5. Применение графики в HTML – документе | 121 |
| 6. Выполнение практических работ на языке HTML | 124 |

| | |
|---|-----|
| 7. Организация гиперссылок в HTML..... | 126 |
| 8. Каскадные таблицы стилей CSS (Cascading Style Sheets) | 130 |
| 9. Типы селекторов в каскадных таблицах стилей | 133 |
| 10. Селекторы в CSS. Идентификационные и контекстные селекторы..... | 135 |
| 11. Добавление элементов мультимедиа в HTML | 140 |
| 12. Использование скриптов | 144 |
| 13-14. Выполнение практической работы в HTML | 149 |
| 15. Разработка базы данных | 153 |
| 16. Связывание Web-страниц с базами данных..... | 157 |
| 17. Действия, выполняемые с базой данных | 160 |

Раздел 5. ИНФОРМАЦИОННЫЕ СИСТЕМЫ

| | |
|---|-----|
| 1. Bigdata | 163 |
| 2. Основные понятия базы данных | 166 |
| 3. Принципы построение баз данных | 171 |
| 4. Реляционная база данных..... | 175 |
| 5. Создание однотабличной базы данных | 179 |
| 6. Создание многотабличной базы данных..... | 184 |
| 7. Создание формы для ввода данных | 187 |
| 8. Создание формы в режиме Конструктор. Практическая работа | 191 |
| 9-10. Простой метод построения отчета..... | 192 |
| 11. Структурированные запросы | 197 |
| 12-13. Использование языка структурированных запросов (SQL) для данных ВЫБОР | 201 |
| 14. Практическая работа по Sql – запросам..... | 207 |
| Глоссарий..... | 208 |
| Использованная литература..... | 213 |

Кадиркулов Роман Алауович
Нурмуханбетова Гулира Кенжеевна

ИНФОРМАТИКА

Жаратылыстану-математика бағытындағы жалпы білім беретін мектептің 10-сынып оқушыларына арналған оқулық Учебник для учащихся 10 класса общеобразовательной школы естественно-математического направления

Редакторы / Редактор – Е. А. Морозова
Спецредакторлары / Спецредакторы Б. А. Айтуллина, В. Г. Архипова
Әдіскер / Методист – О. С. Дзержинская
Суретшісі / Художник – Г. М. Хасенов
Мұқаба / Обложка: А. М. Әбдіразақ, Б. Б. Булатов, Е. С. Жузбаев
Беттеуші / Верстка – С. М. Ахметовой

Басуға 18.06.2019 ж. қол қойылды.
Пішімі 70x100 $\frac{1}{16}$, Есептік баспа табағы 12,36.
Шартты баспа табағы 17,42. Офсеттік басылым.
Әріп түрі «Open Sans». Офсеттік қағаз.
Таралымы 15 000 дана. Тапсырыс № 2151.

Сапасы жөнінде мына мекемеге хабарласыңыз:
Қазақстан Республикасы,
050012, Алматы қаласы, Жамбыл көшесі, 111-үй,
«АЛМАТЫКІТАП БАСПАСЫ» ЖШС,
тел. +7 (727) 250 29 58, факс +7 (727) 292 81 10.
e-mail: info@almatykitap.kz

Сапа және қауіпсіздік стандарттарына сай.
Сертификация қарастырылмаған.
Сақтау мерзімі шектелмеген.

Подписано в печать 18.06.2019 г.
Формат 70x100 $\frac{1}{16}$. Уч.-изд. л. 12,36.
Усл. печ. л. 17,42. Печать офсетная.
Гарнитура «Open Sans». Бумага офсетная.
Тираж 15 000 экз. Заказ № 2151.

С претензиями по качеству обращаться:
Республика Казахстан,
050012, г. Алматы, ул. Жамбыла, 111,
ТОО «АЛМАТЫКІТАП БАСПАСЫ»,
тел. +7 (727) 250 29 58; факс +7 (727) 292 81 10.
e-mail: info@almatykitap.kz

Соответствует всем стандартам качества и безопасности.
Сертификация не предусмотрена.
Срок годности не ограничен.

Түркияда басылды / Отпечатано в Турции
Elma Basım Yayın ve İletişim Hizmetleri San.Tic. Ltd.Şti.
Tevfikbey Mahallesi Halkalı Caddesi No:162/7 Sefaköy-Küçükçekmece-İstanbul

Приобрести учебную и художественную литературу можно в книжных магазинах «АЛМАТЫКІТАП»:

г. Нур-Султан:

- ул. Иманова, 10, тел.: +7 (7172) 53 70 84, 27 29 54;
- пр. Б. Момышулы, 14, тел.: +7 (7172) 42 42 32, 57 63 92;
- пр. Женис, 67, тел.: +7 (7172) 29 93 81; 29 02 12.

Коммерческий отдел, тел.: +7 (727) 292 92 23, 292 57 20, e-mail: sale1@almatykitap.kz

Интернет-магазин: www.flip.kz
Электронные учебники: www.opiq.kz

Об имеющихся книгах и новинках вы можете узнать на сайте www.almatykitap.kz

г. Алматы:

- пр. Абая, 35/37, тел.: +7 (727) 267 13 95, 267 14 86;
- ул. Гоголя, 108, тел.: +7 (727) 279 29 13, 279 27 86;
- ул. Кабанбай батыра, 109, тел.: +7 (727) 267 54 64, 272 05 66;
- ул. Жаңдосова, 57, тел.: +7 (727) 303 72 33, 374 98 59;
- пр. Гагарина, 76, тел. +7 (727) 338 50 52;
- ул. Майлина, 224а, тел. +7 (727) 386 15 19;
- ул. Толе би, 40/1, тел.: +7 (727) 273 51 38, 224 39 37.