

Рекомендовано Министерством образования и науки
Республики Казахстан

Г.И. Салгараева
К.А. Калымова
Ж.А. Орынтаева

ИНФОРМАТИКА

Учебник для 9 класса общеобразовательной школы

9



УДК 373.167.1
ББК 32.973 я 72
С 16

Научный консультант:

Ж.У. Кобдикова – доктор педагогических наук.

С 16 Салгараева Г.И. и др.
Информатика: Учебник для 9 кл. общеобразоват. шк. /
Г.И. Салгараева, К.А. Калымова, Ж.А. Орынтаева. – Нур-Султан:
Издательство «Арман-ПВ», 2019. – 176 стр.

ISBN 978-601-318-238-4

Учебник «Информатика» для 9 класса разработан в соответствии с Типовой учебной программой основного среднего образования по обновленному содержанию с учетом возрастных особенностей учащихся. Материал учебника изложен доступным языком, содержание включает дополнительные сведения.

УДК 373.167.1
ББК 32.973 я 72

© Салгараева Г.И.,
Калымова К.А.,
Орынтаева Ж.А., 2019

ISBN 978-601-318-238-4

© Издательство «Арман-ПВ», 2019

Репродуцирование (воспроизведение) любым способом данного издания без договора с издательством запрещается.

УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

Задания для самостоятельного усвоения темы – задания для формирования функциональной грамотности

1

Отвечаем на вопросы

2

Думаем и обсуждаем

3

Анализируем и сравниваем

4

Выполняем в тетради

5

Выполняем на компьютере

6

Делимся мыслями

7

Домашнее задание

Вспомните!

Вопросы по пройденному материалу, направленные на изучение новой темы

Вы узнаете:

Ожидаемые результаты освоения материала; учебные цели

Творческое задание

Задания на реализацию творческих способностей

Это интересно!

Дополнительная информация, относящаяся к содержанию темы

Словарь

Научные термины на трех языках

Важная информация

Сведения для углубленного изучения темы

Внимание

При необходимости вы всегда сможете найти CD с электронным приложением на сайте *arman-pv.kz* и загрузить его на свой компьютер для дальнейшей работы

Введение

Дорогие ученики!

В этом учебном году вы продолжите изучение курса информатики. Содержание учебника охватывает различные области предмета «Информатика».

Учебник состоит из пяти разделов. В первом разделе вы познакомитесь с понятием информации, свойствами информации, узнаете о критической оценке рисков при длительном использовании компьютера и соблюдении мер безопасности в Сети, изучите правила сетевого этикета. Также в данном разделе вы узнаете об облачных технологиях, рассмотрите виды облачных ресурсов и преимущества использования онлайн-ресурсов.

В процессе изучения второго раздела вы ознакомитесь с основными параметрами конфигурации компьютера и характеристиками, влияющими на выбор компьютера.

Содержание третьего раздела посвящено изучению баз данных и их возможностей, рассмотрению типов данных и объектов.

В четвертом разделе вы продолжите расширять свои знания в области программирования на языке Python.

Пятый раздел содержит информацию о создании 2D игр с помощью библиотеки PyGame на языке программирования Python.

Каждый параграф содержит комплекс заданий «Отвечаем на вопросы», «Думаем и обсуждаем», «Анализируем и сравниваем», «Выполняем в тетради». Задания «Выполняем на компьютере» предназначены для закрепления полученных знаний на практике, а задания «Делимся мыслями» – для переноса знаний и умений на решение проблемных ситуаций в повседневной жизни. «Домашнее задание» предназначено для повторения и усвоения темы самостоятельно.

К учебнику прилагается электронное приложение (CD-диск), которое содержит интерактивные и тестовые задания.

Желаем успехов!

РАЗДЕЛ I

РАБОТА С ИНФОРМАЦИЕЙ

Изучив раздел, вы научитесь:

- определять свойства информации;
- критически оценивать риски, связанные с продолжительным использованием компьютеров;
- осуществлять работу с документами с использованием облачных технологий;
- рассуждать о последствиях нарушения этических и правовых норм в сети.

§ 1. Свойства информации

Вспомните!

- Что такое информация?
- Какие виды информации вы знаете?
- Что такое объем информации?
- Какие свойства информации вы знаете?

Вы узнаете:

- об основных свойствах информации;
- как используются свойства информации;
- о свойствах информации в различных ситуациях.

Свойства информации –
Ақпараттың қасиеттері –
Information Properties

Основные свойства информации

Как и любой другой объект, информация тоже имеет свои отличительные свойства. Отличительной чертой информации от объектов природы и общества является ее дуализм (двойственность): с одной стороны, информация объективна в силу объективности данных, с другой – субъективна, в силу субъективности применяемых для работы с ней методов.

В информатике выделяют множество свойств информации: объективность, надежность, полнота, точность, актуальность, полезность, ценность, своевременность, достоверность, доступность, ясность и т.д.

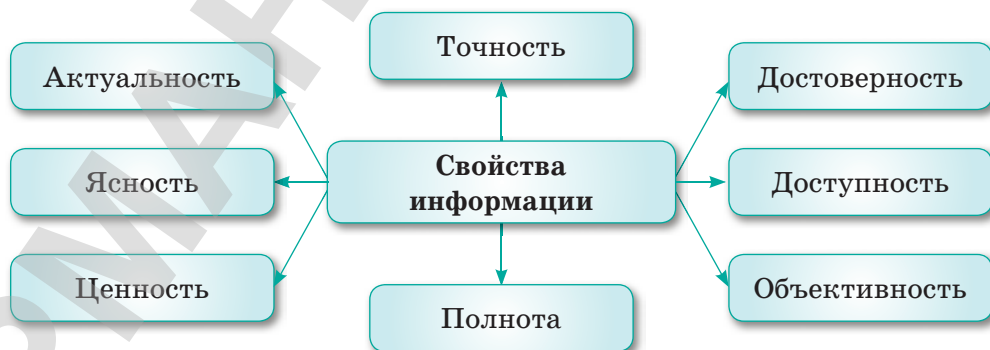


Схема 1. Основные свойства информации

Актуальность. Информация, предоставленная вовремя, дающая точные сведения: *например*, «В данное время на вашем счете 1360 тенге».

Точность. Это свойство характеризует степень соответствия информации реальности в необходимой степени. Точность – это основное свойство памяти: *например*, « $2 * 25 = 50$ ».

Достоверность. Достоверная информация – это информация, отражающая истинное положение дел. *Например*, «В сутках 24 часа» – достоверная, а «Жизнь на Марсе существует» – недостоверная информация.

Ценность информации позволит решить поставленную задачу, достичь цели: «Возьмите зонт сегодня, так как ожидается дождь».

Полнота характеризует достаточность данных для принятия решения или создания новых данных на основе имеющихся. *Например*, полная информация – «4 июня – день рождения мамы»; не полная – «День рождения мамы летом».

Объективность и субъективность. Объективная информация – это информация, которую можно проверить и измерить. В отличие от субъективной, эта информация не зависит от человеческого фактора. *Например*, объективная информация – «Нур-Султан является столицей Казахстана», а информация «Все мальчики в 9 классе учатся хорошо» – субъективная.

Доступность. Возможность получения информации при необходимости. *Например*, если у вас нет компьютера, то информация из книги вам более доступна, чем с любого CD-диска.

Ясность. Информация ясна, если она выражена на языке, доступном для получателя. *Например*, ясной будет информация об эвакуации при пожаре, написанная на вашем родном языке; непонятная информация: 火灾 报警 信号 撤离.

В разных ситуациях одна и та же информация может обладать разными свойствами. Человек или техническое устройство из потока информации в информационном процессе определяет свойства информации для каждого отдельного случая.

1

Отвечаем на вопросы

1. Перечислите свойства информации.
2. Можно ли оценить информацию по ее свойствам?
3. Можно ли определить информационное содержание сообщений?

2

Думаем и обсуждаем

1. Почему свойства информации важны для человека?
2. В чем причина изменения свойств информации?

3

Анализируем и сравниваем

Сравните различия между следующими свойствами информации: точностью и ценностью, доступностью и актуальностью.

4

Выполняем в тетради

1. Предоставьте информационную характеристику следующих информационных объектов:
 - а) эссе по теме;
 - б) геометрические задачи;
 - в) реферат по истории.
2. Установите соответствия между свойствами информации и их определениями.

Свойство информации	Определение свойства информации
Актуальность	Достаточность информации для принятия решения
Ясность	Нет зависимости от мнения других
Объективность	Отражение истинного положения дел
Доступность	Информация, выраженная на понятном для получателя языке
Полнота	Степень соответствия информации данному моменту времени
Точность	Возможность получения информации
Достоверность	Степень соответствия информации конкретному объекту

5

Выполняем на компьютере

- Разделитесь на группы и выполните следующие задания.
1. Подготовьте презентацию по свойствам информации.

8

2. Наберите таблицу в MS Word.

Свойства информации	Определение	Примеры
Актуальность		
Точность		
Объективность		
Доступность		
Полнота		
Ценность		
Достоверность		
Ясность		

6

Делимся мыслями

С какими свойствами информации вы сталкиваетесь в повседневной жизни?

7

Домашнее задание 

Выполните задания на CD-диске, связанные со свойствами информации.

Творческое задание:

Дополнительно изучите другие свойства информации. Приведите примеры.

§ 2. Критическая оценка рисков, связанных с продолжительным использованием компьютера

Вспомните!

- Какие свойства информации вы знаете?
- Можно ли оценить информацию по ее свойствам?
- Почему человеку важна информация?

Вы узнаете:

- о негативном воздействии компьютера на организм человека;
- о методах защиты от негативного воздействия компьютера;
- об опасности длительного использования компьютера.

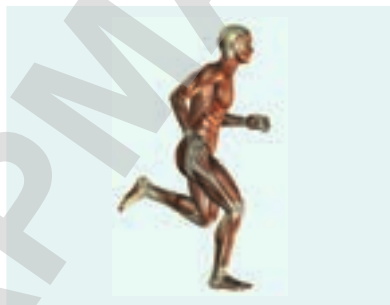
Интерфейс – Интерфейс – *Interfae*

Человеко-компьютерное взаимодействие – Адам мен компьютердің өзара әрекеттесуі – *HCI (human-o mputer interat ion)*

С развитием научно-технического прогресса в конце XX века стремительными темпами стали появляться компьютеры и компьютерные технологии. Сложно представить современную жизнь без ноутбуков, планшетов, мобильных телефонов. Они внедряются в нашу жизнь, занимая свое место в нашем сознании, и мы порой не осознаем, что становимся зависимыми от них.

Влияние компьютера на здоровье человека является важной проблемой современности. По словам специалистов, общение с компьютером должно быть строго дозировано, необходимо следовать элементарным правилам работы за компьютером. Длительное времяпровождение у компьютера может привести к проблемам со здоровьем всего организма в целом (*таблица 1*).

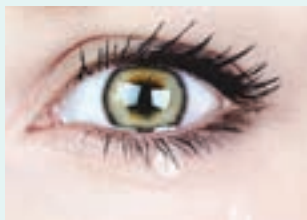
Таблица 1. Влияние компьютера на организм человека



Проблемы с мускулатурой. При длительном малоподвижном сидении за компьютером в организме человека нарушается циркуляция крови. Без движения мышцы теряют тонус и атрофируются. Чем меньше тонус, тем больше становится нагрузка на кости, суставы и позвоночник



Проблемы с позвоночником. Сидячая работа рано или поздно приводит к позвоночному синдрому, болям в спине, остеохондрозу и сколиозу. Любая поза при длительной фиксации вредна для опорно-двигательного аппарата, а малоподвижный образ жизни вредит физическому здоровью человека



Влияние на зрение. Во время работы с компьютером происходит колоссальное напряжение сетчатки глаза и глазных мышц. Появляется резь в глазах, глаза начинают слезиться, изображение на мониторе становится неясным. Глаза утомляются, а зрение ухудшается



Влияние на пищеварение. У человека при постоянном взаимодействии с компьютером нарушается режим питания. А нерегулярное питание и малоподвижный образ жизни могут привести к ожирению и болезням желудочно-кишечного тракта



Влияние на психику. ПК – это техника, которая может медленно загружаться, давать сбои, или просто выйти из строя. Люди при этом могут испытывать постоянный стресс, становятся раздражительными, появляются повышенная агрессивность и тревожно-депрессивное состояние



Электромагнитное излучение, как показывают современные исследования, может вызвать нарушение центральной нервной системы, стать причиной значительных изменений гормонального состояния и возникновения опасных заболеваний. Человек получает больше негативного электромагнитного излучения от компьютеров, чем от другой бытовой техники



Компьютер и органы дыхания. Заболевания органов дыхания, развивающиеся из-за длительного времени работы с компьютером, носят аллергический характер, так как во время работы компьютера корпус монитора и процессор нагреваются и выделяют в воздух вредные вещества



Кистевой туннельный синдром – болезненное состояние, вызванное прогрессирующим сжатием ключевых нервов в запястье. Человек целый день работает за компьютером, стараясь не замечать чувства онемения в руке, иногда ощущает внезапную, острую и колющую боль и «прострелы» через все запястье



Риск развития компьютерной зависимости – одна из наиболее серьезных проблем современного общества. Она лишает человека таких качеств, как социальное общение и оценка ценностей. В группу риска чаще всего попадают дети, чья нервная система еще неустойчива и нестабильна

Лечебная гимнастика

Существует множество упражнений с поворотами в разные стороны для развития мускулатуры спины. Очень полезны плавание и упражнения на турнике. Работая в офисе, старайтесь чаще делать зарядку, больше двигаться и проветривать помещение.

Упражнения для кистей рук

- С усилием сожмите пальцы рук в кулак и разожмите их;
- потрясите расслабленными кистями, постепенно поднимая их вверх и в стороны;
- переплетите пальцы рук и делайте сжимающие движения;
- сделайте вращательные движения кистями в стороны и внутрь;
- по очереди вытягивайте по одному пальцу вперед и крутите им по часовой стрелке или наоборот.

1

Отвечаем на вопросы

1. Назовите негативные факторы, влияющие на организм человека при работе за компьютером.
2. К каким последствиям приводит долгая работа за компьютером?
3. Какие правила техники безопасности во время работы за компьютером вы знаете?
4. Как предотвратить кистевой туннельный синдром?

2

Думаем и обсуждаем

1. Почему долгое времяпровождение за компьютером может привести к различным заболеваниям?
2. Твой друг, проигрывая в компьютерные игры, становится нервным и раздражительным. Как можно помочь ему?
3. Почему нужно часто проводить гимнастику при работе за компьютером?

3

Анализируем и сравниваем

Проанализируйте отрицательное влияние компьютера на мускулатуру, зрение, нервную систему, органы дыхания, и в целом на организм человека.

4

Выполняем в тетради

1. Разделитесь на группы. Приведите примеры основных факторов негативного влияния компьютера на организм человека.
 - 1) Нагрузка на глаза.
 - 2) Проблемы, связанные с мышцами и суставами.
 - 3) Синдром компьютерного стресса.
 - 4) Влияние на психику.
2. Правила техники безопасности при работе на компьютере. Заполните пустые места.
 - 1) Не работать ... на компьютере.
 - 2) Компьютеры ... не трогать.
 - 3) Не трогать ... монитор.
 - 4) Соблюдать ... при работе на компьютере.
 - 5) Компьютер ... в чистоте.
 - 6) Не выключать ... компьютер.

5

Выполняем на компьютере

1. Составьте в MS Word таблицу, связанную с угрозами для здоровья от длительного использования компьютера, и о том, какие меры защиты существуют.
2. Наберите в MS Word инструкцию по эксплуатации компьютера, в которой рассказывается о профилактике заболеваний, возникающих при длительной работе за компьютером.
3. Выполните задания на CD-диске по данной теме.

6

Делимся мыслями

Что вы узнали на уроке? Чему научились? Поделитесь своим мнением с друзьями. В каких жизненных ситуациях можно применить знания, полученные на уроке? Приведите примеры.

7

Домашнее задание

Запишите сведения о стандартах мониторов, разработанных для предотвращения заболеваний, возникающих при длительной работе за компьютером (составьте сравнительную таблицу).

Творческое задание

Подготовьте презентацию на тему «Негативное воздействие компьютера на организм человека».

§ 3. Совместная работа с документами

Вспомните!

- О негативном воздействии компьютера на организм человека.
- Расскажите о методах защиты от негативного воздействия компьютера.
- К каким последствиям приводит продолжительная работа за компьютером?

Вы узнаете:

- об облачных технологиях;
- об облачных моделях;
- об эффективности облачных технологий;
- о видах облачных ресурсов.



Рис. 1. Модель облака

Облачные технологии –
Бұлттық технологиялар –
Cloud technology

Одна из популярных тенденций в области IT-индустрии в последнее время – это **облачные технологии** (рис. 1).

Облачные технологии, cloud computing (англ. *cloud* – «облако», *computing* – «вычисления», *technology* – «технология») – это сервис, предоставляющий удаленный доступ через Интернет или локальную сеть к определенным ресурсам (программам, данным, файлам и т.д.).

Облачные технологии позволяют пользователю обрабатывать информацию через Интернет в режиме онлайн. Облачные технологии – это обобщенное название для совокупности различных служб.

Основное отличие в работе с облачными технологиями от обычных программ является то, что пользователь использует не только ресурсы своего компьютера, но и интернет-услуги отдаленных мощных серверов. При этом пользователь получает полную возможность работать со своими источниками, но эти источники не могут препятствовать операционной системе, базе программ, работе вычислительных серверов и изменять их.

Облачные технологии, в зависимости от назначения и целей, делятся на четыре вида.

Частное облако (private cloud) – это инфраструктура, предназначенная только для работы отдельных частных лиц (представителей частной организации, компании) и их потребителей. Отдельные облака могут быть установлены на самих серверах этих организаций, а для сторонних лиц – расположены на центральных серверах крупных провайдеров (рис. 2).



Рис. 2. Частное облако

Публичное облако (public cloud) – инфраструктура, предназначенная для свободной работы с ресурсами всех пользователей, имеющих доступ к Интернету. Такие службы, как Google, Yahoo, электронная почта, социальные сети (Facebook, Twitter и др.) можно рассматривать как пример публичного облака (рис. 3).

Общественное облако (community cloud) – инфраструктура, предназначенная для пользования ресурсами конкретным сообществом пользователей из организаций, объединенных общей целью и одинаковыми запросами (рис. 4).



Рис. 3. Публичное облако



Рис. 4. Общественное облако

Гибридное облако (hybrid cloud) – комбинация двух или более видов облаков (частного, публичного, общественного). Данная модель облака может быть использована в филиалах компаний, географически удаленных друг от друга, или в крупных компаниях со множеством программных систем (рис. 5).



Рис. 5. Гибридное облако

По функциональности облачные технологии предоставляют по сути одинаковые услуги обычным пользователям: загрузить и установить небольшую по объему программу, создать папки для сохранения файлов, расположенных в облаке. Для этого пользователю необходимо настроить параметры синхронизации для файлов и папок, расположенных на локальном компьютере, или в облаке. Благодаря web-интерфейсу все сохраненные данные в облаке всегда доступны с любого устройства. Получив соответствующую ссылку через электронную почту, любой человек имеет свободный доступ к использованию этих папок и файлов.

Облачное хранилище файлов – это место хранения информации, доступной в сети Интернет, и расположенное на оборудовании поставщика, оказывающего услуги облачных служб.

Существует множество видов облачных ресурсов. Например: Dropbox, SugarSync, Box.net, GoogleDisk, GoogleDocs, OpenDrive, Windows Live SkyDrive, Яндекс.Диск и др (рис. 6).



Рис. 6. Виды облачных ресурсов

Dropbox – место для хранения файлов с возможностью совместной работы при подключении компьютера к Интернету. Синхронная папка с облачным сервером создается автоматически.

SugarSync – облачное хранение данных, предназначенное для резервного копирования пользовательской информации.

Box.net – облачный сервис, обеспечивающий хранение и размещение файлов для персональных аккаунтов и предприятий.

Windows Live SkyDrive (OneDrive) – служба для совместной работы онлайн с документами Microsoft Office и его web-версией Microsoft Office Web Apps, функционирующими в web-браузерах (рис. 7).

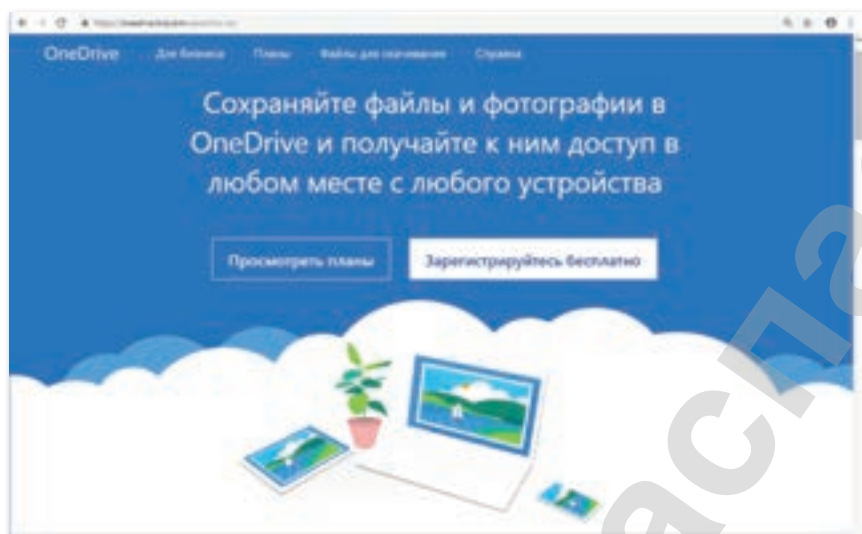


Рис. 7. Личная страница облачного сервиса OneDrive

Яндекс.Диск – облачный сервис, принадлежащий компании Яндекс, который позволяет пользователям хранить и передавать свои данные на облачных серверах.

OpenDrive – облачное хранилище файлов, позволяющее управлять ими, их версиями, выполнять резервное копирование и работать совместно.

Synplicity используется в среде кроссплатформ для синхронизации программного комплекса.

Google Docs – удобный и простой онлайн-офис Google, предназначенный для бесплатной работы пользователя с документами. Подготовленные документы можно через любое оборудование открыть, форматировать, отправить, сохранить в облаке. Для пользователей GoogleDocs создаются специальные персональные приложения вместо различных приложений и сайтов.

Самое главное, нужно пройти процедуру регистрации, чтобы использовать возможности облачного сервиса. К примеру, для того чтобы пользоваться облачным сервисом Google Docs, необходимо создать свой аккаунт.

Аккаунт – это учетная запись, представляющая собой набор данных о пользователе.

Чтобы создать аккаунт в Google Docs, пользователь должен иметь электронную почту. Вам нужно открыть главную

страницу поисковой системы Google на сайте *www.google.com* и выбрать ссылку *Приложения Google*. После этого откроется дополнительное меню, в котором необходимо выбрать ссылку *Документы*. Далее откроется личная страница облачного сервиса Google Docs.

Облачный сервис позволяет легко создавать, обмениваться и изменять документы онлайн в Google Docs.

Используя Google Docs, можно создавать текстовые документы, таблицы, презентации. Чтобы создать таблицу, нужно на главной странице Google Docs нажать на кнопку *Создать* и в дополнительном меню выбрать команду *Таблица*, где автоматически откроется пустая таблица в браузере.

Для создания презентации нажмите на кнопку *Создать*, расположенную на главной странице пользователя, и во всплывающем меню выберите элемент *Презентация*.

В облачном сервисе Google Docs можно использовать веб-приложение, чтобы работать с *графическими изображениями*. При создании графического объекта нужно на главной странице пользователя нажать на кнопку *Новое* и выбрать из меню элемент *Рисунок*. Откроется новый, пустой графический файл.

На уроках информатики мы можем использовать в режиме онлайн облачную службу OneDrive, что позволяет программе Skype (Lync) использовать инновационный сервис для упрощения связи независимо от местонахождения пользователей.

Skype (Lync) имеет следующие преимущества: можно подключаться в любое время и с любого места, где есть связь с Интернетом; имеется возможность автоматической адаптации к условной сети.

Преимущества применения облачных технологий:

- с помощью облачных технологий есть возможность использовать программы различных серверов для решения сложных задач;
- возможность доступа к источникам данных пользователя из любой точки мира, в любое время через компьютер;
- возможность беспрепятственного обмена информацией и совместной работы пользователя с другими пользователями;

- в сравнении с другими программами персонального компьютера, услуги облачных технологий в основном бесплатны либо относительно недорогие;
- одно из преимуществ использования облачных технологий в проектах крупных компаний – экономия средств, затрачиваемых на администрирование, поддержку, обновление, лицензирование программного обеспечения.

Недостатки облачных технологий:

- зависимость пользователя от организации-услугиодателя cloud computing;
- появление новых «облачных» монополистов.

1

Отвечаем на вопросы

1. Перечислите виды облачных технологий.
2. В чем польза облачных технологий?
3. Какие сервисы, предоставляющие услуги облачных технологий, вам известны?
4. Какие процедуры выполняются при регистрации на серверах облачных технологий?

2

Думаем и обсуждаем

1. Как использовать возможности облачных технологий в зависимости от их предназначения?
2. Почему используются частные облака?
3. Какие преимущества имеет Skype (Lync)?
4. Назовите преимущества применения облачных технологий.

3

Анализируем и сравниваем

1. Напишите различия между гибридными и публичными облаками.
2. Сравните облачные ресурсы OneDrive и Google Docs.

4

Выполняем в тетради

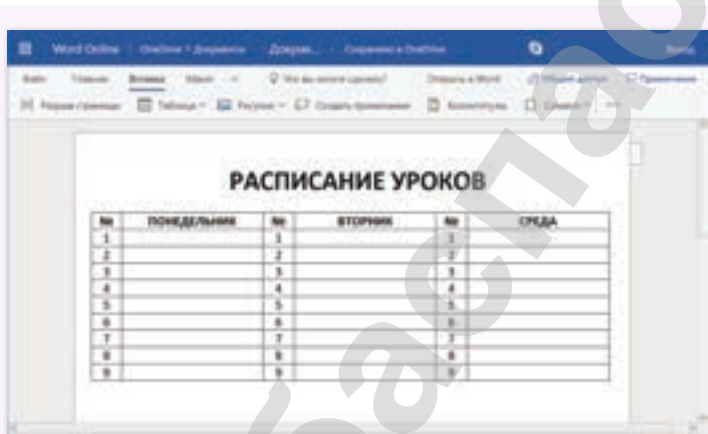
1. Напишите основные преимущества облачных технологий.
2. Напишите основные недостатки облачных технологий.

5

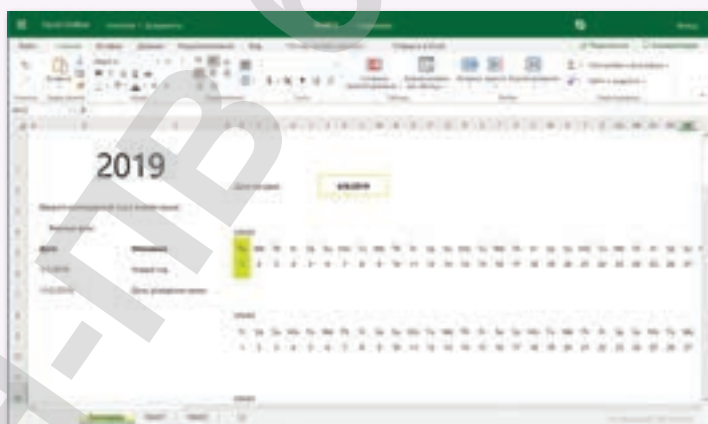
Выполняем на компьютере

Разделитесь на 3 группы, выполните следующие задания в OneDrive:

1. Создайте документ «Расписание уроков класса», используя вкладку Документы в услугах сервиса



2. Создайте «Календарь праздников РК 2020 года», используя услуги сервиса



3. Создайте презентацию на тему «Семья – опора счастья», используя услуги сервиса



22

6

Делимся мыслями

Обсудите с одноклассниками методы работы с облачными ресурсами OneDrive и Google Docs (создание документов, таблиц, презентаций, графических изображений). Какой облачный ресурс оказался наиболее удобным в применении?

7

Домашнее задание

1. Создайте поздравительную открытку или приглашение на торжественное мероприятие, используя графические изображения. Поделитесь ими с классом на одном из облачных ресурсов (GoogleDocs, OneDrive и т.д.).
2. Выполните задания на диске, соответствующие данной теме.

Творческое задание

Подготовьте ответ на вопрос: «Как развиваются облачные технологии в Казахстане?».

§ 4. Сетевой этикет

Вспомните!

- Какие виды облачных технологий вы знаете?
- Назовите модели услуг для потребителей.
- В чем заключается польза облачных технологий?
- Каким может быть объем облачных технологий?

Сетевой этикет – Желілік этикет – *Netiquette*



Вы узнаете:

- что такое сетевой этикет;
- о правилах сетевого этикета;
- что такое флейм.

Интернет стремительно развивается, число его пользователей растет. При работе в Сети, начиная с общения и заканчивая интернет-бизнесом, люди зачастую допускают случайные ошибки, которые возникают от незнания сетевого этикета. Этим вы можете навредить не только собеседнику, но и самому себе. Чтобы избежать неприятных ситуаций в Интернете, нужно придерживаться правил сетевого этикета (схема 2).

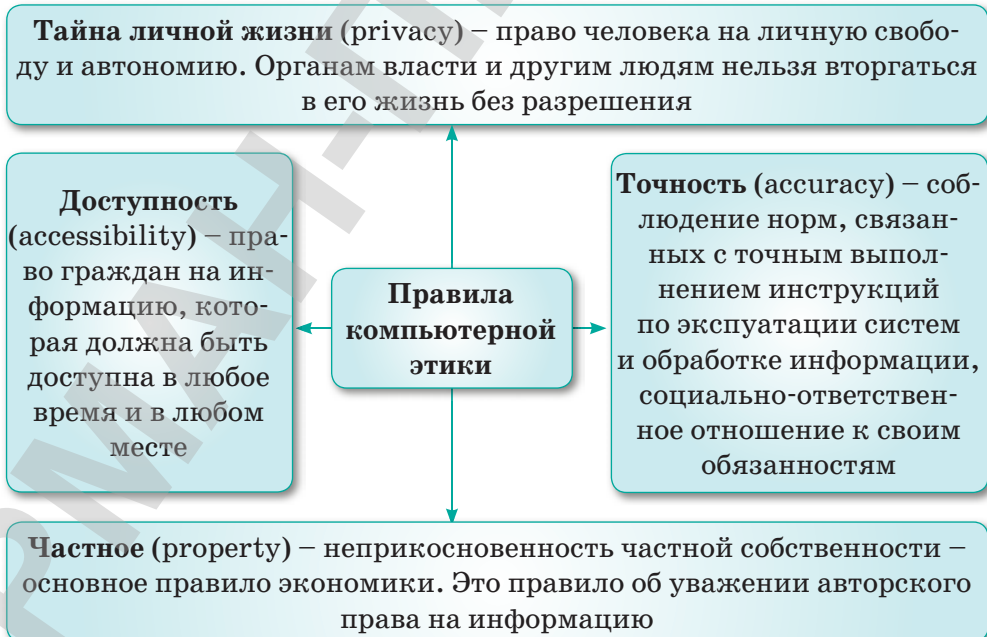


Схема 2. Компьютерная этика

Сетевой этикет. В большинстве случаев несоблюдение сетевого этикета, неприкосновенности частной жизни и прав собственности, отклонения от темы обусловлены рекламой и саморекламой. Также нарушением этикета являются ошибочная, вредоносная либо другая нежелательная информация и плагиат.

Правила этикета в сети:

1. Не забывайте, что разговариваете с человеком.
2. Придерживайтесь правил обычной жизни.
3. Не забывайте, что находитесь в киберпространстве.
4. Уважайте чужое время и возможности.
5. Не теряйте своего достоинства.
6. Помогайте другим людям по мере своих возможностей.
7. Не вмешивайтесь в конфликт и не поддерживайте конфликтующих.
8. Уважайте право на личную переписку.
9. Не злоупотребляйте своими способностями.
10. Учитесь прощать других за их ошибки.

В целом положения сетевого этикета можно разделить на три категории (схема 3).

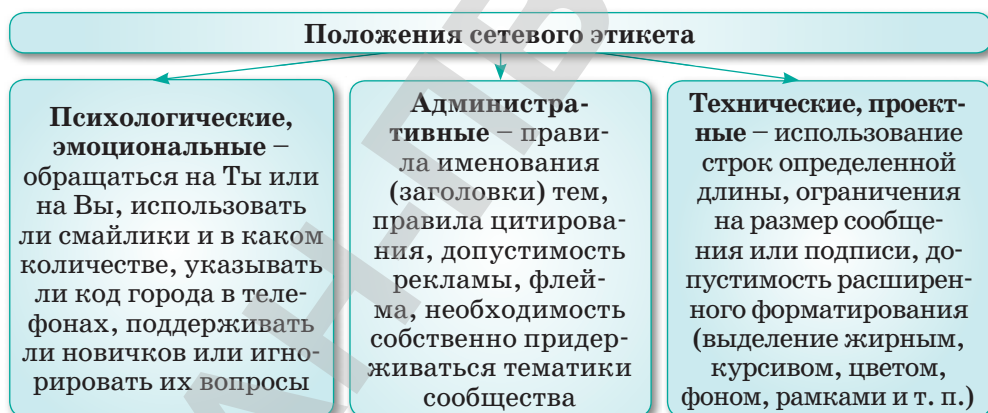


Схема 3. Положения сетевого этикета

Члены одного сетевого сообщества, использующие свои правила, невольно могут нарушить правила другого сообщества. Поэтому следует ознакомиться с правилами всех интернет-сообществ и договориться об их соблюдении. Также есть общие правила, выполняя которые, можно разрешить любые спорные вопросы. Если вы хотите общаться на каком-то форуме или в чате, полезно будет для начала познакомиться с основными правилами данного интернет-сообщества. Прежде

чем написать новое обращение и отправить запрос, не поленились изучить список популярных вопросов пользователей.

Флейм – это эмоциональные замечания, часто высказанные без учета мнения других участников разговора. Это сообщения, где такт – не самое главное, а цель – вызвать реакцию пользователей, «спор ради спора». Крайняя степень выраженности флейма проявляется в случае, когда все забывают, из-за чего начался разговор и начинают ожесточенно ругаться друг с другом. В Интернете флейм обычно наказывается модераторами ресурса. Следовательно, во избежание нежелательных последствий общения, в Интернете следует придерживаться правил сетевого этикета.

1

Отвечаем на вопросы

1. Можно ли отнести к этикету нормы и требования, регулирующие отношения между людьми?
2. Какое отношение имеет слово «этикет» к информатике?
3. Какие слова и обращения являются недопустимыми в Интернете?
4. В чем отличие правил частной жизни и частной собственности?

2

Думаем и обсуждаем

1. Почему нельзя отправлять рекламу в непредназначенные для этого места?
2. Как строятся взаимоотношения между людьми в реальной жизни?
3. Как нарушение этикета влияет на характер человека?
4. Зачем нужно соблюдать нормы этики при работе с информацией?

3

Анализируем и сравниваем

Нужно разделиться на две команды, выбрать координаторов групп.

Первая команда «Нападающие» рассказывает о негативных последствиях использования Сети. Вторая команда «Защитники» рассказывает о положительных сторонах Интернета. Каждый раз результаты записываются на доске.

Вопросы:

- Какие профилактические меры необходимы для предотвращения рисков при пользовании Интернетом?

- Существуют ли риски при работе в Сети?
- Какие нарушения прав пользователей при работе в Сети вам известны?
- Какие меры приняты обществом и государством против нарушения прав пользователей в Сети?
- Вы автор проекта. Опишите ваши действия для его защиты. Каковы ваши права?
- Кто знает больше правил этикета? Подведите итоги.

Задание. Найдите в Интернете законы, указы, решения, касающиеся авторских прав.

4

Выполняем в тетради

1. Из скольких частей состоят нормативные правила?
2. Запишите правила компьютерной этики.

5

Выполняем на компьютере

Создайте таблицу в Word и разделите опасные и безопасные действия в Сети.

- Держите свой пароль в тайне.
- Обновите антивирусную программу.
- Личные данные держите в секрете.
- В социальных сетях ведите себя вежливо.
- Пишите письма друзьям.
- Не заполняйте анкеты в социальных сетях без участия взрослых.
- Не оскорбляйте в социальных сетях других людей.
- Открывайте подозрительные сообщения.
- Разговаривайте с чужими людьми в режиме онлайн.
- Открывайте сообщения подозрительного характера.
- Участвуйте в платных играх, конкурсах, лотереях.

6

Делимся мыслями

Приведите примеры, показывающие последствия нарушений этических и правовых норм при работе в Сети.

7

Домашнее задание

1. Напишите, какие еще правила поведения пользователей в Сети вы бы добавили.
2. Выполните задания на диске, соответствующие данной теме.

ИТОГОВЫЕ ТЕСТОВЫЕ ЗАДАНИЯ К РАЗДЕЛУ I

1. **«Общественное облако» – это вид облака, ...**
 - A) предназначенный для работы частного предприятия и их представителей и потребителей.
 - B) предназначенный для свободного пользования Интернетом обществом.
 - C) предназначенный для общественного пользования лиц с общими целями.
 - D) представляющий собой смешанную комбинацию двух или более видов облаков.
 - E) предлагающий комбинацию нескольких частей облака.

2. **«Гибридное облако» – это вид облака, ...**
 - A) предназначенный для работы частного предприятия и их представителей и потребителей.
 - B) предназначенный для свободного пользования Интернетом обществом.
 - C) предназначенный для общественного пользования лиц с общими целями.
 - D) представляющий собой смешанную комбинацию двух или более видов облаков.
 - E) представляющий собой комбинацию нескольких частей локальной и глобальной сетей.

3. **Облачное хранилище данных, предназначенное для резервного копирования пользовательских данных.**
 - A) SugarSync.
 - B) GoogleDocs.
 - C) Dropbox.
 - D) OpenDrive.
 - E) Vox.net.

4. **Флейм – ...**
 - A) один из видов сетевого общения.
 - B) вход в личное пространство пользователя.
 - C) частная неприкосновенность.
 - D) задания для обработки информации.
 - E) доступность в любое время.

5. Информацию, отражающую истинное положение дел, реальную действительность, называют ...
- A) понятной. D) актуальной.
B) достоверной. E) полной.
C) объективной.
6. Взаимодействие между человеком и компьютером – это ...
- A) интерфейс. D) ярлык.
B) файл. E) папка.
C) каталог.
7. «Облачные технологии» по назначению делятся на ...
- A) частное, общее, публичное, смешанное.
B) полное, смешанное, выборочное.
C) обновленное, противоречивое, выборочное.
D) ценное, целостное, полное, точное.
E) актуальное, смешанное, надежное, полное.
8. Наука, изучающая информацию и информационные процессы, – ...
- A) информатика. D) кибернетика.
B) криптоанализ. E) криптология.
C) криптография.
9. Речь человека относится к ... информации.
- A) тактильной. D) звуковой.
B) графической. E) осязательной.
C) текстовой.
10. Информацию, не зависящую от частного мнения или суждения, можно назвать ...
- A) объективной. D) полной.
B) достоверной. E) полезной.
C) актуальной.
11. Информацию, с помощью которой можно решить поставленную задачу, называют ...
- A) понятной. D) полезной.
B) актуальной. E) объективной.
C) достоверной.

12. Dropbox – ...

- A) место для хранения файлов с возможностью совместной работы при подключении компьютера к Интернету.
- B) облачный сервис, обеспечивающий хранение и размещение файлов для персональных аккаунтов и предприятий.
- C) облачное хранение данных, предназначенное для резервного копирования пользовательской информации.
- D) доступность информации пользователю.
- E) показатель полезности либо бесполезности информации.

13. При общении в Сети нужно ...

- A) проявлять уважение к другим пользователям.
- B) быть твердым в своем убеждении и отстаивать свою правоту.
- C) быть зарегистрированным под своим именем.
- D) регулярно отвечать на сообщения.
- E) быть в постоянном доступе.

14. Ценность информации – ...

- A) степень ее близости к реальному состоянию объекта, процесса, явления и т.п.
- B) уровень соответствия образа, создаваемого с помощью информации, реальному объекту, процессу, явлению.
- C) характеризует качество и достаточность информации.
- D) доступность информации пользователю.
- E) показатель полезности либо бесполезности информации.

15. Сетевой этикет – это ...

- A) правила поведения в Сети.
- B) правила поведения в обществе.
- C) правила отправки сообщений.
- D) правила посещения сайтов.
- E) правила входа в Интернет.

РАЗДЕЛ II

ВЫБИРАЕМ КОМПЬЮТЕР

Изучив раздел, вы научитесь:

- выбирать конфигурацию компьютера в зависимости от его назначения;
- выбирать ПО в зависимости от потребности пользователя;
- разрабатывать и исследовать модели процессов в электронных таблицах.

§ 5. Конфигурация компьютера

Вспомните!

- Что такое процессор?
- В чем предназначение компьютера?

Вы узнаете:

- что такое конфигурация;
- как выбрать конфигурацию компьютера в зависимости от его назначения.

Конфигурация компьютера –
Компьютердің
конфигурациясы –
Computer configuration

В области информационных и компьютерных систем под *конфигурацией* понимают определенный набор комплектующих, основанный на назначении, количестве и ключевых характеристиках компонентов. В базовую конфигурацию компьютера входит минимальный набор комплектующих, состоящий из системного блока, монитора, клавиатуры, мыши. Внутренние устройства, входящие в состав системного блока с базовой конфигурацией, – это материнская плата, процессор, оперативная память (RAM), жесткий диск, видеокарта, блок питания. Конфигурация влияет на функционирование и производительность компьютера (схема 4).



Схема 4. Внутренний состав системного блока

Материнская плата – это главная микросхема компьютера, на которой установлены все основные детали и узлы. При выборе материнской платы нужно обращать внимание на частоту системной шины. Количество различных слотов, которыми оборудуется материнская плата, также имеет немаловажное значение.

Процессор отвечает за обработку информации. Именно в нем производится большинство вычислительных операций при работе машины. Важнейшей его характеристикой является тактовая частота. Чем она выше, тем лучше. Современные процессоры для ПК состоят из 2–6 ядер (двухядерный, четырехядерный и т.д.).

Оперативная память (ОЗУ или RAM). Необходимый ее объем зависит только от рода задач, которые должен будет решать ваш компьютер. В большинстве случаев достаточно комфортная работа на ПК обеспечивается уже с 4 Гб памяти, но для игр и обработки видео ее можно и увеличить. Учитывается также описание в зависимости от выбранной материнской платы.

Накопитель на жестких магнитных дисках, или НЖМД (англ. hard (magnetic) disk drive, HDD, HMDD), жесткий диск, винчестер – запоминающее устройство (устройство хранения информации) произвольного доступа, основанное на принципе магнитной записи. Жесткий диск – аккумулятор, состоящий из нескольких взаимосвязанных дисков. Стороны дисков разделены на концентрационные дорожки, а дорожки – на секторы.

Видеокарта (графическая карта, видеоадаптер) обрабатывает графическую информацию и выводит ее на монитор. В современных играх очень важно качество и скорость обработки информации.

Для более серьезной работы, а также для развлечений и игр необходима дополнительная видеокарта, имеющая собственную память и видеопроцессор. Ее применение существенно повысит качество изображения и улучшит общую производительность системы.

Блок питания. Для большинства конфигураций вполне достаточно мощности 450 Вт. Однако, если на вашем компьютере установлена игровая приставка с улучшенной видеосистемой, возможно, потребуется более мощный источник питания.

Звуковая карта – устройство, позволяющее обрабатывать звук (выводить на акустические системы и/или записывать).

Акустическая система и микрофон используется для воспроизведения звука с компьютера с помощью звуковых колонок и наушников, а также для записи звука.

Конфигурация персонального компьютера выбирается в зависимости от основного функционального назначения, и самое большое разнообразие возможных конфигураций имеют настольные персональные компьютеры. Их классифицируют на офисные, домашние, дизайнерские, игровые. Кроме того, компьютеры из последней группы дополнительно классифицируют по уровню производительности, выделяя начальный, средний, высший уровень.

Офисные персональные компьютеры предназначены главным образом для работы с офисными программами и приложениями, обычно подключаются к локальным сетям и не отличаются высокой производительностью или скоростью работы. Все, что от них требуется, это стабильность и бесперебойность работы.

Персональные компьютеры для дома используются в основном для решения задач, не требующих большой мощности и производительности. Это поиск информации, работа с текстовыми документами, общение в социальных сетях и на форумах, прослушивание музыки и т.п. Тем не менее, конфигурация домашних персональных компьютеров обычно включает качественный монитор, достаточно хороший процессор, видеокарту, DVD привод, акустику, предусматривается подключение к Интернету, при необходимости обеспечивается подключение к телевизору. Дополнением к конфигурации выступают сканер, принтер, ТВ-тюнер, веб-камера и др.

Игровые персональные компьютеры отличаются от остальных наличием мощной графической подсистемы. Главными элементами их конфигурации являются мощный процессор, мощная видеокарта, например Radeon или GeForce, и оперативная память достаточного объема. Комплектация подразумевает обязательное включение игровых аксессуаров (джойстика, руля, педалей и т.д.).

Дизайнерские персональные компьютеры, или графические станции, предназначены для работы с объемными и сложными графическими приложениями и качественной

обработкой фото- и видеофайлов. Конфигурация может быть изменена в зависимости от конкретно выполняемых с его помощью задач: работа с 3D-графикой требует мощной видеокарты, работа с видеофайлами – высокопроизводительного процессора и т.д.

В *схеме 5* представлены программы, позволяющие определять конфигурацию компьютера.

CPU-Z (Central Processing Unit-Z) – программа, определяющая техническую характеристику составляющих персонального компьютера: системного процессора, материнской платы и BIOS, оперативной памяти, видеокарты, кроме жесткого диска.

EVEREST Ultimate Edition – программа, предназначенная для диагностики, тестирования и настройки оптимальной работы аппаратных и программных средств компьютера.

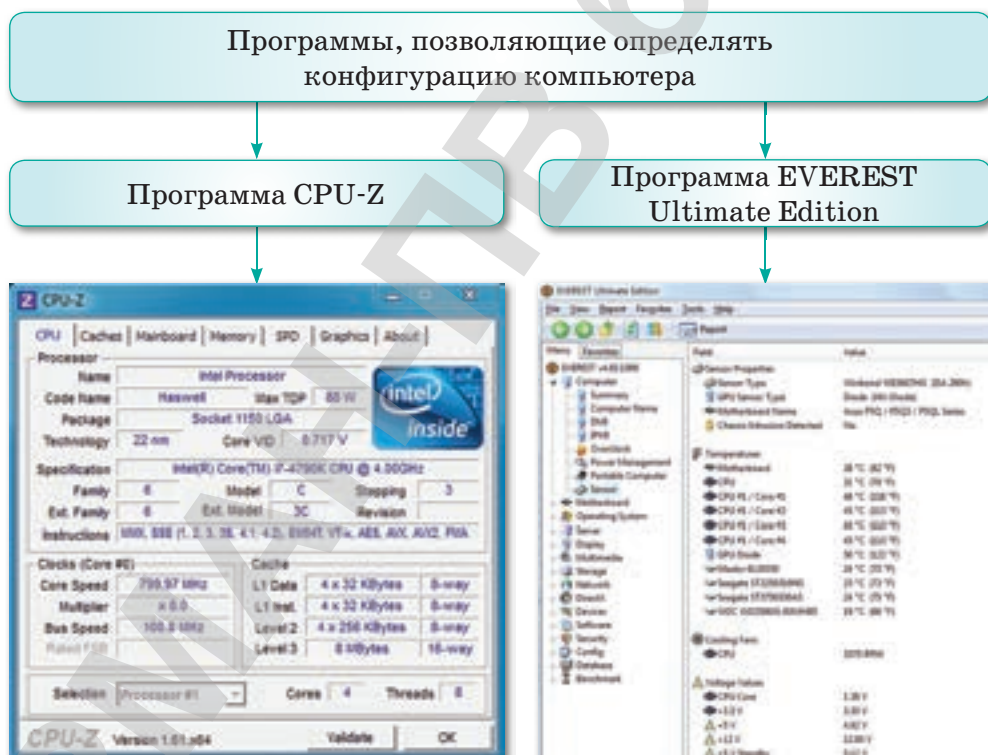


Схема 5. Программы, позволяющие определять конфигурацию компьютера

1

Отвечаем на вопросы

1. Что такое конфигурация?
2. Как выбрать конфигурацию компьютера?
3. Для чего используются персональные компьютеры?
4. Каков минимальный набор комплектующих персонального компьютера?
5. Каковы компоненты материнской платы персонального компьютера?
6. Каковы функции персональных компьютерных шин?
7. Какие параметры характеризуют производительность процессора?

2

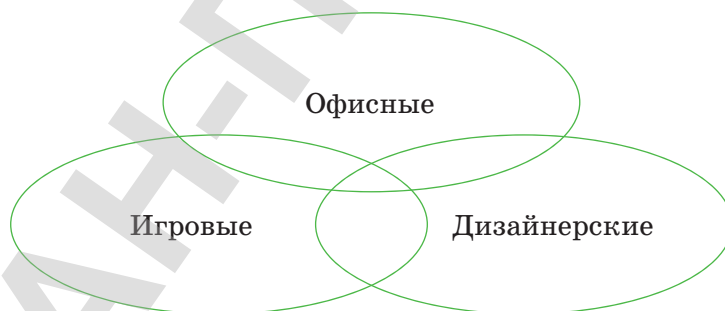
Думаем и обсуждаем

1. Какая конфигурация компьютера используется в повседневной жизни?
2. Каково функциональное назначение составляющих компьютера?
3. В чем отличие игровых компьютеров от дизайнерских?

3

Анализируем и сравниваем

1. Используя диаграмму Венна, сравните компьютеров по их классификации:



2. Сравните, чем отличается персональный компьютер от ноутбука. Проведите анализ.

4

Выполняем в тетради

Какие виды современных компьютеров вы знаете? Найдите в Интернете характеристики их конфигураций и запишите в тетради.

5

Выполняем на компьютере

Задание 1.

Подберите комплектующие для компьютера, предназначенного для решения определенного круга задач (игровой компьютер, офисный компьютер). Рассчитайте стоимость данного компьютера. Для подбора различных вариантов решения указанной задачи используйте табличный процессор (электронные таблицы). Все компоненты должны быть совместимы с материнской платой через интерфейс соединения и пропускной способности.

Задание 2.

а) Найдите в Интернете, газетах и других источниках информацию о **блоке питания** для вашего компьютера и заполните таблицу.

Прејскурант и номер модели	Характеристика	Цена

б) Найдите в Интернете, газетах и других источниках информацию о **процессоре** для вашего компьютера и заполните таблицу.

Прејскурант и номер модели	Характеристика	Цена

6

Делимся мыслями

Что вы узнали, чему научились на уроке? Обсудите с одноклассниками. В каких жизненных ситуациях можно применить знания, полученные на уроке?

7

Домашнее задание

1. Опишите характеристики конфигурации компьютера (ноутбука), который вы используете (найдите информацию в Интернете).
2. Выполните задания на диске, соответствующие данной теме.

§ 6. Практикум. Конфигурация компьютера

Компьютеры можно классифицировать в зависимости от функционального назначения, производительности и габаритов.

На сегодняшний день компьютерные фирмы «Срау», «NEC» «Fujitsu» являются лидерами в производстве мощных суперкомпьютеров. Для них требуются специальные помещения с определенным температурным режимом. Температурный режим суперкомпьютера требует охлаждения либо водой, либо жидким азотом.

Задание 1. Заполните *таблицу 2*.

Таблица 2. Устройства

Устройства	Основное устройство ввода	Вспомогательное устройство вывода	Основное устройство вывода	Вспомогательное устройство ввода
Монитор				
Мышь				
Принтер				
Клавиатура				
Плоттер				
Сканер				
Микрофон				

Задание 2. Включив компьютер, выясните функцию и характеристику устройств (*таблица 3*).

Таблица 3. Функции устройства

Функции устройства	Устройство внутреннего хранения	Устройство внешнего хранения
Флэш-память		
Оперативная память		

Функции устройства	Устройство внутреннего хранения	Устройство внешнего хранения
CD-диск		
DVD-диск		
Жесткий диск		
Постоянная память		

Задание 3. Выполните задание, используя прием «Кубизм».

Возьмите обычный бумажный куб, на гранях которого записаны следующие задания:

1. *Изучите* конфигурацию компьютера.
2. *Опишите* созданные разработки Артура Беркса, Германа Голдстайна, Джона фон Неймана.
3. *Сравните* двоичный принцип и принцип однородности памяти.
4. *Примените:* какой принцип предпочтительнее?
5. *Объясните:* приведите аргументы.
6. *Предложите:* какие дополнительные устройства в будущем вы бы добавили к персональному компьютеру?

§ 7. Выбор программного обеспечения

Вспомните!

- Что такое конфигурация?
- Как выбрать конфигурацию в зависимости от назначения компьютера?

Вы узнаете:

- что такое программное обеспечение;
- о видах программного обеспечения;
- о критериях качества программного обеспечения.

Программное обеспечение –
Программалық
жасақтама – *Software*

Как вы знаете, совокупность программ, предназначенных для решения задач на персональном компьютере, называется *программным обеспечением*.

Программное обеспечение (software) – программа или множество программ, используемых для управления компьютером.

Программное обеспечение подразделяется на 3 категории в зависимости от их функций: системное ПО, прикладное ПО, инструментальное ПО (системы программирования) (*схема 6*).



Схема 6. Программное обеспечение

Системное программное обеспечение – совокупность программ, обеспечивающих управление работой компьютерных систем и его технической функциональности.

Прикладное программное обеспечение – совокупность программ, решающих определенные задачи для конкретных групп пользователей компьютера.

Инструментальное программное обеспечение обеспечивает разработку новых программ для компьютера на языке программирования.

Разновидности инструментальных программ: Borland Delphi, Borland C ++ Builder, Microsoft Visual Basic, Microsoft Visual C ++.

Программы по их правовому статусу можно разделить на три большие группы: *лицензированные, условно-бесплатные и свободно распространяемые.*

1. **Лицензированные программы.** Пользователь получает программное обеспечение вместе с лицензией, которая предоставляет ему право использовать программный продукт при условии выполнения положений о лицензировании. Как правило, эти условия ограничивают возможности пользователя передавать программный продукт другим пользователям, изменять код.
2. **Условно-бесплатные программы.** Пользователь обычно получает бесплатно на определенный испытательный период демонстрационную версию программного продукта с несколько ограниченными возможностями, а после его окончания обязан либо приобрести продукт, либо деинсталлировать его.
3. **Свободно распространяемые программы.** Часть программного обеспечения поставляется со свободной лицензией. Такие лицензии позволяют распространять программное обеспечение, также модифицировать его.

Критерии качества современного программного обеспечения:

- функциональность;
- надежность;
- эргономичность;
- эффективность;
- мобильность.

Функциональность – способность программного обеспечения выполнять набор функций (действий), удовлетворяющих заданным или подразумеваемым потребностям пользователей. Набор указанных функций определяется во внешнем описании программного обеспечения.

Надежность – это способность с достаточно большой вероятностью безотказно выполнять определенные функции при заданных условиях и в течение заданного периода времени.

Эффективность – соотношение уровня услуг, предоставляемых пользователю программным обеспечением в заданных условиях, и объема используемых для этого ресурсов. К числу таких ресурсов могут относиться требуемые аппаратные средства, время выполнения программ, затраты на подготовку данных и интерпретацию результатов.

Эргономичность – характеристики программного обеспечения, которые позволяют минимизировать усилия пользователя по подготовке исходных данных, применению программного обеспечения и оценке полученных результатов, а также вызывать положительные эмоции определенного или подразумеваемого пользователя.

Мобильность – способность программного обеспечения быть перенесенным из одной среды в другую, в частности, с одной аппаратной платформы на другую.

Дополнительно к программным обеспечениям рассмотрим серверное программное обеспечение.

Серверное программное обеспечение – программный компонент вычислительной системы, выполняющий сервисные (обслуживающие) функции по запросу клиента, предоставляя ему доступ к определенным ресурсам или услугам.

Для запуска нескольких операционных систем на одном компьютере часто используются виртуальные машины.

Виртуальная машина – это программа, используемая для работы компонентов компьютера, в результате которой создается виртуальная среда, позволяющая установить и запустить любую операционную систему. Можно создать несколько виртуальных машин, в которых можно установить различные операционные системы, одновременно активировать их и создать виртуальную локальную сеть.

1

Отвечаем на вопросы

1. Какова функция программного обеспечения компьютера?
2. Какие виды программного обеспечения вы знаете?
3. Какие требования предъявляются к качеству ПО?
4. Каковы функциональные цели программного обеспечения?
5. Расскажите о видах программ по их правовому статусу.

2

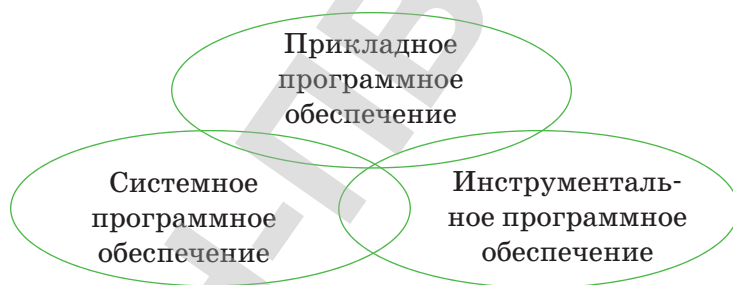
Думаем и обсуждаем

1. Почему виды программного обеспечения зависят от потребностей пользователя?
2. Какова продолжительность лицензии для операционной системы компьютера?

3

Анализируем и сравниваем

Сравните и проанализируйте виды программного обеспечения.



4

Выполняем в тетради

1. Составьте таблицу, описывающую критерии качества современного программного обеспечения, характеристики его деятельности, заполните таблицу.
2. Какие виды ПО компании бесплатно предоставляют пользователям для рекламы своей продукции?

5

Выполняем на компьютере

Постройте таблицу в MS Word. Какое программное обеспечение нужно людям в следующих ситуациях?

Ситуация	Системное ПО	Прикладное ПО	Инструментальное ПО
Профессиональный программист по заказу крупной компании создал компьютерную программу			
Выпускники 11 класса подготовили фотоальбом о школьной жизни			
Учитель подготовил тестовые задания по своему предмету на компьютере			
Производители смартфонов презентуют дизайн новых телефонов			

6

Делимся мыслями

1. Проанализируйте с одноклассниками критерии качества современного программного обеспечения.
2. Обсудите с классом классификацию программ по их правовому статусу.
3. Проанализируйте, к каким видам программного обеспечения относятся программы на данных иллюстрациях.



7

Домашнее задание

Приведите примеры лицензированных, бесплатных и свободно распространяемых программ.

44

§ 8. Расчет стоимости компьютера

Вспомните!

- Что такое компьютерное программное обеспечение? Какова его роль?
- Какие виды ПО вы знаете?
- Каковы функциональные цели ПО?
- Назовите критерии качества программного обеспечения.

Вы узнаете:

- что такое модели;
- об этапах моделирования;
- что такое информационная модель.

Физическая модель –

Физикалық модель –
Physical model

Биологическая модель –

Биологиялық модель –
Biological model

Экономическая модель –

Экономикалық модель –
Economic model

Человечество в своей деятельности всегда создавало и использовало модели окружающей среды.

Модели дают нам возможность наглядно представить сложные для понимания объекты и процессы. Наглядные модели часто используются в учебном процессе (глобусы, образцы молекул, кристаллические решетки и т.д.).

Модели играют очень важную роль в проектировании и построении технического оборудования, машин и механизмов. Невозможно представить себе развитие науки, изучение строения реальных объектов, их свойств и характера без теоретических моделей (теории, законы, гипотезы).

Модель – это определенный новый образец, показывающий важнейшие отличия объекта, явления или процесса. Модель обеспечивает исследователя информацией о свойствах моделируемых объектов.

Моделирование – процесс создания и использования моделей для решения практических задач.

Основные этапы моделирования:

1. Постановка задачи.
2. Создание модели.
3. Проверка реализации задачи.
4. Анализ результатов.

Рассмотрим на примерах следующие виды моделей:

- экономическая модель;
- физическая модель;
- биологическая модель.

Для моделирования выберем среду табличного процессора. В этой среде информационная и математическая модели объединяются в таблицу, которая содержит две области:

- исходные данные;
- расчетные данные (результаты).

1. Как наука, экономическая теория имеет свои особые методы исследования. Важнейшим методом является построение экономических моделей.

Экономическая модель состоит из двух или более переменных, которые:

- а) описывают взаимосвязь между переменными;
- б) отображают экономический результат этой взаимосвязи;
- в) предсказывают, как влияют изменения значений переменных на экономический результат.

Экономические модели используются для обобщения основных характеристик сложных явлений, чтобы упростить их и сделать доступными для анализа.

Из повседневной практики видно, что современному экономисту нужны всесторонние математические знания. Это подтверждается тем, что при планировании и исследовании экономико-математических моделей нужно установить связь между экономическими составляющими. Табличный процессор является очень удобным инструментом для экономистов, бухгалтеров, инженеров, ученых, которым приходится работать с большим количеством числовых данных.

Рассмотрим пример расчета стоимости компьютера как пример экономической модели.

На сегодняшний день в продаже имеется множество видов различных компьютеров (ноутбуков, нетбуков и т.д.). Поэтому при покупке компьютера нужно сначала определиться, для каких целей он будет использоваться. Очевидно, возникнут сложности при выборе нового компьютера. Важно правильно выбрать внутреннее устройство компьютера, а также монитор, мышь, клавиатуру и т.д. Каждое устройство выбирается отдельно, при этом нужно учитывать качество и цену товаров.

Для выбора компьютера необходимо знание следующих характеристик:

- | | |
|------------------------|--------------------------|
| 1) тип процессора; | 5) жесткий диск; |
| 2) оперативная память; | 6) операционная система; |
| 3) видеокарта; | 7) система охлаждения; |
| 4) материнская плата; | 8) монитор. |

Необходимо учитывать тип и тактовую частоту процессора. Например, если тактовая частота высокая, то компьютер будет работать производительней. Следует также уделить внимание числу ядер компьютера. Чем мощнее видеокарта, тем качество изображений будет лучше. При покупке компьютера важен выбор самого главного устройства – материнской платы.

К ней подсоединяются все компоненты компьютера.

Пример. Определить, какую сумму необходимо подготовить для приобретения нужного нам компьютера, предварительно имея перечень необходимых устройств и их стоимость. Рассчитаем стоимость компьютера (рис. 8).

Для решения построим *таблицу 4*, используя MS Excel:

Таблица 4. Расчетная стоимость компьютера

Расчет стоимости компьютера					
№	Наименование товара	Цена		Гарантия	Примечание
		в розницу	оптом		
1	Процессор	22185	21749	1	
2	Кулер	2170	1887	1	
3	Видеокарта	17330	16975	3	
4	Материнская плата	15550	15255	3	
5	ОЗУ	8655	8241	1	
6	Жесткий диск	14175	14045	1	
7	Звуковая плата	6825	6501		
8	Монитор	23990	23067		
9	Клавиатура	2926	2787	1	
10	Мышь	1800	1600	1	
11	Операционная система	10000	9000		
12	Система охлаждения	5046	4806	1	
Всего		130652	125913		

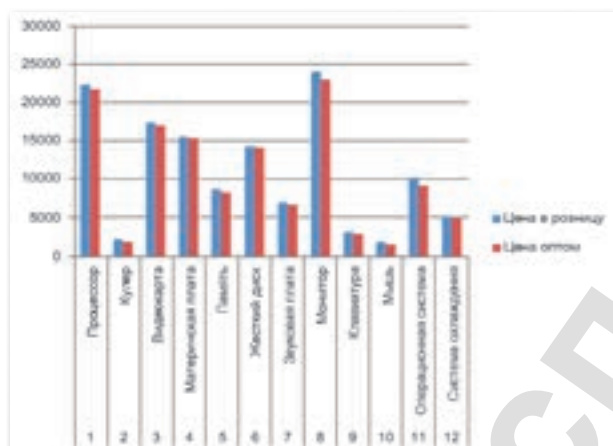


Рис. 8. Экономическая модель расчета стоимости компьютера

2. Физико-математическое моделирование также является очень важным методом исследований.

Например: построим информационную модель движения тела, брошенного под углом к горизонту.

Мячик мал по сравнению с Землей, поэтому его можно считать материальной точкой. Изменение высоты мячика мало, поэтому ускорение свободного падения можно считать постоянной величиной $g = 9,8 \text{ м/с}^2$ и движение по оси OY можно считать равноускоренным.

Скорость броска тела мала, поэтому сопротивлением воздуха можно пренебречь и движение по оси OX можно считать равномерным.

Используя электронную таблицу, сделайте следующие действия:

- 1) объедините ячейки с A1 по C1;
- 2) введите текст «Движение тела, брошенного под углом к горизонту»;
- 3) откройте ячейки B и C, так как текст находится в ячейках с A1 по C1;
- 4) в ячейки A2, A3 и A4 введите $V_0 =$, $A =$, $G =$ соответственно;
- 5) в ячейки C2, C3 и C4 введите м/с, град, м/с² соответственно;
- 6) ячейки B2, B3 и B4 отформатируйте, для ввода десятичных чисел;
- 7) в ячейки B2, B3 и B4 введите 18,0; 35,0; 9,8 соответственно;
- 8) в ячейки A5-введите t , в B5 $X=v_0 \cdot \cos(a) \cdot t$, в C5 $Y = v_0 \cdot \sin(a) \cdot t - (g \cdot t^2) / 2$;
- 9) ячейки в диапазоне с A6 по C19 отформатируйте для ввода десятичных чисел, с одним десятичным знаком после запятой;

- 10) в ячейку A6 введите номер 0,0;
- 11) выбрав ячейки с A6 по A19, заполните их временем с интервалом 0,2;
- 12) в ячейку B6 введите формулу = $\$B\$2*\text{COS}(\text{радианы}(\$B\$3))*A6$;
- 13) в ячейку C6 введите формулу = $\$B\$2*\text{SIN}(\text{радианы}(\$B\$3))*A6 - (\$B\$4*A6^2)/2$;
- 14) скопируйте формулы соответственно в ячейки B7:B19 и C7:C19;
- 15) выберите ячейки с A5 по C19 и добавьте границы для таблицы;
- 16) данная компьютерная модель позволяет решить поставленную задачу по исследованию движения тела, брошенного под углом к горизонту, подобрать оптимальные значения начальной скорости и угла броска;
- 17) сохраните работу под названием «Физическая модель» в своей папке (рис. 9).

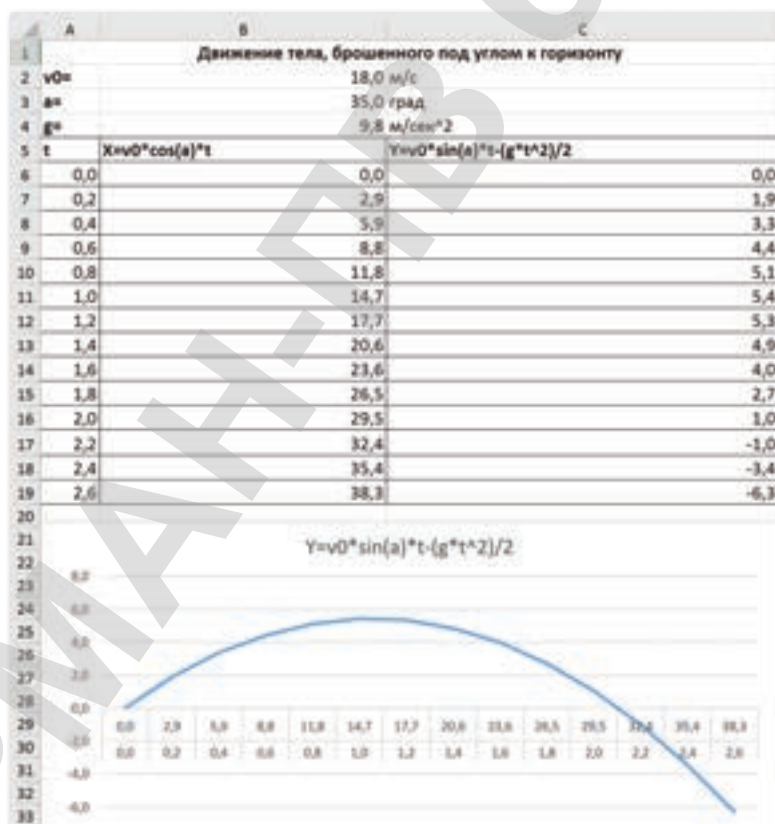


Рис. 9. Физическая модель

3. Существует гипотеза о том, что жизнь человека подчиняется трем циклическим процессам, называемым биоритмами.

Биоритмы – это периодически повторяющиеся изменения характера и интенсивности биологических процессов и явлений в живом организме. Эти циклы описывают три стороны самочувствия человека: физическую, эмоциональную и интеллектуальную. Биоритмы характеризуют подъемы и спады нашего состояния. Считается, что «взлетам» графика, представляющего собой синусоидальную зависимость, соответствуют более благоприятные дни. Дни, в которые график переходит через ось абсциссы, считаются неблагоприятными. Не все считают эту теорию строго научной, но многие верят в нее.

Теории «трех биоритмов» около ста лет. Интересно, что ее авторами стали три человека. Психолог Герман Свобода, отоларинголог Вильгельм Флисс, открывшие эмоциональный и физический биоритмы и получившие одинаковые результаты независимо друг от друга, а также педагог Фридрих Тельчер, исследовавший интеллектуальный ритм. Тельчер заметил, что желание и способность обучающихся воспринимать, систематизировать и использовать информацию, генерировать идеи время от времени изменяются, т.е. имеют ритмический характер.

Сопоставив даты рождений студентов, экзаменов, их результаты, он открыл интеллектуальный ритм с периодом в 33 дня.

Австрийский психолог Г. Свобода, анализируя поведение своих пациентов, начало и развитие болезней, заметил цикличность сердечных и астматических приступов. Результатом этих исследований стало открытие ритмичности физических (23 дня) и психических (28 дней) процессов.

Немецкого доктора Вильгельма Флисса заинтересовала сопротивляемость организма человека болезням. Почему дети с одинаковыми диагнозами в одно время имеют иммунитет, а в другое – тяжело болеют? Собрав данные о начале болезни, температуре и результатах лечения, он связал их с датой рождения. Расчеты показали, что изменения иммунитета можно прогнозировать с помощью 23-дневного физического и 28-дневного эмоционального биоритмов.

За точку отсчета всех трех биоритмов берется день рождения человека. Момент рождения для человека очень труден, ведь все три биоритма в этот день пересекают ось абсциссы. С точки зрения биологии, это достаточно правдоподобно, ведь ребенок, появляясь

на свет, меняет водную среду обитания на воздушную. Происходит глобальная перестройка всего организма. Физический биоритм характеризует жизненные силы человека, т.е. его физическое самочувствие. Его периодичность составляет 23 дня. Эмоциональный биоритм характеризует внутренний настрой человека, его способность эмоционального восприятия окружающего. Продолжительность периода эмоционального цикла равна 28 дням. Третий биоритм характеризует мыслительные способности, интеллектуальное состояние человека. Цикличность его – 33 дня.

Например, возможно осуществить моделирование биоритмов для конкретного человека от указанной текущей даты (дня отсчета) на месяц вперед с целью дальнейшего анализа модели.

Математическая модель. Представим математическую модель следующим образом:

- физический биоритм $f(x)=\sin(2*\pi*x/23)$;
 - эмоциональный биоритм $f(x)=\sin(2*\pi*x/28)$;
 - интеллектуальный биоритм $f(x)=\sin(2*\pi*x/33)$,
- где x – возраст человека в днях.

Заполните таблицу по образцу (рис. 10).

	А	В
1		
2	Биоритм	
3	Исходные данные	
4	Дата рождения	06.03.2004
5	Дата отсчета	01.04.2019

Рис. 10. Информационная модель

Составьте компьютерную модель по образцу. Заполните данные в ячейках с формулами (рис. 11):

Ячейка	Формула	
A9	=B\$5	(1)
A10	=A9+1	(2)
B9	=SIN(2*ПИ()*(A9-B\$4)/23)	(3)
C9	=SIN(2*ПИ()*(A9-B\$4)/28)	(4)
D9	=SIN(2*ПИ()*(A9-B\$4)/33)	(5)

Рис. 11. Исходные данные

Компьютерная модель биоритма (рис. 12):

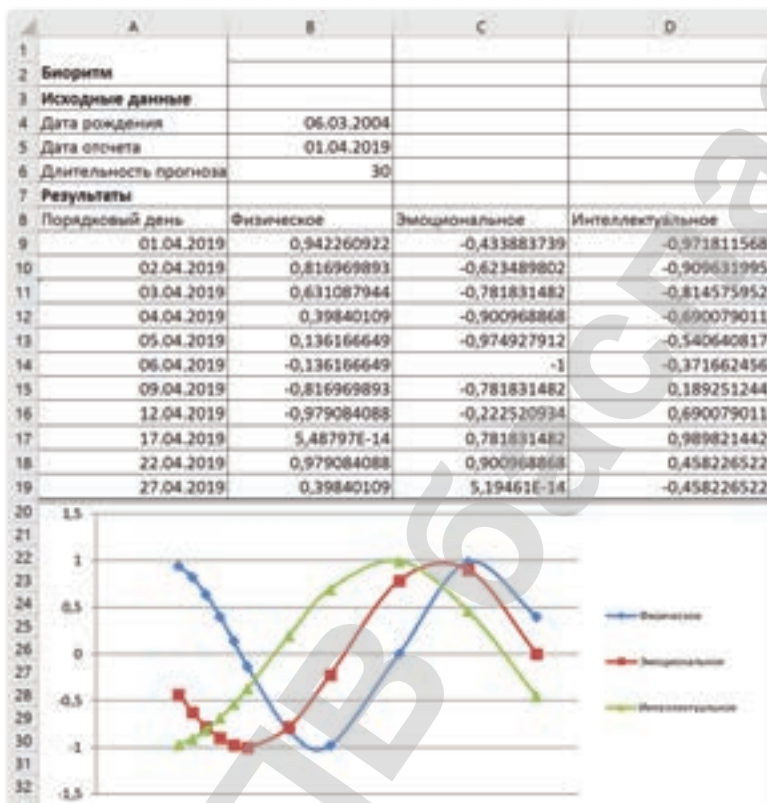


Рис. 12. Биологическая модель

Сохраните работу с своей папке под названием «Биологическая модель».

1

Отвечаем на вопросы

1. Сможете ли вы создать физическую, биологическую и экономическую модель?
2. Можно ли вычислить любые физические расчеты в электронной таблице?
3. Что вы знаете о диаграммах биоритмов?
4. Какие виды экономических расчетов вы знаете?

2

Думаем и обсуждаем

1. Зачем нужны модели?
2. Какова причина появления биологических моделей?

3. В чем разница физической модели от биологической модели?
4. Какова цель экономического моделирования?

3

Анализируем и сравниваем

Расскажите о методах построения моделей. Проанализируйте метод построения экономической модели.

4

Выполняем в тетради

1. Подготовьте расчеты по физическому моделированию.
2. Напишите об эффективном использовании физического, биологического, экономического моделирования.

5

Выполняем на компьютере

1. Если два человека имеют одинаковые или очень похожие таблицы для одного, двух или даже трех биоритмов, то мы можем видеть высокую совместимость этих людей. Составьте модель физической, эмоциональной и интеллектуальной совместимости двух друзей.
2. Предположим, необходимо распределить премию 15 сотрудникам. Каждому сотруднику начисляется определенный процент от начисленного. Требуется определить, какой процент премии необходимо установить, чтобы уложиться в премиальный фонд 150 000 тг?

6

Делимся мыслями

Обсудите с классом, могут ли объекты иметь несколько моделей? Приведите примеры моделирования в различных областях жизни человека.

7

Домашнее задание

1. Подготовьте презентацию на тему «Значимость создания физической модели процессов в электронной таблице».
2. Выполните задания на диске, соответствующие данной теме.

ИТОГОВЫЕ ТЕСТОВЫЕ ЗАДАНИЯ К РАЗДЕЛУ II

1. **Что не входит в состав системного блока?**
 - А) Жесткий диск.
 - В) Клавиатура.
 - С) Процессор.
 - Д) Кэш-память.
 - Е) Оперативная память.

2. **Винчестер – это устройство ...**
 - А) вывода информации.
 - В) обработки информации.
 - С) хранения информации.
 - Д) ввода информации.
 - Е) прослушивания информации.

3. **Устройство, не выводящее информацию –**
 - А) принтер.
 - В) динамики.
 - С) дисплей.
 - Д) плоттер.
 - Е) сканер.

4. **Основные устройства компьютера для обработки информации –**
 - А) процессор, память, принтер, HDD, монитор.
 - В) чипсет и оперативная память.
 - С) микропроцессор и контроллер устройств.
 - Д) центральный процессор и оперативная память.
 - Е) центральный процессор и чипсет.

5. **Основными характеристиками процессора являются –**
 - А) модель, тактовая частота.
 - В) модель, тактовая частота, разрядность адресной шины.
 - С) модель, тактовая частота, разрядность.
 - Д) тактовая частота, разрядность, объем оперативной памяти.
 - Е) модель, тактовая частота, скорость обращения к внешним устройствам.

6. Как называется оперативная память компьютера?
- A) CD-ROM.
 - B) RAM (Random Access Memory).
 - C) ROM (Read Only Memory).
 - D) Hardware.
 - E) Software.
7. Какое устройство компьютера выполняет функцию, подобную мозгу человека?
- A) Принтер.
 - B) Дисковод.
 - C) Процессор.
 - D) Клавиатура.
 - E) Модем.
8. Кэш-память расположена ...
- A) в оперативной памяти.
 - B) на дисках.
 - C) в запоминающем устройстве.
 - D) в процессоре.
 - E) в сканере.
9. При отключении компьютера вся информация ... удаляется.
- A) на жестком диске.
 - B) на гибком диске.
 - C) в постоянной памяти.
 - D) на CD-ROM диске.
 - E) в оперативной памяти.
10. Оперативная память необходима для ...
- A) хранения неизменной информации.
 - B) долгосрочного хранения информации.
 - C) краткосрочного хранения информации.
 - D) удаления информации.
 - E) долгосрочного хранения неизменной информации.
11. Основные составляющие программного обеспечения –
- A) языки программирования и компьютерные вирусы.
 - B) инструментальные средства и системное ПО.

- С) компилятор и интерпретатор.
- Д) драйвер и операционная система.
- Е) прикладное, системное и инструментальное ПО.

12. Производительность работы компьютера зависит от

- А) комплектующих системного блока.
- В) установленного ПО.
- С) скорости интернет-соединения.
- Д) стоимости компьютера.
- Е) срока производства.

13. За минимальную единицу измерения количества информации принято считать

- А) байт.
- В) килобит.
- С) бит.
- Д) ноль.
- Е) единицу.

14. ... – программа, предназначенная для диагностики, тестирования и настройки оптимальной работы аппаратных и программных средств компьютера.

- А) CPU-Z.
- В) Виртуальная машина.
- С) EVEREST Ultimate Edition.
- Д) Серверное программное обеспечение.
- Е) CPU-R.

15. ... – программа, определяющая техническую характеристику составляющих персонального компьютера.

- А) CPU-Z.
- В) Виртуальная машина.
- С) EVEREST Ultimate Edition.
- Д) Серверное программное обеспечение.
- Е) CPU-R.

РАЗДЕЛ III

БАЗЫ ДАННЫХ

Изучив главу, вы научитесь:

- объяснять термины «база данных, запись, поле»;
- создавать базы данных в электронной таблице;
- осуществлять поиск, сортировку и фильтрацию данных.

§ 9. База данных: основные понятия и определения

Вспомните!

- Что относится к информационным объектам?
- Информация какого рода хранится в табличных процессорах?
- Виды табличных процессоров.

База данных – Деректер базасы – *Databae*

Вы узнаете:

- что такое база данных; понятие СУБД;
- для чего предназначены базы данных;
- об основных функциях баз данных.

В повседневной жизни люди постоянно находятся в большом потоке информации самого разного рода и характера. Использование систем управления базами данных для оперативности и эффективности поиска информации из этого потока становится особенно актуальным.

Система управления базами данных (СУБД) – это программное обеспечение для работы с базами данных, комплекс программных и языковых средств, используемых для создания, поддержки и функционирования баз данных.

Для свободной работы с СУБД необходимо дать четкое представление о некоторых понятиях, связанных с базами данных.

СУБД имеет много возможностей (*схема 7*).



Схема 7. Возможности СУБД

База данных (БД) – это набор данных и информации, организованный для быстрого поиска и получения нужной информации с помощью компьютера. Например, база данных «Школа» хранит всю информацию об учителях, учащихся, родителях, каждый из которых имеет фамилию, имя, телефон и т.д. База данных «Библиотека» содержит всю информацию о книгах, каждая из которых имеет свое название, автора, год издания и т.д.

В зависимости от структуры в основном известны реляционные, иерархические и сетевые БД.

Реляционные БД представляют собой таблицу или несколько таблиц, связанных в определенном отношении, где строки таблицы являются *записями*, а столбцы *полями* БД. Таблицы в реляционных базах данных – это набор упорядоченных и связанных данных.

Иерархические БД имеют древовидную структуру и построены с учетом зависимости последующих элементов от предыдущего, где каждый из элементов (узлов) верхнего уровня может быть связан с одним или несколькими узлами нижнего уровня.

Сетевые БД более сложные по своей структуре и представляют собой графическую модель баз данных со свободными связями между узлами. К основным понятиям иерархической и сетевой моделей БД относятся: *элемент (узел), уровень, связь*.

При упоминании баз данных (БД) сразу на ум приходят следующие программы: Access, Oracle, SQL, DataBase и т.д. Безусловно, это очень мощные и недешевые программы, способные автоматизировать обработку и хранение большого объема данных в БД. Но дело в том, что иногда такая мощь просто не нужна. И многие пользователи по всему миру предпочитают создавать простые базы данных в Excel.

Любой пользователь, владеющий инструментарием программы **Microsoft Excel**, может смело начинать работу по вводу, поиску и обработке данных в табличном процессоре.

Это интересно

В Исландии создана всеобъемлющая генеалогическая база данных **Íslendingabók**, содержащая информацию о родственных связях всех исландцев, начиная с XVIII века. Каждый гражданин Исландии имеет доступ к этой базе данных и может проверить степень своего родства с известными личностями.

1

Отвечаем на вопросы

1. Используем ли мы базы данных в повседневной жизни?
2. Для чего предназначены системы управления базами данных?
3. Перечислите основные модели и функции баз данных.
4. Расскажите о возможностях системы управления базами данных.
5. Дайте краткое описание структуры базы данных.

2

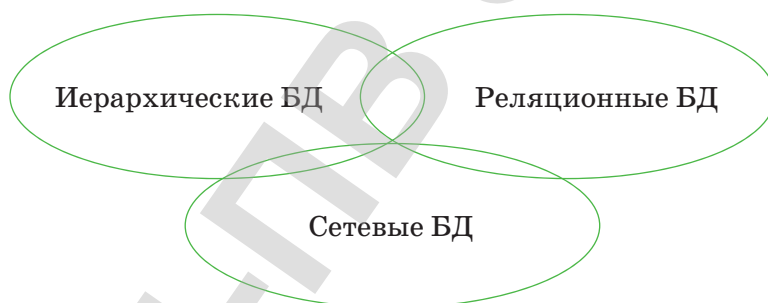
Думаем и обсуждаем

1. Для чего нужно создавать БД?
2. Почему большинство пользователей предпочитают создавать базы данных в Excel?

3

Анализируем и сравниваем

Используя диаграмму Венна, сравните реляционные, иерархические и сетевые базы данных.



4

Выполняем в тетради

1. Приведите различные примеры из жизни разных видов баз данных (реляционные, сетевые и т.д.).
2. Составьте кроссворд или головоломку с объектами БД: таблица, форма, запросы, отчеты и т.д.

5

Выполняем на компьютере

Подготовьте сведения (данные) в определенной интересующей вас предметной области для создания БД. В программе MS Excel составьте структуру вашей будущей

БД (из каких столбцов и строк будет состоять таблица, что будет основным, какие данные будут использованы чаще и т.д.).

6

Делимся мыслями

Обсудите со своими одноклассниками, где чаще всего применяются базы данных. Насколько они упрощают повседневную жизнь?

7

Домашнее задание

1. Найдите в Интернете БД классификации различных профессий.
2. Выполните задания на диске, соответствующие данной теме.

Творческое задание

Используя материалы из Интернета, подготовьте демонстрационную презентацию на тему «Преимущества различных видов БД (реляционные, сетевые и т.д.) и области их применения».

§ 10. Таблица. Понятие поле и записи

Вспомните!

- Что такое база данных?
- С какой целью создаются базы данных?
- Основные функции баз данных.

Вы узнаете:

- о терминах «поле», «запись»;
- какие типы данных различают в БД;
- об операциях с данными;
- о новых возможностях программы MS Excel.

Поле – Әріс – Field

Запись – Жазба – Records

Реляционные базы данных

Табличную базу данных можно представить в виде двумерной таблицы: в каждой ее строке последовательно размещаются записи, а в каждом столбце – свойства (характеристика) объекта или явления в соответствии с заданным именем поля.

Основными составляющими компонентами базы данных являются таблица, поле, запись.

Поле БД – это столбец таблицы, содержащий определенное свойство объекта.

Запись БД – это строка таблицы, содержащая набор значений свойств, размещенных в полях базы данных.

Ключевое поле – это поле, значение которого однозначно определяет каждую запись в таблице.

Каждое поле таблицы содержит одну характеристику объекта и заданный тип данных (текстовый, числовой или дата/время).

Типы данных БД:

- **текстовый** – обычная строка текста;
- **числовой** – число любого типа (вещественное, целое и т. п.);
- **дата/время** – дата и/или время.

Приведем пример отношения между записью и полем в таблице БД «Школа»:

Таблица 5. БД «Школа»

Запись	Ключевое поле	Поле 1	Поле 2	Поле 3	Поле 4
	<i>Порядковый номер</i>	<i>Фамилия, Имя, Отчество</i>	<i>Дата рождения</i>	<i>Контакты</i>	<i>Родители</i>
Запись 1	1	Абылай Венера Бериковна	22.03.2004	8****	Абзалов Б.А.
Запись 2	2	Жапаров Кайрат Талгатович	02.12.2004	8****	Кокенова А.А.
Запись 3	3	Раева Жанна Жакеевна	13.04.2003	8****	Мелибеков К.М.

В таблице 5 приведен пример отношения между записями и полями в таблицах БД «Школа», где **жирным шрифтом** отмечен **Ученик**, *курсивом* – относящиеся к нему характеристики: **Дата рождения, Контакты, Родители**. В отношении Ученик (Порядковый номер, Фамилия, Имя, Отчество, Дата рождения, Контакты, Родители) ключевым полем является **Порядковый номер**.

В каждой строке таблицы содержатся данные об одном объекте (например, об учащихся, компьютере, товарах), а в столбцах таблицы описываются характеристики этих объектов (например, анкетные данные, серия и марка процессора, контакты поставщиков или клиентов).

1

Отвечаем на вопросы

1. В каких областях используется реляционная БД?
2. Объясните новые термины: поле, запись, ключевое поле.
3. Перечислите типы данных в MS Excel.
4. Что такое тип данных в БД?

2

Думаем и обсуждаем

В приведенном примере покажите, что является полем, а что записью? Какие типы данных принимают значения в полях?

№	Планета	Средняя температура, °С	Площадь, тыс. км ²
1	Земля	14,8	510 072
2	Марс	-63,1	144 370
3	Меркурий	67	74 800
4	Венера	463	460 000

3

Анализируем и сравниваем

Составьте кроссворд или ребус с новыми терминами БД: поле, запись, ключ, типы данных и т.д. Проанализируйте и сравните эти термины.

4

Выполняем в тетради

Продолжите ассоциативный ряд примеров в скобках.
В БД чаще всего используются следующие типы данных:

- *текстовый (фамилия, название, ...);*
- *числовой (температура, стоимость, ...);*
- *дата/время (день/месяц/год, ...).*

5

Выполняем на компьютере

В MS Excel создайте БД «Мои друзья» на примере следующей таблицы.

№	Фамилия	Имя	Отчество	Дата рождения	Город
1	Абылай	Венера	Бериқовна	22.03.2004	Нур-Султан
2	Жапаров	Қайрат	Абаевич	02.12.2004	Алматы
3	Раева	Жанна	Жақеевна	13.04.2003	Алматы

6

Делимся мыслями

Обсудите с одноклассниками, с какими новыми понятиями познакомились на уроке? Какие типы данных существуют в MS Excel?

7

Домашнее задание

Подготовьте сообщение на тему «Типы данных в различных версиях MS Excel».

Творческое задание

Подготовьте презентацию на тему «Возможности и достоинства MS Excel в создании БД».

§ 11. База данных. Создание базы данных в электронных таблицах

Вспомните!

- Что такое реляционные БД?
- Понятия поле, запись, ключ.
- Типы данных MS Excel.

Вы узнаете:

- как создавать БД в электронных таблицах;
- о принципах работы в MS Excel;
- о понятиях диапазон, список.

Электронная таблица –
Электронды кесте –
Spreadsheet

На этом уроке вы приобретете практические навыки по созданию и работе с БД в программе MS Excel. Запустим программу по команде: **Пуск** ⇒ **Программы** ⇒ **Microsoft Excel**.

После выполнения команды **Файл** ⇒ **Создать** в программе MS Excel мы сохраняем файл под именем «Дневник» на внешнем носителе либо в папке «Моя база данных».

Порядок работы по созданию БД.

1. Задайте структуру двух будущих таблиц следующим образом:
 - 1) **Фамилия, Имя, Отчество, Адрес, Контакты, Увлечения** (по ученикам). Полученную таблицу назовем **Учащиеся** (таблица 6).

Таблица 6. БД «Учащиеся»

№	Фамилия	Имя	Отчество	Адрес	Контакты	Увлечения
1	Касымов	Ерлан	Болатович	Абая 50	8****	Футбол
2	Абылай	Венера	Бериковна	Байзакова 25	8****	Плавание
3	Жапаров	Кайрат	Абаевич	Ауэзова 15	8****	Программирование
4	Раева	Жанна	Жакеевна	Гоголя 5	8****	Фотографировать
5	Дархан	Айя	Сериковна	Муканова 38	8****	Танцевать
6						

- 2) Фамилия, Имя, Отчество, Название предмета (по учителям). Полученную таблицу назовем **Учителя** (таблица 7).

Таблица 7. БД «Учителя»

№	Фамилия	Имя	Отчество	Адрес	Контакты	Название предмета
1	Абенова	Лаура	Азаматовна	Толе би 70	8****	Информатика
2	Вагнер	Вера	Петровна	Гагарина 4	8****	Русский язык
3						

- 3) Заполните поля таблицы записями данных ваших одноклассников и данных ваших учителей. Например, на отдельных листах Лист 1, Лист 2.

2. Присвойте диапазону базы данных имя.

Для этого выделяем область базы данных вместе с заголовками столбцов A2...G20; в главном меню выбираем команду **Формула** ⇒ **Присвоить имя ...** (либо через контекстное меню с помощью правой кнопки мыши). В диалоговом окне **Создание имени** пишем имя, например, БД1 и нажимаем на кнопку **ОК**. Диапазону присвоено имя (рис. 13).

Аналогично для второй таблицы.

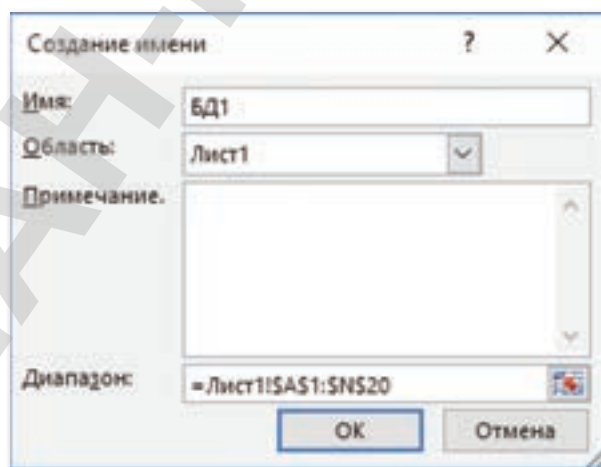


Рис. 13. Присвоение имени

3. Объявите диапазон базы данных списком.

Для этого вновь выделяем область базы данных вместе с заголовками столбцов; в главном меню выбираем команду **Данные** ⇒ **Проверка данных**.

Во всплывающем окне **Проверка вводимых значений** на вкладке **Параметры** в **Типе данных** выбрать **Список**. В поле **Источник** ввести диапазон названий. Нажимаем на кнопку **ОК** список создан (рис. 14).

4. Создание базы данных в MS Excel завершено.

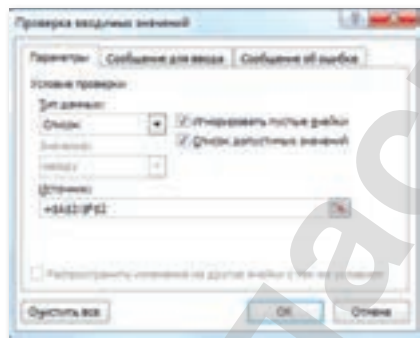


Рис. 14. Создание списка

Важная информация

Каждый столбец содержит только один тип данных – или текст, или числа, или даты.

1

Отвечаем на вопросы

1. Опишите принцип работы по созданию БД.
2. Что такое диапазон? Что такое список?
3. Возможно ли отформатировать отдельные ячейки?
4. Могут ли в одном столбце храниться данные разных типов?

2

Думаем и обсуждаем

Выберите ключевое поле и дополните таблицу.

№	Автор	Название произведения	Цена	Дата издания	Издательство
1	М.О. Ауэзов	Путь Абая	3500	2008	Жазушы

- Порядковый номер
- Автор
- Название произведения
- Цена
- Дата издания
- Издательство

3

Анализируем и сравниваем

Поэтапно опишите действия по созданию БД в MS Excel.

4

Выполняем в тетради

Определите типы данных в полях таблицы и запишите их в тетрадь.

№	Автор	Название произведения	Цена	Дата издания	Издательство
1	М.О. Ауэзов	Путь Абая	3500	2008	Жазушы
2	Л.Н. Толстой	Война и мир	1800	2004	Азбука
3	Э.М. Ремарк	Три товарища	2300	2017	АСТ
4	Ф.С. Фицджеральд	Великий Гэтсби	1950	2001	Эксмо

5

Выполняем на компьютере

Для закрепления изученного на уроке материала дополните БД «Дневник» следующей информацией:

№	Фамилия	Имя	Отчество	Дата рождения	Город
1	Касымов	Ерлан	Болатович	13.11.2004	Нур-Султан
2	Абылай	Венера	Бериковна	22.03.2004	Нур-Султан
3	Жапаров	Кайрат	Абаевич	02.12.2004	Алматы
4	Раева	Жанна	Жакеевна	24.01.2003	Алматы
5	Дархан	Айя	Сериковна	18.03.2004	Нур-Султан

6

Делимся мыслями

Что нового вы узнали на уроке? Обсудите со своими одноклассниками, насколько популярно использование программы MS Excel в создании БД среди пользователей.

7

Домашнее задание

В программе MS Excel создайте шаблон базы данных на примере прочитанных вами книг.

§ 12. Методы поиска информации

Вспомните!

- Способы создания БД.
- Что такое диапазон, список?
- Каковы требования к созданию БД?

Вы узнаете:

- о различных видах представления информации;
- о методах поиска информации в БД;
- как применять изученный материал на практике.

Форма – Форма – *Form*

Поиск – Іздеу – *Search*

Представление информации в БД

При работе со значительными объемами информации вне зависимости от того, какое программное обеспечение используется, поступают следующим образом:

1. Создают таблицы базы данных.
2. Создают формы для ввода данных в таблицы.
3. Создают необходимые запросы к таблицам базы данных.
4. Формируют отчеты на основании запросов для вывода на печать.

Формирование базы данных с использованием СУБД начинается с задания структуры БД: определения полей базы данных, их типов и ввода имен. Далее методом прямого ввода в таблицы или через формы осуществляется ввод, просмотр и редактирование записей базы данных. После этого можно приступить к сортировке и поиску данных в созданной базе данных.

Представление записей базы данных с помощью формы

Форма – объект, предназначенный для поочередного ввода, просмотра и редактирования записей в БД. Формы отображают одну запись из таблицы и имеют кнопки перехода от одной записи к другой.

Задача формы – предоставить пользователю простой, понятный и удобный интерфейс в виде раскрывающегося окна, который обеспечит в режиме диалога возможность эффективно осуществлять ввод данных пользователем в программу и вывод программой результатов для считывания пользователем.

В MS Excel ввод и просмотр записей можно осуществлять как в режиме таблицы, так и в режиме формы. В процессе создания формы можно задать, какие поля БД включить в форму, а также само расположение полей в окне формы.

Для вызова формы активируем (щелкаем мышью) любую ячейку внутри таблицы базы данных и нажимаем на иконку **Форма** главного меню программы. В появившемся окне, представляющем собой форму с именем БД1 (по имени листа книги Excel), нажимаем кнопку **Добавить**. Заполняем окна данными новой записи. Переходим от окна к окну при помощи клавиатуры, нажимая клавишу **Tab**.

Появится форма, содержащая записи данных, как на *рис. 15*.

№	Фамилия	Имя	Отчество	Адрес	Контакты	Увлечение
1	Касымов	Ерлан	Болатович	Абая 50	8****	Футбол
2	Абылай	Венера	Бериковна	Байзанова 25	8****	Плавание
3	Жапаров	Кайрат	Абаевич	Ауезова 15	8****	Программирование
4	Раева	Жанна	Жакеевич	Гоголя 5	8****	Фото
5	Дархан	Айя	Сериковна	Муканова 38	8****	Танцы

БД1 ? X

№:

Фамилия:

Имя:

Отчество:

Адрес:

Контакты:

Увлечение:

1 из 2

Рис. 15. Отображение записей БД в виде формы

Главным преимуществом использования формы в БД является возможность видеть одновременно все поля одной записи на экране, расположенные построчно.

Поиск информации с помощью фильтров

Одно из основных предназначений БД – это поиск необходимой информации. Поиск информации осуществляется с помощью фильтров. В таблицах БД можно осуществлять поиск данных (записей), удовлетворяющих определенным условиям, который называется **фильтром**.

Фильтры – это отбор записей, удовлетворяющих заданному условию или совокупности условий. Условия отбора записей создаются с помощью операторов сравнения: для числовых данных (=, >, <, >=, <=, <>) и для текстовых данных (содержит, не содержит, начинается с ..., не начинается с ..., заканчивается на ..., не заканчивается на ...).

Фильтры подразделяются на следующие виды:

- 1) простые (условие отбора записей только для одного поля);
- 2) сложные (несколько условий для различных полей).

В результате применения сложного фильтра будут отобраны только те записи, которые удовлетворяют одновременно всем условиям.

Фильтры в MS Excel предоставляют поиск данных по записям в пределах одной таблицы. Они не сохраняются как отдельный объект, и в них не создаются вычисляемые поля.

Поиск информации с помощью функций

В MS Excel необходимые данные можно быстро и эффективно находить в списке с помощью функций поиска. Для решения задач поиска и отображения результатов используются функции ВПР или сочетание функций ИНДЕКС и ПОИСКПОЗ.

Когда вам требуется найти данные по строкам в таблице или диапазоне, используйте функцию ВПР – одну из функций поиска. Например, можно найти цену автомобильной детали по ее номеру и т.п.

Самая простая функция **ВПР** означает следующее:

=ВПР (искомое значение; диапазон для поиска значения; номер столбца в диапазоне с возвращаемым значением; точное или приблизительное совпадение – указывается как 0/ЛОЖЬ или 1/ИСТИНА).

Функция **ПОИСКПОЗ** выполняет поиск указанного элемента в диапазоне ячеек и возвращает относительную позицию этого элемента в диапазоне. Например, если диапазон A1:A3 содержит значения 5, 25 и 38, то формула **=ПОИСКПОЗ(25;A1:A3;0)** возвращает значение 2, поскольку элемент 25 является вторым в диапазоне.

1

Отвечаем на вопросы

1. Для чего используются формы в MS Excel?
2. Что такое фильтр?
3. Как применяются текстовые и числовые фильтры?
4. Каким образом фильтры помогают при работе с БД?
5. В каких случаях необходимо использование функций поиска?

2

Думаем и обсуждаем

Дайте краткое описание программного инструментария приложения MS Excel, связанного с работой по поиску и фильтрации данных. Опишите последовательность действий для применения фильтров в имеющейся БД «Дневник».

3

Анализируем и сравниваем

Ученики делятся на группы. Каждой группе раздаются карточки с новыми терминами: формы, фильтры, диапазон, списки и т. д. Учащимся отводится время на подготовку и обмен информацией между собой в группах. Условие задания для групп: составить предложения с новыми терминами так, чтобы получились ложные или истинные утверждения.

4

Выполняем в тетради

Подготовьте ряд примеров применения функций поиска **ВПР**, **ИНДЕКС**, **ПОИСКПОЗ** на основе различных БД из Интернета.

5

Выполняем на компьютере

Дана таблица некоторой базы данных «Численное представление казахов в мире».

№	Страны	Численность	Год переписи
1	Республика Казахстан	12 000 000	2017
2	Китайская Народная Республика	1 462 588	2010
3	Российская Федерация	647 732	2010
4	Республика Узбекистан	803 000	2017
5	Киргизская Республика	34 122	2015

Применив поиск к данной таблице, определите количество записей в этой таблице, удовлетворяющих условию «Численность больше 1 млн чел.».

6

Делимся мыслями

Чему вы научились на сегодняшнем уроке? Обсудите со своими одноклассниками, насколько упрощают работу в БД изученные методы поиска информации. Приведите примеры их применения на практике.

7

Домашнее задание

1. Подготовьте имеющуюся базу данных «Дневник». В таблице «Учащиеся», используя функции поиска, отберите в поле «Адрес» всех учащихся, проживающих в городе Алматы. Полученную таблицу сохраните на электронном носителе либо в «облаке».
2. Выполните задания на диске, соответствующие данной теме.

Творческое задание

В собственной модели базы данных в MS Excel примените изученные методы поиска информации.

§ 13. Сортировка и фильтрация данных

Вспомните!

- Информационные системы данных.
- Представление информации в БД.
- Способы поиска информации в БД.

Вы узнаете:

- о способах сортировки и фильтрации;
- как применять изученный материал на практике.

Сортировка – Сұрыптау –
Sorting

Фильтр – Сүзгі – *Filter*

Сортировка и фильтрация данных

Сортировка данных. MS Excel позволяет осуществлять сортировку данных, т.е. располагать данные в определенном порядке (по возрастанию или убыванию). Например, при выполнении сортировки по возрастанию: 1) числа располагаются от наименьшего к наибольшему; 2) текст – в алфавитном порядке; 3) дата и время – в хронологическом порядке. Соответственно, при выполнении сортировки по убыванию – в обратном порядке.

Фильтрация данных. Если требуется отобразить записи, соответствующие определенным условиям и операторам сравнения, то используется фильтрация. Например, чтобы сократить число отображаемых записей, нужно выбрать в контекстном меню поле, значение которого будет выбрано в качестве условия отбора, и установить в нижней части контекстного меню параметр **Равно**, **Не равно**, **Содержит** или **Не содержит**.

В отличие от поиска, фильтрация отображает только записи, ограниченные согласно заданному условию. Включая или отключая фильтры, можно быстро переходить от отфильтрованного к неотфильтрованному виду одних и тех же данных БД.

На вкладке **Главная** в опциях параметров группы **Сортировка и фильтр** нажмите кнопку **Фильтровать** и выберите нужный фильтр. Таким же образом выполняются сортировка **По возрастанию** или **По убыванию**.

Практические примеры работы с информацией

Ниже приведены примеры сортировки и фильтрации данных БД.

1. Таблица 8 в базе данных представлена следующим образом:

Таблица 8. БД «Зимние Олимпийские игры-2018»

Место	Страна	Золотых	Серебряных	Бронзовых	Общее
1	Норвегия	14	14	11	39
2	Германия	14	10	7	31
3	Канада	11	8	10	29
4	США	9	8	6	23
5	Нидерланды	8	6	6	20
6	Швеция	7	6	1	14
7	Южная Корея	5	8	4	17
8	Швейцария	5	6	4	15
9	Франция	5	4	6	15
10	Австрия	5	3	6	14

Для закрепления теоретического материала на практике выполните следующие задания:

- 1) отсортировать записи **По возрастанию** в поле *Золотых*;
 - 2) выбрать записи, удовлетворяющие условию «*Серебряных* ≤ 5 »;
 - 3) выбрать записи, удовлетворяющие условию «*Бронзовых* > 10 ».
2. Таблица 9 в базе данных представлена следующим образом:

Таблица 9. БД «Учащиеся»

№	Фамилия	Имя	Отчество	Дата рождения	Город
1	Абылай	Венера	Бериковна	22.03.2004	Нур-Султан
2	Жапаров	Кайрат	Талгатович	02.12.2004	Алматы
3	Раева	Жанна	Жакеевна	13.04.2003	Алматы
4	Дархан	Айя	Сериковна	23.03.2004	Нур-Султан

- 1) найдите в *таблице 9* записи, которые содержат город **Нур-Султан**;
- 2) найдите записи, в которых фамилия начинается с «А».

3. Таблица 10 в базе данных представлена следующим образом:

Таблица 10. БД «Вселенная»

№	Планета	Средняя температура, °С	Площадь, тыс. км ²
1	Земля	14,8	510 072
2	Марс	-63,1	144 370
3	Меркурий	67	74 800
4	Венера	463	460 000

- 1) отсортируйте записи **По убыванию** в поле *Средняя температура*;
- 2) назовите количество записей, удовлетворяющих условию «Площадь более 500 млн км²».

В результате выполнения всех заданий должны быть найдены и отмечены только те записи, по которым заданы условия поиска или фильтрации.

1

Отвечаем на вопросы

1. Опишите действия по сортировке данных.
2. Опишите применение расширенного фильтра.

2

Думаем и обсуждаем

1. Для чего используется сортировка и фильтрация в базе данных?
2. Каким образом помогают методы сортировки и фильтрации данных MS Excel при работе с БД?

3

Анализируем и сравниваем

Проанализируйте виды и функции сортировки и фильтрации в базе данных.

4

Выполняем в тетради

Составьте небольшой кроссворд с новыми терминами БД: фильтр, сортировка, поиск данных и т.д.

5

Выполняем на компьютере

В таблице «Численное представление казахов в мире» с помощью фильтра определите:

№	Страны	Численность	Год переписи
1	Республика Казахстан	12 000 000	2017
2	Китайская Народная Республика	1 462 588	2010
3	Российская Федерация	647 732	2010
4	Республика Узбекистан	803 000	2017
5	Киргизская Республика	34 122	2015

Количество записей, удовлетворяющих условию «Численность больше 1 млн чел.» равно: 1) 1; 2) 2; 3) 3; 4) 4.

6

Делимся мыслями

Что нового вы узнали на сегодняшнем уроке? Обсудите со своими одноклассниками, насколько упрощают методы сортировки и фильтрации данных при работе с БД.

7

Домашнее задание

Подготовьте имеющуюся базу данных «Дневник». Отсортируйте в ней по возрастанию/по убыванию (в алфавитном порядке) таблицу «Учащиеся». Используя фильтрацию, отберите в поле ФИО всех девочек/мальчиков среди учащихся. Полученную таблицу сохраните на внешнем носителе либо в своей папке.

Творческое задание

В собственной модели базы данных в MS Excel примените методы фильтрации.

§ 14. Практикум. Работа с базой данных. Создание базы данных «Предметная олимпиада»

Этапы разработки БД:

1) Определение задач.

Обозначение структуры и состава БД, назначение и цели ее создания, виды осуществляемых работ в БД (ввод, редактирование данных, вывод данных или отчета и т.д).

2) Анализ объекта.

Определение составляющих объектов создаваемой БД, свойств этих объектов. Рассмотрение типов данных БД.

3) Синтез модели.

Выбор БД. Схематическое представление модели БД с указанием связей между таблицами.

4) Способы представления информации.

- С использованием форм.
- Без использования форм.

5) Синтез модели объекта и технология его создания.

Шаг 1. Запуск программы Microsoft Excel.

Шаг 2. Создание начальной таблицы или таблиц.

Шаг 3. Создание форм.

Шаг 4. Заполнение БД информацией.

6) Работа с созданной базой данных.

- Поиск необходимых данных.
- Сортировка данных.
- Фильтрация данных.
- Вывод на печать.
- Изменение и редактирование данных.

Порядок выполнения работы:

1. Создайте отдельную папку для работы с БД. Создайте на рабочем диске каталог **STUDY**, а в каталоге **STUDY** – собственный каталог, например, **MYWORK**.

2. Создайте БД «Предметная Олимпиада».

1) Запустите приложение Microsoft Excel.

2) Создайте таблицу **Участники** в БД «Предметная Олимпиада».

3) Задайте поля – **Порядковый номер/№**, **ФИО**, **Предмет**, **Город**.

4) Поле **Порядковый номер/№** установите как основное поле.

- 5) Сохраните созданную *таблицу 11* под именем **Участники**.
- 6) Таблицу заполните следующими записями:

Таблица 11. БД «Участники»

№	Ф.И.О	Предмет	Город
1	Иса Назерке Кайратовна	История Казахстана	Нур-Султан
2	Кулахметова Анар Кыдырхановна	Математика	Алматы
3	Караш Айым Ауезовна	Физика	Алматы
4	Айтбай Елжас Ерболатович	Иностранный язык	Кызылорда
5	Жаксылык Динара Ардаковна	Казахский язык и литература	Павлодар
6	Еркин Зилола Еркиновна	Русский язык и литература	Шымкент
7	Ерболатова Актоты Шахановна	Химия	Орал
8	Кайрат Аслан Темирханович	Биология	Шымкент
9	Батырова Ажар Ерболовна	Математика	Тараз

- 7) Выполните **сортировку** таблицы по полю **ФИО** в алфавитном порядке.
- 8) Для этого выделите таблицу, нажмите на кнопку **Сортировка** во вкладке **Данные** на панели инструментов. Затем в диалоговом окне **Сортировка** выберите необходимый столбец **ФИО** и порядок сортировки.
- 9) Таблицу **Участники** можно вставить записи следующих типов:

Таблица 12. Дополнение

№	Ф.И.О	Предмет	Город
10	Созакбаева Куралай Маратовна	История Казахстана	Кызылорда
11	Марат Аяулым Жаксылыковна	Химия	Актобе
12	Омаров Алмас Сабиржанович	Физика	Нур-Султан
13	Абдуллаева Адия Рахатовна	География	Караганда

- 10) Используя форму, измените предмет у Батыровой А.Е. на **Казахский язык и литература**.
- 11) Добавьте в *таблицу 12* еще одно поле **Язык обучения** и введите соответствующие данные.
- 12) Создайте форму **Учащийся**.
- 13) Предоставьте сведения по всем формам.
- 14) Сохраните БД «Предметная Олимпиада».
- 15) В результате в БД *таблица 13* имеет следующий вид:

Таблица 13. БД «Предметная Олимпиада»

№	Ф.И.О.	Предмет	Язык обучения	Город
1	Иса Назерке Кайратовна	История Казахстана	казахский	Нур-Султан
2	Кулахметова Анар Кыдырхановна	Математика	казахский	Алматы
3	Караш Айым Ауезовна	Физика	казахский	Алматы
4	Айтбай Елжас Ерболатович	Иностранный язык	русский	Кызылорда
5	Жаксылык Динара Ардаковна	Казахский язык и литература	казахский	Павлодар
6	Еркин Зилола Еркиновна	Русский язык и литература	казахский	Шымкент
7	Ерболатова Актоты Шахановна	Химия	казахский	Уральск
8	Кайрат Аслан Темирханович	Биология	русский	Шымкент
9	Батырова Ажар Ерболовна	Казахский язык и литература	казахский	Тараз
10	Созакбаева Куралай Маратовна	История Казахстана	казахский	Кызылорда
11	Марат Аяулым Жаксылыковна	Химия	казахский	Актобе
12	Омаров Алмас Сабиржанович	Физика	казахский	Нур-Султан
13	Абдуллаева Адия Рахатовна	География	русский	Караганда

§ 15. Практикум. Работа по поиску и фильтрации данных в базе данных

Пункт 1. Откройте для работы БД «Предметная Олимпиада» в папке MYWORK.

Пункт 2. Поиск и фильтрация данных БД «Предметная Олимпиада».

- 1) Запустите приложение Microsoft Excel.
- 2) В таблице **Участники** добавьте новое поле **Результаты**.
- 3) Заполните следующими записями:

Таблица 14. Добавление нового поля

№	ФИО	Результаты (баллы)
1	Иса Назерке Кайратовна	78
2	Кулахметова Анар Кыдырхановна	118
3	Караш Айым Ауезовна	109
4	Айтбай Елжас Ерболатович	89
5	Жаксылык Динара Ардаковна	114
6	Еркин Зилола Еркиновна	121
7	Ерболатова Актоты Шахановна	96
8	Кайрат Аслан Темирханович	84
9	Батырова Ажар Ерболовна	69
10	Созакбаева Куралай Маратовна	101
11	Марат Аяулым Жаксылықовна	92
12	Омаров Алмас Сабиржанович	77
13	Абдуллаева Адия Рахатовна	105

- 4) Во вкладке **Диаграммы** постройте гистограмму с полями **ФИО** и **Результаты**, добавленных в область ряда. Измените тип и цвет линий самой гистограммы.
- 5) Прodelайте 4 пункт для полей **Предмет** и **Результат**.
- 6) Произведите с помощью фильтра поиск всех результатов баллов выше 90.
- 7) Отберите по отдельности записи, содержащие участников:
 - 1) с казахским языком обучения и 2) с русским языком обучения.
- 8) Найдите всех участников по предмету **Физика** (используя фильтр **Начинается с ...**).
- 9) Предоставьте результаты работы. Сохраните базу данных.

ИТОГОВЫЕ ТЕСТОВЫЕ ЗАДАНИЯ К РАЗДЕЛУ III

1. Упорядоченную совокупность данных в определенной предметной области называют
 - А) электронной таблицей.
 - В) определенным списком.
 - С) базой данных.
 - Д) распределенными данными.
 - Е) иерархической системой.

2. Программа Excel – это
 - А) текстовый редактор.
 - В) электронная таблица.
 - С) система управления базами данных.
 - Д) графический редактор.
 - Е) программа проектирования приложений.

3. Сортировка – это
 - А) процесс поиска наибольшего и наименьшего значений массива.
 - В) процесс непрерывного упорядочивания некоторого множества.
 - С) процесс замены элементов некоторого множества.
 - Д) процесс упорядочения записей по значениям одного из полей.
 - Е) процесс отбора элементов множества, удовлетворяющих определенному условию.

4. В реляционных базах данных записью называют
 - А) ячейку таблицы.
 - В) строку таблицы.
 - С) столбец таблицы.
 - Д) имя поля.
 - Е) таблицу.

5. В MS Excel столбец однотипных данных является
 - А) полем.
 - В) записью.
 - С) формой.

- D) отчетом.
E) ключом.
- 6. База данных – это**
A) совокупность взаимосвязанных данных о некотором объекте.
B) программа для сбора и хранения информации.
C) таблица с данными в формате Excel.
D) набор произвольной информации.
E) интерфейс, поддерживающий наполнение данными.
- 7. Набор значений свойств элемента, размещенных в полях таблицы, это**
A) вектор.
B) атрибут.
C) запись.
D) таблица.
E) ключ.
- 8. Основным объектом для хранения информации в реляционных базах данных является**
A) таблица.
B) запрос.
C) форма.
D) отчет.
E) модули.
- 9. Строка, описывающая свойства элемента таблицы базы данных, называется**
A) полем.
B) бланком.
C) записью.
D) связью.
E) ключом.
- 10. Тип поля реляционной базы данных определяется**
A) именем поля.
B) типом данных.
C) именем ячейки.

- D) типом таблицы.
- E) типом ключа.

11. Поле типа Дата/Время в таблице MS Excel содержит... .

- A) многострочный текст до 65535 символов.
- B) число любого типа (вещественное, целое и т. п.).
- C) обычную строку текста до 255 символов.
- D) логические значения данных (истина/ложь).
- E) данные формата дата и/или время.

12. Что из перечисленного не является объектом Excel?

- A) Таблицы.
- B) Ключи.
- C) Формы.
- D) Отчеты.
- E) Запросы.

13. Для применения фильтра необходимо выбрать следующее:

- A) Данные ⇒ Сортировка и фильтрация ⇒ Фильтр.
- B) Формулы ⇒ Сортировка и фильтрация ⇒ Фильтр.
- C) Вставка ⇒ Сортировка и фильтрация ⇒ Фильтр.
- D) Главная ⇒ Сортировка и фильтрация ⇒ Фильтр.
- E) Фильтр ⇒ Сортировка и фильтрация ⇒ Фильтр.

14. СУБД служит

- A) для предоставления возможности легко управлять базами данными.
- B) для распределения данных между пользователями.
- C) для хранения больших данных.
- D) для описания данных.
- E) для выполнения инженерных и научных расчетов.

15. Наиболее распространенная модель базы данных?

- A) Распределенная.
- B) Иерархическая.
- C) Сетевая.
- D) Реляционная.
- E) Постреляционная.

РАЗДЕЛ IV

ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

Изучив главу, вы научитесь:

- создавать программы на языке программирования Python (пайтон) с использованием одномерных и двумерных массивов;
- применять алгоритмы сортировки.

§ 16–17. Одномерные массивы

Вспомните!

- Типы данных в Python.
- Виды алгоритмов в Python.
- Операторы выбора в Python.
- Операторы цикла в Python.

Вы узнаете:

- что такое массив;
- для чего предназначены индексы;
- какие типы данных принимает массив, индекс.

Массив – Массив – *Array*

Типы данных – Деректер типі – *Data Type*

Индекс – Индекс – *Index*

Перейдем к изучению нового понятия – массив. До этого мы уже познакомились с понятиями переменной, типами данных, значения которых могут принимать переменные `int` – целые, `float` – дробные, они же вещественные, `str` – набор символов и так далее.

Каждый тип переменной хранит данные определенного типа и занимает свой объем оперативной памяти компьютера. Например, `int` может занимать в памяти от 1 до 4 байта. Различают одномерные и многомерные массивы.

Массив – это совокупность переменных, содержащих данные одного типа, объединенных общим именем. Каждая отдельно взятая переменная называется элементом массива. Доступ к элементам массива осуществляется через указание индексов, которые стоят внутри квадратных скобок после имени и обязательно нумеруются, начиная с нуля. Поэтому массивы иногда называют переменными с индексом.

Одномерные массивы

Одномерный массив – это множество однотипных элементов, объединенных под одним именем, где каждый элемент имеет свой порядковый номер.

Порядковый номер – это индекс элемента массива. Элементы массива располагаются по порядку один за другим.

При объявлении массива указывается имя массива и тип его элементов. Размер массива не задается, он может быть расширен новыми элементами любое время. Для использования

массива в Python понадобится импортирование стандартного модуля `array`.

Служебные слова в Python:

'and', 'or', 'in', 'is', 'not', 'if', 'else', 'elif', 'assert', 'break', 'continue', 'def', 'del', 'for', 'from', 'while', 'yield', 'import', 'print', 'return'.

Модули – это готовые программы, включающие функции, объекты и классы. Например:

```
import array as arr
a = arr.array('i'),
```

где `arr` – название, которое мы задаем сами для дальнейшего использования модуля `array`;

`a` – имя массива (в данном случае);

`i` – тип элементов массива (`integer`).

Обращение к каждому элементу массива осуществляется с помощью указания имени массива и индекса элемента (рис.16).

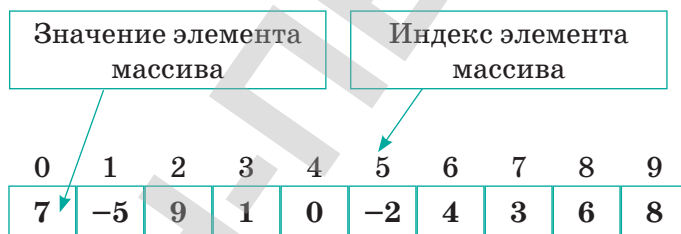


Рис. 16. Одномерный массив

В работе с одномерными массивами нет ничего сложного:

```
#объявляем массив firstArray
firstArray = arr.array('i')
```

Массив объявляется путем указания типа его элементов, в нашем случае это целые (`int`), затем указывается его имя, в нашем случае это `firstArray`. Мы хотим, чтобы массив состоял из 8 элементов. Нужно изменить (задать) эти значения, т.е. выполнить инициализацию массива. Ее проще всего выполнять с помощью цикла, в котором мы последовательно проходим

по всем элементам массива. Инициализируем, применяя формулу $i * 2 - 1$, где i – значение индекса массива:

```
#инициализируем массив данными по формуле
for i in range(0,8):
    firstArray.append(i * 2 - 1)
```

Важная информация

В Python очень важно правильно оформлять программу. Нужно принимать во внимание, что табуляция и пробелы в строках и между ними имеют определенный смысл. В отступах рекомендуется использовать 4 пробела.

Ввод и вывод одномерного массива

Для ввода и вывода значения элементов массива используется цикл с параметром.

```
#заполнение в цикле всех элементов массива
for i in range(0,10):
    testArray.append(i)
#вывод в цикле содержимого массива
for i in range(0,10):
    print(testArray[i])
```

Массивы применяют в тех случаях, когда нельзя обойтись одной-двумя переменными или когда имеется определенная последовательность элементов одного типа. Допустим, пользователю необходимо обратиться к переменным с определенным местом в последовательности, в начале или середине, поменять их значения местами, отсортировать по возрастанию/по убыванию и т.д.

Рассмотрим описанное выше на примере следующей программы:

```
#Программа запрашивает Выставленные оценки
по предмету.
point1 = input("Vvedite ocenku: ")
point2 = input("Vvedite ocenku: ")
point3 = input("Vvedite ocenku: ")
point4 = input("Vvedite ocenku: ")
point5 = input("Vvedite ocenku: ")
print("Byli vvedeny sleduyushchie ocenki: ")
```

```
print(point1+'\t',point2+'\t',point3+'\t',
point4+'\t',point5+'\t')
```

Результат работы программы (рис. 17):



```
C:\Windows\System32\cmd.exe
Vvedite ocenku: 4
Vvedite ocenku: 5
Vvedite ocenku: 2
Vvedite ocenku: 5
Vvedite ocenku: 4
Byli vvedeny sleduyuschie ocenki:
4      5      2      5      4
C:\Users\Gulnur\Desktop\Informatics\source>
```

Рис. 17. Результат программы

Программа, приведенная выше, запрашивает у пользователя пять оценок, которые сохраняются в пяти переменных:

```
point1, point2, point3, point4, point5;
```

А теперь предложим вариант этой же программы, но с использованием массива. Вот что получится:

```
import array as arr
point = arr.array('i')
for i in range(0,5):
    point.append(int(input("Vvedite ocenku: ")))
```

#input всегда возвращает данные с типом str, поэтому мы использовали функцию int() для конвертации в числа.

```
print("Byli vvedeny sleduyuschie ocenki: ")
for i in range(0,5):
    print(point[i], end='\t')
```

Эта программа выведет такой же результат, что и в первом варианте, но ее запись короче. Такой вариант программирования более удобен и компактен за счет использования массива.

Типы массивов

Теперь перейдем к рассмотрению основных типов массивов, применяемых в программировании.

По размерности массивы бывают следующих типов:

- одномерные;
- двумерные, или матрицы;
- трехмерные и т.д.

На практике редко используют массивы размерностью более трех (хотя и трехмерные используются не очень часто). Примером трехмерного массива является учебник или книга (1-й индекс – страницы в книге, 2-й индекс – строки на страницах, 3-й индекс – буквы в строке). Массивы размерностью более одного по-другому еще называют – **многомерными** (двумерные, трехмерные массивы и т.д.).

В языке программирования Python нет классических многомерных массивов. Однако базовые возможности этой платформы легко позволяют построить многомерный список (list).

Важная информация

Нумерация элементов массива начинается с нуля. Например, если мы имеем массив, состоящий из пяти элементов, то доступ к ним можно получить таким образом: A[0]; A[1]; A[2]; A[3]; A[4].

Объявление (задание) массива в среде программирования Python

Для того чтобы объявить массив, нужно указать имя массива и тип переменных.

Например, объявим массив типа float:

```
import array as arr
a = arr.array('f')
```

Инициализация массива в среде программирования Python

После того, как мы объявим массив, мы можем использовать его в своей программе, однако, если мы его не проинициализировали (не задали начальные значения его элементам), то это будет бессмысленно. Инициализацию массива можно выполнить несколькими способами. Например, при объявлении int:

```
numbers_array = arr.array('i', [7, -5, 9, 1, 0,
-2, 4, 3, 6, 8])
```

Массив `numbers_array` имеет 10 элементов. Размер массива можно узнать, используя функцию `len()`:

```
len(numbers_array)
```

Если нужно, чтобы изначально все элементы массива размерностью 9 были единицами, то используем цикл:

```
import array as arr
b = arr.array('i')
for i in range(0,9):
    b.append(1)
```

Можно задать массив и таким способом:

```
b = arr.array('i', [10, 20, 42])
```

В Python индексы последовательностей начинаются с нуля. Отрицательные индексы указывают на элемент с конца последовательности (`-1` – последний элемент). Например:

```
>>> s = [1, 2, 3, 4, 5]
>>> print s[0], s[2], s[-1]
1 3 5
```

Это интересно!

Язык Python был разработан Гвидо Ван Россумом в 90-х годах. Крупнейшие компании мира (NASA, Google, Yahoo, Microsoft и т. д.) применяют язык Python для своих разработок.

1

Отвечаем на вопросы

1. Что такое массив?
2. Как объявляется массив? Как форматируется массив?
3. Какие типы данных может содержать массив?

2

Думаем и обсуждаем

1. Поясните различие между индексом и элементом массива.
2. Расскажите о преимуществах применения массивов. Дайте краткое описание того, как задаются массивы: как они объявляются, как происходит обращение к элементам массива, как производится ввод и вывод элементов массива.

3

Анализируем и сравниваем

Перечислите типы данных Python, приведите примеры массивов с этими типами данных.

4

Выполняем в тетради

1. Прделайте несколько операций с массивом Array.
 - 1) Объявите массив типа float с числом элементов 10 и присвойте ему нулевые начальные значения.
 - 2) Выведите 3-й элемент массива Array = 10.
 - 3) Присвойте 8-му элементу массива значение 3,14.
 - 4) Увеличьте значение 1-го элемента массива на 1.
2. Составьте кроссворд или ребус с новыми терминами: *массив, индекс, тип данных и т.д.*

5

Выполняем на компьютере

Проверьте результаты вычислений.

```
import array as arr
Array = arr.array('f')
for i in range(0,10):
    Array.append(0)
print("Array[2]: ",Array[2])
Array[7] = 3.14
Array[0]+=1
```

6

Делимся мыслями

Что нового вы узнали на уроке? Обсудите со своими одноклассниками, как массивы применяются в программировании. Есть ли необходимость в использовании массивов?

7

Домашнее задание

Прделайте следующие операции с массивом Array.

1. Объявите массив типа int с числом элементов 20 и присвойте ему нулевые начальные значения.
2. Присвойте элементам массива значение их индекса.

Творческое задание:

Используя материалы из Интернета, подготовьте презентацию на тему «Применение массивов на практике».

§ 18–19. Поиск элементов с заданными свойствами

Вспомните!

- Что такое массив?
- Для чего используются индексы?

Вы узнаете:

- как решать стандартные задачи по массивам с условиями;
- как работает сумматор накопления;
- как осуществляется накопление произведения.

Положительные элементы –
Оң элементтер – *Positive elements*

Отрицательные элементы – Тегіс
элементтер – *Negative elements*

Четные индексы – Жұп индекстер –
Even indexes

Нечетные индексы – Тақ индекстер –
Odd indexes

При работе с одномерными массивами часто приходится осуществлять поиск элементов по определенным заданным условиям. Это задачи на поиск наибольшего/наименьшего элемента среди множества элементов (максимум, минимум); вычисление суммы положительных или отрицательных чисел, четных или нечетных и т.д.

Примеры одномерных массивов

Рассмотрим практические примеры применения массивов. Задайте одномерный массив, содержащий 20 элементов, взятых случайным образом из интервала $[1, 100]$.

Вычислите и выведите результат:

1. Сумму всех четных значений элементов массива (S1).
2. Сумму всех нечетных значений элементов массива (S2).
3. Значение наибольшего элемента в массиве (max).
4. Значение наименьшего элемента в массиве (min).
5. Сумму элементов массива, значения которых меньше 30 (Sum).
6. Произведение элементов массива с индексами 2 или 9 (P).

Последовательно рассмотрим и разберем программы приведенных задач.

Итак, сначала задаем массив и переменные, которые мы будем использовать для хранения результатов работы с элементами массива

```

size = 20
a = arr.array('i', )
S1 = 0
S2 = 0
Sum = 0
P = 1

```

Мы объявили массив, теперь нужно выполнить его инициализацию. По заданию массив нужно заполнить случайными значениями от 1 до 100. Задаем нашему одномерному массиву случайные значения:

```

for i in range(size):
    a.append(random.randint(1,100))

```

Естественно, это удобнее всего делать в цикле, проходя по всем элементам массива последовательно и присваивая каждому свое случайное значение. Для того чтобы функция `randint()` генерировала случайные величины в интервале от 1 до 100, нужно задать начальное и конечное значения диапазона. Теперь переходим к основной задаче – вычислению нужных сумм и произведений. Для прохода по массиву используем цикл `for`:

```

for i in range(size):
    if a[i]%2 == 0:
        S1 += a[i]
    if a[i]%2 != 0:
        S2 += a[i]
    if a[i] < 30:
        Sum += a[i]
    if i==2 or i==9:
        P *= a[i]

```

Как обычно, для прохода по всему одномерному массиву мы используем цикл `for` в интервале от 0 до `size`, где `size` является размером массива. Условие используется строгое, без включения самого `size`, т.к. нумерация элементов массива начинается с нуля, т.е. если размер массива равен 20, то его индексами будут числа от 0 до 19 включительно, что есть 20 элементов.

- 1) Далее по условию задания нам нужно найти сумму всех четных значений элементов массива. Для этого мы будем использовать соответствующее условие выборки:

```
if a[i]%2 == 0:
    S1 += a[i]
```

Если остаток от деления значения элемента на 2 равен нулю, то значение этого элемента массива четное и, соответственно, его можно прибавлять к общей сумме. Так как мы используем переменную $S1$ для накопления общей суммы, то при объявлении мы инициализировали ее значением 0. Напоминаем, что это сокращенное выражение $S1+=a[i]$; в полной форме оно будет выглядеть, как $S1=S1+a[i]$.

- 2) Далее по условию задания нам нужно найти сумму всех нечетных значений элементов массива. Для нахождения этой суммы используется обратное условие, т.е. если остаток от деления не равен 0, то выполняем суммирование в общую переменную накопления:

```
if a[i]%2 != 0:
    S2 += a[i]
```

- 3–4) Переходим к задачам нахождения наибольшего (max) и наименьшего (min) элементов в массиве.

Алгоритм следующий: 1) перед циклом необходимо занести первый элемент массива в переменные `minimum` и `maximum`, они будут хранить минимальное и максимальное значения изначально; 2) во время цикла переменные поменяют его, если найдется значение меньше для `minimum` или больше для `maximum`; 3) вывод переменных с максимальным и минимальным элементом.

Идет сравнение каждого элемента массива: если элемент больше переменной с максимальным значением, то значение записывается в переменную `maximum`; если элемент меньше переменной с минимальным значением, то значение записывается в переменную `minimum`:

```

maximum = a[0] //Помещаем значения 1-го элемента
массива в переменные
minimum = a[0]
for i in range(size):
#если значение элемента больше значения переменной
max, то записываем это значение в переменную
    if a[i] > maximum:
        maximum = a[i]
#аналогично и для min
    if a[i] < minimum:
        minimum = a[i]

```

Также в Python есть встроенные функции `max()`, `min()`:

```

maximum_1 = max(a)
minimum_1 = min(a)

```

- 5) Для подсчета суммы всех элементов, значения которых меньше 30, используем соответствующее условие выборки:

```

if a[i] < 30:
    Sum += a[i]

```

- 6) И последнее, что нам осталось сделать, это найти произведение элементов массива с индексами 2 или 9. Задаем условие, что если в данный момент мы находимся на элементах массива 2 или (or) 9, то выполнить умножение:

```

if i==2 or i==9:
    P *= a[i]

```

Ниже приводится полный код данной программы:

```

#Пример работы с элементами одномерного массива
import array as arr
import random
random.seed(None)

#объявляем необходимые переменные
size = 20

```



```
a = arr.array('i', )
S1 = 0
S2 = 0
Sum = 0
P = 1

#функция вывода массива на экран
def print Array(array):
for i in range(len(array)):
    print(array[i], end=" ")
    print('\n')

#наполняем массив случайными величинами
for i in range(20):
    a.append(random.randint(1,100))

#выводим содержимое массива на экран
printArray(a)

#выполняем необходимые подсчеты в цикле
for i in range(size):
    if a[i]%2 == 0:
        S1 += a[i]
    if a[i]%2 != 0:
        S2 += a[i]
    if a[i] < 30:
        Sum += a[i]
    if i==2 or i==9:
        P *= a[i]

maximum_1 = max(a)
minimum_1 = min(a)
maximum = a[0]
minimum = a[0]
for i in range(len(a)):
    if a[i] > maximum:
        maximum = a[i]
    if a[i] < minimum:
        minimum = a[i]
```

```
#выводим результаты работы программы
print("Summa chetnyh: ", S1)
print("Summa nechetnyh: ", S2)
print("Min: ", minimum, minimum_1)
print("Max: ", maximum, maximum_1)
print("Summa elementov do 30: ", Sum)
print("Proizvedeniya elementov s indexami 2 ili
9: ", P)
```

1

Отвечаем на вопросы

1. Как искать элементы массива с определенными параметрами?
2. Дайте описание алгоритма нахождения элементов max и min в массиве.

2

Думаем и обсуждаем

Дайте краткое описание того, как осуществляется поиск элементов массива с определенными параметрами. В чем заключается суть решения этих задач? Как происходит обращение к элементам массива? Как производится вычисление суммы и произведения элементов массива?

3

Анализируем и сравниваем

Сумма всех элементов одномерного массива рассчитывается довольно просто. Нет никаких условий, все элементы массива складываются последовательно.

```
sum += massiv[i];
```

Как будет выглядеть формула вычисления произведения всех элементов массива? Сравните.

4

Выполняем в тетради

- Проделайте несколько операций с массивом Array.
1. Найдите сумму всех элементов массива (S).
 2. Найдите произведение всех элементов массива (P).

5

Выполняем на компьютере

1. Проверьте результаты вычислений.

```
for i in range(size):  
    S += Array[i]  
for i in range(size):  
    P1 *= Array[i]
```

2. Выполните задания на диске, соответствующие данной теме.

6

Делимся мыслями

Что нового вы узнали на уроке? Обсудите со своими одноклассниками, какие у вас возникли сложности при освоении нового материала? На что нужно обратить внимание?

7

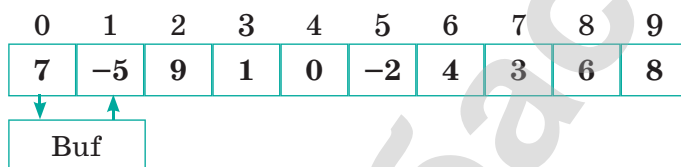
Домашнее задание

Проделайте несколько операций с массивом Array.

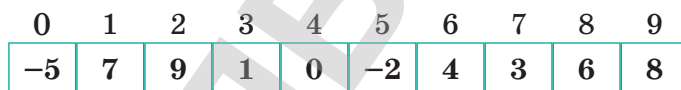
1. Найдите сумму положительных элементов массива (S).
2. Найдите произведение отрицательных элементов массива (P).

§ 20. Практикум. Перестановка элементов массива

Рассмотрим алгоритм перестановки элементов массива. Для этого необходимо использовать ячейку-переменную (буфер), в которую сначала временно перемещаем значение первой переменной. Затем в первую переменную помещаем значение второй переменной. Сохраненное значение первой переменной из буфера переносим во вторую переменную. Схема этой процедуры может быть представлена и записана следующим образом:



```
buf=a[0]
a[0]=a[1]
a[1]=buf
```



Задание 1. В массиве x замените элементы таким образом, чтобы элементы с четными индексами встали на место нечетных и наоборот, и занесите их в массив y .

```
import array as arr

x = arr.array('i')
print("x:", end="")
n = 10
x = input().split()[:n]
y = x

for i in range(10):
    if i % 2 == 0:
        c = x[i]
        x[i]=x[i+1]
        x[i+1]=c
```

```

    y[i] = x[i]
print("y: ", end="")
for i in range(len(y)):
    print(y[i], end=" ")

```

Результат работы программы (рис. 18):

```

===== RESTART: C:\Users\Admin\Documents\123.py =====
x: 7 1 8 2 9 3 6 4 9 5
y: 1 7 2 8 3 9 4 6 5 9
>>>|

```

Рис. 18. Результат программы

Задание 2. Поменяйте местами наибольший и наименьший элементы (max и min) в заданном массиве.

```

import array as arr

x = arr.array('i')
print("x=", end="")
n = 10
x = input().split()[:n]
max = x[0]
min = x[0]
for i in range(len(x)):
    if x[i] > max:
        max = x[i]
    elif x[i] < min:
        min = x[i]
imax=x.index(max)
imin=x.index(min)
print("Max: ", max, "Index of max element:", imax)
print("Min: ", min, "Index of min element:", imin)

b = x[imin]
x[imin] = x[imax]
x[imax] = b

for i in range(len(x)):
    print(x[i],end=' ')

```

```
print()
```

Результат работы программы (рис. 19):

```
===== RESTART: C:\Users\Admin\Documents\max min.py =====
x=7 8 9 -4 5 6 1 2 3 0
Max: 9 Index of max element: 2
Min: -4 Index of min element: 3
7 8 -4 9 5 6 1 2 3 0
>>> |
```

Рис. 19. Результат программы

Задание 3. Дан одномерный массив числовых значений, насчитывающий n элементов. Выполните перемещение элементов массива по кругу вправо.

$A[1] \rightarrow A[2]; A[2] \rightarrow A[3]; \dots A[n] \rightarrow A[1]$

```
import random

n=random.randint(1,20)
arr=[random.randint(0,100) for i in range(n)]
print(arr)

for i in range(n):
    arr[-i], arr[-i-1]=arr[-i-1], arr[-i]
arr[0], arr[1]=arr[1], arr[0]

print(arr)
```

§ 21. Практикум. Сортировка элементов массива

Под сортировкой элементов массива подразумевается расположение в порядке возрастания или убывания элементов массива. Сортировка осуществляется с помощью перестановки элементов массива различными методами.

Рассмотрим метод сортировки данных, который называется **пузырьковой сортировкой** (также его называют **методом обмена**). Будет приведен алгоритм и его реализация на языке программирования Python.

Упорядоченный массив создается на том же участке памяти, где находится исходная последовательность. Идея метода состоит в том, чтобы попарно сравнивать соседние элементы.

Каждый проход начинается с начала последовательности. Первый элемент массива сравнивается со вторым, если порядок между ними нарушается, то они меняются местами. Затем сравниваются второй с третьим, третий с четвертым и так далее: элементы с неправильным порядком в паре меняются местами до конца массива. В итоге после первого прохода, максимальный (или минимальный, в зависимости от вида сортировки: по возрастанию/по убыванию) элемент будет находиться на последнем месте в массиве, он как бы «всплывет» наверх. Именно поэтому этот метод называется пузырьковой сортировкой. На следующем проходе рассматривается последовательность от 1 до $N-1$, затем от 1 до $N-2$ и так до конца. После каждого прохода можно проверять, выполнялись ли перестановки элементов. Если не выполнялись, то сортировка завершена.

Задание 1. Реализация алгоритма сортировки на языке программирования Python.

```
import array as arr
#n - количество элементов
n = int(input("Number of elements: "))
mass = arr.array('i')
#заполняем массив
for i in range(n):
    print(i+1, end="-")
    mass.append(int(input("element: ")))
#извлечение исходного массива
```



```

print("Initial array: ", end=" ")
for i in range(len(mass)):
    print(mass[i], end=" ")

print("\n")
#сортировка массива в порядке убывания
for i in range(n):
    for j in range(n-1):
        if mass[j] < mass[j+1]:
            #перестановка
            temp = mass[j]
            mass[j] = mass[j+1]
            mass[j+1]=temp
#извлечение отсортированного массива
print("Final array: ", end = " ")
for i in range(len(mass)):
    print(mass[i], end=" ")

```

Результат работы программы (рис. 20):

```

C:\Windows\System32\cmd.exe
Number of elements: 7
1-element: 2
2-element: 6
3-element: 9
4-element: 1
5-element: 3
6-element: 7
7-element: 4
Initial array: 2 6 9 1 3 7 4
Final array: 9 7 6 4 3 2 1
C:\Users\Gulnur\Desktop\Informatics\source>_

```

Рис. 20. Результат программы

В языке программирования Python сортировка осуществляется также через функцию `sorted()`. Для представления элементов последовательности в обратном порядке используется `reverse`.

По возрастанию:

```

my_list = [2, 6, 9, 1, 3, 7, 4]
a = sorted(my_list)

```

```
print(a) # [1,2,3,4,6,7,9]
```

По убыванию:

```
my_list = [2,6,9,1,3,7,4]
a = sorted(my_list, reverse = True)
print(a) # [9,7,6,4,3,2,1]
```

Задание 2. Пример консольной программы на языке программирования Python, которая считывает массив, сортирует его пузырьковым методом и выводит результат на экран.

```
import array as arr

n = int(input("Number of elements: "))
mass = arr.array('i')

for i in range(n):
    print(i+1, end="-i")
    mass.append(int(input("element: ")))

print("Initial array: ", end = " ")
for i in range(n):
    print(mass[i], end=" ")
print("\n")

noSwap = True
passnum = n-1
while passnum > 0 and noSwap:
    noSwap = False
    for i in range(passnum):
        if mass[i] > mass[i+1]:
            noSwap = True
            temp = mass[i]
            mass[i] = mass[i+1]
            mass[i+1] = temp
    passnum = passnum - 1

print("Final array: ", end = " ")
for i in range(n):
    print(mass[i], end=" ")
```

Разберем код программы. Подключим необходимые библиотеки (строка 1), предложим пользователю ввести число N (количество элементов в массиве) и считаем число N (строка 3), создаем массив (строка 4), предложим пользователю ввести элементы массива и считаем элементы массива (строки 6–8), выполним сортировку пузырьковым методом (строки 16–26), выведем отсортированный массив на экран (29–31). Работа программы представлена на скриншоте ниже (рис. 21):

```

C:\Windows\System32\cmd.exe
Number of elements: 5
1-i element: 9
2-i element: -6
3-i element: 1
4-i element: 8
5-i element: -7
Initial array: 9 -6 1 8 -7
Final array: -7 -6 1 8 9
C:\Users\Gulnur\Desktop\Informatics\source>

```

Рис. 21. Результат программы

Задание 3. Даны расстояния от Нур-Султана до некоторых городов Казахстана. Расставьте значения расстояний до городов в порядке отдаленности: $S = \{1224, 196, 1505, 426, 276\}$.

Города РК	Алматы	Караганда	Шымкент	Павлодар	Кокшетау
S, км	1224 км	196 км	1505 км	426 км	276 км

§ 22. Практикум. Удаление и вставка элемента

Что значит удаление элемента из одномерного массива? Удаление должно привести к физическому «уничтожению» элемента из массива, при этом общее количество элементов должно быть уменьшено. А при вставке элемента в массив общее количество элементов должно быть увеличено.

Исходная последовательность:

-5	7	9	1	0	-2	4	3	6	8
----	---	---	---	---	----	---	---	---	---

Последовательность после удаления элемента массива:

-5	7	9	1	0	-2	4	3	6
----	---	---	---	---	----	---	---	---

Последовательность после вставки элемента массива:

-5	7	9	1	0	-2	4	3	6	8	-1
----	---	---	---	---	----	---	---	---	---	----

Для удаления и вставки элементов в массивы в языке программирования Python есть прописанные функции, которые можно использовать.

Для вставки элемента используются методы `insert()` и `append()`, если необходимо вставить несколько элементов, используется метод `extend()`.

Функция `x.insert(i, n)` позволяет вставить элемент `n` по указанному индексу `i` в массиве `x`.

Можно удалить элементы массива, используя оператор `del`. Также можно использовать метод `remove()` для удаления элемента или метод `pop()` для удаления элемента по указанному индексу.

Функция `x.remove(n)` позволяет удалить элемент `n` из массива `x`.

Рассмотрим эти функции на примере:

```
import array as arr

#Введите количество элементов
n = int(input("Number of elements: "))
mass = arr.array('i')
#Функция для вывода массива на экран
```

```
def print Array(a):
    for i in range(n):
        print(a[i], end=" ")
    print('\n')

#Инициализируем массив
for i in range(n):
    print(i+1, end="-i")
    mass.append(int(input("element: ")))
print("Initial array: ", end=" ")
print Array(mass)

#Введите номер элемента, который хотите удалить
m = int (input("Enter the number of element for
delete: "))

#Удаление элемента с массива
del mass[m-1]
#Вставка элемента в начало массива
mass.insert(0,1)

#Вставка элемента в конец массива
mass.append(9)
print("Array: ", end=" ")
print Array(mass)

print("extend([5, 6, 7]) appends iterable to the
end of the array")

#Вставляем в конец массива элементы 5, 6, 7
mass.extend([5, 6, 7])
print("Extended array: ", end=" ")
print Array(mass)

#Удаление 3-го элемента массива
print("Remove 3d element")
mass.pop(2)

#Удаление элемента массива числа 7
mass.remove(7)
```

```
print("Final Array: ", end=" ")
print Array(mass)
```

Результат программы (рис. 22):

```
===== RESTART: C:\Users\Admin\Documents\delete_insert.py =====
Number of elements: 5
1-i element: 1
2-i element: 2
3-i element: 3
4-i element: 4
5-i element: 5
Initial array: 1 2 3 4 5

Enter the number of element for delete: 2
Array: 1 1 3 4 5 9

extend ([5, 6, 7]) appends iterable to the end of the array
Extended array: 1 1 3 4 5 9 5 6 7

Remove 3d element
Final Array: 1 1 4 5 9 5 6
```

Рис. 22. Результат программы

Задание 1. В произвольно заданном массиве удалите все четные значения элементов.

```
import random

a = []
for i in range(20):
    a.append(random.randint(1,100))
print("A =", a)

b = []
i = 0
while i < len(a):
    if a[i] %2 == 0:
        b.append(a[i])
        del a[i]
    else:
        i += 1

print("A =", a)
print("B =", b)
```

Задание 2. Дан массив размера n и целое число k ($1 \leq k \leq N$). Перед элементом массива с порядковым номером k вставьте новый элемент с нулевым значением.

```
import array as arr

n = int(input("Number of elements: "))
k = int(input("Number of elements: "))
mass = arr.array('i')

def printArray(a):
    for i in range(len(a)):
        print(a[i], end=" ")
    print('\n')

for i in range(n):
    print(i+1, end="-i ")
    mass.append(int(input("element: ")))

print("Initial array: ", end = "")
printArray(mass)

mass.insert(k,0)

print("Array: ", end = "")
printArray(mass)

print("Final Array: ", end = "")
printArray(mass)
```


§ 23. Двумерные массивы

Вспомните!

- Какие типы данных вам известны?
- Какие функции выполняют операторы цикла *while*, *for*?
- О вводе и выводе одномерных массивов.

Вы узнаете:

- о понятии «многомерный массив»;
- что такое списки, размерность списков;
- как вводить и выводить списки.

Список – Tізім – List

В задачах часто приходится хранить данные в виде таблиц. Такой вид данных называется **матрицами**, или **двумерными массивами**. В языке программирования Python нет многомерных массивов, однако любая таблица может быть представлена списком списков, то есть каждый элемент списка является списком.

Основы списка (List)

Список в Python – это упорядоченная коллекция элементов любого типа. Для сравнения, массив – это упорядоченная коллекция элементов одного типа, поэтому список принципиально более гибок, чем массив,

но именно эта гибкость делает вещи немного сложнее, если вы хотите работать с обычной структурой. Список также является динамически изменяемым типом, и это означает, что вы можете добавлять и удалять элементы из списка в любое время.

Рассмотрим пример записи двумерного массива:

```
my_list1 = [[11, 12, 5, 2], [15, 6, 10, 5], [10, 8, 12, 5], [12, 15]]
```

Следует отметить, что двумерный массив может иметь разное количество столбцов и строк.

Двумерные массивы могут содержать разные типы данных.

```
my_list2 = [[11, 12, 5, 2], ['hello', 'hi', 'salem'], [10.1, 8.2, 12.4, 3.5], [12, 15]]
```

Данный массив `my_list2` состоит из вложенных списков разных типов данных, как `int`, `string`, `float`.

Результатом вывода `my_list2` будет (рис. 23):



```
C:\Windows\System32\cmd.exe
11 12 5 2
hello hi salem
10.1 8.2 12.4 3.5
12 15
C:\Users\Volodar\Desktop\informatics\sources_
```

Рис. 23. Результат программы

Доступ к значениям в двумерном массиве

Элементы данных в двумерных массивах могут быть доступны с использованием двух индексов. Один индекс ссылается на основной массив, а другой – на позицию элемента данных во внутреннем массиве. Если мы упомянем только один индекс, тогда весь внутренний массив будет напечатан для этой позиции индекса. Пример ниже иллюстрирует, как это работает:

```
print(my_list1[0]) - выводит на экран [11, 12, 5, 2]
print(my_list1[1][2]) - выводит на экран 10
```

Чтобы распечатать весь двумерный массив, мы можем использовать цикл `for`:

```
for i in my_list1:
    for j in i:
        print(j, end=" ")
    print()
```

Приведенный выше код дает следующий результат (рис. 24):



Рис. 24. Результат программы

1

Отвечаем на вопросы

1. Что такое список?
2. Из каких типов данных могут состоять списки?
3. Как производится ввод и вывод элементов двумерного массива?
4. Как можно обратиться к элементам двумерного массива?

2

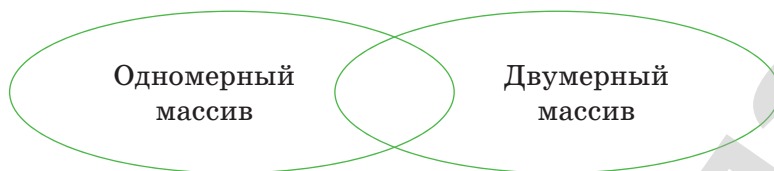
Думаем и обсуждаем

Расскажите о преимуществах работы со списками. Как определяются вложенные списки в массивах? Как осуществляется доступ к элементам двумерного массива?

3

Анализируем и сравниваем

Сопоставьте одномерный и двумерный массивы с помощью диаграммы Венна.



4

Выполняем в тетради

Запишите в тетради, что получится в результате выполнения этого кода.

```
my_list = [[7, 12, 4], [9, 50, 3], [45, 0, -1]]
print(my_list[1])
print(my_list[2])
print(my_list[-1])
```

5

Выполняем на компьютере

Запустите программу на компьютере.

```
my_list1 = [[7, 12, 4], [9, 50, 3], [45, 0, -1]]
my_list2 = [[7, 18, 6], [11, 3]]
my_list3 = [['Almaty', 'Dubai', 'London'],
['Kazakstan', 'UAE', 'England']]
print(my_list1[1])
my_list2[1] = [11, 3, 5]
print(my_list2)
print(my_list3[1][0])
```

6

Делимся мыслями

Что нового вы узнали на уроке? Насколько отличаются двумерные массивы от одномерных?

7

Домашнее задание

Напишите программу на языке Python, которая выведет двумерный массив в следующем виде:

```
1
2 3
4 5 6
7 8 9 10
```

§ 24. Практикум. Вставка и удаление элемента в двумерном массиве

Вставка значений в двумерный массив

Мы можем вставить новые элементы в определенную позицию, используя метод `insert()` и указав индекс.

Задание 1.

В приведенном ниже примере новый элемент данных вставляется в позиции индекса 2.

```
from array import *
my_list = [[11, 12, 5, 2], [15, 6, 10], [10, 8, 12, 5], [12, 15]]

my_list.insert(2, [0, 5, 11, 13, 6])

for i in myList:
    for j in i:
        print(j, end=" ")
    print()
```

Результат приведен ниже (рис. 25):



```
C:\Windows\System32\cmd.exe
11 12 5 2
15 6 10
0 5 11 13 6
10 8 12 5
12 15
C:\Users\Galnur\Desktop\Informatics\source>
```

Рис. 25. Результат программы

Удаление значений в двумерном массиве

Используя метод `del()`, можно удалить весь внутренний массив или конкретные элементы данных внутреннего массива путем переопределения значений с помощью индекса. Но если нам нужно удалить определенные элементы данных в одном

из внутренних массивов, следует использовать процесс обновления, описанный ниже.

Задание 2.

В этом примере выполняется удаление элемента массива.

```
from array import *
my_list = [[11, 12, 5, 2], [15, 6, 10], [10, 8, 12, 5], [12, 15]]

del my_list[3]

for i in my_list:
    for j in i:
        print(j, end=" ")
    print()
```

Результат программы (рис. 26):



```
C:\Windows\System32\cmd.exe
11 12 5 2
15 6 10
10 8 12 5
C:\Users\Gulnur\Desktop\Informatics\source>
```

Рис. 26. Результат программы

Задание 3.

Произвольно задайте двумерный массив в виде матрицы 5x5. Замените все значения элементов четных столбцов на 0, а нечетных столбцов на 1.

§ 25. Практикум. Задачи на двумерные массивы

Задание 1. Напишите программу в Python, которая принимает две цифры m (строка) и n (столбец) в качестве входных данных и генерирует двумерный массив. Значение элемента в строке i и столбце j массива должно быть $i * j$.

Листинг программы

```
row = int(input("Input number of rows: "))
column = int(input("Input number of columns: "))
my_list = [[0 for col in range(column)] for r in
range(row)]

for i in range(row):
    for j in range(column):
        my_list[i][j]= i*j

print(myList)
```

Запрашиваем количество строк и столбцов. Заполняем двумерный массив размером $row * column$ нулями. Заменяем каждый элемент на пересечении индексов строк и столбцов их произведениями. Например, значение элемента матрицы на 2-й строке и в 3-м столбце выводится таким образом: индекс второй строки равен 1, так как индексация в массиве начинается с 0; соответственно индекс 3-го столбца равен 2. Значение элемента $my_list[1][2] = 1 * 2 = 2$. Выводим массив на экран.

Результат работы программы (рис. 27):

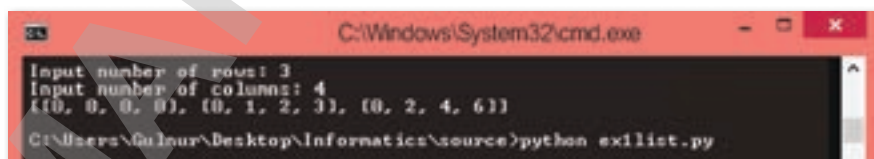


Рис. 27. Результат программы

Задание 2. Напишите программу, которая считает размерность двумерного массива и заполнит его случайными числами между 0 и 50. Выведите на экран сумму каждого столбца и произведение каждой строки в двумерном массиве.

Листинг программы

```

import random
random.seed(None)
row = int(input("Input number of rows: "))
column = int(input("Input number of columns: "))
my_list = [[0 for col in range(column)] for r in
range(row)]

sum_column = [0 for col in range(column)]
product_row = [1 for row in range(row)]

for i in range(row):
    for j in range(column):
        my_list[i][j]= random.randint(0,50)
print(my_list)

for i in range(column):
    for j in range(row):
        sum_column[i] += my_list[j][i]

for i in range(row):
    for j in range(column):
        product_row[i] *= myList[i][j]

print("Sum of column elements: ",sum_column)
print("Product of row elements: ",product_row)

```

Результат работы программы (рис. 28):

```

C:\Windows\System32\cmd.exe
Input number of rows: 2
Input number of columns: 3
[[40, 2, 31], [6, 41, 49]]
Sum of column elements: [46, 43, 80]
Product of row elements: [2480, 12054]
C:\Users\Galnur\Desktop\Informatics\source>_

```

Рис. 28. Результат программы

Задание 3.

Разработайте программу, которая выводит таблицу умножения. Полученный результат должен быть представлен в виде списка. Дополните программу вопросом с выводом ответа: «Чему равно произведение $5 * 5$?» и т.п.

ИТОГОВЫЕ ТЕСТОВЫЕ ЗАДАНИЯ К РАЗДЕЛУ IV**1. В какой из записей указано объявление массива?**

- A) `var A: array [1..10] of integer.`
- B) `for i in range(10): if A[i]<min: min=A[i].`
- C) `S=0 for i in range(10): S+=array[i].`
- D) `for i in range(10): A[i]=random.randint(1,100).`
- E) `a = arr. array('i').`

2. В какой из записей показан ввод элементов массива?

- A) `a = arr.array('i').`
- B) `for i in range(10): if A[i]<min: min=A[i].`
- C) `S=0 for i in range(10): S+=array[i].`
- D) `for i in range(10): A[i]=random.randint(1,100).`
- E) `var A: array [1..10] of integer.`

3. В какой из записей показан вывод элементов массива?

- A) `for i in range(10): print(A[i]).`
- B) `for i in range(10): if A[i]<min: min=A[i].`
- C) `S=0 for i in range(10): S+=array[i].`
- D) `for i in range(10): A[i]=random.randint(1,100).`
- E) `var A: array [1..10] of integer.`

4. В какой записи указано положительное число?

- A) `B[i]%2!=0.`
- B) `B[i]%2==0.`
- C) `B[i]>0.`
- D) `B[i]<0.`
- E) `B[i]<>0.`

5. В какой записи указано отрицательное число?

- A) `B[i]>0.`
- B) `B[i]%2==0.`
- C) `B[i]%2!=0.`
- D) `B[i]<0.`
- E) `B[i]<>0.`

6. В какой записи указано четное число?

- A) `B[i]>0.`
- B) `B[i]%2==0.`

- C) $B[i] \% 2 \neq 0$.
- D) $B[i] < 0$.
- E) $B[i] = 0$.

7. В какой записи указано нечетное число?

- A) $B[i] \% 2 \neq 0$.
- B) $B[i] > 0$.
- C) $B[i] \% 2 == 0$.
- D) $B[i] < 0$.
- E) $B[i] = 0$.

8. Согласно какой записи 4-й элемент массива равен 7?

- A) $B[3] = 7$.
- B) $B[4] = 7$.
- C) $B[7] = 4$.
- D) $B[7] = 3$.
- E) $B[i] = 7$.

9. Согласно какой записи идет увеличение каждого элемента массива в два раза?

- A) `for i in range(n): A[2*i]=A[i].`
- B) `for i in range(n): A[i]=A[i]/2.`
- C) `for i in range(n): A[i]=A[i]+2.`
- D) `for i in range(1,n): A[i]=A[i]+2.`
- E) `for i in range(n): A[i]=A[i]*2.`

10. Согласно какой записи идет уменьшение каждого элемента массива в два раза?

- A) `for i in range(1,n): A[i]=A[i]+2.`
- B) `for i in range(n): A[i]=A[i]*2.`
- C) `for i in range(n): A[i]=A[i]+2.`
- D) `for i in range(n): A[i]=A[i]/2.`
- E) `for i in range(n): A[2*i]=A[i].`

11. Найдите произведение элементов массива.

- A) `P=1 for i in range(0,n): P*=A[i].`
- B) `P=0 for i in range(0,n) P*=A[i].`
- C) `P=1 for i in range(1,n) P*=A[i].`
- D) `P=0 for i in range(1,n) P*=A[i].`
- E) `P=n for i in range(0,n) P*=A[i].`

12. Найдите сумму отрицательных элементов массива.

- A) `S=0 for i in range(0,n): if A[i]>0: S+=A[i].`
- B) `S=0 for i in range(0,n): if A[i]<0: S+=A[i].`
- C) `S=1 for i in range(0,n): if A[i]<0: S+=A[i].`
- D) `S=0 for i in range(1,n): if A[i]<0: S+=A[i].`
- E) `S=0 for i in range(0,n): if A[i]<>0: S+=A[i].`

13. Найдите произведение положительных элементов массива.

- A) `P=0 for in in range(0,n): if A[i]<0: P*=A[i].`
- B) `P=1 for in in range(0,n): if A[i]<0: P*=A[i].`
- C) `P=1 for in in range(0,n): if A[i]>0: P*=A[i].`
- D) `P=1 for in in range(1,n): if A[i]>0: P*=A[i].`
- E) `P=1 for in in range(0,n): if A[i]<>0: P*=A[i].`

14. Найдите сумму четных элементов массива.

- A) `S=0 for in in range(0,n): if A[i]%2 == 0: S+=A[i].`
- B) `S=0 for in in range(0,n): if A[i]%2 = 0: S+=A[i].`
- C) `S=1 for in in range(0,n): if A[i]%2 == 0: S+=A[i].`
- D) `S=0 for in in range(1,n): if A[i]%2 == 0: S+=A[i].`
- E) `S=0 for in in range(0,n): if A[2*i] == 0: S+=A[i].`

15. Найдите произведение нечетных элементов массива.

- A) `P=1 for in in range(0,n): if A[i]%2 != 0: P*=A[i].`
- B) `P=1 for in in range(0,n): if A[i]%2 !== 0: P*=A[i].`
- C) `P=0 for in in range(0,n): if A[i]%2 != 0: P*=A[i].`
- D) `P=1 for in in range(0,n): if A[i]%2 == 0: P*=A[i].`
- E) `P=1 for in in range(0,n): if A[i]%2 <> 0: P*=A[i].`

РАЗДЕЛ V

СОЗДАНИЕ 2D ИГРЫ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

Изучив раздел, вы научитесь:

- подключать библиотеку PyGame;
- использовать готовые модули библиотеки PyGame для создания окна для игры;
- создавать задний фон игры;
- загружать готовые персонажи для игры;
- программировать движение персонажа;
- управлять персонажами с клавиатуры;
- разрабатывать игру по готовому сценарию;
- реализовывать алгоритм подсчета результатов игры.

§ 26. Библиотека PyGame. Подключение библиотеки

Вспомните!

- О типах алгоритмов в Python.
- Об операторах ветвления и цикла.
- О типах данных в Python.

Вы узнаете:

- что такое PyGame;
- как подключается библиотека PyGame;
- о функциях модулей PyGame.

Библиотека PyGame –
PyGame кітапханасы –
PyGame Library

Запустить – Қосу – *Initiate*

Событие – Оқиға – *Event*

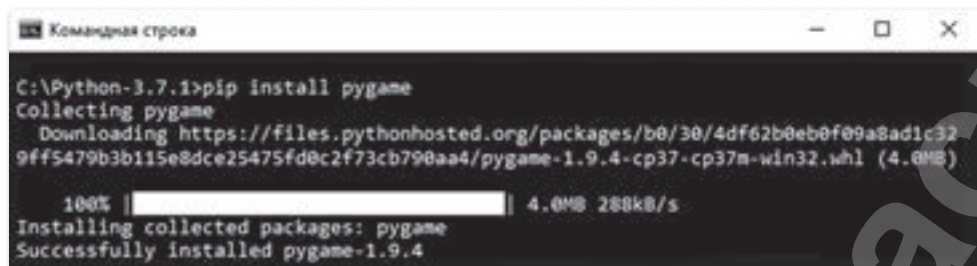
Библиотека – это собрание кода функций и классов. В Python термином, используемым для описания библиотеки с кодом, является *модуль (module)*. Например, модуль `pygame` содержит подмодули `pygame.draw`, `pygame.image`, `pygame.mouse` и др.

Библиотека PyGame – это модуль для языка программирования Python, предназначенный для разработки 2D игр и других мультимедийных приложений. PyGame разработан на основе другой библиотеки для создания игр SDL (Simple DirectMedia Layer), что позволяет работать на многих платформах и операционных системах. PyGame является привязкой к SDL, которая дает возможность программистам Python использовать мощные ресурсы данной библиотеки.

PyGame не входит в стандартную библиотеку Python, то есть не поставляется с установочным пакетом, а требует отдельной установки.

Вы можете скачать PyGame для вашей операционной системы с сайта www.pygame.org или в командной строке задать следующую команду (рис. 29):

```
Python - m pip install - U pygame - user
или
Python - m pip install pygame
```



```

Командная строка

C:\Python-3.7.1>pip install pygame
Collecting pygame
  Downloading https://files.pythonhosted.org/packages/b0/30/4df62b0eb0f09a8ad1c329ff5479b3b115e8dce25475fd0c2f73cb798aa4/pygame-1.9.4-cp37m-win32.whl (4.0MB)
    100% |#####| 4.0MB 288kB/s
Installing collected packages: pygame
Successfully installed pygame-1.9.4

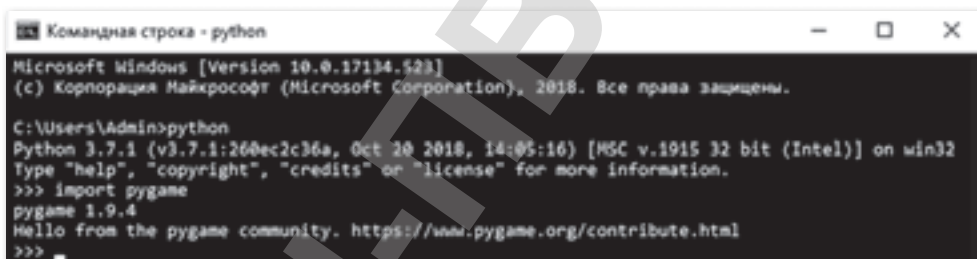
```

Рис. 29. Установка PyGame

После того, как вы установили соответствующий пакет, вы можете проверить его, открыв интерпретатор Python и введя следующую команду:

```
import pygame
```

Если библиотека была успешно установлена, вы должны увидеть версию PyGame, как показано на *рисунке 30*.



```

Командная строка - python

Microsoft Windows [Version 10.0.17134.523]
(c) Корпорация Майкрософт (Microsoft Corporation), 2018. Все права защищены.

C:\Users\Admin>python
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import pygame
pygame 1.9.4
Hello from the pygame community. https://www.pygame.org/contribute.html
>>> _

```

Рис. 30. Проверка установки PyGame

После импорта библиотеки PyGame ее необходимо инициализировать:

`pygame.init ()` – инициализирует каждый из подмодулей в пакете PyGame, который может загружать драйверы и запрашивать оборудование, чтобы PyGame был готов использовать все устройства на вашем компьютере.

Библиотека PyGame состоит из нескольких модулей, которые можно использовать по отдельности. Существуют модули для

каждого из устройств, которые вы можете использовать при создании игры.

В *таблице 15* предоставлен список и назначение некоторых модулей. Полный список и документацию можно найти по ссылке www.pygame.org/docs/. Для получения доступа к данным модулям используется обращение через PyGame.

Таблица 15. Модули библиотеки PyGame

Название модуля	Назначение
pygame.cdrom	Доступ к дисководам компакт-дисков и управление ими
pygame.cursors	Загружает изображения курсора
pygame.display	Доступ к дисплею
pygame.draw	Рисует фигуры, линии и точки
pygame.event	Управляет внешними событиями
pygame.font	Использует системные шрифты
pygame.image	Загружает и сохраняет изображение
pygame.key	Считывает нажатия клавиш с клавиатуры
pygame.mixer	Загружает и воспроизводит мелодии
pygame.mouse	Управляет мышью
pygame.movie	Воспроизводит видеофайлы
pygame.music	Работает с музыкой и аудио
pygame.rect	Управляет прямоугольными областями
pygame.sndarray	Манипулирует звуковыми данными
pygame.surface	Управляет изображениями и экраном
pygame.transform	Изменяет размеры и перемещает изображения
pygame.time	Управляет временем и частотой кадров

Создание и сохранение программы в Python с использованием библиотеки PyGame выполняется стандартным образом. Для этого в интегрированной среде разработки IDLE откроем меню **File** и выберем пункт **New File**, после чего откроется

дополнительное окно. В этом окне откроем меню **File** и выберем пункт **Save As...**. Затем выберите место и сохраните файл с расширением «*.py» (рис. 31).

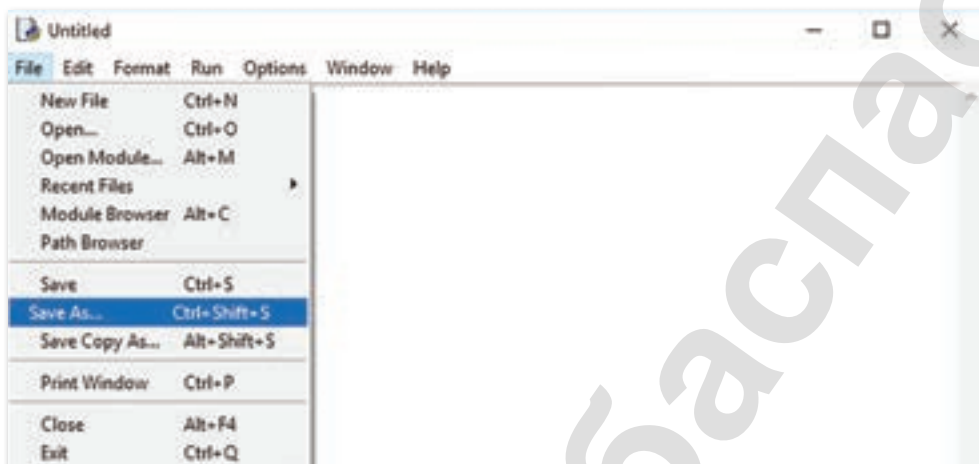


Рис. 31. Сохранение файла в Python

1

Отвечаем на вопросы

1. Что такое библиотека в Python?
2. Как осуществляется подключение библиотеки PyGame?
3. В чем заключается функция библиотеки PyGame?
4. Перечислите модули PyGame.

2

Думаем и обсуждаем

1. Является ли PyGame стандартной библиотекой Python?
2. Почему используются команды `import pygame`?
3. Для чего нужно деление модулей PyGame на несколько типов?

3

Анализируем и сравниваем

Проведите сравнительный анализ разных версий Python и версий модуля PyGame по выбору.

4

Выполняем в тетради

Запишите в тетради алгоритм установки библиотеки PyGame.

5

Выполняем на компьютере 

1. Установите PyGame на компьютере в классе.
2. Проверьте, установлена ли программа PyGame.
3. Укажите, как сохранить программу, созданную в библиотеке PyGame.

6

Делимся мыслями

Что нового вы узнали на уроке? Обсудите со своими одноклассниками, где и на каких сайтах или интернет-ресурсах можно ознакомиться с программами, разработанными на PyGame.

7

Домашнее задание 

1. Составьте кроссворд, содержащий термины Python и PyGame.
2. Выполните задания на диске, соответствующие данной теме.

§ 27. Использование модулей библиотеки PyGame для создания окна для игры

Вспомните!

- Что такое библиотека, модуль в Python?
- Типы модулей PyGame.
- Как назначаются модули PyGame?

Вы узнаете:

- о назначении модуля `pygame.display`;
- о функциях модуля `pygame.display`;
- о написании программы по созданию окна.

Экран – Экран – Screen

Задать режим – Режим кою – Set mode

Выход – Шығу – Quit

Первым шагом при разработке игры является создание окна, то есть площадки для игры. Библиотека PyGame имеет готовый модуль `pygame.display`, который предназначен для управления окном дисплея и экраном.

В таблице 16 приведены некоторые функции данного модуля:

Таблица 16. Функции `pygame.display`

Функции	Назначение
<code>pygame.display.init</code>	Инициализирует модуль дисплея
<code>pygame.display.quit</code>	Закрывает модуль дисплея
<code>pygame.display.flip</code>	Полностью обновляет поверхность дисплея
<code>pygame.display.set_mode</code>	Инициализирует окно или экран для отображения
<code>pygame.display.get_surface</code>	Получает ссылку на текущую установленную поверхность отображения
<code>pygame.display.toggle_fullscreen</code>	Переключает между полноэкранным и оконным дисплеями
<code>pygame.display.set_caption</code>	Присваивает заголовок окна
<code>pygame.display.get_caption</code>	Возвращает текущий заголовок окна

PyGame имеет одну поверхность отображения, которая либо содержится в окне, либо работает в полноэкранном режиме. Начало отображения, где $x = 0$ и $y = 0$ – это верхний левый угол экрана. Обе оси увеличиваются в положительном направлении в нижней правой части экрана.

В PyGame может быть активен только один дисплей. Создание нового элемента с помощью функции `pygame.display.set_mode ()` закрывает предыдущий экран.

Для примера создадим окно *screen* (это название окна, которое мы задаем сами) для игры размером 600x500. Для этого используем следующую функцию:

```
pygame.display.set_mode(resolution = (x, y),
flags = 0, depth = 0)
```

Аргументом resolution является пара чисел, представляющих ширину и высоту. **Аргумент flags** – это набор дополнительных опций. **Аргумент depth** – это количество битов, используемых для цвета.

Назовем нашу игру «My First Game». Используя описанную ниже функцию, зададим название окна:

```
pygame.display.set_caption("Some title")
```

Код программы:

```
import pygame
pygame.init()
screen = pygame.display.set_mode((600, 500), 0, 32)
pygame.display.set_caption("My First Game")
```

Основной цикл программы – это ключевой блок, где прописывается логика игры:

```
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            exit()
        screen.blit(background, (0, 0))
    pygame.display.update()
```

Цикл `while` имеет значение `True` в качестве условия, что он будет повторяться непрерывно до тех пор, пока мы не выйдем из него или не заставим его выйти каким-либо другим способом.

Для того, чтобы фоновое изображение полностью покрывало окно, используем следующие действия:

```
screen.blit(фон, (0,0,0))
```

В конце обновляем экран для получения нового изображения:

```
pygame.display.update()
```

Цикл программы состоит из трех основных разделов:

- связь с пользователем: используется для постоянного «прослушивания» пользовательских входов и реагирования на них. Это возможно, когда пользователь использует клавиатуру или мышь;
- реализация игровой логики;
- обновление экрана путем перерисовки сцены и персонажей.

На выходе мы получаем окно заданного размера с названием «My First Game».

Рассмотрим пример использования некоторых функций данного модуля (рис. 32):

```
1) print("get_caption: ", pygame.display.get_
caption())
```

```
2) print("get_surface: ", pygame.display.get_
surface())
```

1. Функция `pygame.display.get_caption()` возвращает заголовок и название иконки для поверхности дисплея. Они часто будут иметь одинаковое значение.

2. Функция `pygame.display.get_surface()` возвращает ссылку на текущую установленную поверхность дисплея. Если режим отображения не был установлен, возвращается `None`.

```
===== RESTART: C:\Users\Admin\Documents\display.get.py =====
pygame 1.9.4
Hello from the pygame community. https://www.pygame.org/
contribute.html
get_caption: ('My First Game', My First Game')
get_surface: <Surface (600x500x32 SW) >
>>>
```

Рис. 32. Использование дополнительных функций модуля

1

Отвечаем на вопросы

1. В чем основное назначение модуля `pygame.display`?
2. Каково назначение функции `pygame.display.set_mode()`?
3. Каково назначение функции `pygame.display.flip()`?
4. Каково назначение функции `pygame.quit()`?

2

Думаем и обсуждаем

Какие две операции должны быть выполнены для использования функции `pygame.display.set_mode()`?

3

Анализируем и сравниваем

Проанализируйте и объясните разницу между системой координат компьютерных экранов и стандартной прямоугольной системой координат.

4

Выполняем в тетради

Запишите в тетрадь команду цикла `for event in pygame.event.get()`:

```
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            exit()
```

5

Выполняем на компьютере

В рассмотренном на уроке примере попробуйте изменить параметры экрана на `(300,200)`, или `(200,300)`: `screen = pygame.display.set_mode ((x, y), 0,32)`. Какую разницу вы заметили?

6

Делимся мыслями

Что нового вы узнали на уроке? Обсудите со своими одноклассниками значимость модуля `pygame.display` и используемых функций этого модуля.

7

Домашнее задание

Подготовьте ответ на вопрос: «Какими командами можно дополнить таблицу функций `pygame.display`?».

§ 28. Знакомство с другими готовыми модулями библиотеки PyGame

Вспомните!

- Как подключается модуль `pygame.display`?
- В чем основное назначение модуля `pygame.display`?
- О функциях модуля `pygame.display`.

Вы узнаете:

- о готовых модулях PyGame;
- о функциях `pygame.surface`, `pygame.image`;
- о функциях `pygame.draw` и др.

Изображение –
Кескін – *Image*

Загрузить – Жүктеу – *Load*

Рисовать – Сурет
сапу – *Draw*

В этом параграфе вы ознакомитесь с некоторыми наиболее распространенными модулями PyGame: `pygame.surface`, `pygame.image`, `pygame.event`, `pygame.draw`, `pygame.mouse`.

Модуль `surface` – объект PyGame для представления изображений.

`Surface.blit (source, dest, area=None, special_flags = 0)` – рисует одно изображение на другом. Рисует заданную поверхность (`source`) поверх базовой (`Surface`), где `dest` – кортеж (`x, y`), начало координат отрисовки, `area (width, height)` – размеры `source` поверхности.

`Surface.get_rect ()` – возвращает кортеж вида (`x, y, width, height`), где `x, y` – координаты левого верхнего угла поверхности (`Surface`), а `width, height` – ее размеры.

Модуль `image` – объект, содержащий функции для загрузки и сохранения изображений. Изображения загружаются как объект `surface`. Класс `surface` позволяет манипулировать ими (рисовать линии, устанавливать пиксели, захватывать области и т. д.).

`Pygame.image.load ()` – загружает новое изображение из файла.

Модуль `event` – объект PyGame для взаимодействия с событиями и очередями.

PyGame обрабатывает все свои сообщения о событиях через очередь событий. Подпрограммы в этом модуле помогают управлять этой очередью событий. Любое событие в PyGame, например, нажатие клавиши, помещается в список, состоящий

из объектов `Event objects`. Все эти «событийные объекты» имеют тип, доступ к которому можно получить с помощью метода `event.type`.

`pygame.event.get ()` – с помощью метода `get ()` можно получить список событий из очереди.

Модуль `mouse` – объект `PyGame` для работы с мышью.

`pygame.mouse.set_pos` – устанавливает положение курсора мыши;

`pygame.mouse.set_cursor` – устанавливает изображение для системного курсора мыши.

Модуль `draw` – объект `PyGame` для рисования фигур.

Разберем этот модуль и его функции более подробно (таблица 17).

Таблица 17. Рисование основных фигур с `pygame.draw`

Функция	Назначение
<code>pygame.draw.rect</code>	Рисование прямоугольника
<code>pygame.draw.polygon</code>	Рисование многоугольника
<code>pygame.draw.circle</code>	Рисование окружности
<code>pygame.draw.ellipse</code>	Рисование эллипса
<code>pygame.draw.arc</code>	Рисование дуги эллипса
<code>pygame.draw.line</code>	Рисование отрезка прямой линии
<code>pygame.draw.lines</code>	Рисование нескольких смежных отрезков
<code>pygame.draw.aaline</code>	Рисование сглаженных линий
<code>pygame.draw.aalines</code>	Рисование последовательности сглаженных линий

Модуль для рисования можно дополнить цветами. Как же на компьютерах воспроизводятся цвета? Компьютерные экраны состоят из пикселей, которые состоят из трех цветов RGB: красного (Red), зеленого (Green) и синего (Blue). RGB – это цветовая модель, представляющая метод получения всех цветов и их оттенков путем смешивания трех главных компонентов в диапазоне между 0 и 255.

Вот таблица цветов, которую мы можем применять в дальнейшем (таблица 18).

Таблица 18. Таблица цветов

Название цвета	Формат цвета RGB	Цвет
BLACK	(0, 0, 0)	
WHITE	(255, 255, 255)	
RED	(255, 0, 0)	
GREEN	(0, 255, 0)	
BLUE	(0, 0, 255)	
CYAN	(0, 255, 255)	
MAGENTA	(255, 0, 255)	
YELLOW	(255, 255, 0)	

В схеме 8 показано слияние цветов и получение нового цвета в формате RGB:

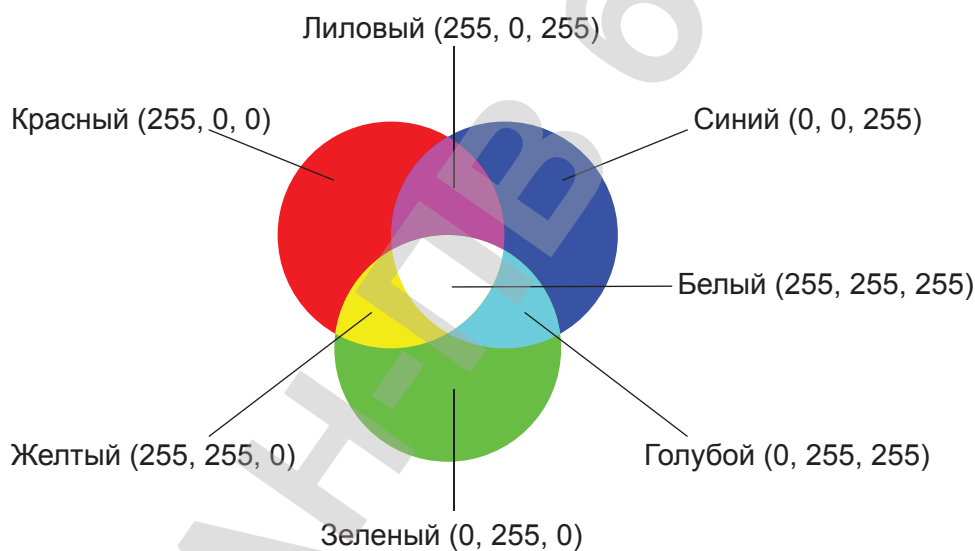


Схема 8. Цветовая модель RGB

Рассмотрим применение функций рисования `pygame.draw` на примере:

Листинг программы:

```
import pygame
from math import pi
```



```
pygame.init()
#определение цветов в формате RGB
BLACK = ( 0, 0, 0)
WHITE = (255, 255, 255)
BLUE = ( 0, 0, 255)
GREEN = ( 0, 255, 0)
RED = (255, 0, 0)

#обозначение высоты и ширины экрана
size = [400, 300]
screen = pygame.display.set_mode(size)

#обозначение названия окна
pygame.display.set_caption("Пример рисования")

#обработка событий
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            exit()

#Очистка экрана и установка белого фона экрана
screen.fill(WHITE)

#Нарисуем на экране черную линию от точки (0,0)
до (40,40) шириной в 4 пикселя
pygame.draw.line(screen, BLACK, [0, 0], [40,40], 4)

#Нарисуем на экране несколько смежных отрезков
шириной в 4 пикселя
pygame.draw.lines(screen, GREEN, False, [[60,
20], [100, 20], [140, 5], [180, 20], [220,20]], 4)

#Нарисуем на экране сплошную линию от точки (0,50)
до (40,90) шириной в 2 пикселя
pygame.draw.aaline(screen, BLACK, [0, 50],[40,
90], 2)

#Нарисуем прямоугольник, начиная с точки (60,60),
с длиной и шириной сторон в 60 пикселей, толщиной
линии в 1 пиксель
```

```
pygame.draw.rect(screen, BLUE, [60,60,60,60], 1)
```

#Нарисуем сплошной залитый прямоугольник, начиная с точки (140,60), с длиной и шириной сторон в 60 пикселей

```
pygame.draw.rect(screen, BLUE, [140, 60,60,60])
```

#Нарисуем окружность с центром в точке (90,170) и радиуса 30 пикселей

```
pygame.draw.circle(screen, BLACK, [90, 170], 30,1)
```

#Нарисуем сплошной залитый круг с центром в точке (170,170) и радиуса 30 пикселей

```
pygame.draw.circle(screen, BLACK, [170, 170], 30)
```

#Нарисуем эллипс, используя прямоугольник

```
pygame.draw.ellipse(screen, RED, [60,220,60,30], 1)
```

#Нарисуем сплошной залитый эллипс, используя прямоугольник

```
pygame.draw.ellipse(screen, RED, [140,220,60,30])
```

#Нарисуем треугольник с помощью функции рисования многоугольника

```
pygame.draw.polygon(screen, BLACK, [[170, 25], [140, 60], [200, 60]], 5)
```

#Нарисуем дугу эллипса, используя радианную меру угла

```
pygame.draw.arc(screen, RED, [210, 75, 150, 125], 0, pi/2, 3)
```

```
pygame.draw.arc(screen, GREEN, [210, 75, 150, 125], pi/2, pi, 3)
```

```
pygame.draw.arc(screen, BLUE, [210, 75, 150, 125], pi, 3*pi/2, 3)
```

```
pygame.draw.arc(screen, BLACK, [210, 75, 150, 125], 3*pi/2, 2*pi, 3)
```

#Обновим экран после использования функций рисования

```
pygame.display.flip()
```

```
#Завершим работу
pygame.quit()
```

Результат программы (рис. 33):



Рис. 33. Результат программы

1

Отвечаем на вопросы

1. Какие готовые модули библиотеки PyGame вы знаете?
2. Опишите функции модулей `pygame.surface`, `pygame.image`, `pygame.event`, `pygame.mouse`.
3. В чем основное назначение модуля `pygame.draw`?
4. Перечислите функции `pygame.draw`.
5. Где применяется формат цвета RGB?

2

Думаем и обсуждаем

Что означает данный код: `pygame.draw.line(screen, green, [x0, y0], [x1, y1], 5)`? Ответьте на вопросы:

1. Что означает `green`?
2. Для чего нужны `[x0, y0]` и `[x1, y1]`?
3. Какую единицу измерения обозначает число 5?

3

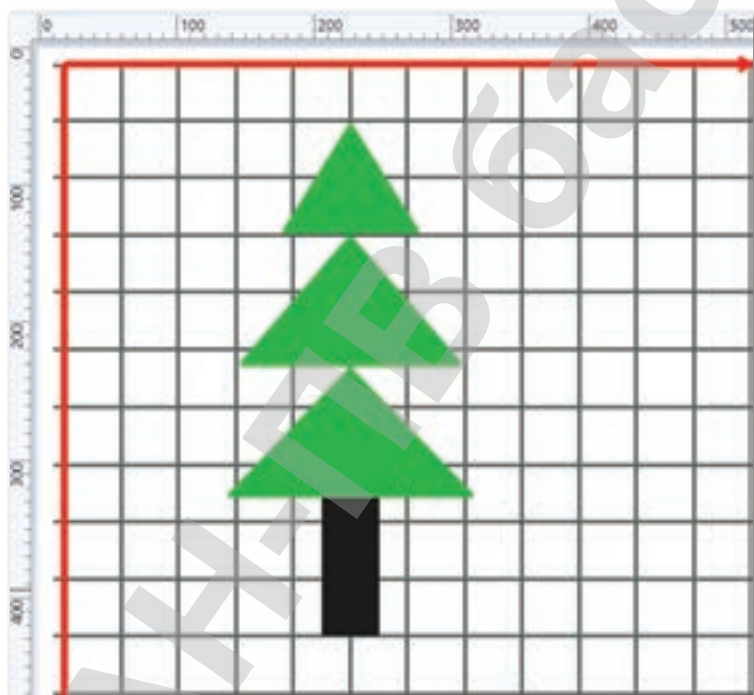
Анализируем и сравниваем

Что происходит во время рисования прямоугольника, окружности или эллипса, если заданная ширина линии равна нулю?

4

Выполняем в тетради

Для того чтобы свободно рисовать на компьютере, нужно произвести следующие действия: начертите в тетради координатную плоскость с определенным масштабом, нарисуйте несколько моделей рисунков (елку, заборчик и др.), используя геометрические фигуры.



5

Выполняем на компьютере

1. Нарисуйте рисунок по данному фрагменту кода программы:

```
pygame.draw.polygon(screen, GREEN,
[[50,100], [0,200], [150,200], [100,100]], 3)
```

Какая фигура у вас получилась?

2. Начертите эллипс по данному фрагменту кода программы:

```
pygame.draw.ellipse(screen, BLACK  
[30, 30, 200, 100], 1).
```

Укажите начальные координаты, длину и ширину, используемого прямоугольника.

6

Делимся мыслями

Что нового вы узнали на уроке? Обсудите со своими одноклассниками, какие готовые модули нам необходимы для создания приложений в Python? Как можно работать с графикой в PyGame? Каковы возможности функций модуля рисования?

7

Домашнее задание 

1. Нарисуйте светофор, используя функции модуля draw и формат цвета RGB.
2. Выполните задания на диске, соответствующие данной теме.

§ 29. Задний фон и персонажи игры

Вспомните!

- О модулях *PyGame*.
- О функциях модулей *pygame.surface*, *pygame.image*.
- О функциях модуля *pygame.draw* и др.

Вы узнаете:

- как заполнить задний фон цветом;
- как загрузить изображение на задний фон;
- как работать с модулем *pygame.time.clock*.

Заполнить – Толтыру – *Fill*

Цвет – Түс – *Color*

Фон – Фон – *Background*

Рассмотрим более подробно, как создавать задний фон игры. Для этого можно заполнить фон любым цветом или загрузить изображение. К примеру, для заливки фона каким-либо цветом используется следующая функция:

```
pygame.Surface.fill(color, rect=None, special_flags=0)
```

Эта функция позволяет заполнить поверхность сплошным цветом. Если прямой аргумент для *rect* не задан, вся поверхность будет заполнена. Аргумент *rect* ограничит заливку определенной областью. Заливка также будет содержаться в области обрезки поверхности. Аргумент *color* может быть задан последовательностью RGB, RGBA или индексом цвета.

Вспомним, что *Surface* также является объектом *PyGame* для представления изображений.

Пример загрузки изображения на задний фон:

```
background_image = 'green.jpg'
import pygame
from pygame.locals import *
from sys import exit
pygame.init()
screen = pygame.display.set_mode((600, 500), 0, 32)
pygame.display.set_caption("My First Game")
background = pygame.image.load(background_image).
convert()
```

```

while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            exit()
    screen.blit(background, (0,0))
    pygame.display.update()

```

Наша игра будет иметь однотонный задний фон зеленого цвета. Для этого нам понадобится изображение зеленого цвета размером не меньше 600x500. В нашем листинге мы дали ему название `background_image`. Для загрузки данного изображения используем модуль `pygame.image`, предназначенный для передачи изображений. В частности, функция `pygame.image.load(filename)` загружает новое изображение из файла. PyGame автоматически определит тип и создаст новый объект `Surface`.

Возвращаемая поверхность будет содержать тот же цветовой формат, цветовой ключ и альфа-прозрачность, что и файл, из которого она получена. Функция `convert()` вызывается без аргументов, чтобы создать копию, которая будет рисоваться на экране быстрее. Она используется для преобразования `pygame.Surface` в тот же формат пикселей, что и тот, который был создан из `pygame.display.set_mode()`. Если вы не вызываете его, преобразование пикселей будет требоваться каждый раз, что очень замедляет процесс (рис. 34).

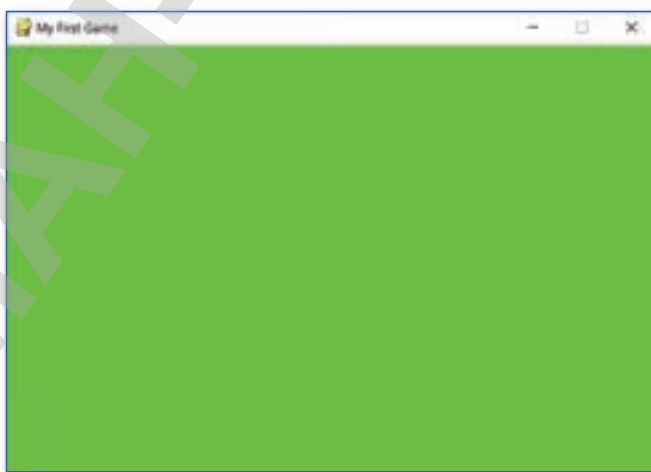


Рис. 34. Результат листинга

Вы можете задать любое изображение для заднего фона вашей игры.

Метод `fill()` для объектов `Surface`:

```
#Нарисовать белый фон на поверхности
windowSurface.fill (WHITE)
```

Мы хотим заполнить всю поверхность, которая хранится в `windowSurface`, белым цветом. Функция `fill()` полностью покрывает поверхность `Surface` цветом, который мы передаем в качестве параметра.

Обратите внимание, что в `Pygame` окно на экране не изменится при вызове метода `fill()` или любой другой функции рисования. Скорее, они изменяют объект `Surface`, и вы должны отобразить новый объект `Surface` на экране с помощью функции `pygame.display.update()`, чтобы увидеть изменения.

Это связано с тем, что изменение объекта `Surface` в памяти компьютере происходит намного быстрее, чем изменение изображения на экране. Поэтому гораздо эффективнее рисовать на экране только после того, как все наши функции рисования будут нарисованы на объекте `Surface`.

Метод `fill()` является не функцией, а методом объектов `pygame.Surface`. Он полностью заполнит весь объект `Surface` любым значением цвета, которое вы передадите для параметра цвета.

Для лучшего понимания работы с другими модулями библиотеки `PyGame` предлагаем рассмотреть простой код, обобщающий изученные ранее модули и их функции. Запустим данный код и разберем полученный результат. Ниже приведено построчное пояснение кода программы.

Листинг программы `HelloWorld.py`:

```
background_image = 'plate.png'
mouse_image = 'apple.jpg'
import pygame
from pygame.locals import *
from sys import exit
pygame.init()
screen = pygame.display.set_mode((640, 480), 0, 32)
pygame.display.set_caption("Hello, World!")
background = pygame.image.load(background_image).
convert()
```



```

mouse_cursor = pygame.image.load(mouse_image).
convert_alpha()
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            screen.blit(background, (0,0))
x, y = pygame.mouse.get_pos()
x-= mouse_cursor.get_width() / 2
y-= mouse_cursor.get_height() / 2
screen.blit(mouse_cursor, (x, y))
pygame.display.update()

```

Результат программы (рис. 35):



Рис. 35. Результат программы

Для данного примера нам нужны два изображения: одно – для использования в качестве фона, а другое – для использования в качестве курсора мыши. В данном случае фон – изображение тарелки, а курсор, который следует за движением курсора мыши, – изображение яблока. Первые две строки листинга задают данные изображения. Далее строки 3–6 импортируют `import pygame` и инициализирует нужные модули `pygame.init()`.

После инициализации нам нужно создать окно. В 7–8 строках создается окно размером 640x480. Задается имя окна "Hello, world!". Далее используется функция `load()` для загрузки изображения для заднего фона и курсора мыши:

```
background = pygame.image.load(background_image).
convert()
mouse_cursor=pygame.image.load(mouse_image).
convert_alpha()
```

Для передвижения картинки с помощью курсора мыши удалим исходные координаты и рассчитываем последние данные:

```
x, y = pygame.mouse.get_pos()
x-= mouse_cursor.get_width()/2
y-= mouse_cursor.get_height()/2
screen.blit(mouse_cursor, (x, y))
```

Также можно использовать функцию `screen.blit()`. Обновим экран для получения нового изображения:

```
pygame.display.update()
```

1

Отвечаем на вопросы

1. Как выполняется заполнение заднего фона игры?
2. Опишите процедуру загрузки изображения на задний фон игры.
3. Изменяются ли размеры окна при использовании метода `fill()`?

2

Думаем и обсуждаем

1. Подумайте, почему `fill(color)` является методом, а не функцией?
2. В чем заключается особенность метода `fill()` для объекта `Surface`?

3

Анализируем и сравниваем

Сравните и проанализируйте особенности заполнения заднего фона: цветом и изображением.

4

Выполняем в тетради

Выпишите в тетрадь изученные на уроке методы и приведенные рекомендации.

5

Выполняем на компьютере

1. В модуле `pygame.display` есть функция `set_caption()`. С ее помощью задается строка, которая устанавливается в качестве заголовка окна. Сделайте так, чтобы каждую секунду заголовок окна изменялся.
2. Выполните задания на диске, соответствующие данной теме.

6

Делимся мыслями

Что нового вы узнали на уроке? Обсудите со своими одноклассниками важность загрузки изображения для фона игры.

7

Домашнее задание

Фон игры можно сделать интереснее, если в качестве обоев использовать разные изображения. Создайте несколько вариантов фоновых обоев и сделайте так, чтобы фоновое изображение менялось.

§ 30. Анимирование персонажей. Создание спрайтов. Загрузка готовых персонажей для игры

Вспомните!

- Как заполнить задний фон цветом?
- Как загрузить изображение на задний фон?

Вы узнаете:

- как создаются спрайты;
- как загружают готовых персонажей для игры;
- как написать программу для создания игры.

Класс – Класс – *Class*

Время – Уақыт – *Time*

Автомобиль – Көлік – *Car*

Спрайты – это графические объекты в игре, чаще всего персонажи. Как правило, спрайты создаются заранее (то есть это готовые на момент запуска игры изображения), а не рисуются, как графические примитивы в программном коде. Ими можно манипулировать и перемещать по экрану как единое целое. Спрайт рассматривается как объект с разными свойствами (ширина, высота, цвет и т. д.) и методами.

Начнем с главного класса игры (назовем его `Game`), который будет управлять другим кодом. В классе `Game` есть функция `__init__` для инициализации игры, а также главный цикл игровой анимации.

Для примера рассмотрим создание простой гоночной игры.

Игрок будет контролировать автомобиль и может передвигаться между линиями, используя клавиши влево и вправо, а также ускоряться и замедляться при помощи клавиш вверх и вниз.

Создайте класс `Car.py`. Данный класс представляет описание объекта *автомобиль*. Он происходит от класса `Sprite` в `PyGame`. Автомобиль мы рисуем как прямоугольник. Однако можно загрузить и подходящее изображение машины, используя функцию `image.load()`.

```

import pygame

WHITE = (255, 255, 255)

class Car(pygame.sprite.Sprite):

    def __init__(self, color, width, height):

        super().__init__()
        self.image = pygame.Surface([width, height])
        self.image.fill(WHITE)
        self.image.set_colorkey(WHITE)

        pygame.draw.rect(self.image, color, [0, 0,
width, height])
        self.rect = self.image.get_rect()

```

Создадим окно размером 600x500. Задаем имя «My First Game». В качестве фонового рисунка загружаем зеленый фон. Рисуем машину размером 20x30 и задаем координаты (200; 450). Далее рисуем дорогу размером 200x450, начальные координаты (100, 0). Также добавляем разделительную полосу посередине дороги.

Листинг программы:

```

background_image = 'green.jpg'

import pygame, random
from Car import Car
from pygame.locals import *
from sys import exit
pygame.init()

RED = (255, 0, 0)
GREEN = (20, 255, 140)
GREY = (210, 210, 210)

screen = pygame.display.set_mode((600, 500), 0, 32)
pygame.display.set_caption("My First Game")

```

```
background = pygame.image.load(background_image).
convert()

cars_list = pygame.sprite.Group()
playerCar = Car(RED, 20, 30)
playerCar.rect.x = 200
playerCar.rect.y = 450

cars_list.add(playerCar)

check = True
clock=pygame.time.Clock()

while check:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            check=False

    all_sprites_list.update()

    #Рисуем на экране
    screen.fill(GREEN)
    #Рисуем дорогу
    pygame.draw.rect(screen, GREY, [100,0, 200, 450])
    #Draw Line painting on the road
    pygame.draw.line(screen, WHITE, [200,0], [200,
450], 5)

    cars_list.draw(screen)

    #Обновление экрана
    pygame.display.flip()
    clock.tick(60)

pygame.quit()
```

Результат программы (рис. 36):



Рис. 36. Результат программы

1

Отвечаем на вопросы

1. Где применяются спрайты?
2. Как осуществляется добавление персонажа на фон при создании игры?
3. Для чего используется класс `Sprite` в PyGame?

2

Думаем и обсуждаем

1. Насколько необходимо добавление персонажа на фон при создании игры?
2. Почему функция запуска игры `__init__` в классе `Game` считается важной?

3

Анализируем и сравниваем

1. В чем эффективность добавления персонажа в игру через спрайт?
2. Проанализируйте преимущества и недостатки программы по добавлению готового персонажа в созданной игре.

4

Выполняем в тетради

Запишите в тетрадь алгоритм создания простой гоночной игры.

5

Выполняем на компьютере

Напишите пример анимации в PyGame в программу. Проанализируйте результат.

```
import sys, pygame
pygame.init()
size = [600, 500]
speed = [0, 2]
white = (255, 255, 255)

screen = pygame.display.set_mode(size)

ball = pygame.image.load("ball.bmp")
ballrect = ball.get_rect()
while 1:
    for event in pygame.event.get():
        if event.type == pygame.QUIT: sys.exit()

        ballrect = ballrect.move(speed)
        if ballrect.left < 0 or ballrect.right > width:
            speed[0] = -speed[0]
        if ballrect.top < 0 or ballrect.bottom > height:
            speed[1] = -speed[1]

    screen.fill(white)
    screen.blit(ball, ballrect)
    pygame.display.flip()
```

6

Делимся мыслями

Что нового вы узнали на уроке? Обсудите со своими одноклассниками, в каких жизненных ситуациях можно применить знания, полученные на уроке? Приведите примеры.

7

Домашнее задание

Используя `pygame.time.clock`, создайте светофор, который меняет цвета.

§ 31–32. Управление и движение спрайтов с клавиатуры

Вспомните!

- Как можно использовать спрайты?
- Что значит добавить персонаж в игру?
- Для чего используется класс *Sprite*?

Вы узнаете:

- как управлять спрайтами при помощи клавиатуры;
- о назначении метода *moveRight*;
- как предотвратить столкновения спрайтов.

Игра – Ойын – *Game*

Переместить –
Жылжыту – *Move*

Клавиатура –
Пернетақта – *Keyboard*

Для перемещения автомобиля влево, вправо, вперед и назад нам нужно добавить методы в наш класс *Car*. Затем мы добавим event handlers в основной цикл программы для реагирования на события нажатия клавиш. Когда игрок использует клавиши со стрелками на клавиатуре, мы будем вызывать наши методы для перемещения автомобиля на экране.

Метод *moveRight()* отвечает за перемещение вправо и принимает два аргумента. Первый аргумент неизвестен и называется *self*. Он относится к текущему объекту. Второй называется *pixels* и относится к количеству пикселей, которые мы будем использовать для перемещения автомобиля. Добавление пикселей к текущей координате объекта *x*:

```
def moveRight(self, pixels):
    self.rect.x += pixels
```

По такому же принципу добавляем метод *moveLeft()*:

```
def moveLeft(self, pixels):
    self.rect.x -= pixels
```

Обновляем наш главный цикл:

```
while check:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
```

```

        check=False
    elif event.type==pygame.KEYDOWN:
        if event.key==pygame.K_x:
            check = False

    keys = pygame.key.get_pressed()
    if keys[pygame.K_LEFT]:
        playerCar.moveLeft(5)
    if keys[pygame.K_RIGHT]:
        playerCar.moveRight(5)

```

При нажатии клавиш **ВПРАВО** и **ВЛЕВО** машина двигается на 5 пикселей по оси *x*.

Для нашей автомобильной гонки мы добавим еще несколько машин. Каждый автомобиль – это объект, который будет создан из класса `Car`.

Добавляем несколько свойств и методов в класс `Car`. Это необходимо сделать для того, чтобы у следующих автомобилей были разные размеры (ширина и высота), цвета и разная скорость.

Затем в нашей основной программе мы создадим четыре дополнительных автомобиля и добавим их в список под названием `all_comming_cars`. Каждому автомобилю будет присвоен свой цвет, скорость и разное начальное положение (координаты *x* и *y*).

Общая скорость игры будет сохранена в переменной, называемой `speed`. Обработчики новых событий будут определять, когда пользователь нажимает клавиши **ВВЕРХ** или **ВНИЗ** на клавиатуре. Они увеличивают или уменьшают переменную `speed` на **0,05**.

Листинг программы:

```

import pygame, random
#Let's import the Car Class
from Car_new import Car
pygame.init()

GREEN = (20, 255, 140)
GREY = (210, 210, 210)
WHITE = (255, 255, 255)
RED = (255, 0, 0)

```

```
MAGENTA = (255, 0, 255)
YELLOW = (255, 255, 0)
CYAN = (0, 255, 255)
BLUE = (100, 100, 255)
speed = 1
colorList = (RED, GREEN, MAGENTA, YELLOW, CYAN, BLUE)
SCREENWIDTH=600
SCREENHEIGHT=600
size = (SCREENWIDTH, SCREENHEIGHT)
screen = pygame.display.set_mode(size)
pygame.display.set_caption("My First Game")
cars_list = pygame.sprite.Group()
playerCar = Car(RED, 60, 80, 70)
playerCar.rect.x = 160
playerCar.rect.y = SCREENHEIGHT - 100
car1 = Car(PURPLE, 60, 80, random.randint(50,100))
car1.rect.x = 60
car1.rect.y = -100
car2 = Car(YELLOW, 60, 80, random.randint(50,100))
car2.rect.x = 160
car2.rect.y = -600
car3 = Car(CYAN, 60, 80, random.randint(50,100))
car3.rect.x = 260
car3.rect.y = -300
car4 = Car(BLUE, 60, 80, random.randint(50,100))
car4.rect.x = 360
car4.rect.y = -900
cars_list.add(playerCar)
cars_list.add(car1)
cars_list.add(car2)
cars_list.add(car3)
cars_list.add(car4)
all_coming_cars = pygame.sprite.Group()
all_coming_cars.add(car1)
all_coming_cars.add(car2)
all_coming_cars.add(car3)
all_coming_cars.add(car4)
check = True
clock=pygame.time.Clock()
```

```

while check:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            check=False
        elif event.type==pygame.KEYDOWN:
            if event.key==pygame.K_x:
                playerCar.moveRight(10)
    keys = pygame.key.get_pressed()
    if keys[pygame.K_LEFT]:
        playerCar.moveLeft(5)
    if keys[pygame.K_RIGHT]:
        playerCar.moveRight(5)
    if keys[pygame.K_UP]:
        speed += 0.05
    if keys[pygame.K_DOWN]:
        speed -= 0.05
    for car in all_coming_cars:
        car.moveForward(speed)
        if car.rect.y > SCREENHEIGHT:
            car.changeSpeed(random.randint(50,100))
            car.repaint(random.choice(colorList))
            car.rect.y = -200
    car_collision_list=pygame.sprite.spritecollide
(playerCar,all_coming_cars,False)
    for car in car_collision_list:
        print("Car crash!")
        check=False
    cars_list.update()
    screen.fill(GREEN)
    pygame.draw.rect(screen, GREY, [40,0,
400,SCREENHEIGHT])
    pygame.draw.line(screen, WHITE, [140,0],
[140,SCREENHEIGHT],5)
    pygame.draw.line(screen, WHITE, [240,0],
[240,SCREENHEIGHT],5)
    pygame.draw.line(screen, WHITE, [340,0],
[340,SCREENHEIGHT],5)

    cars_list.draw(screen)

```

```
pygame.display.flip()
clock.tick(60)

pygame.quit()

Car_new.py
import pygame
WHITE = (255, 255, 255)

class Car(pygame.sprite.Sprite):

    def __init__(self, color, width, height, speed):

        super().__init__()

        self.image = pygame.Surface([width, height])
        self.image.fill(WHITE)
        self.image.set_colorkey(WHITE)

        self.width=width
        self.height=height
        self.color = color
        self.speed = speed

        pygame.draw.rect(self.image, self.color,
[0, 0, self.width, self.height])

        self.rect = self.image.get_rect()

    def moveRight(self, pixels):
self.rect.x += pixels

    def moveLeft(self, pixels):
self.rect.x -= pixels

    def moveForward(self, speed):
self.rect.y += self.speed * speed / 20

    def moveBackward(self, speed):
self.rect.y -= self.speed * speed / 20
```

```

def changeSpeed(self, speed):
    self.speed = speed

def repaint(self, color):
    self.color = color
    pygame.draw.rect(self.image, self.color, [0,
0, self.width, self.height])

```

Результат программы (рис. 37):



Рис. 37. Результат программы

Следующее дополнение к нашей игре используется в большинстве двухмерных игр: цель состоит в том, чтобы проверить, сталкивается ли спрайт с другими спрайтами (в данном случае это автомобили, сохраненные в списке `all_coming_cars`). Если произошло столкновение, выводится запись, что игра окончена (рис. 38):



Рис. 38. Результат программы

1

Отвечаем на вопросы

1. Что такое управление спрайтами через обработчиков событий нажатия клавиш?
2. Что нужно учитывать при программировании передвижения автомобиля влево, вправо, вперед и назад?
3. В чем особенность передвижения спрайтов?

2

Думаем и обсуждаем

1. Насколько необходимо добавление обработчиков события `event handlers`?
2. Как влияют столкновения спрайтов на ход игры?

3

Анализируем и сравниваем

При использовании клавиатуры проанализируйте, как будет себя вести ваш объект при движении и управлении за пределами игровой площадки экрана.

4

Выполняем в тетради

Запишите в тетрадь методы, вызываемые для передвижения автомобиля по экрану.



5

Выполняем на компьютере

Создайте программу, которая будет выполнять следующие команды:

1. Создайте как минимум два разных элемента, которые рисуют объект на экране. Не рисуйте те же графические примитивы, которые мы уже делали ранее. Создайте свой уникальный предмет: корабль, солнце, дом и др. Если у вас есть рисунок, самостоятельно созданный на предыдущих занятиях, то можно использовать его.
2. Мы говорили о перемещении графики с помощью клавиатуры. Используйте этот метод для управления двумя различными элементами на экране.

6

Делимся мыслями

Обсудите со своими одноклассниками обработчики событий, с которыми вы ознакомились на уроке. Каковы последствия столкновений при неправильном управлении с помощью клавиатуры?

7

Домашнее задание

Попрактикуйтесь в создании игр, используя знания по движению и управлению спрайтов с помощью клавиатуры.

§ 33–34. Практикум. Программирование условий. Разработка игры по готовому сценарию

Любая игра PyGame базируется на основном цикле программы.

Структура главного цикла

1. Начало (инициализация игры, запускается один раз при запуске игры):
 - импортирование и инициализация библиотеки PyGame;
 - задание значений и открытие нового окна;
 - инициализирование персонажей (спрайты) игры.
2. Основной цикл `while` (повторяется пока удовлетворяет условие):
 - взаимодействие с пользователем (с помощью клавиатуры, мыши и т.д.);
 - логика игры;
 - обновление экрана (перерисовка персонажей и фона).
3. Конец.

Напишите игру по данному сценарию: в верхней части окна имеется несколько рядов кирпичей.

Цель игры – управляя мячом, разрушить все кирпичи. За каждый уничтоженный кирпич начисляется очко. При падении мяча игрок теряет одну жизнь. Игрок имеет три жизни. [Алгоритм: <http://codeNtronix.com>]. Полный листинг приведен ниже.

```
import sys
import pygame

SCREEN_SIZE = 640, 480

#Размеры объектов
BRICK_WIDTH = 60
BRICK_HEIGHT = 15
PADDLE_WIDTH = 60
PADDLE_HEIGHT = 12
BALL_DIAMETER = 16
BALL_RADIUS = BALL_DIAMETER // 2

MAX_PADDLE_X = SCREEN_SIZE[0] - PADDLE_WIDTH
```

```

MAX_BALL_X = SCREEN_SIZE[0] - BALL_DIAMETER
MAX_BALL_Y = SCREEN_SIZE[1] - BALL_DIAMETER
PADDLE_Y = SCREEN_SIZE[1] - PADDLE_HEIGHT - 10
BLACK = (0,0,0)
WHITE = (255,255,255)
BLUE = (0,0,255)
BRICK_COLOR = (200,200,0)

#состояния игры
STATE_BALL_IN_PADDLE = 0
STATE_PLAYING = 1
STATE_WON = 2
STATE_GAME_OVER = 3

class Bricka:

    def __init__(self):
        pygame.init()

        self.screen = pygame.display.set_mode(SCREEN_SIZE)
        pygame.display.set_caption("IGRA")

        self.clock = pygame.time.Clock()

        if pygame.font:
            self.font = pygame.font.Font(None,30)
        else:
            self.font = None

        self.init_game()

    def init_game(self):
        self.lives = 3
        self.score = 0
        self.state = STATE_BALL_IN_PADDLE

        self.paddle = pygame.Rect(300,PADDLE_Y,PADDLE_
WIDTH,PADDLE_HEIGHT)
        self.ball = pygame.Rect(300,PADDLE_Y - BALL_
DIAMETER,BALL_DIAMETER,BALL_DIAMETER)

```

```

self.ball_vel = [5,-5]
self.create_bricks()

def create_bricks(self):
    y_ofs = 35
    self.bricks = []
    for i in range(7):
        x_ofs = 35
        for j in range(8):
self.bricks.append(pygame.Rect(x_ofs,y_ofs,BRICK_
WIDTH,BRICK_HEIGHT))
            x_ofs += BRICK_WIDTH + 10
            y_ofs += BRICK_HEIGHT + 5

def draw_bricks(self):
    for brick in self.bricks:
        pygame.draw.rect(self.screen, BRICK_COLOR,
brick)

def check_input(self):
    keys = pygame.key.get_pressed()

    if keys[pygame.K_LEFT]:
        self.paddle.left -= 5
        if self.paddle.left < 0:
            self.paddle.left = 0

    if keys[pygame.K_RIGHT]:
        self.paddle.left += 5
        if self.paddle.left > MAX_PADDLE_X:
            self.paddle.left = MAX_PADDLE_X

    if keys[pygame.K_SPACE] and self.state ==
STATE_BALL_IN_PADDLE:
        self.ball_vel = [5,-5]
        self.state = STATE_PLAYING
    elif keys[pygame.K_RETURN] and (self.state ==
STATE_GAME_OVER or self.state == STATE_WON):
        self.init_game()

```

```
def move_ball(self):
    self.ball.left += self.ball_vel[0]
    self.ball.top += self.ball_vel[1]

    if self.ball.left <= 0:
        self.ball.left = 0
        self.ball_vel[0] = -self.ball_vel[0]
    elif self.ball.left >= MAX_BALL_X:
        self.ball.left = MAX_BALL_X
        self.ball_vel[0] = -self.ball_vel[0]

    if self.ball.top < 0:
        self.ball.top = 0
        self.ball_vel[1] = -self.ball_vel[1]
    elif self.ball.top >= MAX_BALL_Y:
        self.ball.top = MAX_BALL_Y
        self.ball_vel[1] = -self.ball_vel[1]

def handle_collisions(self):
    for brick in self.bricks:
        if self.ball.colliderect(brick):
            self.score += 3
            self.ball_vel[1] = -self.ball_vel[1]
            self.bricks.remove(brick)
            break

    if len(self.bricks) == 0:
        self.state = STATE_WON

    if self.ball.colliderect(self.paddle):
        self.ball.top = PADDLE_Y - BALL_DIAMETER
        self.ball_vel[1] = -self.ball_vel[1]
    elif self.ball.top > self.paddle.top:
        self.lives -= 1
        if self.lives > 0:
            self.state = STATE_BALL_IN_PADDLE
        else:
            self.state = STATE_GAME_OVER

def show_stats(self):
```

```

    if self.font:
        font_surface = self.font.render("SCORE: " + str
(self.score) + " LIVES: " + str(self.lives), False, WHITE)
        self.screen.blit(font_surface, (205,5))

def show_message(self,message):
    if self.font:
        size = self.font.size(message)
        font_surface = self.font.render(message, False,
WHITE)
        x = (SCREEN_SIZE[0] - size[0]) / 2
        y = (SCREEN_SIZE[1] - size[1]) / 2
        self.screen.blit(font_surface, (x,y))

def run(self):
    while 1:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                sys.exit()

        self.clock.tick(50)
        self.screen.fill(BLACK)
        self.check_input()
        if self.state == STATE_PLAYING:
            self.move_ball()
            self.handle_collisions()
        elif self.state == STATE_BALL_IN_PADDLE:
            self.ball.left = self.paddle.left + self.
paddle.width / 2
            self.ball.top = self.paddle.top - self.ball.
height
            self.show_message("PRESS SPACE TO LAUNCH
THE BALL")
        elif self.state == STATE_GAME_OVER:
            self.show_message("GAME OVER. PRESS ENTER
TO PLAY AGAIN")
        elif self.state == STATE_WON:
            self.show_message("YOU WON! PRESS ENTER TO PLAY
AGAIN")
            self.draw_bricks()

```

```

        pygame.draw.rect(self.screen, BLUE,
self.paddle)

        pygame.draw.circle(self.screen, WHITE,
(self.ball.left + BALL_RADIUS, self.ball.top + BALL_
RADIUS), BALL_RADIUS)

        self.show_stats()

        pygame.display.flip()

if __name__ == "__main__":
    Bricka().run()

```

Результат программы (рис. 39):



Рис. 39. Результат программы

Для реализации алгоритма подсчета очков вводится переменная $score=0$, а также переменная $lives = 3$, которая отвечает за количество оставшихся жизней. Переменная $state$ показывает, в каком состоянии находится игра. Всего определено 4 состояния:

STATE_BALL_IN_PADDLE = 0 – мяч на доске.
STATE_PLAYING = 1 – идет игра
STATE_WON = 2 – выигрыш
STATE_GAME_OVER = 3 – конец игры

```

self.lives = 3
self.score = 0
self.state = STATE_BALL_IN_PADDLE

```

Данная функция производит подсчет очков. Если мяч попадает в кирпич, к общему количеству очков прибавляется 3 очка, и кирпич удаляется с поля игры.

Если количество кирпичей на экране равно 0, состояние игры переходит в состояние «выигрыш» и объявляется, что игрок выиграл.

В случае падения мяча с доски проверяется количество жизней. Если переменная `lives` больше нуля, игра переходит в состояние «мяч на доске» и игра возобновляется. Если же жизней больше нет, объявляется «конец игры».

```

def handle_collisions(self):
    for brick in self.bricks:
        if self.ball.colliderect(brick):
            self.score += 3
            self.ball_vel[1] = -self.ball_vel[1]
            self.bricks.remove(brick)
            break

    if len(self.bricks) == 0:
        self.state = STATE_WON

if self.ball.colliderect(self.paddle):
    self.ball.top = PADDLE_Y - BALL_DIAMETER
    self.ball_vel[1] = -self.ball_vel[1]
elif self.ball.top > self.paddle.top:
    self.lives -= 1
    if self.lives > 0:
        self.state = STATE_BALL_IN_PADDLE
    else:
        self.state = STATE_GAME_OVER

```

ИТОГОВЫЕ ТЕСТОВЫЕ ЗАДАНИЯ К РАЗДЕЛУ V

1. ... – загрузка и воспроизведение звуков.

- A) `pygame.mixer.`
- B) `pygame.cdrom.`
- C) `pygame.image.`
- D) `pygame.display.`
- E) `pygame.key.`

2. ... – рисование фигур и линий.

- A) `pygame.key.`
- B) `pygame.cdrom.`
- C) `pygame.image.`
- D) `pygame.display.`
- E) `pygame.draw.`

3. ... – управление и взаимодействие событиями.

- A) `pygame.display.`
- B) `pygame.cdrom.`
- C) `pygame.image.`
- D) `pygame.event.`
- E) `pygame.key.`

4. ... – преобразование поверхностей.

- A) `pygame.image.`
- B) `pygame.cdrom.`
- C) `pygame.transform.`
- D) `pygame.display.`
- E) `pygame.key.`

5. ... – управление временем и частотой кадров.

- A) `pygame.key.`
- B) `pygame.cdrom.`
- C) `pygame.image.`
- D) `pygame.display.`
- E) `pygame.time.`

6. `pygame.rect` – ...

- A) рисование фигур, линий и точек.
- B) загрузка значка курсора.

- C) реакция на нажатие клавиши на клавиатуре.
- D) управление изображением и экраном.
- E) управление прямоугольной областью.

7. `pygame.surface` – ...

- A) загрузка значка курсора.
- B) управление изображением и экраном.
- C) реакция на нажатие клавиши на клавиатуре.
- D) управление прямоугольной областью.
- E) рисование фигур, линий и точек.

8. ... – инициализация окна или экрана для отображения.

- A) `pygame.display.set_caption`.
- B) `pygame.display.get_surface`.
- C) `pygame.display.set_mode`.
- D) `pygame.display.get_caption`.
- E) `pygame.display.quit`.

9. ... – ссылка на текущую установленную поверхность отображения.

- A) `pygame.display.set_mode`.
- B) `pygame.display.get_surface`.
- C) `pygame.display.set_caption`.
- D) `pygame.display.get_caption`.
- E) `pygame.display.quit`.

10. ... – переключение между полноэкранным и оконным дисплеями.

- A) `pygame.display.toggle_fullscreen`.
- B) `pygame.display.get_surface`.
- C) `pygame.display.set_caption`.
- D) `pygame.display.get_caption`.
- E) `pygame.display.quit`.

11. ... – установка текущего заголовка окна.

- A) `pygame.display.toggle_fullscreen`.
- B) `pygame.display.get_surface`.
- C) `pygame.display.set_caption`.
- D) `pygame.display.get_caption`.
- E) `pygame.display.quit`.

12. `pygame.display.get_caption` – это ...

- А) настройка модуля дисплей.
- В) полное обновление экрана дисплея.
- С) получение текущего заголовка окна.
- Д) ссылка на текущую установленную поверхность отображения.
- Е) установка текущего заголовка окна.

13. `pygame.display.get_surface` – это ...

- А) настройка модуля дисплей.
- В) полное обновление экрана дисплея.
- С) получение текущего заголовка окна.
- Д) ссылка на текущую установленную поверхность отображения.
- Е) установка текущего заголовка окна.

14. `pygame.display.flip` – это ...

- А) настройка модуля дисплей.
- В) полное обновление экрана дисплея.
- С) получение текущего заголовка окна.
- Д) ссылка на текущую установленную поверхность отображения.
- Е) установка текущего заголовка окна.

15. Спрайт – это ...

- А) настройка модуля дисплей.
- В) графические объекты в компьютерной графике, персонажи.
- С) получение текущего заголовка окна.
- Д) управление движением вправо.
- Е) установка текущего заголовка окна.

Глоссарий

Аккаунт – это учетная запись, представляющая собой набор данных о пользователе.

Анимация – поочередный вывод на экран набора изображений с большой скоростью, создающей иллюзию движения.

Базы данных (БД) – информационная модель, позволяющая упорядоченно хранить данные о группе объектов, обладающих одинаковым набором свойств.

Библиотека – это собрание кода функций и классов. В Python термином, используемым для описания библиотеки с кодом, является модуль.

Библиотека PyGame – это модуль для языка программирования Python, предназначенный для разработки 2D игр и других мультимедийных приложений.

Биоритмы – это периодически повторяющиеся изменения характера и интенсивности биологических процессов и явлений в живом организме.

Виртуальная машина – это программа, используемая для работы компонентов компьютера, в результате которой создается виртуальная среда, позволяющая установить и запустить любую операционную систему.

Гибридное облако – комбинация двух или более видов облаков (частного, публичного, общественного).

Достоверность информации – показатель качества информации, означающий ее полноту и общую точность. Информация, которая не вызывает абсолютно никаких сомнений, является реальной, подлинной.

Жесткий диск – основной накопитель данных в большинстве компьютеров, запоминающее устройство (устройство хранения информации) произвольного доступа.

Звуковая карта – дополнительное оборудование компьютера или ноутбука, позволяющее обрабатывать звук.

Изображение – картинка на экране компьютера.

Импортирование – операция, которая делает модуль доступным для использования в программе.

Инициализация – установка начального состояния объекта (свойств объекта при его создании).

Информация – сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состояниях, которые

воспринимаются в процессе жизнедеятельности и работы живыми организмами, управляющими машинами и др.

Кадр – одно из изображений, составляющих анимацию.

Ключевое поле – одно или несколько полей, комбинация значений которых однозначно определяет каждую запись в таблице.

Конфигурация – определенный набор комплектующих, основанный на назначении, количестве и ключевых характеристиках компонентов.

Массив – набор однотипных компонентов, расположенных в памяти непосредственно друг за другом; доступ к ним осуществляется по индексу.

Мастер – программный модуль, инструмент, помогающий пользователю быстро и наглядно осуществить те или иные настройки.

Материнская плата – это главная микросхема компьютера, на которой установлены все основные детали и узлы.

Модули – процедуры на языке VBA.

Моделирование – процесс создания и использования моделей для решения практических задач.

Модель – это определенный новый образец, показывающий важнейшие отличия объекта, явления или процесса.

Облачное хранилище файлов – это место хранения информации, доступной в сети Интернет, и расположенное на оборудовании поставщика, оказывающего услуги облачных служб.

Облачные технологии – технология распределенной обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователю как интернет-сервис.

Общественное облако – вид инфраструктуры, предназначенный для использования конкретным сообществом потребителей, имеющих общие задачи.

Одномерный массив – фиксированное количество элементов одного типа, объединенных одним именем, причем каждый элемент имеет свой уникальный номер, и номера элементов идут подряд.

Оперативная память – относительно быстрая энергозависимая память компьютера с произвольным доступом, в которой осуществляются большинство операций обмена данными между устройствами.

Пиксель – самая маленькая точка на экране, которую компьютер способен отобразить.

Порядковый номер – это индекс элемента массива. Элементы массива располагаются по порядку один за другим.

Прозрачность – часть изображения, которая не выводится на экран и поэтому не заслоняет графические объекты заднего плана.

Программное обеспечение – программа или множество программ, используемых для управления компьютером.

Публичное облако – инфраструктура, предназначенная для свободной работы с ресурсами всех пользователей, имеющих доступ к Интернету.

Серверное программное обеспечение – программный компонент вычислительной системы, выполняющий сервисные (обслуживающие) функции по запросу клиента, предоставляя ему доступ к определенным ресурсам или услугам.

Синтаксис – набор правил по составлению текстов программ на языке программирования.

Событие – нечто, происходящее во время выполнения программы.

Сортировка – упорядочение записей по значениям одного из полей.

Спрайт – персонаж или иной графический объект в компьютерной игре.

Столкновение – ситуация в компьютерных играх, когда один персонаж игры сталкивается с другим персонажем или каким-либо экранным объектом.

Фильтр – вывод нужных пользователю данных в результате созданного им запроса.

Флейм – это эмоциональные замечания, часто высказанные без учета мнения других участников разговора.

Форма – объект, предназначенный для ввода и отображения данных в базах данных.

Функциональность – способность программного обеспечения выполнять набор функций (действий), удовлетворяющих заданным или подразумеваемым потребностям пользователей.

Частное облако – это инфраструктура, предназначенная только для работы отдельных частных лиц (представителей частной организации, компании) и их потребителей.

Эргономичность – характеристики программного обеспечения, которые позволяют минимизировать усилия пользователя

по подготовке исходных данных, применению программного обеспечения и оценке полученных результатов.

Этикет – установленный порядок поведения, форм обслуживания.

Эффективность – соотношение уровня услуг, предоставляемых пользователю программным обеспечением в заданных условиях, и объема используемых для этого ресурсов.

CPU-Z (Central Processing Unit-Z) – программа, определяющая техническую характеристику составляющих персонального компьютера.

EVEREST Ultimate Edition – программа, предназначенная для диагностики, тестирования и настройки на оптимальную работу аппаратных и программных средств компьютера.

Google Docs – удобный и простой онлайн-офис Google, предназначенный для бесплатной работы пользователя с документами.

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода.

Windows Live SkyDrive (OneDrive) – служба для совместной работы онлайн с документами Microsoft Office и его web-версией Microsoft Office Web Apps, функционирующими в web-браузерах

Использованная литература

1. Безручко В.Т. Информатика (курс лекций): Учебное пособие. – М.: ИД. «Форум»: ИНФРА – М. 2014. – 432 с.
2. Семакин И. Информатика. Базовый курс. 7–9 классы / И. Семакин, Л. Залогова, С. Русаков. – М.: Бином. Лаборатория знаний, 2015. – 390 с.
3. Хахаев И.А. – Практикум по алгоритмизации и программированию по Python 2014. – 240 с.
4. Босова Л.Л., Босова А.Ю., Информатика: учебник для 9 класса. М.: БИНОМ. Лаборатория знаний, 2016. – 214 с.
5. Соболев Б.В. и др. Информатика. Учебник. / 3-е изд., доп. и перераб. – Ростов н/Д: Феникс, 2007. – 446 с.
6. Казиев В.М. Информатика в примерах и задачах: кн. для уч-ся 10–11 кл. / В. М. Казиев. – М. : Просвещение, 2007. – 304 с.: ил. – (Профильная школа).
7. Информатика: программы для общеобразоват. учреждений: 2–11 кл. / сост. М.Н. Бородин. – 5-е изд., испр. – М.: БИНОМ. Лаборатория знаний, 2008. – 463 с.
8. Златопольский Д.М. Основы программирования на языке Python. М.: ДМК Пресс, 2017. – 284 с.: ил.

Электронные ресурсы

www.intuit.ru

<http://school-collection.edu.ru/>

<https://infourok.ru/>

<http://www.klyaksa.net/>

<https://www.yaklass.ru/>

<https://pythontutor.ru/>

<https://pythonworld.ru/bookshop>

<https://comprost.ru/komplektuiushchie/sborka-pc/>

https://5urokov.ru/bosova_9_uch/1_6

http://www.booksgid.com/operating_systems_databases/

<https://www.tiensmed.ru/programmer1.html>

Содержание

Введение	4
РАЗДЕЛ . РАБОТА С ИНФОРМАЦИЕЙ	5
§ 1. Свойства информации	6
§ 2. Критическая оценка рисков, связанных с продолжительным использованием компьютера	10
§ 3. Совместная работа с документами	15
§ 4. Сетевой этикет	24
РАЗДЕЛ II. ВЫБИРАЕМ КОМПЬЮТЕР	31
§ 5. Конфигурация компьютера	32
§ 6. Практикум. Конфигурация компьютера	38
§ 7. Выбор программного обеспечения	40
§ 8. Расчет стоимости компьютера	45
РАЗДЕЛ III. БАЗЫ ДАННЫХ	57
§ 9. База данных: основные понятия и определения	58
§ 10. Таблица. Понятие поле и записи	62
§ 11. База данных. Создание базы данных в электронных таблицах	65
§ 12. Методы поиска информации	69
§ 13. Сортировка и фильтрация данных	74
§ 14. Практикум. Работа с базой данных. Создание базы данных «Предметная олимпиада»	78
§ 15. Практикум. Работа по поиску и фильтрации данных в базе данных	81
РАЗДЕЛ IV. СПРОГРАММИРОВАНИЕ АЛГОРИТМОВ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON	85
§ 16–17. Одномерные массивы	86
§ 18–19. Поиск элементов с заданными свойствами	93
§ 20. Практикум. Перестановка элементов массива	100
§ 21. Практикум. Сортировка элементов массива	103
§ 22. Практикум. Удаление и вставка элемента	107
§ 23. Двумерные массивы	111
§ 24. Практикум. Вставка и удаление элемента в двумерном массиве	114
§ 25. Практикум. Задачи на двумерные массивы	116

РАЗДЕЛ V. СОЗДАНИЕ 2D ИГРЫ

НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON	121
§ 26. Библиотека PyGame. Подключение библиотеки	122
§ 27. Использование модулей библиотеки PyGame для создания окна для игры	127
§ 28. Знакомство с другими готовыми модулями библиотеки PyGame	131
§ 29. Задний фон и персонажи игры	139
§ 30. Анимирование персонажей. Создание спрайтов. Загрузка готовых персонажей для игры	145
§ 31–32. Управление и движение спрайтов с клавиатуры	150
§ 33–34. Практикум. Программирование условий. Разработка игры по готовому сценарию	158
Глоссарий	168
Использованная литература	172

Учебное издание

**Гульназ Ибрагимовна Салгараева
Кульзия Акрашевна Калымова
Жанна Абдыхамитовна Орынтаева**

ИНФОРМАТИКА

Учебник для 9 класса общеобразовательной школы

Главный редактор К. Караева
Редактор Н. Хасенова, Г. Маликова
Технический редактор В. Бондарев
Художественный редактор Е. Мельникова
Бильд-редактор Ш. Есенкулова
Художник-оформитель О. Подопригора
Дизайн обложки В. Бондарев
Дизайн О. Подопригора
Верстка Л. Костина, С. Сулейменова
Г. Илишева, Т. Макарова



Внимание

При необходимости вы всегда сможете найти CD с электронным приложением на сайте *arman-pv.kz* и загрузить его на свой компьютер для дальнейшей работы

По вопросам приобретения обращайтесь по следующим адресам:

г. Нур-Султан, м-н 4, д.2, кв.55

Тел.: 8 (7172) 92-50-50, 92-50-54. E-mail: astana@arman-pv.kz

г. Алматы, м-н Аксай 1А, д.28Б

Тел./факс: 8 (727) 316-06-30, 316-06-31. E-mail: info@arman-pv.kz

Книжный магазин «Арман-ПВ»

г. Алматы, ул. Алтынсарина, д.87

Тел: 8 (727) 303-94-43.

Сдано в набор 07.05.18. Подписано в печать 06.06.19. Формат 70 x 100¹/₁₆.

Бумага офсетная. Гарнитура «ММ Мектептік» Печать офсетная.

Объем 14,19 усл.печ.л. Тираж 37000 экз.

Артикул 809-007-001р-19