

Н. С. Кольева, Е. В. Шевчук, Э. Б. Ержанов

ИНФОРМАТИКА

Учебник для 9 классов
общеобразовательных школ

*Утверждено Министерством
образования и науки Республики Казахстан*



Алматы "Мектеп" 2019

УДК 373.167.1
ББК 32.973я72
К62

Кольева Н. С., и др.
К62 **Информатика.** Учебник для 9 классов общеобразовательных школ. /
Н. С. Кольева, Е. В. Шевчук, Э. Б. Ержанов — Алматы: Мектеп, 2019 — 112 с.
ISBN 978—601—07—1261—4

К $\frac{4306020500-060}{404(05)-19}$ 23(1)—19

УДК 373.167.1
ББК 32.973я72

© Кольева Н. С., Шевчук Е. В.,
Ержанов Э. Б., 2019
© Издательство "Мектеп",
художественное оформление, 2019
Все права защищены
Имущественные права на издание
принадлежат издательству "Мектеп"

ISBN 978—601—07—1261—4

ВВЕДЕНИЕ

Наше общество живет в период, характеризующийся небывалым ростом объема информации. Применение современной компьютерной техники дает возможность переложить трудоемкие, рутинные операции на автоматические или автоматизированные устройства, которые могут работать со скоростью, превышающей скорость обработки информации человеком в миллионы раз.

Современный специалист должен быть готов к работе в новых условиях высокоинформатизированного общества, а значит, должен быть компетентным в области информатики и владеть практическими навыками при работе с компьютерной техникой и современными информационными технологиями, знать основные принципы работы с информационными системами, уметь оценивать точность и полноту информации, влияющей на принятие управленческих решений.

Этот учебник вам в этом поможет.

В учебнике используются условные обозначения, которые помогут вам ориентироваться в параграфе.

Желаем вам успехов!

Условные обозначения



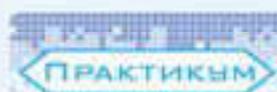
— сведения, которые надо знать.



— проверьте себя и убедитесь, что теперь знаете.



— контрольные вопросы и задания.



— практическая работа за компьютером.

Уровень А — учебно-познавательные задания;

Уровень В — частично-исследовательские задания;

Уровень С — задания, требующие самостоятельной творческой работы.

РАЗДЕЛ

1

Работа с информацией

Из данного раздела вы узнаете:

- ▶ свойства информации;
- ▶ облачные технологии Google Диск для работы с совместными электронными документами;
- ▶ что такое аккаунт.

Вы научитесь:

- ▶ определять свойства информации;
- ▶ создавать аккаунт в Google;
- ▶ создавать собственные слайды в общей коллективной презентации с использованием сервиса Google;
- ▶ работать с совместными электронными документами;
- ▶ управлять настройками доступа к документам, хранящимся в облачных хранилищах;
- ▶ рассуждать о последствиях нарушений этических и правовых норм в сети.

§1

Свойства информации

Вы научитесь:

- ▶ определять свойства информации (актуальность, точность, достоверность, ценность);
- ▶ критически оценивать риски, связанные с продолжительным использованием компьютеров.

Ключевые понятия:

- ▶ информация
- ▶ объективность информации
- ▶ точность информации
- ▶ ценность информации
- ▶ актуальность информации
- ▶ достоверность информации
- ▶ понятность информации
- ▶ репрезентативность
- ▶ адекватность

В повседневной жизни от информации зависят экономическое развитие общества, жизнь и здоровье людей. В любой ситуации приходится анализировать информацию, чтобы оценить, насколько она понятна, актуальна и полезна для окружающих, насколько достоверны содержащиеся в ней сведения.



Любая ли информация помогает нам в этом? Подумайте, почему одни люди сразу реагируют на определенную информацию, а другие остаются к ней равнодушными?

Значимость тех или иных свойств информации определяется конкретной ситуацией. Люди по-разному реагируют на одну и ту же информацию. Например, когда по телевидению передают курс обмена иностранной валюты, то некоторые даже не обратят на это внимание, а некоторые поспешат в банк для ее обмена. А происходит это потому, что информация обладает свойствами, которые представляют различную ценность для разных людей. В одних ситуациях важны актуальность и достоверность информации (например, о курсе валют). В других ситуациях важную роль играют такие свойства, как доступность и понятность (например, инструкции по использованию какого-либо устройства печатают на нескольких языках.)

Эффективность использования информации связана с такими ее свойствами, как актуальность, доступность (понятность), достоверность и полнота.

Информация для одного человека может оказаться ценной (полезной), а для другого — бесполезной (не несет никакой ценной информации). Например, информация о погоде в Париже для многих будет бесполезна, но для людей, которые вылетают в Париж, эта информация будет ценной. **Ценность** (полезность) — это одно из свойств информации.

Другим не менее важным свойством информации является ее **актуальность**. Актуальность информации определяется тем, насколько важны для человека или общества данные сведения, могут ли они быть использованы в конкретной ситуации для решения проблемы.

Это свойство во многом зависит от интервала времени, прошедшего с момента появления данной информации, а также от того, насколько быстро изменяется ситуация.

Таким образом, своевременность информации предполагает ее поступление не позже заранее назначенного момента времени, согласованного со временем решения поставленной задачи.

Только актуальная, вовремя полученная информация может принести пользу людям. Недаром прогноз погоды сообщают накануне, а не в тот же день.

Доступность информации обеспечивается за счет ее преобразования в понятную форму. При этом одну и ту же информацию можно представить в разной форме — в зависимости от ее получателя.

Информация становится понятной, если она выражена в той форме и тем языком, который воспринимает тот, кому она предназначена.

Например, учебник по физике для 10 классов совершенно непонятен восьмикласснику, так как в нем содержатся незнакомые термины и формулы, а учебник по физике для 8 классов содержит доступную информацию для восьмиклассника, но десятиклассник не найдет в нем ничего нового.

В книжном магазине вы найдете отдел детской литературы, где на каждой книге будет указан возраст ребенка, на который она ориентирована. Это означает, что информация в этих книгах представлена так, чтобы быть доступной и понятной читателю данного возраста.

Широко внедряемая сейчас в библиотеках информационно-поисковая система библиотечного каталога спроектирована так, чтобы предоставить читателю информацию о наличии книг по запрошенной тематике в доступной и удобной для восприятия форме. В соответствии с этим же правилом ученые стараются найти более надежные способы предупреждения о землетрясениях, ураганах и других стихийных бедствиях.

Достоверность информации определяется ее свойством отражать состояние реально существующего объекта, процесса или явления. Недостоверная информация может привести к неправильному пониманию ситуации и, как следствие, к принятию неправильного решения.

Полнота (достаточность) информации означает, что она содержит минимальный, но достаточный для принятия правильного решения набор данных. О полноте информации можно говорить, когда какая-либо дополнительная информация об объекте будет уже избыточна.

Понятие полноты информации связано с ее смысловым содержанием.

Как неполная, так и избыточная информация снижает эффективность решений, принимаемых человеком на ее основе.

Таким образом, информация должна быть актуальной, доступной, достоверной и полной.

ЭТО ИНТЕРЕСНО!

Насколько достоверна информация в Википедии?

Если задать вопрос в Google "Насколько точна Википедия?", то первый результат, который вы увидите, чего, в принципе, и следовало ожидать, будет статья самой Википедии по этой теме ("Надежность Википедии"). В 2005 г. ученые вывели качество научных статей в Википедии. В итоге, специалисты обнаружили несколько серьезных ошибок среди 42 статей, которые анализировали эксперты.

Называемая "свободной энциклопедией, редактировать которую может каждый", Wikipedia показала себя достаточно хорошо и в большинстве других проведенных исследований, сравнивающих ее точность с другими специализированными на тех или иных вопросах энциклопедиями.

Тем не менее, поскольку любой человек может отредактировать запись, данные энциклопедии легко могут быть заменены на неверные.



Адекватность информации — это соответствие образа, создаваемого с помощью полученной информации (информационной модели), реальному объекту, процессу или явлению. В реальной жизни вряд ли возможна ситуация, когда существует полная адекватность информации. Всегда присутствует большая или меньшая степень неопределенности. Степень адекватности информации реальному состоянию объекта тоже влияет на правильность решений, принятых человеком.

Например, вы успешно окончили школу и желаете продолжить образование по экономическому направлению. Поговорив с друзьями, вы узнаете, что подобную подготовку можно получить в разных вузах. В результате таких бесед вы получаете весьма разноречивые сведения, которые не позволяют вам принять решение в пользу того или иного варианта, т. е. полученная информация не соответствует реальному состоянию дел. Вы можете посетить официальные сайты вузов, различные сайты рейтингов экономических специальностей вузов, где можно получить более достоверную информацию.

Репрезентативность информации связана с правильностью ее отбора и формирования для адекватного отражения свойств объекта. Непременным условием определения свойства репрезентативности информации является поступление похожей информации из разных источников. Понятно, что полного совпадения по всем источникам информации

никогда не будет. Однако, если все сделано правильно, то полученная информация будет отражать самые важные характеристики объекта.

Важнейшее значение здесь имеют выбранные методики и методы отбора информации и ее оценки.

Например, перед социальной службой города поставлена задача: выяснить, сколько денежных средств еженедельно каждая семья в среднем тратит на питание. Невозможно представить, что для этого социальные работники будут опрашивать всех жителей города. Значит, необходимо выбрать наиболее типичную группу людей, которая и будет опрошена. В результате опроса будет сформирован массив информации, называемый *выборкой*. Необходимо также определить методику опроса, методы обработки собранных данных, их оценки и анализа результатов. Если полученные результаты отражают ситуацию, характерную для большинства жителей города, то говорят о репрезентативности информации, полученной в результате опроса отобранной группы людей. Заключение об адекватности и репрезентативности информации могут быть сделаны на основании специальных методов, которыми занимаются такие науки, как статистика и математическая статистика.

Кроме того, информация может быть **объективной** или **субъективной** (зависящей или независящей от чьего-либо мнения). Сообщение “зима в Европе холодная” является субъективным. Зима в Европе кому-то действительно может показаться холодной (например, для жителей Южного Казахстана), а вот, например, северянам покажется очень даже теплой. Сообщение гидрометеослужбы “температура воздуха — -20°C ” дает объективную информацию, из которой каждый делает для себя определенные выводы.



Одна и та же информация может обладать разными свойствами для разных людей.

Рассмотрим несколько примеров, где выделены наиболее важные свойства для конкретной ситуации.

Пример 1. Утром, когда вы собираетесь в школу, вы обязательно смотрите на часы: вам нужна только достоверная информация. Скорее всего, вы выглянете в окно или посмотрите на термометр, чтобы решить, что следует надеть. Здесь важна актуальность информации. Затем вы идете в школу и находите кабинет, в котором проходит урок согласно расписанию. Вам нужна полная и достоверная информация, иначе невозможно найти нужный кабинет.

Пример 2. Вам необходимо воспользоваться географической картой для определения маршрута поездки, знакомства с новой страной,

изучения исторических событий. Карта всегда служила человеку источником получения необходимой информации. Она также является важным инструментом для исследований в различных областях. Такие задачи, как соотнесение с реальной местностью и координирование строительных работ, решаются с помощью карт. Поэтому здесь жизненно важна адекватность информации, содержащейся в картах реальной местности.

Пример 3. Сейчас популярны геоинформационные системы — живые карты на компьютере или мобильных устройствах. В них обрабатывается и анализируется информация, поступающая со спутников. Такие системы позволяют решать нетрадиционные задачи:

- прогнозировать объем продаж и потенциал рынка, так как могут отображать демографические данные и информацию о расположении магазинов, ассортименте товаров;

- анализировать последствия экологических аварий и выбирать оптимальные решения для их ликвидации;

- строить модели гидрографических сетей и определять участки затоплений;

- строить модели рельефа поверхности Земли.

Все карты “описаны” специальным языком, который понятен лишь специалисту. Это означает, что данная информация доступна не всем. Каждый символ для специалиста несет большой объем достоверной, объективной и понятной информации, которая недоступна тем, кто не знает данного языка.

Пример 4. В современных “космических технологиях” решающую роль играет информация, полученная с помощью различных приборов. Например, расположение станции относительно Солнца важно для работы солнечных батарей. Малейшая неточность — и космический корабль лишится энергии. Такая информация должна быть актуальной, достоверной и полной.

Контрольные вопросы



1. Что такое информация в общепринятом смысле?
2. Какие свойства информации вы знаете? Опишите каждое из них.
3. Что означает полная и неполная информация, достоверная и недостоверная информация?
4. Когда у одной и той информации для разных людей приоритетными становятся разные свойства? Приведите примеры.

Практикум

Уровень А

В следующих примерах определите свойства встречающейся информации:

— На следующий день учительница вывесила правильные решения всех задач...

— Идет контрольная работа по математике. Вы попросили у соседа решение задачи. Шпаргалка содержала полное и правильное решение, но... почерк был непонятен.

— Один персидский царь, собираясь завоевать соседнее государство, обратился к оракулу с вопросом: “Что произойдет, если я со своим войском переправлюсь через пограничную реку?” Оракул ответил: “Государь, ты разрушишь великое царство”. Удовлетворившись таким предсказанием, завоеватель переправился со своим войском через реку и был разгромлен войском противной стороны. В гневе он обратился к оракулу, обвиняя того в обмане. На что оракул ответил: “Государь, а разве твое царство было не велико?”

Уровень В

Приведите примеры:

1. Достоверной, но неактуальной информации;
2. Актуальной, но недостоверной информации;
3. Полной, достоверной, но бесполезной информации;
4. Актуальной, но непонятной информации;
5. Актуальной, не репрезентативной информации.

Уровень С

Прочитайте статью из сборника материалов Международной научной конференции. Сохранив стиль текста, измените его содержание так, чтобы информация в нем стала недостоверной и неточной.

“Развитие многоядерных процессоров привело к увеличению производительности при тех же размерах оборудования, снижению стоимости оборудования, и как следствие — эксплуатационных расходов, снижению энергопотребления облачной системы, что для большинства Центров Обработки Данных (ЦОД) является большой проблемой при наращивании мощностей. Увеличение емкостей носителей информации, и как следствие — снижение стоимости хранения 1Мб информации привело к безграничному увеличению объемов хранимой информации, снижению стоимости обслуживания храни-

лиц информации при значительном увеличении объемов хранимых данных. Развитие технологии многопоточного программирования привело к эффективному использованию вычислительных ресурсов многопроцессорных систем, гибкому распределению вычислительных мощностей “облака”. Развитие технологии виртуализации привело к возможности создания виртуальной инфраструктуры, гибкому масштабированию и наращиванию систем, снижению расходов на организацию и сопровождение систем, доступности виртуальной инфраструктуры через сеть Интернет. Увеличение пропускной способности сети привело к увеличению скорости обмена данными, снижению стоимости интернет-трафика, доступности облачных технологий. Все эти факторы привели к повышению конкурентоспособности облачных технологий в сфере Информационных Технологий”.

(Ширакова Е. А. Облачные технологии // Современные тенденции технических наук.

Рефлексия:

▶ Я узнал(а), что такое... и могу применить для того, чтобы...

§2-3 Совместная работа с документами

Вы научитесь:

- ▶ осуществлять совместную работу с документами с использованием облачных технологий.

Ключевые понятия:

- ▶ облачные технологии
- ▶ совместная работа с документами



Как вы думаете, насколько эффективно работать всем классом одновременно с одним документом? Например, один занимается редактированием текста, другой добавляет таблицы, а третий в это время работает с картинками. В результате за короткий период времени получается готовый документ с картинками и рисунками.

В этом параграфе рассмотрим принципы работы с Google документами: о том, как совместно редактировать тексты, таблицы, а также предоставлять доступ к чтению и редактированию общих файлов.

Бесплатный сервис облачного хранения Google Диск позволяет хранить в Интернете до 15Гб информации. И при желании можно поделиться информацией с каждым. Для работы с Google Диском у

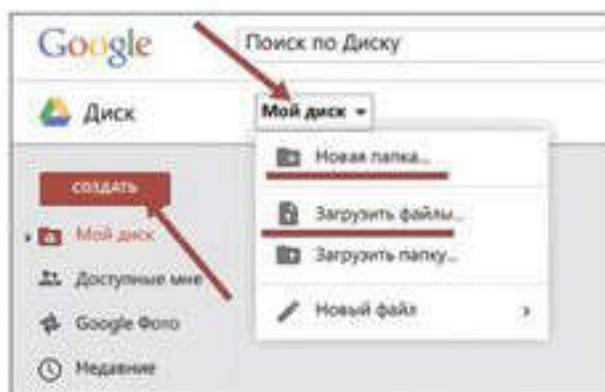


Рис. 2.1. Загрузка файлов

вас должен быть аккаунт Google. Он создается автоматически, когда вы создаете почтовый ящик на gmail.com. Можно создать аккаунт Google, привязав к нему имеющийся почтовый ящик любого почтового сервера. Для того, чтобы загрузить файл на Google Диск, необходимо войти в свой аккаунт. Выбрать Мой Диск или Создать, как на рисунке 2.1.

Более того, Google Диск имеет свой собственный набор приложений для работы с тестом, электронными таблицами и презентациями (рис. 2.2).

Работать с офисными документами можно как самостоятельно, так и совместно с другими пользователями.

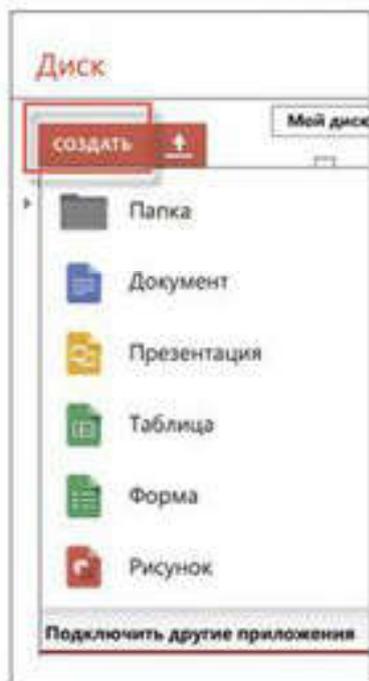


Рис. 2.2. Набор приложений для работы с текстом

Для настройки доступа с документами (текст, таблица или презентация) другим пользователям нажмите на кнопку *Настройка доступа*.

Затем указать, какие именно действия можно выполнить с вашими файлами. Так, есть возможность открыть доступ конкретным людям только для просмотра содержимого, т. е. для чтения, для просмотра с возможностью оставлять комментарии, или же для последующего редактирования (рис. 2.3).

Для этого необходимо ввести список электронных адресов пользователей, которым будет предоставлен доступ, и из выпадающего списка выбрать уровень доступа.

Чуть выше есть кнопка *доступ по ссылке*. Далее можно настроить опцию, выбрав действия, которые будут доступны всем обладателям ссылки на ваш документ.

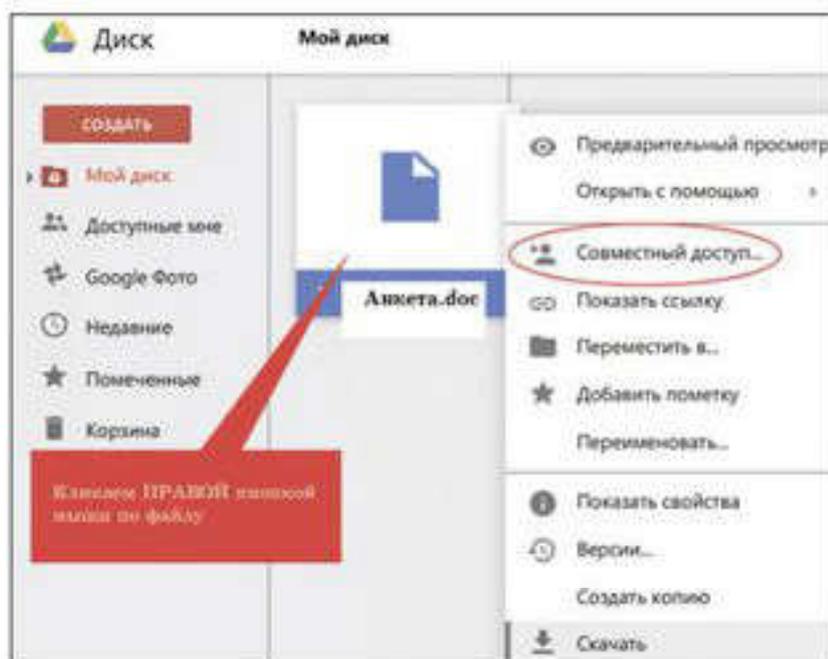


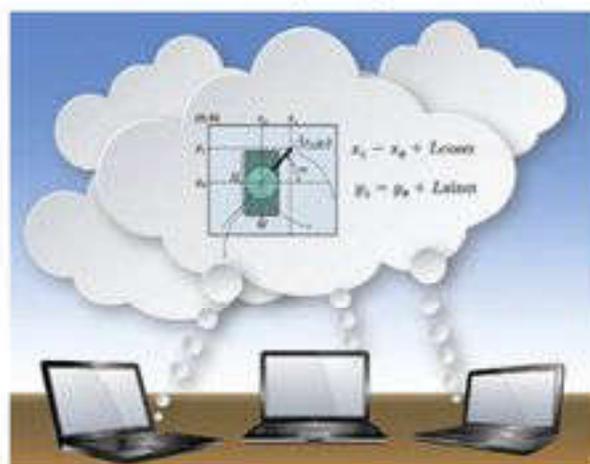
Рис. 2.3. Настройка доступа к документу

Плюсом данного способа является отсутствие необходимости добавлять каждого конкретного участника вручную. Однако есть и “минус” — любой, случайно попавший на страницу, также получит доступ к документу. Но иногда бывает удобно сделать документ публичным, открыв доступ по ссылке только для чтения, а затем дополнительно указать список электронных адресов тех, кто будет работать над редактированием файла. Для этого необходимо нажать на ссылке *Расширенные* и увидеть список пользователей, которым в настоящее время уже открыт доступ к документу, а также, какими именно правами эти пользователи обладают.

Здесь же вы, как владелец документа, можете изменить права любого участника, а также добавить или удалить пользователя (рис. 2.4).

После предоставления доступа новому участнику, каждый из них получит соответствующее уведомление на электронную почту и сможет перейти по ссылке к новому файлу.

При совместной работе с документами on-line всегда можно видеть иконки пользователей, одновременно работающих с файлом. А также есть возможность устроить обсуждение документа, нажав на кнопку чата. Также можно оставлять комментарии.



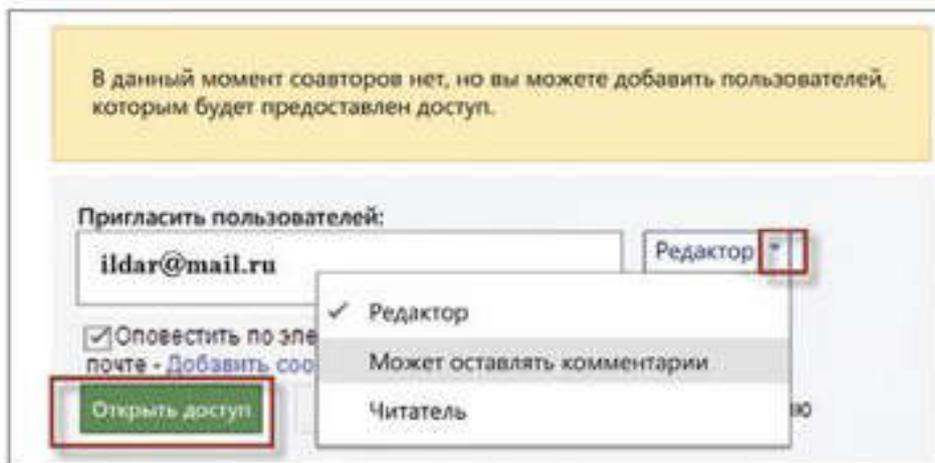


Рис. 2.4. Функции доступа или удаления пользователя

В документе отображаются действия в реальном времени, можно увидеть, где именно находится курсор того или иного пользователя, а также те действия, которые он производит: удаление, редактирование и многое др.

Таким образом, имеется возможность совместно работать с любыми документами в реальном времени. Это быстро и удобно, а главное, работать с документами можно с любых устройств.

Но при этом нужно помнить, что существуют риски, связанные с продолжительной и непрерывной работой за компьютером или с мобильными устройствами:

- может ухудшиться зрение, так как работа за компьютером предполагает длительные и интенсивные зрительными нагрузки;
- идет нагрузка на опорно-двигательный аппарат, так как пользователь в течение длительного времени находится в одном и том же положении — это создает большую статическую нагрузку на позвоночник и некоторые группы мышц, а также приводит к нарушениям нормального кровообращения;
- работа за компьютером сопровождается напряжением нервно-эмоциональной сферы, так как требуется постоянная концентрация внимания и переработка большого объема информации.

ЭТО ИНТЕРЕСНО!

Основатель компании Microsoft Билл Гейтс для своих детей ввел ограничения для работы с компьютером и мобильными устройствами: 45 минут в день в будни и по часу в день на выходных для игр. Использование компьютера допускалось детьми только для выполнения школьных заданий.





Облачные технологии (вычисления) — это технологии распределенной обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователю как интернет-сервис

Контрольные вопросы



1. Что такое *облачные технологии* ?
2. Какие возможности предоставляют облачные технологии?
3. Что бы вы отнесли к достоинствам, а что — к недостаткам облачных технологий?

Практикум

Уровень А

1. Создайте аккаунт Google или войдите в существующий.
2. Создайте несколько папок, загрузите данные, переместите по разным папкам, скачайте файл, удалите файл с Google Диска.

Уровень В

“Создание совместной презентации”.

Используя облачный сервис Google, создайте презентацию о своем классе. Каждый создает страничку “О себе”, используя различные объекты: фото, видео, таблицы и др. Сначала совместно обсудите оформление и наполнение слайдов, после этого приступайте к выполнению проекта.

Уровень С

“Проведение интернет-опроса с помощью Google Форм”

1. Создайте или войдите в свой аккаунт Google;
2. Разработайте форму опроса по исследованию предпочтений потребителей в каком-либо сегменте рынка (продукт, услуга, торговая марка);
3. В названии обязательно укажите свою фамилию;
4. Используйте разные типы вопросов (не менее трех);
5. Примените тему для формы;

6. В описании формы поблагодарите респондента за потраченное время. Отметьте, что участие в данном опросе очень важно для вас;

7. В тексте подтверждения поблагодарите за участие в опросе.

Рефлексия:

▶ Я узнал(а), что такое... и могу применить для того, чтобы...

§4 Сетевой этикет

Вы научитесь:

▶ рассуждать о последствиях нарушения этических и правовых норм в сети.

Ключевые понятия:

▶ сетевой этикет
▶ компьютерная этика
▶ сетикет (нетикет)

В Интернете, как и в обычной жизни, существуют определенные правила общения. Если в обычной жизни это называется *этикой общения*, то в информатике — это *компьютерная этика*.



Компьютерная этика занимается исследованием поведения людей, использующих компьютер.

Например, при копировании материалов из Интернета нужно помнить про авторские права. Любая информация в Интернете (тексты, рисунки, фотографии, программы, игры) защищается авторским правом. И использовать эту информацию в своих целях можно только с согласия автора, например, если по электронной почте вы получите письменное разрешение автора. Иногда на веб-страницах публикуются условия использования материала, где указывается, можно ли сохранять и копировать материал.

Если вы готовите реферат по какому-либо предмету, и информацию нашли в Интернете, то обязательно нужно указать автора (если он известен) и источник, например, адрес сайта.

В Законе Республики Казахстан “Об авторском праве и смежных правах” определено, в каких случаях можно использовать информацию без согласия автора.



Что такое нетикет?

Это понятие появилось в середине 80-х годов XX в.



Сетикет, нетикет (от англ. net — “сеть”) — правила поведения, общения в Сети, традиции и культура интернет-сообщества, которых придерживается большинство.

Виртуальное общение отличается от общения, которое происходит в реальной жизни. Многим людям бывает легче общаться с малознакомыми людьми, которые наверняка им больше никогда не встретятся, — с попутчиками, случайными собеседниками и т. п. В таком общении исчезают зависимость, предубеждение, контакт начинается “с чистого листа”. Примерно то же самое происходит и при общении в Сети: собеседники не знают друг о друге практически ничего, ни достоинств, ни недостатков; никого не интересует, например, какие у собеседника оценки в школе, или, умеет ли он плавать и т. п. Все собеседники изначально равны между собой. Это раскрепощает, но везде нужна мера. Иногда один из участников виртуальной беседы, понимая безнаказанность своего поведения, может оскорбить других, возмутить, разочаровать или просто надоест им. При работе в Сети необходимо руководствоваться определенными негласными, но общеизвестными нормами поведения, этикой общения. Правила нетикета не являются всеобщими. Каждое сетевое сообщество формирует свои правила, и для каждого способа сетевого общения существуют свои правила.

Рассмотрим некоторые из них.

Общение в чатах, форумах — самый демократичный способ сетевого общения, очень распространенный среди молодежи. При общении в чатах желательно выполнять следующие правила:

— используйте язык, на котором общается большинство присутствующих. Это касается как иностранных языков, так и сленга;

— не “умничайте”. Это может возбудить против вас собеседника. Посылайте простые и ясные сообщения;

— если собеседники по каким-либо причинам вас не устраивают, лучше покинуть чат;

— если вы находитесь на форуме, который специализируется на определенной тематике, придерживайтесь темы разговора. Для желающих просто пообщаться существуют специальные разделы форума;

— не повторяйте реплики по несколько раз, если на них нет ответа, чтобы не показаться назойливым или нетерпеливым. Возможно, реакция на вашу реплику последует позже;

— не используйте ненормативную лексику. Во многих чатах следят за ходом общения и применяют к хулиганам наказание в виде ограничения доступа на сервер.

ЭТО ИНТЕРЕСНО!

У постоянных участников тематических чатов, форумов существует словарь сокращений, который помогает отвечать кратко, не загромождая сообщения лишними фразами.

Сокращения известных словоформ	
Словоформа	Сокращение (пишется прописными буквами)
По моему скромному мнению	IMHO (In My Humble Opinion)
Поговори позже	TTYL (Talk To You Later)
К вашему сведению	FYI (For Your Information)
Кстати	BTW (By The Way)
Насколько я знаю	AFAIK (As Far As I Know)
Увидимся	SYL (See You Later)



Общение по электронной почте — популярный и доступный способ общения через Интернет. Переписка может быть деловой (например, “учитель — ученик”, “учитель — мама ученика”), или частной (“мама — сын” или между друзьями) .

Если письмо частное, то к нему не предъявляется жестких требований с точки зрения этики. Письмо, как обычное, так и электронное, вашему близкому другу может быть написано любимыми словами, которые согласен читать ваш друг. Стиль письма и манера изложения в частной переписке зависят от степени знакомства с адресатом.

В обычной деловой переписке есть общепринятые “золотые” правила, которые необходимо знать и которыми следует руководствоваться при любой форме переписки:

- изясняйтесь кратко, выражая только суть;
- пишите грамотно;
- требуя вежливости (честности, благожелательности), будьте вежливы (честны, благожелательны) сами;
- не “отписывайтесь” общими фразами. Если вам нечего сказать, сообщите об этом в корректной форме и прекратите переписку;

— заполняйте поле Тема. Из-за опасения получить в письме компьютерный вирус многие пользователи удаляют сообщения без темы или с подозрительной темой, даже не читая их;

— придерживайтесь объявленной темы;

— подписывайте свои послания;

— не пересылайте большие файлы без предварительной архивации;

— не злоупотребляете рассылкой по нескольким адресам, если ваше письмо не содержит важной и полезной для всех адресатов информации. Ваш адрес могут занести в “черный список” нежелательных адресатов.

Отправка писем, которые могут быть неинтересны получателю (например, различная реклама), является нарушением **правил сетевого этикета** и называется *электронным спамом*. В зависимости от грубости нарушения спаммер может быть наказан. Сервер поставщика услуг Интернета, через которого в сеть поступило сообщение, устанавливается очень просто, а через него легко найти и конкретного нарушителя.

Есть также и свои правила ответа на сообщения электронной почты. Содержание ответа зависит от того, получено ли оно от известного источника или от неизвестного.

- Не затягивайте с ответом. Отвечать знакомым людям следует так, как обычно отвечают по телефону. Если в данный момент выполнить поступившую просьбу нельзя, следует прямо об этом сообщить, не затягивая с ответом более, чем на сутки. Если обращение поступило от незнакомого лица, следует ответить в течение трех дней.

- Если обращение содержит информацию, которая вам неинтересна (например, это реклама, незнакомый файл), на него нельзя отвечать. Любой ответ, даже резко негативный, является для спаммера подтверждением действительности почтового адреса и вызывает новый поток сообщений от него и от других спаммеров. Действительные электронные адреса являются предметом торговли среди дельцов от рекламы, особенно если о владельце адреса известны дополнительные сведения (пол, возраст, образование, личные интересы). По этим же причинам не следует заполнять анкеты, поступающие по электронной почте, даже если в них обещается огромный выигрыш в лотерее.

- Сообщения электронной почты могут иметь вложения (например, фотографии, текстовые файлы). К этим вложениям надо

подходить, как подходят к посылке, полученной от неизвестного доброжелателя. Если письмо с вложением получено от знакомого человека, значит, в тексте письма он обязательно должен был указать, что он приложил и зачем. Если письмо с вложением получено от незнакомого лица, файл вложения следует удалить, не читая и не просматривая, каким бы заманчивым ни было предложение им воспользоваться. Программные файлы, рассылаемые таким образом, часто содержат вирусы, а файлы документов — макровирусы.

ЭТО ИНТЕРЕСНО!

В электронной переписке вместо слов, выражающих эмоции, часто используют сочетания символов (смайлики — ☺, англ. smiley, “улыбочка”). Смайлик — это “выражение лица”, составленное из различных символов: двоеточие — глаза, дефис — нос, скобка — рот. Если посмотреть на эти сочетания сбоку, то становится очевидным “настроение” смайлика — радость, печаль, например:

- :-) — улыбка
- :(— хмурый взгляд
- :~)) — хохот
- :- (— грусть



Сетевой этикет не является догмой. Можно не знать правил общения в Интернете и при этом успешно общаться с виртуальными собеседниками, следуя известному лозунгу “Требуй вежливости от других, будь вежлив сам”.

Контрольные вопросы



1. Почему надо соблюдать этику сетевого общения?
2. Каковы общесетевые правила общения?
3. Какими основными правилами надо руководствоваться при электронной переписке?
4. Какие ограничения накладываются на присоединяемые к электронному письму файлы?
5. Как следует вести себя в чате?
6. Какими могут быть последствия нарушения этических и правовых норм в Сети?

Практикум

Уровень А

Зайдите на сайт <http://onicon.ru/pravjla-obchweniya-v-jnternet> и ознакомьтесь с правилами общения в Интернете.

Уровень В

Законом РК “Об авторском праве и смежных правах” определено, что право на произведение принадлежит тому, кто его создал, или тому, кому автор эти права передал. Фотографии, рисунки, музыка, литературные произведения, сохраненные в электронной форме или размещенные на сайте в Интернете, также являются объектами авторского права. Это значит, что без согласия автора нельзя использовать его произведения: показывать, воспроизводить, распространять, продавать, перерабатывать и т. д.

Проведите исследовательскую работу: найдите в сети Интернет текст Закона РК “Об авторском праве и смежных правах” и подготовьте ответ на следующий вопрос: Можно ли и при каких условиях сохранять на своем компьютере музыкальные файлы, видеофайлы, картинки из Интернета?

Уровень С

Законом РК “Об авторском праве и смежных правах” определено, что право на произведение принадлежит тому, кто его создал, или тому, кому автор эти права передал. Фотографии, рисунки, музыка, литературные произведения, сохраненные в электронной форме или размещенные на сайте в Интернете, также являются объектами авторского права. Это значит, что без согласия автора нельзя использовать его произведения: показывать, воспроизводить, распространять, продавать, перерабатывать и т. д.

Проведите исследовательскую работу: найдите в сети Интернет текст Закона РК “Об авторском праве и смежных правах” и подготовьте ответ на следующий вопрос: Представьте, что вы создаете сайт. Можно ли и при каких условиях размещать на своем сайте музыкальные файлы, видеофайлы, картинки из Интернета?

Рефлексия:

▶ Я узнал(а), что такое... и могу применить для того, чтобы...

РАЗДЕЛ

2

Выбираем компьютер

Из данного раздела вы узнаете:

- ▶ о конфигурации ПК;
- ▶ что программное обеспечение определяет сферу применения компьютера;
- ▶ о прикладных программах;
- ▶ о правилах выбора компьютера.

Вы научитесь:

- ▶ самостоятельно определять конфигурацию компьютера;
- ▶ различать типы программного обеспечения;
- ▶ рассчитывать стоимость компьютера.

§5

Конфигурация компьютера

Вы научитесь:

- ▶ выбирать конфигурацию компьютера в зависимости от его назначения.

Ключевые понятия:

- | | |
|---------------------|---------------------|
| ▶ компьютер | ▶ видеокарта |
| ▶ монитор | ▶ звуковая карта |
| ▶ процессор | ▶ сетевая карта |
| ▶ материнская плата | ▶ внешняя память |
| | ▶ внутренняя память |



Кто-нибудь принимал участие в покупке компьютера? На что вы обращали внимание? Задумывались ли вы над тем, стоит ли покупать компьютер той или иной конфигурации? Как вы думаете, что понимают под конфигурацией компьютера?

На эти вопросы вы найдете ответы в данном параграфе.

При покупке компьютера необходимо учитывать два основных момента. Во-первых, необходимо определить, для решения каких задач предназначен компьютер. В зависимости от функционально-

го назначения условно компьютеры можно разделить на офисные, учебные, игровые, домашние, мультимедийные и т. д.

Приведем правила выбора комплектующих компьютера, на которые необходимо обращать внимание:

— “компьютер для офиса” предназначен в основном для обработки текстовой и числовой информации, сканирования документов, выхода в Интернет. Важно: скорость обработки информации и надежность хранения данных;

— “компьютер для дизайнера” предназначен для обработки графической информации, видеoinформации, web-конструирования, сканирования документов, выхода в Интернет. Важно: разрешение экрана монитора, видеокарта, жесткий диск;

— “компьютер для компьютерного клуба”. Важно: мощность процессора, качество видеокарты и звуковой карты, скорость передачи информации по сети;

— “компьютер для учащегося” предназначен для помощи при выполнении домашних заданий и организации досуга, работы с текстовой и числовой информацией, просмотра видео, прослушивания музыки, выхода в Интернет. Важно: невысокая цена.

Во-вторых, на что еще нужно обратить внимание? Совместимость отдельных устройств с материнской платой. Это, прежде всего, относится к совместимости по интерфейсу подключения. Существует несколько различных процессорных интерфейсов, для каждого из которых выпускаются свои модели материнских плат.

Например, для процессора фирмы Intel используются интерфейсы Socket 1155, LGA 2011-3 и др., а для процессоров фирмы AMD — SocketAM 3, SocketFM 2, SocketS -AM2. Выбирая материнскую плату, обратите внимание на ее процессорный интерфейс.

Также при комплектации компьютера нужно учитывать, что некоторые компоненты, например, видеокарта, звуковая карта, сетевая карта уже встроены в материнскую плату.

Таким образом, под конфигурацией компьютера понимают набор комплектующих, исходя из их назначения.

Самая важная информация о комплектующих компьютера — это тип и частота работы центрального процессора (CPU), объем оперативной памяти (ОЗУ, RAM), объем (размер) жесткого диска (HDD), тип и модель видеокарты, наличие привода для дисков (CD-ROM, DVD-RW, Blu-ray).



Под конфигурацией компьютера понимают набор комплектующих, исходя из их назначения.

Конфигурацию компьютера можно определить самостоятельно. Для этого можно воспользоваться двумя способами:

1. Пуск → Средства администрирования Windows → Сведения о системе.

2. Пуск → В строке поиска введите msinfo32.

Затем вы получите необходимые данные, которые можно поместить в текстовый документ.



Контрольные вопросы

1. Что понимают под *конфигурацией компьютера* ?
2. Как определить конфигурацию компьютера?

Практикум

Уровень А

Определите конфигурацию вашего домашнего компьютера (ноутбука) и заполните таблицу в текстовом процессоре MS Word:

Наименование комплектующих	Характеристика
Процессор	Тактовая частота —
Оперативная память	Объем —
Видеокарта	
Жесткий диск	Объем —
Монитор	Диагональ

Допишите те комплектующие, которые не были указаны в таблице.

Уровень В

Найдите информацию, воспользовавшись сайтами интернет-магазинов или прайс-листами компьютерных магазинов, и заполните таблицу в текстовом процессоре MS Word:

Наименование комплектующих	Компьютер для учащегося		Компьютер для офиса		Игровой компьютер	
	характеристика	цена	характеристика	цена	характеристика	цена
Процессор						
Оперативная память						
Видеокарта						
Жесткий диск						

Допишите те комплектующие, которые, по вашему мнению, необходимы.

Уровень С

В компьютерную фирму обращается клиент, желающий приобрести компьютер. Клиент планирует его использовать для развлечений, иметь возможность подключения компьютера к телевизору для просмотра фильмов, и к Интернету. Дополнительным оборудованием для домашнего компьютера должны быть сканер, струйный фотопринтер, web-камера. Какую комплектацию вы бы порекомендовали? Ответ оформите в виде таблицы “Наименование комплектующих — Характеристика — Цена”.

Рефлексия:

▶ Я узнал(а), что такое... и могу применить для того, чтобы...

§6

Выбор программного обеспечения

Вы научитесь:

- ▶ определять, для чего создается программное обеспечение.

Ключевые понятия:

- ▶ прикладное программное обеспечение (ППО)
- ▶ системное программное обеспечение (СПО)
- ▶ системы программирования

Основу компьютера составляет аппаратное обеспечение (в переводе с английского *hardware* — “твердое изделие”) и программное обеспечение (в переводе с английского *software* — “мягкое изделие”). Для того, чтобы компьютер можно было использовать при решении каких-либо задач, на него устанавливают программное обеспечение — программы, выполняющие ввод, обработку и вывод данных.



Рис. 6.1. Основы компьютера



Программное обеспечение (ПО) — совокупность всех программ, которые хранятся на всех устройствах долговременной памяти компьютера.

Являясь основной частью компьютерной системы, программное обеспечение определяет сферу применения компьютера. Используя имеющееся программное обеспечение и устанавливая на компьютере новые программы, можно:

- превратить его в рабочее место бухгалтера или конструктора, директора школы или агронома;
- решать различные задачи: подготовить реферат, отредактировать документ, отправить письмо другу и многое др.

Каждая программа предназначена для решения определенной задачи. При этом она тесно взаимодействует с аппаратными устройствами компьютера: центральным процессором, устройствами ввода/вывода, запоминающим устройством и др.

Обычно выделяют три вида программного обеспечения: прикладные программы, системные программы и системы программирования (рис. 6.2).



Рис. 6.2. Классификация программного обеспечения

ЭТО ИНТЕРЕСНО!



Лидером производства программного обеспечения является фирма Билла Гейтса "Microsoft". Билл Гейтс — один из самых богатых людей планеты.



Прикладное программное обеспечение (ППО) предназначено для решения конкретных задач. На сегодняшний день имеется большое количество прикладных программ. ППО подразделяется на



Рис. 6.3. Программы общего назначения

на программы общего и специального назначения (рис. 6.3). К программам общего назначения относят:

- калькуляторы;
- графические редакторы: Paint, Photoshop, Gimp, Inkscape;
- текстовые редакторы: Блокнот, WordPad в операционной системе Windows; nano, gedit, KWrite и Kate в операционной системе Linux;
- текстовые процессоры: OpenOffice.org Writer, Microsoft Word;
- электронные таблицы: Excel, OpenOffice.org Calc;
- базы данных и СУБД: Access, OpenOffice.org Base;
- программы разработки презентаций: PowerPoint, OpenOffice.org Impress.
- игры (головоломки, стратегии и другие) и программы для проф.дизайна.

К программам специального назначения относят (рис. 6.4):

- компьютерные словари и энциклопедии;
- бухгалтерские программы.

Системное программное обеспечение (СПО) выполняет функции посредника между аппаратурой компьютера и программами, управляет его устройствами. Часто при покупке нового компьютера в стоимость входит и операционная система, но может быть и ситуация, когда установлена только MS DOS (дискетная операционная



Рис. 6.4. Программы специального назначения

система), которая позволяет в дальнейшем загрузить ОС с графическим интерфейсом.

Системы программирования позволяют программистам разрабатывать различные программы (Visual Basic, Delphi, Pascal, C++ и др.).

ЭТО ИНТЕРЕСНО!

Прикладные программы используются в самых необычных областях жизни человека. Так, японскими производителями игрушек Takara выпущен переводчик кошачьего мяуканья.

Раньше программное обеспечение было "привязано" к определенной операционной системе. Например, одни программы работали только под управлением Windows, другие – только под управлением Linux.



Кроссплатформенная программа — это программа, у которой есть версии для различных операционных систем.

В настоящее время разработано много кроссплатформенных приложений, у которых есть версии для разных операционных систем.



По своему назначению все ПО можно разделить на прикладное, системное и системы программирования:

Прикладное ПО (программы общего и специального назначения).

Системное ПО (операционные системы, антивирусные программы и др.).

Системы программирования (системы программирования и языки программирования).

Контрольные вопросы



1. Что такое *программное обеспечение компьютера* ?
2. Чем различаются типы программного обеспечения?
3. Какие вам известны прикладные программы? Для чего они предназначены?
4. Что обозначают термины *hardware* и *software*?
5. Что такое *кроссплатформенное приложение* ?

Практикум

Уровень А

Проведите мини-исследование и выясните, какое программное обеспечение установлено на вашем компьютере. Определите, к какому классу программного обеспечения относится та или иная программа. Результат оформите в виде таблицы:

Программа	Класс программного обеспечения

Уровень В

Проведите мини-исследование: найдите в сети Интернет примеры кроссплатформенных приложений. Опишите, к какому классу ПО они относятся. Результат оформите в виде таблицы.

Программа	Класс программного обеспечения

Уровень С

Напишите эссе “ПО для программистов и опытных пользователей”.

Рефлексия:

▶ Я узнал(а), что такое... и могу применить для того, чтобы...

§7

Выбираем компьютер

Вы научитесь:

- ▶ выбирать конфигурацию компьютера в зависимости от его назначения.

Ключевые понятия:

- ▶ компьютер для офиса
- ▶ компьютер для дизайнера
- ▶ универсальный компьютер
- ▶ игровой компьютер



Многих людей ставит в тупик вопрос “как выбрать компьютер?” В этом параграфе мы рассмотрим, на какие параметры обратить внимание при выборе компьютера для дома или офиса.

Для начала необходимо определиться, какой тип компьютера нужен для дома — стационарный (десктоп) или миниатюрный (ноутбук, моноблок, неттоп). Они различаются не только размерами. Ноутбук, моноблок или неттоп гораздо компактнее и их можно брать с собой куда угодно. Настольные компьютеры более производительны, чем ноутбуки. В стационарном ПК можно заменить практически любые комплектующие, но обновить ноутбук намного сложнее. Можно заменить оперативную память и жесткий диск, а также приобрести внешнюю видеокарту, но даже это возможно не для каждой модели ноутбука.

Прежде чем купить компьютер, важно определиться: при выборе комплектующих обратиться к специалистам или подобрать самостоятельно. И в первом и во втором случае имеются определенные плюсы и минусы.

1. Сборка на заказ

Как советуют многие специалисты, нужно заказывать в том магазине, который имеет хорошую репутацию. При заказе компьютера необходимо сказать менеджеру или сборщику, что вы ожидаете от нового компьютера и на какую сумму рассчитываете.

 — возможность выбора характеристик и дизайна компьютера под свои потребности и вкус;

 — правильный подбор комплектующих, качество сборки и надежность компьютера сильно зависят от квалификации сборщика

2. Самостоятельная сборка

Если вы уверены в своих базовых знаниях и навыках, то можете самостоятельно собрать компьютер под свои потребности и вкус.

 — те же, что и при сборке на заказ, но при этом минимально возможная цена, а также уверенность в том, что “все собрано идеально, так как делалось для себя”;

 — при отсутствии необходимых знаний и навыков высока вероятность получить несбалансированную систему или допустить критические ошибки.

В зависимости от цены компьютеры условно можно разделить на четыре группы:

— до 150 тыс. тенге — начальный уровень (**бюджетные компьютеры**). Компьютеры этой группы предназначены для простых задач — работа с текстом, Интернетом, просмотр видео, простые игры типа “Зума”, а также некоторые современные игры на самых минимальных настройках качества. Используются обычно в офисах.

Приведем пример расчета бюджетного бесшумного компьютера (табл. 7.1).

Таблица 7.1

Пример расчета бюджетного бесшумного компьютера

Название	Цена, у. е.
Корпус Morex CASO-25 60W (или аналогичный бесшумный mini-ITX корпус с внешним блоком питания 60—90 Вт)	45
SSD-диск Kingston SSDNow UV400 120 Gb (SUV400S37/120G)	25
Материнская плата со встроенным 4-ядерным процессором Asus N3150I-C	90
Оперативная память Kingston ValueRAM DDR3 4Gb 1600GHz (KVR16N11S8/4)	30
Итого	190

За небольшие деньги можно собрать небольшой (22 x 19 x 6 см), надежный, полностью бесшумный и быстрый в работе четырех-ядерный компьютер для использования в офисе, серфинга по Интернету. В данную комплектацию входит быстрый SSD-диск, поэтому загрузка операционной системы происходит за считанные секунды, а офисные программы и Интернет работают быстро.

150—400 тыс. тенге — средний уровень (**универсальные и недорогие игровые компьютеры**). Компьютеры этой группы наиболее распространены, так как вычислительной мощности процессоров и возможностей видеокарт компьютеров этой группы достаточно не только для простых задач, но и для большинства мощных ресурсоемких программ, они позволяют в большинстве случаев запускать современные игры на средних и максимальных настройках качества. Оптимальный выбор для большинства домашних пользователей (табл. 7.2).

Таблица 7.2

Расчет стоимости универсального компьютера

Название	Цена, у. е.
1	2
Корпус Zalman Z1 Black + блок питания Chieftec APS-550SB (или корпус другого производителя + блок питания от 400 Вт + кулер)	95
HDD-диск WD Caviar 1000GB WD10EZRZ	45
Оперативная память, комплект Kingston HyperX DDR4 2 x 4Gb 2666GHz (HX426C15FBK2/8)	75
На AMD	
Процессор AMD Ryzen 3 2200G BOX	95

Продолжение

1	2
Материнская плата Asus A320M-K	65
На Intel	
Процессор Intel i3-8100 BOX	130
Материнская плата Asus PRIME H310M-K	65
Итого на AMD	375
Итого на Intel	410

Тихий, быстрый и недорогой компьютер: отличные процессоры, 8 Гб оперативной памяти и производительное встроенное видео позволяют комфортно чувствовать себя практически во всех популярных программах, а также в большинстве современных игр на минимальных, а некоторых — и на средних настройках качества.

Выше 400 тыс. тенге – высший уровень (средние и мощные игровые компьютеры). Персональные компьютеры этой группы обладают самой лучшей производительностью, мощность процессора позволяет решать современные задачи и новые самые тяжелые игры на максимальных настройках качества. К минусам можно отнести высокую стоимость, а также очень часто — повышенный шум системы охлаждения. В основном используют в компьютерных клубах (табл. 7.3).

Таблица 7.3

Расчет стоимости среднего игрового компьютера

Название	Цена, у.е.
1	2
Корпус Zalman Z1 Black + блок питания Chieftec APS-550SB (или корпус другого производителя + блок питания от 500 Вт + кулер)	95
SSD-диск для системы Samsung 970 EVO 500 Gb M.2 NVMe (MZ-V7E500BW)	140
HDD-диск WD Caviar 2000GB WD20EZRZ	65
Видеокарта Asus PH-GTX1060-6G	310
Оперативная память, комплект Kingston HyperX DDR4 3200GHz 2x8Gb (HX432C16PB3K2/16)	170
На AMD	
Процессор AMD Ryzen 5 2600 BOX	190

Продолжение

1	2
Материнская плата Asus PRIME B350-PLUS	110
На Intel	
Процессор Intel Core i5-8400 BOX	230
Материнская плата Asus Prime H370-PLUS	135
Итого на AMD	1080
Итого на Intel	1145

Игровой компьютер с процессорами средне-высокого уровня, хорошей видеокартой и емким очень быстрым SSD M.2 NVMe под систему, которые вместе смогут позволить наслаждаться комфортной, очень быстрой работой системы и всеми современными играми на максимальных и большинством — на ультра-максимальных настройках качества (табл. 7.4).

Таблица 7.4

Расчет стоимости мощного игрового компьютера

Название	Цена, у. е.
Корпус Zalman Z3 Plus + блок питания Chieftec APS-600SB (или корпус другого производителя + блок питания от 600 Вт + кулер)	115
SSD-диск для системы Samsung 970 EVO 500 Gb M.2 NVMe (MZ-V7E500BW)	140
HDD-диск WD Caviar 2000GB WD20EZRZ	65
Видеокарта Asus GeForce RTX 2070 8GB (DUAL-RTX2070-8G)	600
Оперативная память, комплект Kingston HyperX DDR4 3200GHz 2x8Gb (HX432C16PB3K2/16)	170
На AMD	
Процессор AMD Ryzen 7 2700 + кулер	280
Материнская плата Asus PRIME X370-PRO	140
На Intel	
Процессор Intel Core i7-8700 + кулер	360
Материнская плата Asus TUF Z390-PRO GAMING	200
Итого на AMD	1510
Итого на Intel	1650

Мощный игровой компьютер, который справится с любой самой тяжелой программой или игрой. Такой компьютер понравится любому пользователю — независимо от того, планирует ли он открывать десятки или сотни закладок в Интернете, заниматься обработкой видео или играть в игры на ультра-максимальных настройках качества.

Специализированные компьютеры. В эту группу попадают персональные компьютеры для инженеров, видеомонтажа, которые индивидуально подбираются под конкретные задачи, хотя очень часто бывает, что покупают самый новейший процессор и новейшую видеокарту.

Таким образом, самостоятельно подобрать комплектующие для компьютера даже при таком огромном ассортименте совсем не сложно, если следовать простым советам, указанным в параграфе.

Контрольные вопросы



1. Какие плюсы и минусы самостоятельного комплектования ПК вы знаете?
2. В чем состоит отличие офисного компьютера от специализированного?

Практикум

Уровень А

Используя ресурсы Интернет, рассчитайте стоимость очень мощного игрового компьютера.

Название	Цена, у. е.
1	2
Корпус Zalman Z3 Plus + блок питания Chieftec APS-650SB (или корпус другого производителя + блок питания от 650 Вт + кулер)	
SSD-диск для системы Samsung 970 EVO 1000 Gb M.2 NVMe (MZ-V7E1T0BW)	
HDD-диск WD Caviar 4000GB WD40EZRZ	
Видеокарта Asus GeForce RTX 2080 8GB (DUAL-RTX2080-8G)	
Оперативная память, комплект Kingston HyperX DDR4 3200GHz 4x8Gb (HX432C16PB3K4/32)	

Продолжение

1	2
На AMD	
Процессор AMD Ryzen 7 2700X + кулер	
Материнская плата Asus PRIME X370-PRO	
На Intel	
Процессор Intel Core i9-9900K + кулер	
Материнская плата Asus TUF Z390-PRO GAMING	
Итого на AMD	
Итого на Intel	

Уровень В

Рассчитайте стоимость офисного компьютера, используя прайс-листы интернет-магазинов. Необходимо учесть работу с большими объемами данных и нетребовательные офисные приложения. Выполните расчеты, используя комплектующие разных брендов.

Уровень С

Рассчитайте стоимость компьютера для архитектора. Кроме конфигурации компьютера необходимо учесть стоимость лицензионного специального программного обеспечения.

Рефлексия:

► Я узнал(а), что такое... и могу применить для того, чтобы...

РАЗДЕЛ
3

Базы данных

Из данного раздела вы узнаете:

- ▶ что такое базы данных;
- ▶ как создавать базу данных в электронной таблице;
- ▶ как осуществлять поиск, сортировку и фильтрацию данных.

Вы научитесь:

- ▶ создавать базу данных в электронной таблице;
- ▶ осуществлять поиск, сортировку и фильтрацию данных.

§8

Базы данных

Вы научитесь:

- ▶ объяснять термины "база данных", "запись", "поле".

Ключевые понятия:

- ▶ базы данных
- ▶ запись
- ▶ поле

Нам известно, что важно не только хранить информацию, но и быстро ее находить. До появления компьютеров человечество хранило информацию на различных носителях. В XX в. в основном на бумажных носителях: карточки в поликлинике, каталог в библиотеке, картотека в отделе кадров и др.



Что такое базы данных?

Под **базой данных** понимают совокупность данных об объектах рассматриваемой предметной области, их свойствах и взаимосвязях.

Итак, базой данных можно считать библиотечные каталоги или же папки со сведениями о сотрудниках, лежащие в отделе кадров любого учреждения, или ваш школьный журнал.

Базы данных служат в основном для хранения информации о большом количестве каких-либо объектов и для поиска сведений о них. Так, в базе данных, которая содержит сведения о лекарствах, продаваемых в аптеках города, можно по запросу покупателя найти сведения о месте продажи конкретного лекарства и о его цене.

Основными операциями с базами данных являются ее первичное заполнение, внесение всевозможных изменений в данные (например, сведений о поступлении новых лекарств или об изменении цен на них и т. д.), поиск нужных данных и их упорядочение (сортировка), которые используются для ускорения поиска.



База данных (БД) — информационная модель, которая позволяет в упорядоченном виде хранить данные о группе объектов, обладающих одинаковым набором свойств.

Базы данных бывают сетевыми, иерархическими и реляционными.

Иерархические базы данных графически могут быть представлены как дерево, состоящее из объектов различных уровней. Верхний уровень занимает один объект, второй — объекты второго уровня, и т. д. Такие объекты находятся в отношении предка (объект, более близкий к корню) к потомку (объект более низкого уровня), при этом объект-предок может не иметь потомков или иметь их несколько, тогда как объект-потомок обязательно имеет только одного предка.

В качестве примера иерархической модели данных можно рассматривать генеалогическое дерево (рис. 8.1).



Рис. 8.1. Пример генеалогического дерева

Другой пример — размещение информации (в виде файлов) на компьютере. В качестве дерева выступает определенная структура в виде каталогов (папок).

Сетевая база данных является обобщением иерархической — за счет допущения объектов, имеющих более одного предка.

Примером сетевой базы данных фактически является глобальная компьютерная сеть Интернет. Гиперссылки связывают между собой сотни миллионов документов в единую распределенную сетевую базу данных. Еще один пример сетевой модели представлен ниже (рис. 8.2).

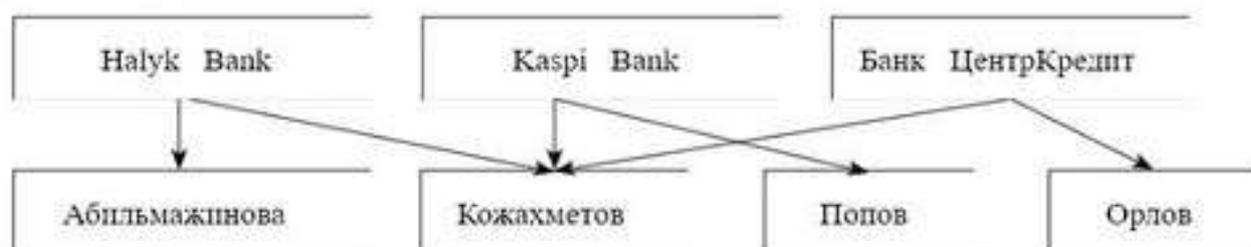


Рис. 8.2. База данных “Вкладчики”

Рассмотрим подробно *реляционную* базу данных

Табличная база данных содержит перечень объектов одного типа, т.е. объектов, которые имеют одинаковый набор свойств. Подобные базы данных представляются в виде двумерной таблицы.

Рассмотрим следующую базу данных:

Записи

Поля

Фамилия	Имя	Адрес	Телефон
Абишев	Ансаган	ул. Г. Мусрепова, д. 364, кв. 512	8(727) 123 45 67
Мусин	Нуржан	ул. М. Жумабаева, д. 968, кв. 245	8(727) 891 01 23
Васильев	Егор	пр. Абая, д. 725, кв. 345	8(727) 456 78 90

Столбцы таблицы называют *полями*. Каждое поле имеет имя и может содержать данные определенного типа. Тип данных и имя — это свойства поля.



Поле базы данных — это столбец таблицы, содержащий значения определенного свойства.

Строки таблицы являются записями об объекте; эти записи разбиты на поля столбцами таблицы. Каждая запись имеет набор значений, содержащихся в полях.



Запись базы данных — это строка таблицы, содержащая набор значений свойств, размещенный в полях базы данных.



Контрольные вопросы

1. Что понимают под *базой данных* ?
2. В чем заключается разница между записью и полем в табличной базе данных?

Практикум

Уровень А

Дана сетевая структура базы данных “Вкладчики” (рис. 8.3). Преобразуйте данную структуру к табличному виду.

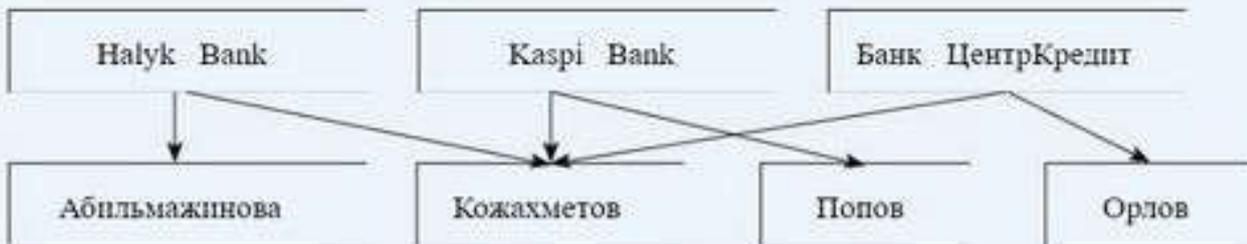


Рис. 8.3. База данных “Вкладчики”

Уровень В

Дана иерархическая структура базы данных “Таблица Менделеева” (рис. 8.4). Преобразуйте данную структуру к табличному виду.

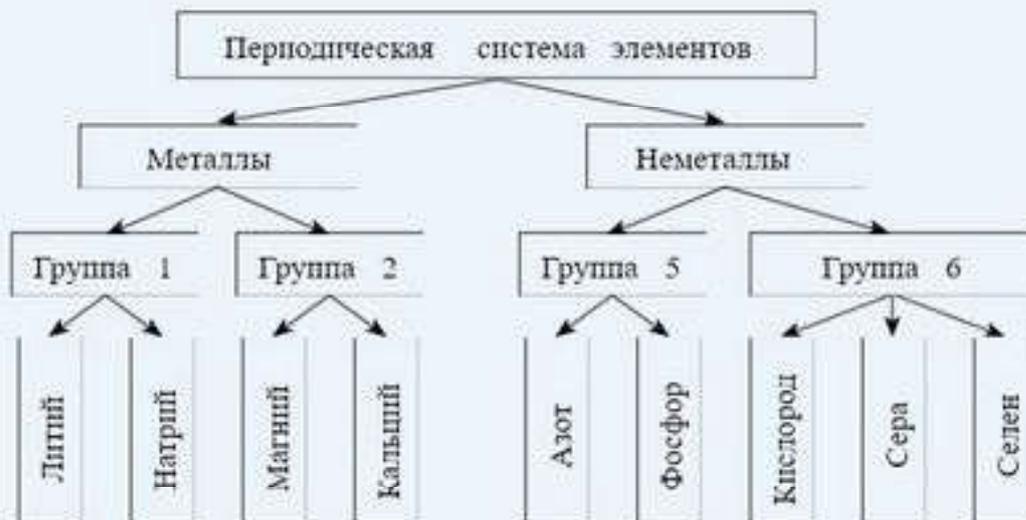


Рис. 8.4. База данных “Таблица Менделеева”

Уровень С

Создайте иерархическую базу данных “Школа” и оформите ее в виде, аналогичном рисунку 8.4.

Рефлексия:

Я узнал(а), что такое... и могу применить для того, чтобы...

§9

Создание базы данных в электронных таблицах

Вы научитесь:

- создавать базы данных в MS Excel.

Ключевые понятия:

- базы данных
- поле
- список
- строка
- запись
- столбец

С помощью Microsoft Excel можно создавать и обрабатывать базы данных. В этом случае электронную таблицу называют *списком* или *базой данных в Excel* и используют терминологию:

- строка списка — запись базы данных;
- столбец списка — поле базы данных.

Название столбца может занимать только одну ячейку и при работе с таблицей называется именем поля. Например, диапазон А3 : F1 можно рассматривать как базу данных (рис. 9.1). Столбцы

1	Планеты Солнечной системы					
2						
3	Планета	Период	Расстояние	Диаметр	Масса	Спутник
4	Меркурий	0,24	58	4,9	0,32	0
5	Венера	0,62	108	12,1	1,86	0
6	Земля	1	150	12,8	6	1
7	Марс	1,8	288	6,8	0,61	2
8	Юпитер	11,86	778	142,6	1906,98	16
9	Сатурн	29,46	1426	120,2	570,9	17
10	Уран	84,01	2869	49	87,24	14
11	Нептун	164,8	4496	50,2	103,38	2

Рис. 9.1. Информация о планетах Солнечной системы

A, B, C, D, E, F этой таблицы называются полями, а строки с 3 по 1 — записями. Диапазон A3 : F3 содержит имена полей.

Существуют ограничения, накладываемые на структуру базы данных в MS Excel:

- первая строка данных должна содержать неповторяющиеся имена полей;
- остальные строки базы данных должны содержать записи;
- не должно быть пустых строк или столбцов;
- информация в одном поле (столбце) должна быть однородной, т.е. только числа или только текст;
- не должно быть объединенных ячеек.

Рассмотрим БД “Планеты Солнечной системы”. Единицы измерения, используемые в таблице:

- период обращения по орбите — в земных годах;
- среднее расстояние от Солнца — в млн. км;
- экваториальный диаметр — в тыс. км;
- масса — $\times 10^{24}$ кг.

Основной отличительной особенностью таблицы, которая рассматривается как база данных, являются ее большие размеры.

Создать базу данных в MS Excel легче, если при вводе данных использовать встроенную форму данных (рис. 9.2).

Таким образом, можно работать над данными с помощью формы и непосредственно в окне таблицы.

Планеты Солнечной системы

Планета: Меркурий 1 из 8

Период: 0,24

Расстояние: 58

Диаметр: 4,9

Масса: 0,32

Спутник: 0

Добавить

Удалить

Вернуть

Назад

Далее

Критерии

Закрыть

Рис. 9.2. Форма для работы с данными

Контрольные вопросы



1. Назовите преимущества и недостатки табличного представления баз данных.
2. Что такое *запись* и что такое *поле в базе данных* ?
3. Поля каких типов могут присутствовать в базе данных Excel?
4. Назовите преимущества и недостатки работы с базами данных с помощью формы?

Практикум

Уровень А

Создание базы данных.

1. Создайте базу данных “*Планеты Солнечной системы*” по предложенному образцу (табл. 9.1)

Таблица 9.1

БД “Планеты солнечной системы”

Планета	Период	Расстояние	Диаметр	Масса	Спутник

2. Для ячеек “*Период*”, “*Расстояние*”, “*Диаметр*”, “*Масса*”, “*Спутник*” установите числовой формат.

4. Внесите первую запись. Начиная со второй записи, заполните таблицу, используя команду *Форма*.

Команды *Форма* нет на Ленте и для того, чтобы отобразить ее на панели быстрого доступа, необходимо:

— в контекстном меню панели быстрого доступа выбрать команду **Настройка панели быстрого доступа**;

— в открывшемся диалоговом окне *Параметры Excel* выбрать раздел *Панель быстрого доступа*;

— из раскрывающегося списка *Выбрать команды из*: выбрать команду *Команды не на ленте*.

— в списке выбрать **Форма** и нажать на кнопку **Добавить**;

— закрыть диалоговое окно *Параметры Excel*.

После выполнения этих шагов ваша панель быстрого доступа будет включать новый значок. При щелчке на нем отображается

форма для ввода данных. Активизируйте любую ячейку в листе и выберите команду *Форма* на панели быстрого доступа.

5. Сохраните базу данных на своем рабочем диске под названием “*Солнечная система.xlsx*”.

Уровень В

1. Создайте базу данных “*Птицы и животные Казахстана*”.
2. При помощи формы создайте базу данных “*Области Казахстана*” с информацией об областных центрах, площади и населении.

Уровень С

Проведите исследование, какой способ создания БД для Вас наиболее удобен, наиболее оптимален по затраченному времени. Подсказка: оцените время, за которое вы создаете две одинаковые БД разными способами. Сделайте выводы.

Рефлексия:

▶ Я узнал(а), что такое... и могу применить для того, чтобы...

§10 Методы поиска информации

Вы научитесь:

- ▶ осуществлять поиск с помощью формы и команды Найти.

Ключевые понятия:

- ▶ список в электронной таблице
- ▶ поиск
- ▶ рабочий лист

Самым простым способом поиска информации в списках Excel является использование формы данных. В основном требуется найти определенные записи в списках.

Для этого необходимо воспользоваться окном *Форма*. Затем в появившемся окне формы выбрать кнопку *Критерии* и ввести признаки, по которым будет производиться поиск. Например, если поиск осуществляется по значениям в текстовом поле, необходимо выбрать первые символы, которые однозначно определяют искомые данные.

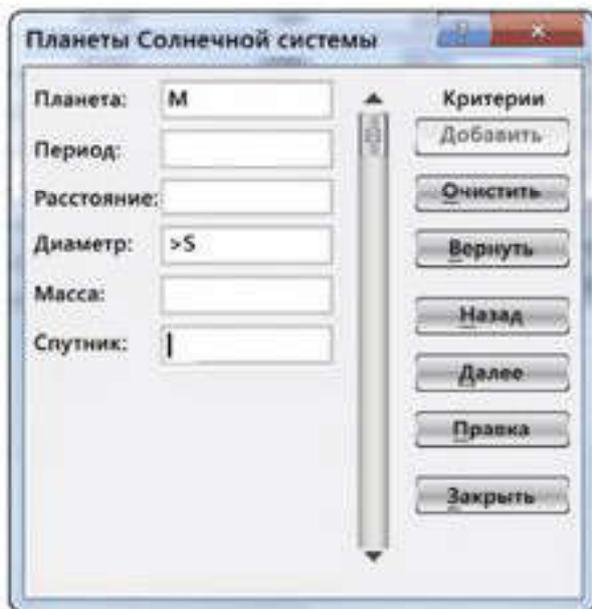


Рис. 10.1. Форма с заданными критериями

Если ввести несколько критериев, то поиск будет произведен по нескольким критериям (рис. 10.1).



Как осуществить поиск с помощью формы, значений и формул?

С помощью формы данных поиск ведется по заданным полям (столбцам). Но если необходимо осуществить поиск по всему рабочему листу какого-либо конкретного значения или формулы, то лучше использовать процедуру поиска.

На ленте **Главная** в группе **Редактирование** выберите команду **Найти и выделить** (рис. 10.2)

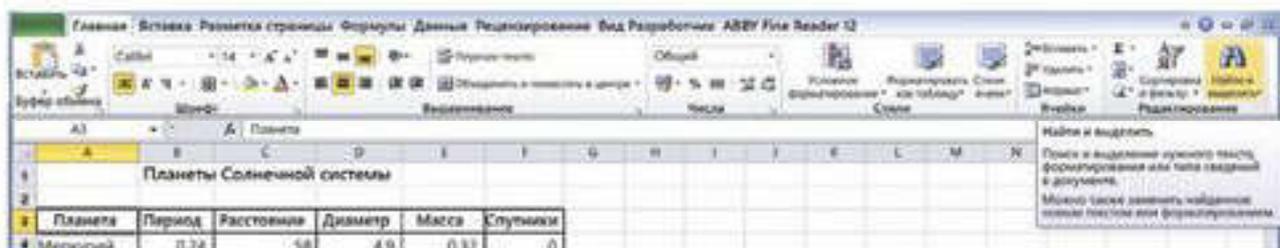


Рис. 10.2. Поиск с использованием команды Найти

В открывшемся диалоговом окне **Найти и заменить** в поле **Найти** введите фрагмент содержимого ячейки. Затем нажмите на кнопку **Найти далее** (рис. 10.3). Если введенный фрагмент есть в таблице, то будет активизирована ячейка, содержащая искомое.

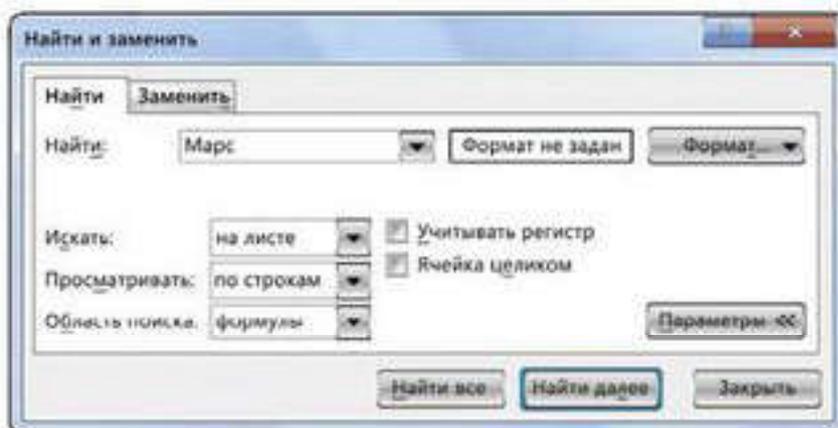


Рис. 10.3. Диалоговое окно Найти и Заменить

По умолчанию поиск ведется по строкам. Можно изменить схему поиска. Для этого необходимо задать в диалоговом окне *Найти* поиск по столбцам. Также можно уточнить, какого рода данные вы ищете: значения или формулы.

Вместо конкретного символического фрагмента для поиска можно использовать символы * и ?.

Символ "*" используется вместо любых символов, т.е., если указать "M*", то программа укажет и ячейку, содержащую текст "Марс" и ячейку с текстом "Меркурий".

Вопросительный знак "?" используется вместо одного символа, т.е., если указать 1?, будут найдены ячейки, содержащие двухзначные числа, начинающиеся с 1.

Контрольные вопросы



1. Что такое *список* в электронной таблице ?
2. Из каких элементов состоят списки?
3. Какие этапы включают процедуры поиска записей?
4. Как выполнить поиск данных на рабочем листе?

Практикум

Уровень А

Осуществите в созданной базе данных поиск по некоторым критериям (например, название по любой начальной букве).

Уровень В

Осуществите поиск по нескольким критериям с помощью общего средства поиска (Найти).

Уровень С

Проведите исследование, какой способ поиска данных в БД для Вас наиболее оптимален, наиболее удобен. Выделите критерии поиска и сделайте выводы.

Рефлексия:

Я узнал(а), что такое... и могу применить для того, чтобы...

§11-12 Сортировка и фильтрация данных

Вы научитесь:

- ▶ осуществлять поиск, сортировку и фильтрацию данных.

Ключевые понятия:

- ▶ сортировка
- ▶ фильтр
- ▶ поиск
- ▶ данные

Сортировка данных является базовой операцией любой таблицы. В Excel предусмотрены средства упорядочивания данных после любых изменений и дополнений, которые можно использовать при необходимости.



Как можно сортировать данные в электронных таблицах (ЭТ)?

Данные в ЭТ можно сортировать по возрастанию или убыванию в алфавитном порядке или области (рис. 11.1).

	A	B	C	D
1	-10	бит	1 января 2017	20:30
2	-5	bit	3 марта 2018	16:30
3	0	S	12 января 2016	12:30
4	1		3 марта 2017	8:30
5	5		12 января 2015	4:30

Рис. 11.1. Сортировка данных



Как вы думаете, что происходит с данными, если выбрать Сортировку по возрастанию?

Сортировка по возрастанию:

- Числа — от наименьшего отрицательного до наибольшего положительного числа;
- Текст — в алфавитном порядке (числа, знаки, латинский алфавит, русский алфавит);
- Дата и время — в хронологическом порядке;
- Пустые ячейки — всегда помещаются в конец списка.

При сортировке по убыванию данные выстраиваются в порядке, обратном указанному.



Сортировка — это изменение относительного положения данных в списке в соответствии со значением и с типом данных.

Существует несколько режимов сортировки. В электронных таблицах есть такой режим сортировки, который после выбора любого столбца расширяет диапазон сортируемых данных (рис. 11.2).

Результат расширенного диапазона приведен на рисунке 11.3.



Рис. 11.2. Режим сортировки

	A	B	C	D
1	5	1	12 января 2015	4:30
2	1	5	3 марта 2017	8:30
3	0	S	12 января 2016	12:30
4	-5	bit	3 марта 2018	16:30
5	-10	бит	1 января 2017	20:30

Рис. 11.3. Расширенный диапазон сортировки

	A	B	C	D
1	-10	бит	1 января 2017	20:30
2	5	бит	3 марта 2018	16:30
3	0	S	12 января 2016	12:30
4	1	S	3 марта 2017	8:30
5	5	L	12 января 2015	4:30

Рис. 11.4. Вид после сортировки в пределах выделения

После сортировки в расширенном диапазоне *изменяется порядок* следования строк, *но сохраняется их целостность* (рис. 11.4).

При необходимости можно выполнить сортировку по нескольким столбцам: *Главная* → *Сортировка и фильтр* → *Настраиваемая сортировка*. Выбрать первый уровень сортировки и настроить его параметры. Добавить следующий уровень сортировки и т. д. (рис. 11.5)

В результате сначала данные будут отсортированы по нескольким столбцам по очереди: сначала они сортируются по первому столбцу,

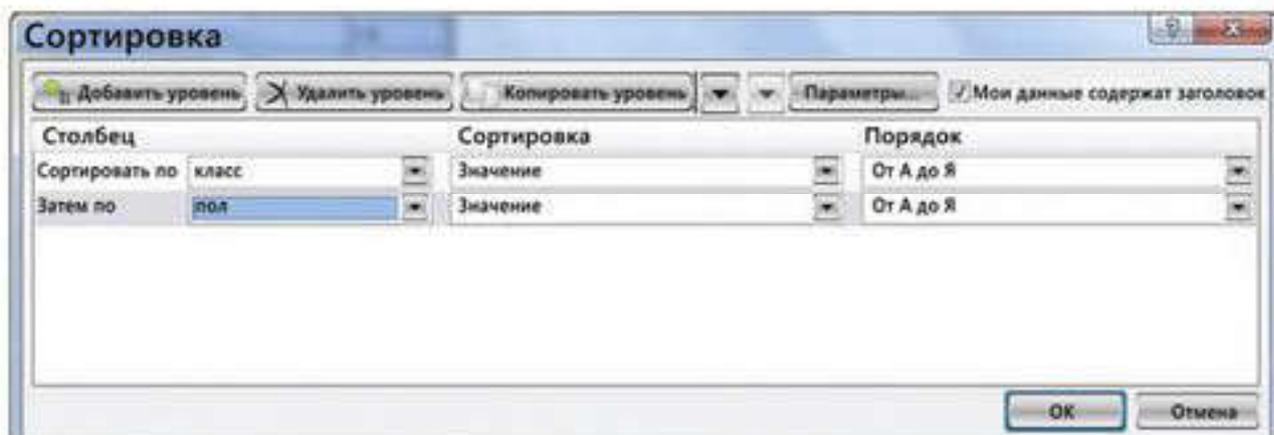


Рис. 11.5. Настраиваемая сортировка по нескольким столбцам

потом — по второму и т. д. Исходная таблица и результат сортировки представлен на рисунке 11.6.

	А	В	С
1	ФИ	пол	класс
2	Ахметов Д.	м	1 А
3	Садыков Т.	м	1 Б
4	Баракова А.	ж	4 Б
5	Ермеков К.	м	2 В
6	Ахметова Г.	ж	1 Б
7	Темиргалиев Б.	м	1 А
8	Сафронов А.	м	1 А

	А	В	С
1	ФИ	пол	класс
2	Ахметов Д.	м	1 А
3	Сафронов А.	м	1 А
4	Темиргалиев Б.	м	1 А
5	Ахметова Г.	ж	1 Б
6	Садыков Т.	м	1 Б
7	Ермеков К.	м	2 В
8	Баракова А.	ж	4 Б

Рис. 11.6. Исходная таблица и отсортированная таблица

Фильтрация. С помощью фильтров можно осуществить поиск данных, которые скрывают в исходной таблице записи, не удовлетворяющие условиям поиска. Например, у нас имеется таблица (рис. 11.7):

	А	В	С	Д
1	ФИО	Оклад, тенге	Возраст	Отдел
2	Ахметов А.А.	150 000	45	бухгалтерия
3	Смирнов Е.В.	120 000	28	производственный
4	Жакеев С.К.	100 000	21	бухгалтерия
5	Еремеева С.С.	180 000	50	склад
6	Балатбек Е.С.	100 000	23	производственный
7	Низамутдинов В.К.	100 000	24	производственный
8	Жабагин С.М.	120 000	29	бухгалтерия
9	Гордиянова Г.В.	120 000	33	склад

Рис. 11.7. Исходная таблица

Используя фильтры (рис. 11.8), можно получить необходимые данные.

Таблица с примененным фильтром, где показана заработная плата > 120 000.



Поиск данных в электронной таблице — это отбор записей (строк), удовлетворяющих условиям поиска, заданным в форме **фильтра**.

	A	B	C	D
1	ФИО	Оклад, тенге	Возраст	Отдел
2	Ахметов А.А.	150 000	45	бухгалтерия
5	Еремеева С.С.	180 000	50	склад

Рис. 11.8. Использование числового фильтра

Условия поиска записей создаются с использованием *операторов сравнения*.

Для текстовых данных операции:

- *равно* (сравниваются все символы);
- *начинается с* и *не начинается с* (сравниваются первые символы);
- *заканчивается на* и *не заканчивается на* (сравниваются последние символы);
- *содержит* и *не содержит* (сравниваются последовательности символов в различных частях текста).

Помимо текстовых фильтров, бывают “Числовые фильтры”, которые в основном принимают те же самые условия, что и текстовые, но имеют и дополнительные, относящиеся только к числам:

— “Выше среднего” и “Ниже среднего” – возвращает значения, которые выше и ниже среднего значения, соответственно. Среднее значение рассчитывается, исходя из всех числовых значений списка столбца;

— “Первые 10...” – клик по данному пункту вызывает окно (рис. 11.9).

В этом окне можно задать, какие элементы вывести: первые из наибольших или первые из наименьших. Также, какое коли-

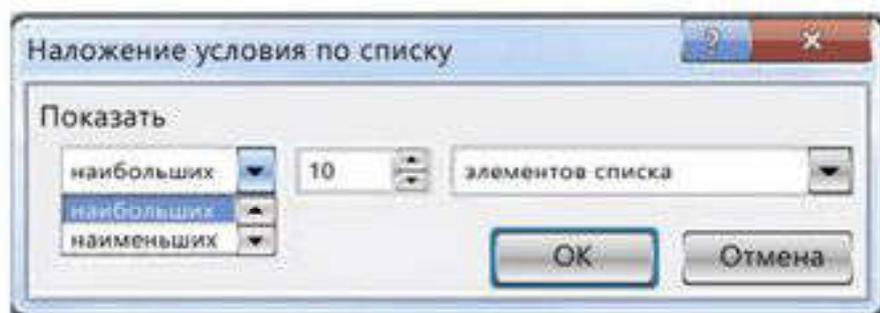


Рис. 11.9. Числовой фильтр “Первые 10...”

чество элементов вывести, если в последнем поле выбран пункт “элементов списка”. Если же выбран пункт “% от количества элементов”, второе значение задает данный процент. То есть если в списке 10 значений, то будет выбрано самое высокое (или самое низкое) значение. Если в списке 1000 значений, то либо первые, либо последние 100.

Если в списке содержатся данные типа Дата и время, то к ним можно применить фильтры по дате:

- *До ...* будут выведены все записи до указанной даты;
- *После ...* будут выведены все записи после указанной даты;
- *Между ...* будут выведены все записи, находящиеся в диапазоне между указанными датами.

Существуют и другие условия фильтрации по дате: все даты за период..., с начала года, на прошлой неделе и т. д.

Контрольные вопросы



1. Как осуществляется сортировка списков?
2. Какими характеристиками обладает список данных?
3. Что необходимо делать, чтобы отсортировать данные по нескольким критериям?

Практикум

Уровень А

Сортировка данных в списке.

1. Запустите MS Excel.
2. Откройте файл “Солнечная система.xlsx”.
3. Выполните сортировку по столбцу “Планета” по возрастанию. Результат сортировки скопируйте на Лист 2.
4. Выполните сортировку по столбцу “Расстояние” по убыванию. Результат сортировки скопируйте на Лист 2.
5. Переименуйте Листы 2 и 3. Назовите Лист 2 — Планеты, Лист 3 — Расстояние.

Уровень В

Примените сортировку и фильтрацию по нескольким критериям (для текстовых данных, для числовых данных и др.).

Уровень С

Проведите исследовательскую работу и выясните, в чем различие между простыми и составными фильтрами. Результат исследования оформите в виде таблицы.

Простой фильтр		Составной фильтр	
условия	особенности	условия	особенности

Рефлексия:

▶ Я узнал(а), что такое... и могу применить для того, чтобы...

§13 Работа с базой данных

Вы научитесь:

- ▶ создавать базу данных в электронной таблице;
- ▶ осуществлять поиск, сортировку и фильтрацию данных.

Ключевые понятия:

- ▶ база данных
- ▶ поле
- ▶ критерий
- ▶ функция
- ▶ аргументы функции

В MS Excel при работе с базами данных используются функции для анализа данных из списков или баз данных. Каждая функция, которая имеет обобщенное название **БД Функция**, использует три аргумента:

- база данных;
- поле;
- критерий.

Данные аргументы ссылаются на интервалы ячеек на рабочем листе, которые используются данной функцией. В MS Excel для работы с базами данных имеется 13 функций.



Синтаксис функций баз данных:
БД Функция (база_данных; поле; критерий).

Под критерием будем понимать интервал ячеек, который содержит задаваемые условия.

Приведем некоторые функции баз данных:

— функция **ДСРЗНАЧ** — усредняет значения в столбце списка (базы данных);

— функция БСЧЕТ — подсчитывает количество ячеек в столбце списка (базы данных);

— функция БСЧЕТА — подсчитывает непустые ячейки в столбце списка (базы данных).

С остальными функциями познакомимся при выполнении практических заданий.

Практикум

Уровень А

Заполните таблицу. При выполнении задания используйте ресурсы Интернет или справочную систему MS Excel.

Наименование функции БД	Назначение функции БД
ДСРЗНАЧ	
БСЧЕТ	
БСЧЕТА	
БИЗВЛЕЧЬ	
ДМАКС	
ДМИН	
БДПРОИЗВЕД	
ДСТАНДОТКЛ	
ДСТАНДОТКЛП	
БДСУММ	
БДДИСП	
БДДИСПП	

Уровень В

В данной базе данных запишите фамилии в порядке:

	А	В	С	Д
1	ФИО	Оклад, тенге	Возраст	Отдел
2	Ахметов А.А.	150 000	45	бухгалтерия
3	Смирнов Е.В.	120 000	28	производство
4	Жакеев С.К.	100 000	21	бухгалтерия
5	Еремеева С.С.	180 000	50	склад
6	Балатбек Е.С.	100 000	23	производство
7	Низамутдинов В.К.	100 000	24	производство
8	Жабагин С.М.	120 000	29	бухгалтерия
9	Гордиянова Г.В.	120 000	33	склад

- 1) Если записи отсортированы по возрастанию по полю “ФИО”.
- 2) Если записи отсортированы по убыванию по полю “Оклад”, а затем по полю “Возраст”.
- 3) Если записи отсортированы по возрастанию по полям “Оклад”, “Возраст” и “Отдел”.

Уровень С

Сколько записей удовлетворяют условию:
“Отдел” = бухг и “Возраст” < 40?

Рефлексия:

Я узнал(а), что такое... и могу применить для того, чтобы...

РАЗДЕЛ

4

Программирование алгоритмов на языке программирования Python

Из данного раздела вы узнаете:

- ▶ об одномерном массиве в программировании;
- ▶ как ввести и вывести одномерный массив;
- ▶ о методах поиска элемента в массиве;
- ▶ о методах перестановки и сортировки элементов в массиве.

Вы научитесь:

- ▶ использовать одномерный массив;
- ▶ решать задачи с использованием одномерного массива.

§14 Одномерный массив

Вы научитесь:

- ▶ создавать программы на языке программирования Python с использованием одномерных массивов.

Ключевые понятия:

- ▶ массив
- ▶ переменная
- ▶ индекс
- ▶ элемент массива

Основное предназначение современных компьютеров — обработка большого количества данных. При этом надо как-то обращаться к каждой из тысяч (или даже миллионов) ячеек с данными. Очень сложно дать каждой ячейке собственное имя и при этом не запутаться. Из этой ситуации выходят так: дают имя не ячейке, а группе ячеек, в которой каждая ячейка имеет собственный номер. Такая область памяти называется *массивом* (или *таблицей*).



Массив — это группа переменных одного типа, расположенных в памяти рядом (в соседних ячейках) и имеющих общее имя. Каждая ячейка в массиве имеет уникальный номер (индекс).

Для работы с массивами нужно, в первую очередь, научиться:

- выделять память нужного размера под массив;
- записывать данные в нужную ячейку;
- читать данные из ячейки массива.

В языке Python нет такой структуры, как “массив”. Вместо этого для хранения группы однотипных объектов используют списки (тип данных `list`).

Список в Python — это набор элементов, каждый из которых имеет свой номер (индекс). Нумерация всегда начинается с нуля.

В отличие от обычных массивов, в большинстве языков программирования список — это динамичная структура, его размер можно изменять во время выполнения программы (удалять и добавлять элементы).

Каждый день приложения в наших устройствах обрабатывают более миллиона данных в секунду. Например, когда приложение “Карты” находит нам подходящий маршрут, используются множество таких переменных, например, какие дороги находятся поблизости, где они пересекаются, что считается зданием, а что нет. Беря в расчет все эти переменные, приложение “Карты” создает нам подходящий маршрут (рис. 14.1).



Как приложения в компьютерах легко работают с большим количеством данных?

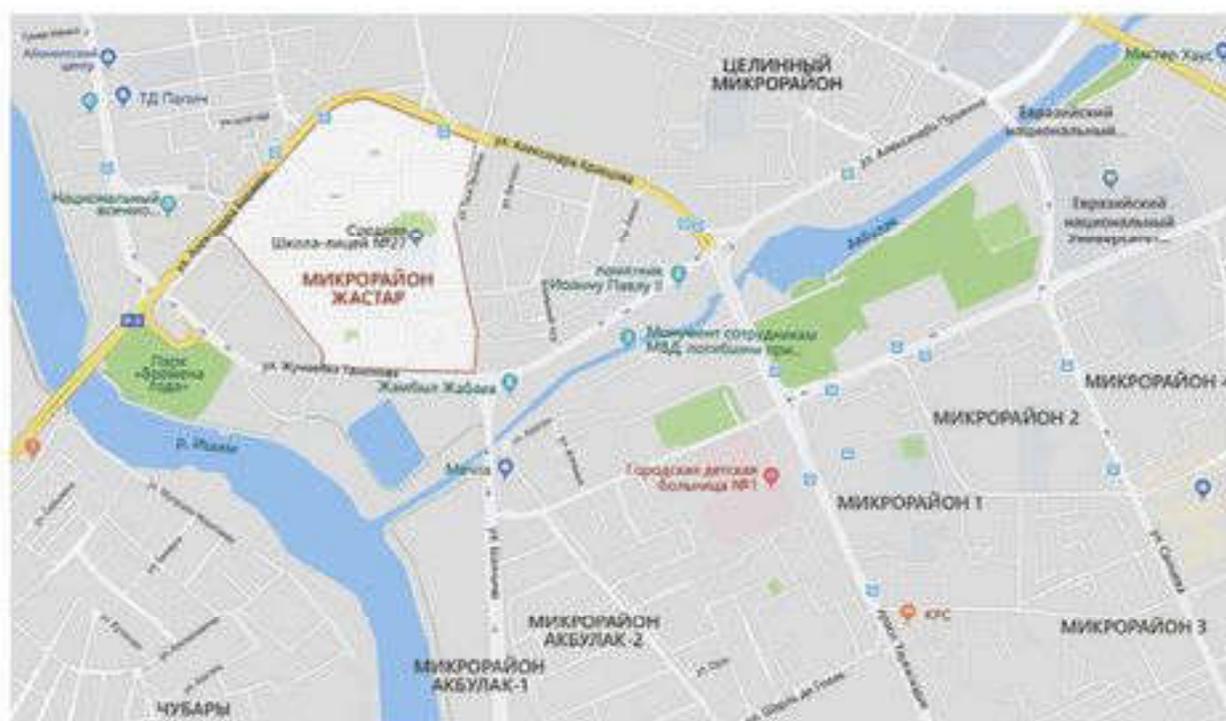


Рис. 14.1. Приложение “Карты”

Как вы знаете, в программировании, если мы хотим сохранить какое-то значение, мы используем переменные. Например в *Python* :

```
name = "Adilhan"
```

Так мы сохранили имя Адильхан в переменной `name`. В программировании мы храним какие-то значения с последующим их применением. А если значений, используемых с одной целью, много? Как хранить такого рода значения? Нелегко для программиста создавать отдельные переменные для каждого значения. Например: если мы будем использовать 5 имен, тогда:

```
name1 = "Adilhan"
name2 = "Dias"
name3 = "Dariga"
name4 = "Nurbahyt"
name5 = "Aruzhan"
```

Как видите, если список будет увеличиваться, для каждого нового имени нужно создавать отдельную переменную. А компьютеры работают с большим количеством значений, иногда даже с миллиардами значений.

Во избежание таких трудностей, в программировании есть такая возможность, как создание массива из данных. Массив хранит в себе множество значений одного типа или предназначения. Например: если мы хотим хранить значения пяти имен, тогда:

```
names=["Adilhan","Dias","Dariga","Nurbahyt","Aruzhan"]
```

Таким образом, мы создали массив из пяти имен. В языке программирования *Python*, как вы знаете, переменные могут хранить такие типы данных, как текст, числовые значения, а также объекты.



Список/массив обозначается скобками `[]`. Все, что находится внутри квадратных скобок `[]`, является элементами массива.

В *Python*, чтобы отобразить элементы из массива, используйте команду `print`. Например:

```
names=["Adilhan","Dias","Dariga","Nurbahyt","Aruzhan"]
print(names)
```

Вывод:

```
['Adilhan', 'Dias', 'Dariga', 'Nurbahyt', 'Aruzhan']
```



У каждого элемента из массива существует его отдельный адрес, также известный как индекс элемента, чтобы отобразить или использовать его. Если мы создали массив, то дальнейшее применение квадратных скобок говорит о том, что мы используем отдельные элементы с индексом.

Пример обращения к элементу по его индексу:

```
names = ["Adilhan", "Dias", "Dariga", "Nurbahyt", "Aruzhan"]
print(names[0])
```

Вывод:

```
Adilhan
```

В примере выше продемонстрировано, как отобразить первое имя из массива *names*.

Пример :

```
names = ["Adilhan", "Dias", "Dariga", "Nurbahyt", "Aruzhan"]
starosta = names[2]
print(starosta)
```

Вывод:

```
Dariga
```

В примере выше мы сохранили третье имя из массива в переменную *starosta*. Затем отобразили значение этой переменной.

Пример :

```
names = ["Adilhan", "Dias", "Dariga", "Nurbahyt", "Aruzhan"]
names[3] = "Daniyar"
print(names)
```

Вывод:

```
['Adilhan', 'Dias', 'Dariga', 'Daniyar', 'Aruzhan']
```

В примере выше мы изменили значение четвертого элемента из массива на Daniyar.

ЭТО ИНТЕРЕСНО!

В языке программирования Python можно вместо одного значения индекса использовать область значений. Например, если мы хотим отобразить элементы со 2-го по 5-й, то мы пишем таким образом:

```
print(names[1:4])
```

Если хотим отобразить значения с 3 по последний, пишем таким образом:

```
print(names[2:])
```

Чтобы отобразить все имена до 4-го имени:

```
print(names[:3])
```

Также в языке программирования Python можно указывать индекс с отрицательным значением. В этом случае он будет указывать на элементы с конца массива. Например:

```
print(names[-1])
```



Контрольные вопросы



1. Зачем нужен массив?
2. Как создать массив?
3. Как указать на отдельные элементы массива?
4. Как изменить значение элемента массива на другое?

Практикум

Уровень А

1. Создайте массив, который хранит названия шести городов Казахстана.
2. Отобразите массив.
3. Отобразите второй город из массива.
4. Отобразите последний город из массива.

5. Создайте переменную `столица`, сохранив в ней название нашей столицы из массива.

Уровень В

1. Создайте массив, который хранит шесть чисел.
2. Отобразите одинаковые числа в массиве.

Уровень С

1. Поменяйте местами первое число с последним из массива.
2. Добавьте в массив еще одно число.

Рефлексия:

Я узнал(а), что такое... и могу применить для того, чтобы...

§15–16

Поиск элемента с заданными свойствами

Вы научитесь:

- ▶ создавать программы на языке программирования Python с использованием одномерных массивов.

Ключевые понятия:

- ▶ элемент
- ▶ массив
- ▶ переменная



Как осуществляется поиск в базе данных?

При программировании приложения перед программистом часто встает вопрос о поиске определенных значений в массиве данных.

Например: найти определенного пользователя с определенным логином. Или же — найти, у скольких пользователей день рождения совпадает с определенной датой.

Логическая реализация поиска в массиве выглядит следующим образом. Например, напишем следующую программу: существует ли число 3 в массиве чисел.

Во-первых, объявим массив данных:

```
a = []
```

Мы создали пустой массив. Теперь начнем заполнять его числами:

```
a = [ int (input()) for i in range(N) ]
```

Чтобы осуществить поиск в одномерном массиве, достаточно просто сравнить число, которое мы ищем, со всеми числами из массива.

```
for i in a:
    if i == 3:
        print("YES")
        break
```

В данном примере мы показываем, если число 3 существует в данном массиве, то отобразить "YES".

Ранее мы уже обсуждали — у *Python* есть множество готовых функций, что упрощает жизнь программиста. Одна из них — для поиска элемента в массиве. "Поиск числа 3 в данном массиве" можно реализовать следующим образом.

```
if 3 in a:
    print("YES")
```

Выражение "3 in a" говорит о том, существует ли число 3 в массиве *a*. Данное выражение возвращает *true* если верно, и *false* — в противном случае.

Если же мы хотим найти, на каком месте стоит данное число в массиве, то существует функция *index()*. Она возвращает индекс данного элемента, если он есть в массиве, и ошибку, если данного элемента не существует. Реализация выглядит следующим образом:

```
a = [ int (input()) for i in range(N) ]
if 3 in a:
    print(«у элемента, равного 3 порядковому номеру=»,
          a.index(3))
```

Чтобы наша программа не вернула нам ошибку, мы в первую очередь должны проверить, существует ли число в массиве, затем — вывести индекс данного элемента из массива.

Контрольные вопросы



1. Как найти элемент в массиве, соответствующий заданным критериям?
2. Какой метод в *Python* реализует поиск элемента с заданным значением?
3. Как вывести индекс элемента с заданным значением?

Практикум

Уровень А

Создайте следующую программу:

1. Введите число N в первой строке ввода.
2. Создайте массив, заполнив его N числами во второй строке ввода.
3. В третьей строке ввода введите число K .
4. Осуществите поиск в массиве элементов, равных числу K . Если элементы в массиве существуют, выведите "YES", в противном случае — "NO".
5. Если существуют элементы с заданным значением во второй строке вывода, то выведите количество этих элементов в массиве.

Пример 1.

Ввод	Вывод
5	NO
3 4 6 7 10	
5	

Пример 2.

Ввод	Вывод
6	YES
10 12 3 4 3 5	2
3	

Уровень В

Создайте программу. В первой строке дано число N . Во второй строке дано N чисел. В третьей строке дано число K . Если числа K не существует, следует вывести длину массива, что осуществляется путем добавления числа K в конец массива.

Пример 1

Ввод	Вывод
5	0
4 3 2 4 5	
4	

Пример 2

Ввод	Вывод
6	3
1 2 3 4 5 6	
4	

Пример 3

Ввод	Вывод
6	6
1 2 3 4 5 6	
7	

Уровень C

Создайте программу, которая осуществляет поиск последней позиции числа K в массиве. В первой строке дано число N . Во второй строке дано N чисел. В третьей строке дано число K . Если числа K не существует, следует вывести длину массива, что осуществляется путем добавления числа K в конец массива.

Пример 1

Ввод	Вывод
5	3
4 3 2 4 5	
4	

Пример 2

Ввод	Вывод
6	4
1 2 3 3 3 6	
3	

Пример 3

Ввод	Вывод
6	6
1 2 3 4 5 6	
7	

Дополнительное задание

Дана задача. Адильхан загадал Диасу число от 1 до 100. Диас предлагает варианты ответа Адильхану. Каждый раз Адильхан говорит — больше или меньше. Как Диасу, в худшем случае, быстро подойти к правильному ответу и за какое количество попыток? Реализуйте данный алгоритм на Python.

Примеры решения задач

Рассмотрим следующую задачу. Дается массив из N чисел. Найдите максимальный элемент из массива. В первой строке — число N . Во второй строке — N чисел.

Пример :

Ввод	Вывод
8	7
1 6 5 7 3 2 4 5	

В первую очередь введем число N :

```
n = int( input () )
```

Затем создадим массив из N чисел:

```
a = [ int (input()) for i in range(n)]
```

Найдем максимальный элемент в данном массиве тем же способом, что и при вычислении определенного элемента из массива. При этом создадим переменную, которая будет хранить максимальный элемент. На первом шаге мы будем хранить значение первого элемента из массива как максимального.

```
mx = a[0]
for i in a:
    if i > mx:
        mx = i
print(mx)
```

Таким образом, если мы находим какой-то элемент, который больше значения переменных, то мы меняем его значение на большее.

ЭТО ИНТЕРЕСНО!

В Python-е также существуют функции нахождения максимума и минимума в массиве. Такие, как `max` и `min`. Пример:

```
print (max (a) )
```

Находит максимум в массиве чисел `a`.

```
print (min (a) )
```

```
Находит( min (a) )
```

Находит минимум в массиве чисел `a`.

Также существует функция `sum` для подсчета суммы чисел в массиве чисел.

**Рефлексия:**

- ▶ Я узнал(а), что такое... и могу применить для того, чтобы...

\$17 Перестановка элементов**Вы научитесь:**

- ▶ создавать программы на языке программирования Python с использованием одномерных массивов.

Ключевые понятия:

- ▶ массив
- ▶ переменная

? *Каким образом можно осуществлять перестановку в массиве?*

Одна из операций с массивом, часто используемая программистами, это перестановка элементов внутри массива. Часто она нужна во время сортировки элементов в массиве. Реализация перестановки элементов в массиве выглядит следующим образом. Например, у нас имеется числовой массив.

```
numbers = [ 3, 4, 10, 23, 21, 15 ]
```

Мы хотим поменять третий элемент с пятым. Аналогичным способом реализуется и перестановка значений у переменных. Ис-

пользуется третья переменная, выполняющая функцию временного хранилища.

```
num1 = 5
num2 = 10
temp = num1
num1 = num2
num2 = temp
```

Только в данном случае мы используем элементы массива с индексами.

```
temp = numbers[2]
numbers[2] = numbers[4]
numbers[4] = temp
```

В языке программирования Python существует готовая функция для перестановки элементов в массиве в обратном порядке. Эта функция называется `reverse()`.

Например, возьмем массив из чисел.

```
numbers = [ 1, 32, 21, 34, 4 , 23]
```

Теперь применим функцию `reverse()` и отобразим массив.

```
numbers.reverse()
for i in numbers:
    print(i, end=" ")
```

Перестановка элементов

Решение задач

Рассмотрим следующую задачу. Дан массив из N чисел. Необходимо произвести циклический сдвиг вправо всех элементов массива.

Пример :

Ввод

6
9 4 5 6 8 7

Вывод

4 5 6 8 7 9

Создадим массив из N чисел следующим образом:

```
N = int( input() )
nums = [ int (input()) for i in range(N) ]
```

Теперь с помощью цикла поменяем соседние элементы массива местами, т.е. переместим последний элемент с конца путем перемены местами соседних элементов.

```
for i in range( N-1, 0, -1 ):
    temp = nums[ i-1 ]
    nums[ i-1 ] = nums[ i ]
    nums[ i ] = temp
```

Теперь отобразим наш массив:

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

Практикум

Уровень А

1. Дан массив из N чисел. Переставьте элементы в массиве в обратном порядке, не используя функцию `reverse()`.

Пример :

Ввод	Вывод
7	8 9 7 3 2 6 4
4 6 2 3 7 9 8	

2. Дан массив из N чисел. Поменяйте местами максимальный и минимальный элементы. Выведите массив в одну сторону.

Уровень В

1. Дан массив из N чисел. Выполните сдвиг элементов в массиве на K элементов вправо и отобразите его. В первой строке дано число N . Во второй — N чисел. На последней дано число K — сдвиг.

Пример 1:

Ввод	Вывод
6	2 8 9 4 6 3
4 6 3 2 8 9	
3	

Пример 2:

Ввод	Вывод
8	4 10 23 45 78 1 2 3
23 45 78 1 2 3 4 10	
2	

2. Напишите следующую программу. Создайте массив из N чисел. Затем переставьте местами соседние элементы в массиве следующим образом: ($A[0]$ с $A[1]$, $A[2]$ с $A[3]$ и т. д.).

Пример

Ввод	Вывод
8	5 7 4 6 8 7 1 9
7 5 6 4 7 8 9 1	

Уровень С

1. Дан список целых чисел. Требуется “сжать” его, переместив все ненулевые элементы в левую часть списка, не меняя их порядок, а все нули — в правую часть. Нельзя изменять порядок ненулевых элементов и использовать дополнительный список.

Пример :

Ввод	Вывод
4 0 5 0 3 0 0 5	4 5 3 5 0 0 0 0

2. Решите предыдущую задачу с перестановкой элементов, не используя во время перестановки дополнительную переменную.

Пример :

Ввод	Вывод
8	5 7 4 6 8 7 1 9
7 5 6 4 7 8 9 1	



Контрольные вопросы

1. Как реализовать перестановку элементов в массиве?
2. Какая функция в языке программирования *Python* переставляет элементы в обратном порядке?

Рефлексия:

▶ Я узнал(а), что такое... и могу применить для того, чтобы...

§18–19 Двумерный массив

Вы научитесь:

- ▶ создавать программы на языке программирования Python с использованием двумерных массивов;
- ▶ использовать итерацию в массиве.

Ключевые понятия:

- ▶ двумерный массив
- ▶ матрица
- ▶ итерация



Что такое двумерный массив?

Двумерный массив можно представить в виде таблицы. У каждой ячейки таблицы есть номер строки и номер столбца. То есть каждая ячейка имеет не один индекс, как в обычном списке, а два: номер строки и номер столбца. Такие списки называют двумерными. А в математике подобные структуры называют “матрицы”.



Размерность массива — это количество индексов, необходимое для однозначной индексации элемента в рамках массива.

Рассмотрим пример матрицы размера 4 x 3:

```
matrix = [[-1, 0, 1],
          [-1, 0, 1],
          [0, 1, -1],
          [1, 1, -1]]
```

Данный оператор можно записать в одну строку:

```
matrix = [[-1, 0, 1], [-1, 0, 1], [0, 1, -1], [1, 1, -1]]
```

Вывод двумерного массива можно осуществить одним оператором.

```
print(matrix)
```

Недостатком данного способа считается то, что это не позволяет выполнять какой-то предварительной обработки элементов:

Результат: `[[−1, 0, 1], [−1, 0, 1], [0, 1, −1], [1, 1, −1]]`



Как обратиться к конкретному элементу?

Рассмотрим пример:

```
a = [[1, 2, 3], [4, 5, 6]]
print (a[0])
print (a[1])
print (a[0][0], a[0][1], a[0][2])
```

Результат:

```
C:\Programs\Python\Python36-32\python.exe
[1, 2, 3]
[4, 5, 6]
1 2 3
↔
Process finished with exit code 0
```

Рассмотрим следующий пример:

```
a = [[1, 2, 3], [4, 5, 6]]
print (a[0])
print (a[1])
print (a[0][0], a[0][1], a[0][2])
print (len(a))
print (len(a[0]))
```

В данном примере `a[0]` — это вся первая строка, функция `len(a)` — количество строк, `len(a[0])` — количество элементов (столбцов) в первой строке.

Результат:

```
C:\Programs\Python\Python36-32\python.exe
[1, 2, 3]
[4, 5, 6]
1 2 3
2
3
Process finished with exit code 0
```

Для обработки и вывода списка, как правило, используются вложенные циклы. Первый цикл — по номеру строки, второй цикл — вложенный — по элементам внутри строки. Например, вывести двумерный числовой список на экран построчно, разделяя числа пробелами внутри одной строки, можно так:

```
for i in range(len(A)):
    for j in range(len(A[i])):
        print(A[i][j], end = ' ')
print()
```

То же самое, но циклы не по индексу, а по значениям списка:

```
for row in A:
    for elem in row:
        print(elem, end = ' ')
print()
```

Кроме того, для вывода одной строки можно воспользоваться методом

```
join:
for row in A:
    print(' '.join(list(map(str, row))))
```

Конечно, заполнять и выводить массив вручную нецелесообразно. Для заполнения массива случайными числами используется следующий алгоритм.

```
1 import random
2 for i in range(N):
3     for j in range(M):
4         matrix[i][j] = random.randint ( 30, 60 )
5         print ( "{:4d}".format(matrix[i][j]), end = "" )
```



Нумерация элементов двумерного массива, как и элементов одномерного массива, начинается с нуля.

Например, `matrix[2][3]` — это элемент третьей строки четвертого столбца.

Рассмотрим следующий пример. Необходимо заполнить и вывести на экран числа от 10 до 99 включительно.

```

a = [] #создаем пустой список
    k = 10 # начальное значение, может быть любым
for r in range(9): # 9 строк
    a.append([]) # создаем пустую строку
    for c in range(10): # в каждой строке – 10 элементов
        a[r].append(k) # добавляем очередной элемент в строку
        k += 1 # увеличиваем значение счетчика
for r in a:
    print(r)

```

Результат:

`C:\Programs\Python\Python36-32\python.exe`

```

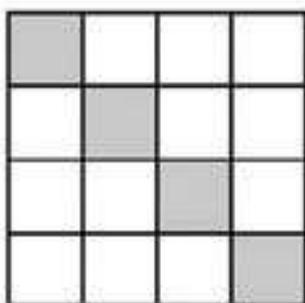
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
[20, 21, 22, 23, 24, 25, 26, 27, 28, 29]
[30, 31, 32, 33, 34, 35, 36, 37, 38, 39]
[40, 41, 42, 43, 44, 45, 46, 47, 48, 49]
[50, 51, 52, 53, 54, 55, 56, 57, 58, 59]
[60, 61, 62, 63, 64, 65, 66, 67, 68, 69]
[70, 71, 72, 73, 74, 75, 76, 77, 78, 79]
[80, 81, 82, 83, 84, 85, 86, 87, 88, 89]
[90, 91, 92, 93, 94, 95, 96, 97, 98, 99]

```

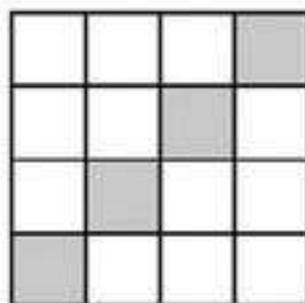
Обработка элементов матрицы

Рассмотрим, как можно обработать (например, сложить) некоторые элементы квадратной матрицы A , содержащей N строк и N столбцов.

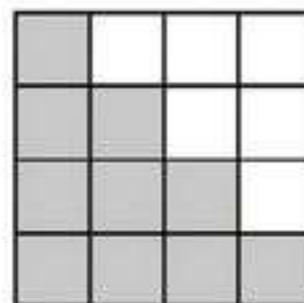
На рисунке *a* выделена главная диагональ матрицы, на рисунке *б* — вторая (побочная) диагональ, на рисунке *в* — главная диагональ и все элементы под ней.



a)



б)



в)

Главная диагональ — это элементы $A[0,0]$, $A[1,1]$, ..., $A[N-1,N-1]$, т.е. номер строки равен номеру столбца. Для перебора этих элементов нужен один цикл:

```
for i in range(N):
    # работаем с A[i][i]
```

Элементы побочной диагонали — это $A[0,N-1]$, $A[1,N-2]$, ..., $A[N-1,0]$. Заметим, что сумма номеров строки и столбца для каждого элемента равна $N-1$, поэтому получаем такой цикл перебора:

```
for i in range(N):
    # работаем с A[i][N-1-i]
```

В случае в (обработка всех элементов на главной диагонали и под ней) нужен вложенный цикл: номер строки будет меняться от 0 до $N-1$, а номер столбца для каждой строки i — от 0 до i :

```
for i in range(N):
    for j in range(i+1):
        # работаем с A[i][j]
```

Чтобы переставить столбцы матрицы, достаточно одного цикла. Например, переставим столбцы 2 и 4, используя вспомогательную переменную c :

```
for i in range(N):
    c = A[i][2]
    A[i][2] = A[i][4]
    A[i][4] = c
```

или, используя возможности Python:

```
for i in range(N):
    A[i][2], A[i][4] = A[i][4], A[i][2]
```

Переставить две строки можно вообще без цикла, учитывая, что $A[i]$ — это ссылка на список элементов строки i . Поэтому достаточно просто переставить ссылки. Оператор

```
A[i], A[j] = A[j], A[i]
```

переставляет строки матрицы с номерами i и j .

Для того, чтобы создать копию строки с номером i , нельзя делать так (подумайте, почему?):

```
R = A[i]
```

а вместо этого нужно создать копию в памяти:

```
R = A[i][:]
```

Построить копию столбца с номером j несколько сложнее, так как матрица расположена в памяти по строкам. В этой задаче удобно использовать генератор:

```
C = [ row[j] for row in A ]
```

В цикле перебираются все строки матрицы A , они по очереди попадают в переменную row . Генератор выбирает из каждой строки элемент с номером j и составляет список из этих значений.

С помощью генератора легко выделить в отдельный массив элементы главной диагонали:

```
D = [ A[i][i] for i in range(N) ]
```

Здесь предполагается, что матрица A состоит из N строк и N столбцов.

Контрольные вопросы



1. Что такое *двумерный массив*?
2. Как создать двумерный массив?
3. Что произойдет, если сложить два массива?

Практикум

Получены значения температуры воздуха за четыре дня с трех метеостанций, расположенных в разных регионах страны:

Номер станции	1-й день	2-й день	3-й день	4-й день
1	-8	-14	-19	-18
2	25	28	26	20
3	11	18	20	25

Т.е. запись показаний в двумерном массиве выглядела бы так:

```
t[1,1]:=-8;      t[1,2]:=-14;      t[1,3]:=-19;      t[1,4]:=-18;
t[2,1]:=25;      t[2,2]:=28;      t[2,3]:=26;      t[2,4]:=20;
t[3,1]:=11;      t[3,2]:=18;      t[3,3]:=20;      t[3,4]:=25;
```

Уровень А

1. Выведите на экран температуру на 2-й метеостанции за 4-й день и на 3-й метеостанции за 1-й день.
2. Создайте пустой массив:
 - Заполните его n -городами.
 - Отобразите каждый второй город из массива.

Уровень В

1. Выведите на экран показания термометров всех метеостанций за 2-й день.
2. Определите среднюю температуру на 3-й метеостанции.
3. Дан массив из N чисел. В первой строке дано число N . Затем в последующих N строках даны числа. Выведите все четные из данного массива.

Пример :

Ввод	Вывод:
6	4
4	2
2	10
3	
1	
10	
11	

Уровень С

1. Выведите на экран, в какие дни и на каких метеостанциях температура была в диапазоне $24—26^{\circ}\text{C}$ тепла.
2. Дан массив из N . В первой строке дано число N . В следующих N строках даны положительные и отрицательные числа массива. Посчитайте количество чисел меньше нуля.

Пример :

Ввод	Вывод :
5	3
2	
-45	

-1
23
-12

Рефлексия:

▶ Я узнал(а), что такое... и могу применить для того, чтобы...

§20–21 Сортировка

Вы научитесь:

▶ сортировать элементы в массиве.

Ключевые понятия:

▶ массив
▶ переменная

? Мы всегда можем отсортировать данные в программе в алфавитном порядке. Каким образом реализуется сортировка во время программирования?

В нашей жизни мы постоянно применяем сортировку. Например, когда ищем в списках себя. Нам легче найти, когда все данные отсортированы в алфавитном порядке: от большего к меньшему и наоборот. В случае, когда определенные данные отсортированы, мы легко можем применить определенную систему для поиска нужного элемента. Компьютеры также нуждаются в определенном порядке, они работают с данными. В программировании существует множество различных алгоритмов сортировки. Математик и теоретик информатики Дональд Эрвин Кнут для того, чтобы описать все алгоритмы сортировки, посвятил целый том только методам сортировки в программировании. Мы рассмотрим самый простейший. Также научимся применять один из быстрых методов сортировки, который существует в языке программирования *Python*.

Сортировка пузырьком. Одной из простейших в понимании и реализации сортировок является сортировка пузырьком. Суть алгоритма заключается в том, что так же, как и в воде, пузырек стремится на поверхность, т. е. большие числа на каждом проходе стремятся к

концу списка в массиве. Реализация выглядит следующим образом. У нас есть массив из 6 чисел.

```
nums = [6, 3, 4, 1, 9, 8]
```

С помощью двух циклов можно пройти по каждому элементу массива, переставляя меньший элемент вверх.

```
for i in range(6):
    for j in range(5, i, -1):
        if nums[j-1] > nums[j]:
            temp = nums[j-1]
            nums[j-1] = nums[j]
            nums[j] = temp
```

Сначала сравниваем последний элемент с предпоследним. Если они стоят неправильно (меньший элемент “ниже”), то меняем их местами. Далее так же рассматриваем следующую пару элементов и т.д., как на рисунке 16.1.

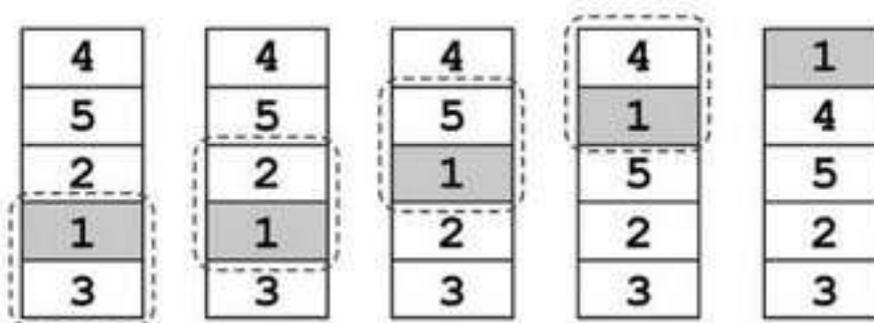


Рис. 16.1. Пример сортировки пузырьком

Функция `sort()`. Как вы уже знаете, *Python* имеет готовые функции, что упрощает работу программиста. Функция `sort()` позволяет отсортировать массив за очень короткое время. Реализация данной функции выглядит следующим образом.

Дан массив из N чисел. Заполните массив числами.

```
nums = [ int( i )
         for i in split().input() ]
```

Затем примените функцию `sort()` к массиву `nums` и выведите массив.

```
nums.sort()
for i in nums:
    print( i , end=" ")
```

Практикум

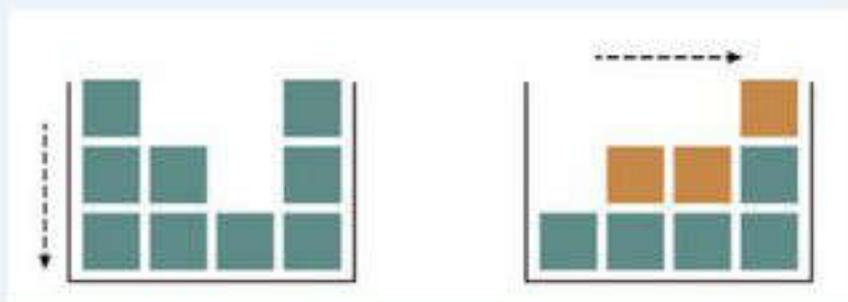
Уровень А

1. Дан массив из N чисел. Отсортируйте по убыванию данный массив и выберите его.

Пример :

Ввод	Вывод
6	8 7 6 4 3 2
3 2 4 8 6 7	

2. В коробке находятся n столбиков игрушечных кубиков, расположенных в ряд. Изначально гравитация в коробке тянет все кубики вниз. Когда Крис переключает гравитацию, она начинает тянуть все кубики к правой стенке коробки. Рисунок показывает начальное и конечное расположение кубиков в коробке: кубики, изменившие свои позиции, выделены оранжевым цветом.



Вам дано изначальное расположение игрушечных кубиков в коробке. Напишите программу, которая найдет количество кубиков в каждом из n столбиков после переключения гравитации.

Пример :

Ввод	Вывод
4	1 2 2 3
3 2 1 2	

Уровень В

1. На Диаса возложена ответственная задача — поддержать только что присоединившихся олимпийцев по программированию.

Сейчас у него имеется n олимпийцев, которым необходимо предоставить поддержку. Каждый олимпиец имеет собственный рейтинг. Диас предпочитает поддерживать олимпийца, если суще-

ствуется хотя бы один из них с рейтингом строго меньшим, чем его, и хотя бы один олимпиец с рейтингом строго большим, чем у него.

Напишите программу: сколько олимпийцев должен поддержать Диас.

Пример 1:

Ввод	Вывод
2	0
1 5	

Пример 2:

Ввод	Вывод
3	1
1 2 5	

Примечание

В первом тестовом примере Диас не может поддержать олимпийца с рейтингом 1, потому что нет олимпийца с рейтингом меньше, чем 1. Также он не может поддержать олимпийца с рейтингом 5, потому что нет олимпийца с рейтингом больше, чем 5.

Во втором тестовом примере Диас может поддержать олимпийца с рейтингом 2, потому что есть олимпиец с рейтингом меньше, чем 2, и есть олимпиец с рейтингом больше, чем 2.

2. Определите, сколько обменов сделает алгоритм пузырьковой сортировки по возрастанию для данного массива.

Пример 1:

Ввод	Вывод
5	0
1 2 3 4 5	

Пример 2

Ввод	Вывод
5	10
5 4 3 2 1	

Уровень С

1. У Алдияра есть n флэш-карт объемом a_1, a_2, \dots, a_n гигабайт и большой файл размера m гигабайт. Найдите минимальное количество флэш-карт, на которые Алдияр может записать файл, если он может разделить свой файл на части произвольного размера.

В первой строке дается число n , во второй — число m , в третьей строке — n флэш карт.

Пример 1:

Ввод	Вывод
3	2
5	
2 1 3	

Пример 2:

Ввод	Вывод
3	3
6	
2 3 2	

Примечание

В первом тестовом примере Алдияру нужно 2 флеш-карты — первая и третья.

Во втором тестовом примере Алдияру нужны все 3 флеш-карты.

2. В одном государстве на прямой Ox живут три друга. Первый живет в точке x_1 , второй — в точке x_2 , а третий — в точке x_3 . Они собираются вместе встретить Новый год, для чего им необходимо всем оказаться в одной точке. Напишите программу, которая вычислит, какое минимальное суммарное расстояние им надо пройти, чтобы собраться вместе в одной точке и встретить Новый год?

Пример 1:

Ввод	Вывод
7 1 4	6

Пример 2:

Ввод	Вывод
20 30 10	20

Контрольные вопросы



1. Как работает сортировка пузырьком?
2. Какая функция в языке программирования *Python* сортирует элементы в массиве?

Рефлексия:

Я узнал(а), что такое... и могу применить для того, чтобы...

§22–23

Удаление и вставка элемента

Вы научитесь:

- составлять программы инвертированной среды разработки с использованием одномерных массивов.

Ключевые понятия:

- массив
- элемент
- индекс



Каковы преимущества использования массивов в программировании?

Библиотека языка программирования Python включает в себя множество готовых функций. Подробную информацию об их использовании можно найти на данном сайте: <https://docs.python.org/>. Эти готовые функции во многом упрощают работу программиста, что делает программирование на языке *Python* намного легче.

Функция `index`. Функция `index` возвращает индекс элемента из массива. Если элемент отсутствует, то в таком случае программа возвращает ошибку. Например:

```
numbers = [ 4, 5, 34, 23, 121]
print( numbers.index(34) )
```

Вывод:

2

Функция `append`. В предыдущих параграфах мы рассматривали методы добавления в массив с помощью метода `append`.

```
numbers += [int (input() )]
```

Мы модифицировали наш массив с помощью сложений данного массива с другим массивом, содержащим вводимое число.

Функция `append` упрощает добавление в конец массива элементов. Выглядит это следующим образом:

```
numbers = [ 1, 2, 45, 3, 6, 4]
numbers.append( 5 )
print (numbers)
```

Вывод:

[1, 2, 45, 3, 6, 4, 5]

Тем самым, в конец массива мы добавили новый элемент. Также добавить элемент можно следующим образом:

```
numbers.append( int (input () ) )
```

Удалить элемент из массива можно функцией *remove ()*.

Функция extend. Переменные в языке программирования *Python* хранят разного рода объекты. Например, также можно хранить массив внутри массива следующим образом:

```
numbers = [4,5,2, [3,4],6]
```

Как мы видим, под 3-м индексом у нас хранится массив [3, 4]. Чтобы сослаться на элементы такого рода, мы используем следующий метод:

```
print( number [3][0] )
```

Если функция *append*, то она добавляет в конец массива любой объект, в том числе и массив, например:

```
numbers = [ 1,2,3 ]
numbers.append( [3,4] )
print(numbers)
```

Вывод:

```
[1,2,3,[3,4]],
```

то функция *extend* сначала разделяет массив на отдельные элементы, затем добавляет их в массив.

Например:

```
numbers = [ 1,2,3 ]
numbers.extend( [3,4] )
print(numbers)
```

Вывод:

```
[ 1,2,3,3,4 ]
```

Функция split. Данная функция разделяет текст на отдельные слова и хранит их в массиве. Например:

```
text = "Hello world! I like coding"
text_new = text.split()
print(text_new)
```

Вывод:

```
['Hello', 'world!', 'I', 'like', 'coding']
```

Как вы видите, в примере была переменная `text`. Затем была создана переменная `text_new` и в ней сохранились слова по отдельности в виде массива. Функцию `split` также можно использовать, если ввести числовые значения в одной строке, так как язык `Python`, в основном, используется при веб-программировании. Объекты в одной строке Python рассматривает как текст, но, используя функцию `split`, текст сначала можно разделить на отдельные слова, затем меняем строковый тип на числовой.

Например :

```
numbers = [ int(i) for i in input().split() ]
```



Контрольные вопросы

1. Какая разница между функциями `append` и `extend`?
2. Как можно ввести числа одной строкой в Python, чтобы числа хранились в виде числовых значений?

Практикум

Уровень А

Задание 1

Создать массив из N чисел.

Использовать функцию `append` для добавления элементов в массив.

Вычислить сумму элементов массива.

Задание 2

Создать массив, который будет хранить текст.

Посчитать количество слов во введенном тексте.

Уровень В

Напишите программу. В одной строке вам даны числа. Вычислить среднее значение данных чисел.

Пример :

Ввод

4 5 6 8 7

Вывод

6

Уровень С

Напишите программу. Даны числа в одной строке. Посчитайте количество чисел, которые больше своих рядом стоящих соседей.

Пример :

Ввод

1 3 2 4 3

Вывод

2

Рефлексия:

▮ Я узнал(а), что такое... и могу применить для того, чтобы...

РАЗДЕЛ
5

Создание 2D игры на языке программирования Python

Из данного раздела вы узнаете:

- ▶ как создавать 2D-игры;
- ▶ о библиотеке PyGame;
- ▶ как анимировать персонажи.

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;
- ▶ программировать движение персонажа, управлять персонажами с клавиатуры;
- ▶ разрабатывать игру по готовому сценарию;
- ▶ реализовать алгоритм подсчета результатов игры.

§24–25 Библиотека PyGame

Вы научитесь:

- ▶ создавать 2D-игры;
- ▶ подключать библиотеку PyGame;
- ▶ использовать готовые модули библиотеки PyGame для создания окна для игры.

Ключевые понятия:

- ▶ библиотека PyGame
- ▶ функция `pygame.display.set_mode(resolution=(0,0), flags=0, depth=0)`
- ▶ функция `pygame.font.SysFont(name, size, bold=False, italic=False)`
- ▶ функция `pygame.display.update()`

Язык Python имеет несколько библиотек, что дает возможность дополнять его различным функционалом. Используя ту или иную

библиотеку, можно в коде программы не выполнять скучные и повторяющиеся действия.

Библиотека PyGame — специальная библиотека для разработки игр с графическим интерфейсом на языке Python. Преимущество PyGame состоит в том, что приложения, разработанные с помощью данной библиотеки, будут работать на мобильных устройствах и компьютерах. С помощью PyGame уже разработано множество игр и приложений (список можно посмотреть на официальном сайте www.pygame.org).

Чтобы установить библиотеку PyGame, необходимо зайти на страничку загрузки PyGame (<http://www.pygame.org/download.shtml>) и выбрать библиотеку, доступную для загрузки.

* Примечание. На момент издания учебника наиболее популярна ветка Python 2.x, чем 3.x (последняя версия ветки 2.x — 7, а ветки 3.x — 7). Все примеры приведены для версии 2.6.

Создадим окно для игры. Для этого требуется инициализировать PyGame, но сначала импортируем эту библиотеку в наш код. Введем следующий код:

```
import pygame, sys
from pygame.locals import *
```

После импорта библиотеки ее можно инициализировать. Для этого используется следующий код.

```
pygame.init()
```

Теперь создадим окно, в котором будет формироваться кадр. Для этого используется функция:

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

— resolution — список чисел, в котором хранится размер окна по оси x и y;

— flag — флаг окна (полный экран, изменяемые размеры и т.д),

— depth — глубина цвета.

```
screen = pygame.display.set_mode((400,100), 0, 32)
```

В нашем примере инициализируется окно размером 400 x 100 с глубиной цвета — 32.

Зададим заголовок для окна. Для этого используется функция:

```
pygame.display.set_caption(title, icontitle=None), где
```

- title — строка с заголовком окна;
- icontitle — иконка в заголовке.

```
pygame.display.set_caption("My First PyGame Windows")
```

Для ввода текста в окне необходимо знать координаты текста и размер шрифта. Для текста определим переменную `helloText`, а для координат — переменные `x` и `y`, для размера шрифта — `fontSize`. Вот что получилось:

```
helloText = "Hello, World and GCUP"
(x, y, fontSize) = (10, 40, 14)
```

Параметры шрифта задаются функцией:

```
pygame.font.SysFont(name, size, bold=False, italic=False), где
```

- name — название шрифта;
- size — кегль;
- bold/italic — начертание.

```
myFont = pygame.font.SysFont("None", fontSize)
```

В данном примере используется стандартный шрифт с названием "None" и размером, который хранится в переменной `fontSize`.

Для цвета шрифта используется функция `fontColor`. Числа — это вектор цвета в системе RGB (0 — минимальное значение, 255 — максимальное). Пример:

```
fontColor = (255, 255, 0)
```

Функция `bgColor` задает цвет фона окна. Пример:

```
bgColor = (255, 255, 255)
```

Для преобразования текста в изображение используется функция `fontImage`:

```
fontImage = myFont.render(helloText, 0, (fontColor))
```

- myFont — преобразует текст в картинку;
- text — текст;
- antialias — сглаживание шрифта;
- color — цвет текста;
- background — цвет фона (не обязательный параметр).

Теперь следует сформировать кадр. Для этого зальем кадр цветом заднего фона, через функцию

```
fill(color,rect=None, special_flags=0)
```

объекта screen. Пример:

```
screen.fill(bgColor)
```

Для отображения кадра используется функция

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

Для данной функции никакие параметры не используются. Пример:

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

Первая программа на PyGame готова. Вот что получилось:

```
import pygame, sys
from pygame.locals import *
pygame.init()
screen = pygame.display.set_mode((400,100),0,32)
pygame.display.set_caption("My First PyGame Windows");
helloText = "Hello, World and GCUP"
(x,y,fontSize) = (10,40,14)
myFont = pygame.font.SysFont("None", fontSize)
fontColor = (255,255,0)
bgColor = (255,255,255)
fontImage = myFont.render(helloText, 0, (fontColor))
mainLoop = True
while mainLoop:
    for event in pygame.event.get():
        if event.type == QUIT:
            mainLoop = False
    screen.fill(bgColor)
    screen.blit(fontImage, (x,y))
    pygame.display.update()
pygame.quit()
```

Контрольные вопросы



1. В чем преимущества библиотеки PyGame?
2. Какие функции предоставляет библиотека PyGame и что вы можете получить, если будете ее использовать?

Практикум

Уровень А

Наберите программу, приведенную в параграфе. Откомпилируйте эту программу и запустите ее.

Уровень В

В коде задачи (из уровня А) измените текст, параметры шрифта. Откомпилируйте эту программу и запустите ее.

Уровень С

Создайте окно размером 600 пикселей по высоте и 300 пикселей по ширине и введите произвольный текст, например, свои ФИО, задайте параметры шрифта (гарнитуру, кегль, начертание, цвет).

Рефлексия:

▶ Я узнал(а), что такое... и могу применить для того, чтобы...

§26–27

Задний фон и персонажи игры. Анимирование персонажей

Вы научитесь:

- ▶ анимировать персонажи;
- ▶ программировать движение персонажа игры;
- ▶ управлять персонажами с клавиатуры;
- ▶ разрабатывать свою игру.

Ключевые понятия:

- ▶ анимирование персонажей
- ▶ программирование движений
- ▶ управление персонажами

Для того, чтобы анимировать персонажей и задавать фон в Python, рассмотрим следующий пример:

```
1 import sys, pygame
2 pygame.init()
3
4 size = width, height = 320, 240
```

```

5   speed = [2, 2]
6   black = 0, 0, 0
7
8   screen = pygame.display.set_mode(size)
9
10  ball = pygame.image.load("ball.bmp")
11  ballrect = ball.get_rect()
12
13  while 1:
14      for event in pygame.event.get():
15          if event.type == pygame.QUIT: sys.exit()
16
17          ballrect = ballrect.move(speed)
18          if ballrect.left < 0 or ballrect.right > width:
19              speed[0] = -speed[0]
20          if ballrect.top < 0 or ballrect.bottom > height:
21              speed[1] = -speed[1]
22
23          screen.fill(black)
24          screen.blit(ball, ballrect)
25          pygame.display.flip()

```

Разберем текст программы:

1-я строка — импорт-пакет, в котором все доступные модули Pygame.

2-я строка — инициализация каждого из этих модулей.

8-я строка — открываем графическое окно.

10-я строка — загрузка изображения мяча. После загрузки изображения мяча объявление переменной ballrect. Pygame поставляется с удобным утилитарным типом объекта, названным Rect, который представляет прямоугольную область. Позднее, в анимационной части кода, мы увидим, чем занимается Rect-объект.

13-я строка — программа считается готовой. Внутри цикла проверяем пользовательский ввод, движение мяча и его отскок.

17—21-е строки — двигаем переменную ballrect с текущей скоростью. Если мяч двигается за границы экрана, мы обращаем скорость в том направлении.

23-я строка — очищаем экран, заполняя его черным цветом.

24-я строка — рисуем изображение мяча на экране. Обрисовка изображения обрабатывается методом "Surface.blit()",

И последнее, что необходимо сделать — обновить видимый дисплей. Pygame управляет дисплеем посредством двойного буфера. Затем, когда мы заканчиваем прорисовку, вызываем метод “pygame.display.flip()”. Он делает все, что мы обрисовывали на поверхности экрана, видимым.



Контрольные вопросы

1. Как изменить фон окна в Python?
2. Как загрузить готовые объекты для игры?

Практикум

Уровень А

Наберите программу, приведенную в параграфе. Откомпилируйте эту программу и запустите ее.

Уровень В

В коде задачи (из уровня А) изменить фон, цвет окна разрабатываемой игры.

Уровень С

В коде задачи (из уровня А) изменить фон (пусть он будет в виде травы) и, для более реалистичного эффекта, для мяча добавьте небольшую тень.

Рефлексия:

- ▶ Я узнал(а), что такое... и могу применить для того, чтобы...

§28–29 Программирование условий

Вы научитесь:

- ▶ Программировать и создавать движущиеся игровые элементы;
- ▶ управлять игровыми элементами с помощью мыши;
- ▶ улучшить навыки разработки игр.

Ключевые понятия:

- ▶ анимирование игровых элементов
- ▶ создание игровых объектов и элементов
- ▶ условный оператор
- ▶ сложные условия

Практически любая программа содержит в себе условные конструкции. Вследствие изменения каких-либо условий программа может реагировать на эти изменения, выполняя определенный участок кода. Условные операторы и конструкции дают возможность выбрать один из вариантов выполнения действий в зависимости от каких-либо условий. Простейшая условная конструкция выглядит следующим образом:

Если <выполняется условие> делать:
какие-то действия.

Например:

```
if a<b:
    print(a+b)
else:
    print(a-b)
```

“Если $a < b$, то вывести сумму $a + b$, в противном случае вывести разность $a - b$ ”

Итак, кратко вспомнив условные конструкции, давайте напишем и разберем код программы “CollidingBalls”, которая представляет собой простую 2D-игру. Для создания игры будем использовать условные конструкции и библиотеки `tkinter` и `random`.

- Создадим окно и научимся перехватывать события от мыши.
- Создадим основной игровой объект — цветной шарик. Заставим его появляться на месте щелчка мышью.
- Научим шарик двигаться и отскакивать от края экрана. Освоим управление движущимся шариком с помощью мыши.
- Создадим “россыпь” неподвижных шаров, которые наш шарик будет “выбивать”. Научимся обрабатывать столкновения.

• Определим “опасные” шары. Если шарик их касается — мы проигрываем.

Рассмотрим код программы. В строках 1—13 описываются основные свойства окна и мячей, которые мы создадим программно и где будет проходить сама игра.

```

1. HEIGHT = 480
2. BG_COLOR = 'white'
3. MAIN_BALL_COLOR = 'blue'
4. MAIN_BALL_RADIUS = 25
5. BAD_COLOR = 'red'
6. COLORS = ['aqua', 'fuchsia', BAD_COLOR, 'pink',
             'yellow', BAD_COLOR, 'gold', 'chartreuse', BAD_COLOR]
7. NUM_OF_BALLS = 9
8. MAX_RADIUS = 35
9. MIN_RADIUS = 15
10. DELAY = 8
11. INIT_DX = 1
12. INIT_DY = 1
13. ZERO = 0

```

В строках 14—46 описывается класс ball, внутри которого описываются основные свойства и поведения мячей, а также их взаимодействия друг с другом и с границами окна.

```

14. # ball class
15. class Ball():
16. def __init__(self, x, y, r, color, dx=0, dy=0):
17. self.x = x
18. self.y = y
19. self.r = r
20. self.color = color
21. self.dx = dx
22. self.dy = dy
23. def draw(self):
24. canvas.create_oval(self.x - self.r, self.y -
                      self.r, self.x + self.r, self.y + self.r, fill=self.
                      color,
                      outline=self.color if self.color != BAD_COLOR else
                      'black')
25. def hide(self):

```

```

26. canvas.create_oval(self.x - self.r, self.y -
    self.r, self.x + self.r, self.y + self.r, fill=BG_
    COLOR, outline=BG_COLOR)
27. def is_collision(self, ball):
28. a = abs(self.x + self.dx - ball.x)
29. b = abs(self.y + self.dy - ball.y)
30. return (a * a + b * b) ** 0.5 <= self.r + ball.r
31. def move(self):
32. # collision with the walls (Столкновение со стеной)
33. if (self.x + self.r + self.dx >= WIDTH) or (self.x -
    self.r + self.dx <= ZERO):
34. self.dx = -self.dx
35. if (self.y + self.r + self.dy >= HEIGHT) or (self.y
    - self.r + self.dy <= ZERO):
36. self.dy = -self.dy
37. # therballscollision (Столкновение с другими мячами)
38. for ball in balls:
39. if self.is_collision(ball):
40. if ball.color != BAD_COLOR: # not a bad ball
        ball.hide()
        balls.remove(ball)
        self.dx = -self.dx
        self.dy = -self.dy
41. else: # badball (реакция на столкновение с плохим мя-
        чиком, т. е. при столкновении с которым мы проигрываем)
        a. self.dx = self.dy = 0
42. self.hide()
43. self.x += self.dx
44. self.y += self.dy
45. if self.dx * self.dy != 0:
46. self.draw()

```

В строках 47—67 описывается реакция игры на взаимодей-
ствие с компьютерной мышью, т. е. что происходит при щелчке
мышью, и т. д.)

```

47. # process the mouse events
48. def mouse_click(event):
49. global main_ball
50. if event.num == 1: # left mouse button
51. if 'main_ball' not in globals(): # старт

```

```

52. main_ball = Ball(event.x, event.y, MAIN_BALL_RADIUS,
    MAIN_BALL_COLOR, INIT_DX, INIT_DY)
53. if main_ball.x > WIDTH / 2:
54.     main_ball.dx = -main_ball.dx
55. if main_ball.y > HEIGHT / 2:
56.     main_ball.dy = -main_ball.dy
57. main_ball.draw()
58. else: # turn left
59. if main_ball.dy * main_ball.dx > 0:
60.     main_ball.dy = -main_ball.dy
61. else:
62.     main_ball.dx = -main_ball.dx
63. elif event.num == 3: # right mouse button: turn right
64. if main_ball.dy * main_ball.dx > 0:
65.     main_ball.dx = -main_ball.dx
66. else:
67.     main_ball.dy = -main_ball.dy

```

В строках 68—81 описывается программный генератор случайной россыпи мячей со случайными свойствами в пределах игрового окна.

```

68. # create a list of objects-balls
69. def create_list_of_balls(number):
70.     lst = []
71.     while len(lst) < number:
72.         next_ball = Ball(random.choice(range(MAX_RADIUS,
            WIDTH - MAX_RADIUS)),
            random.choice(range(MAX_RADIUS, HEIGHT - MAX_
            RADIUS)),
            random.choice(range(MIN_RADIUS, MAX_RADIUS)),
            random.choice(COLORS))
73.         is_collision = False
74.         for ball in lst:
75.             if next_ball.is_collision(ball):
76.                 is_collision = True
77.                 break
78.         if not is_collision:
79.             lst.append(next_ball)
80.         next_ball.draw()
81.     return lst

```

В строках 82—88 описывается программная реализация генератора плохих мячей, при столкновении с которыми игра будет завершена, а игрок проиграет.

```
82. # count the number of bad balls
83. def count_bad_balls(list_of_balls):
84. result = 0
85. for ball in list_of_balls:
86. if ball.color == BAD_COLOR:
87. result += 1
88. return result
```

В строках 89—98 описывается главный программный цикл.

```
89. # games main loop
90. def main():
91. if 'main_ball' in globals():
92. main_ball.move()
93. if len(balls) - num_of_bad_balls == 0:
94. canvas.create_text(WIDTH / 2, HEIGHT / 2, text=»YOU
    WON!», font=»Arial 20», fill=»lime»)
95. main_ball.dx = main_ball.dy = 0
96. elif main_ball.dx * main_ball.dy == 0:
97. canvas.create_text(WIDTH / 2, HEIGHT / 2, text=»YOU
    LOSE!», font=»Arial 20», fill=»red»)
98. root.after(DELAY, main)
```

В строках 99—112 описывается генерация игрового окна с заданными в строках 1—13 свойствами

```
99. # create a window, the canvas and start game
100. root = tkinter.Tk()
101. root.title(«Colliding Balls»)
102. canvas = tkinter.Canvas(root, width=WIDTH,
    height=HEIGHT, bg=BG_COLOR)
103. canvas.pack()
104. canvas.bind('<Button-1>', mouse_click)
105. canvas.bind('<Button-2>', mouse_click, '+')
106. canvas.bind('<Button-3>', mouse_click, '+')
107. balls = create_list_of_balls(NUM_OF_BALLS)
108. num_of_bad_balls = count_bad_balls(balls)
109. if 'main_ball' in globals(): # for restarts
```

```
110. del main_ball
111. main()
112. root.mainloop()
```



Контрольные вопросы

1. Какая библиотека нужна для работы с окнами в Python?
2. Какая библиотека нужна для генерации случайных элементов в Python?
3. Можно ли, используя условные конструкции, управлять поведением игровых элементов?

Практикум

Уровень А

Наберите программу, приведенную в параграфе. Откомпилируйте эту программу и запустите ее.

Уровень В

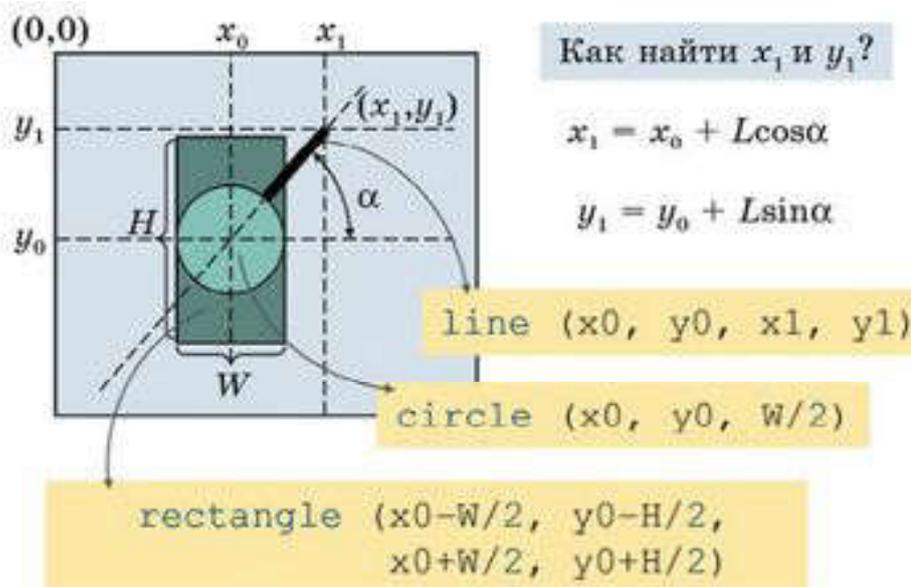
В коде задачи (из уровня А) измените свойства генерируемого окна `BG_COLOR`, `HEIGHT`, поэкспериментируйте с другими параметрами окна.

Уровень С

В коде задачи (из уровня А) измените свойства генерируемых мячей и их поведение на взаимодействие с мышью.

Проектная работа. Разработка 2D-игры “Танк с вращающейся пушкой”

Постановка задачи. Описание математической модели.



Описание программы

```

1: from graph import *
2: import math
3: H = 60; W = 30; L = 40 # размеры танка
4: x0 = 200; y0 = 400; angle = 90 # пушка
5: brushColor("#6b8e23")
6: rectangle(x0-W/2, y0-H/2, x0+W/2, y0+H/2)
7: a = angle*math.pi/180 # в радианы
8: x1 = x0+L*math.cos(a)
9: y1 = y0-L*math.sin(a)
10: penSize(5)
11: line(x0, y0, x1, y1)
12: penSize(1)
13: brushColor("#556b2f")
14: circle(x0, y0, W/2)
15: run()

```

Пояснение к коду:

6-я строка – описание корпуса танка;

7—11-е строки – ствол;

12—15-е строки – башня танка.

Для анимации поворота пушки используется следующий фрагмент кода

```

1: def keyPressed(event):
2:     if event.keycode == VK_LEFT:
3:         drawGun(angle+5) # влево на 5 градусов
4:     elif event.keycode == VK_RIGHT:
5:         drawGun(angle-5) # вправо на 5 градусов
6:     elif event.keycode == VK_ESCAPE:
7:         close()
8:     ...
9: onKey(keyPressed)

```

Рисование и вращение пушки. Идея: в первый раз рисуем, потом просто меняем координаты.

```

1: def drawGun(angleNew):
2:     global angle, gun # глобальные переменные
3:     angle = angleNew # запомнить новый угол
4:     aRad = angle*math.pi/180 # в радианы
5:     x1 = x0 + L*math.cos(aRad)
6:     y1 = y0 - L*math.sin(aRad)
7:     if gun == None: # если в первый раз...
8:         gun = line(x0, y0, x1, y1)
9:     else: # если пушка уже нарисована
10:         changeCoord(gun, [(x0,y0), (x1,y1)] ) # массив
        новых координат

```

Рисование и вращение пушки. Как это работает:

```

1: gun = None # еще не рисовали пушку
2: drawGun(angle) # рисуем в первый раз
3:
4:     gun = line(x0, y0, x1, y1)
5:     # теперь gun содержит адрес линии
6:     ...
7: def keyPressed(event):
8:     drawGun(angle+5) # вращаем
9:
10: changeCoord(gun, [(x0,y0), (x1,y1)] )
11:     # просто меняем координаты

```

Таким образом, представим полную программу

```

1: from graph import *
2: import math

```

```

3: def keyPressed(event):
    ...
4: def drawGun(angleNew):
    ...
5: H = 60; W = 30; L = 40
6: x0 = 200; y0 = 400; angle = 90
7: gun = None
8: brushColor("#6b8e23")
9: rectangle(x0-W/2, y0-H/2, x0+W/2, y0+H/2)
10: penSize(5)
11: drawGun(angle)
12: penSize(1)
13: brushColor("#556b2f")
14: circle(x0, y0, W/2)
15: onKey(keyPressed)
16: run()

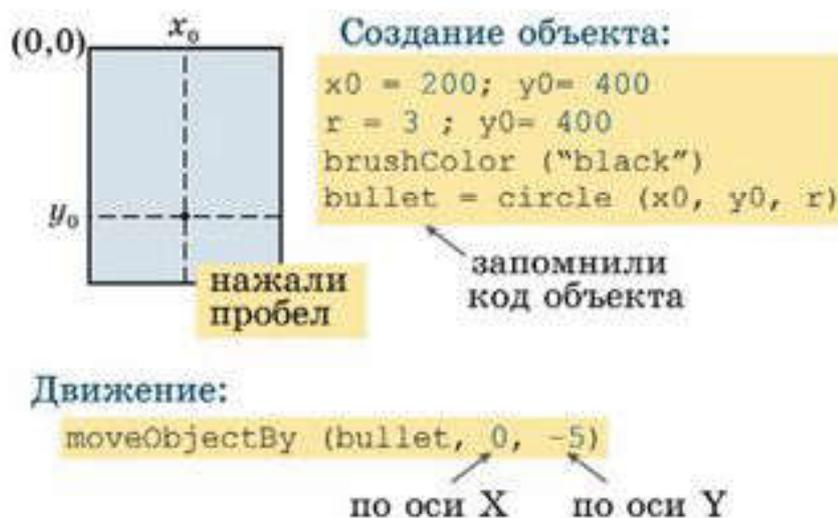
```

Пояснение к коду:

- 2—3-е строки – процедуры;
- 4—6-е строки – начальные значения;
- 8-я строка – корпус;
- 9—10-е строки – пушка;
- 11—14-е строки – башня.

Рассмотрим алгоритм стрельбы из пушки

Постановка задачи. Описание математической модели.



Как организовать анимацию?

```

1: def update():
2:   global isFlying, bullet

```

```

3:         if isFlying: # если летит...
4:             y = coords(bullet)[1]
5:             if y < 0: # если улетел...
6:                 isFlying = False
7:             moveTo(bullet, x0-r, y0-r)
8:     else: # летит дальше...
9:         moveObjectBy(bullet, 0, -5)
...
10: onTimer(update, 30) # вызывается каждые 30 мс

```

Как запустить движение?

```

1: def keyPressed(event):
2:     global isFlying
3:     if event.keycode == VK_SPACE:
4:         isFlying = True # полетели!
5:     elif event.keycode == VK_ESCAPE:
6:         close() # закончить работу
...
7: onKey(keyPressed)

```

Полная программа

```

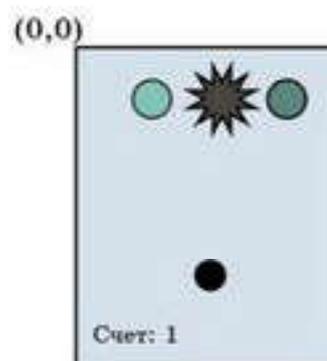
1: from graph import *
2: def update():
...
3: def keyPressed(event):
...
4: x0 = 200; y0 = 400; r = 3
5: brushColor("black")
6: bullet = circle(x0, y0, r)
7: isFlying = False
8: onKey(keyPressed)
9: onTimer(update, 30)
10: run()

```

Стрельба по тарелкам

Постановка задачи. Описание математической модели:

1. Создать объекты тарелки createPlates.
2. Двигать тарелки movePlates.
3. Проверить попадание в какую-нибудь тарелку checkCollision.
4. Проверить попадание в конкретную тарелку hit.



Создание массива тарелок

```

1: def createPlates( N ):
2:     global plates # глобальный массив
3:     yPlates = 100 # у всех одна у-координата
4:     plates = [] # пока массив пустой
5:     for i in range(N):
6:         brushColor( randColor() )
7:         p = circle(randint(0,500), # x центра
8:                 yPlates, # у центра
9:                 randint(10,20)) # радиус
10:    plates.append(p) # добавить в массив
...
11: createPlates( 5 ) # вызов процедуры

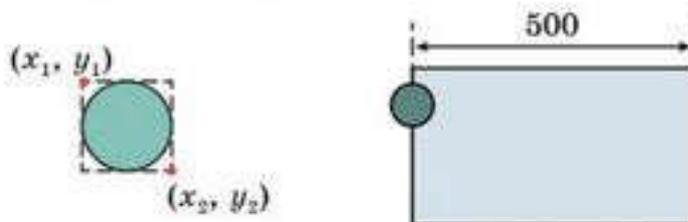
```

Движение тарелок

```

1: def movePlates():
2:     global plates # глобальный массив
3:     for p in plates: # для каждой тарелки
4:         moveObjectBy(p, -2, 0) # сдвиг на 2 влево
5:         x1,y1,x2,y2 = coords(p)
6:         if x1 < 0: # если вышла за границу...
7:             # перескочить вправо на ...
8:             moveObjectBy(p, randint(500,600), 0)

```



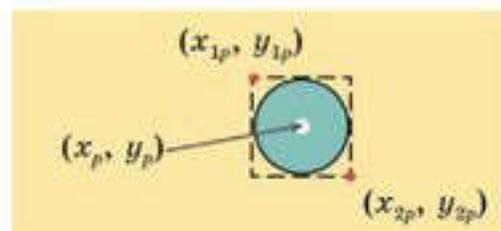
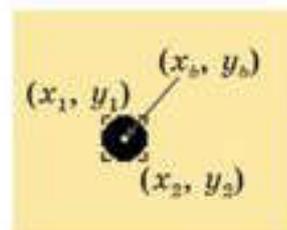
Попал ли снаряд в данную тарелку?



```

1: def hit(p):
2:     global bullet
3:         # координаты снаряда
4:     x1,y1,x2,y2 = coords(bullet)
5:     xb = x1 + r # центр снаряда
6:     yb = y1 + r
7:         # координаты тарелки
8:     x1p,y1p,x2p,y2p = coords(p)
9:     xp = (x1p + x2p) / 2
10:    yp = (y1p + y2p) / 2
11:    Rp = (x2p - x1p) / 2
12:    d2 = (xb-xp)**2 + (yb-yp)**2
13:    return d2 <= (Rp+r)**2

```



Как вызывать эти функции?

```

1: def update():
2:     global isFlying, bullet
3:     movePlates()
4:     if isFlying: # если летит...
5:         y = coords(bullet)[1]
6:         if y < 0: # если улетел...
7:             isFlying = False
8:             moveTo(bullet, x0-r, y0-r)
9:         else: # летит дальше...
10:            moveBy(bullet, 0, -5)
11:            checkCollision()
...
12: onTimer(update, 30) # вызывается каждые 30 мс

```

Как вывести счет игры?



Как и когда изменяется score?

Сначала: `score = 0`

При попадании: `score += 1`

Метка (элемент типа label)

Создание метки:

```
lbl = label("Счет: 0", (10, 200), bg = "white")
```

(x,y)

фон

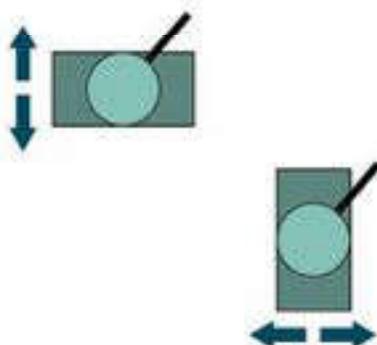
Изменение текста метки:

```
lbl["text"] = "Счет: " + str(score)
```

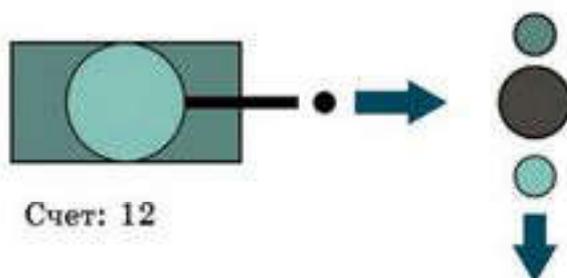
Строка из числа

Дополнительные задания:

1. Сделать два танка, у одного пушка управляется клавишами “влево” и “вправо”, у второго — клавишами “вверх” и “вниз”.

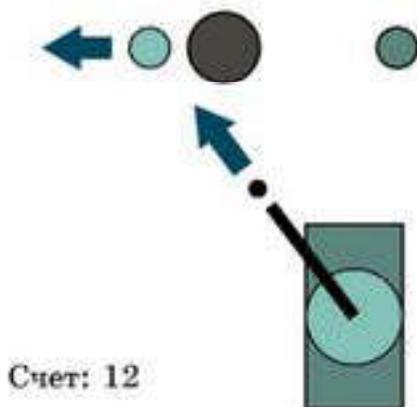


2. Выполнить задание, когда в случае стрельбы слева направо (тарелки летят сверху вниз). Дорисовать танк, из дула которого вылетает снаряд.



Счет: 12

3. Сделать танк с вращающейся пушкой. Снаряд вылетает из ствола в том же направлении. За попадание в более мелкую тарелку игрок получает больше баллов.



Счет: 12

Рефлексия:

▶ Я узнал(а), что такое... и могу применить для того, чтобы...

ГЛОССАРИЙ

E-mail — электронная почта, один из основных видов сети Интернет.

URL — адрес страницы или файла в Интернете.

Web-страница — основной структурный элемент “Всемирной паутины” — документ, который содержит текстовую и (или) графическую информацию, а также ссылки на другие документы Internet.

Алгоритм — совокупность правил выполнения определенных действий, обеспечивающих решение задачи.

Алфавит — определенный набор символов, а символы называются буквами алфавита.

Алфавитно-цифровая информация — символы кодовой таблицы, которые можно ввести в ПК одним нажатием клавиши (буквы, цифры, служебные знаки).

Архитектура компьютера — это описание его организации и принципов функционирования его структурных элементов. Включает основные устройства персонального компьютера и структуру связей между ними.

Байт — совокупность из восьми *бит*, воспринимаемая компьютером как единое целое.

База данных (БД) — информационная модель, которая позволяет в упорядоченном виде хранить данные о группе объектов, обладающих одинаковым набором свойств.

Бит — наименьшая единица информации, известная в природе. Значение бита — 0 или 1, и это можно толковать как альтернативу: “выключено — включено”, “нет — да”, “ложь — истина”.

Буфер обмена — область памяти, которую Windows выделяет на время своей работы. Предназначен для обмена текстовой и графической информацией между документами и приложениями.

Входная информация — информация об объектах, которую получает человек или устройство.

Выражение — форма записи некоторого высказывания. Выражение состоит из операндов, соединенных между собой специальными знаками, которые определяют семантику (смысл) выражения. Например, выражение “Здравствуй, Мир!” состоит из операндов-слов и орфографических знаков “,”, “пробел” и “!”. В прикладной информатике употребляются три типа выражений: арифметическое, условное и логическое. Операнды этих выражений соединяются знаками арифметических операций, знаками отношения и знаками логических операций. Важная особенность этих выражений: при выполнении операций над данными вырабатывается уникальное значение выражения, которое является либо числовым (для арифметического выражения), либо логическим (TRUE или FALSE, 1 или 0) для условного и логического выражения. Последовательность выполнения операций (приоритет) может быть изменена с помощью круглых скобок.

Выходная информация — информация, которая получается в результате преобразования человеком или устройством входной информации.

Данные — обрабатываемая информация, представленная в памяти компьютера в двоичной форме.

Дизъюнкция — это логическая операция, которая каждым двум простым высказываниям ставит в соответствие составное высказывание, являющееся ложным тогда и только тогда, когда все исходные высказывания ложны, и являющееся истинным, когда хотя бы одно из исходных высказываний истинно.

Запись базы данных — это строка таблицы, содержащая набор значений свойств, размещенный в полях базы данных.

Интернет — глобальная сеть, объединяющая компьютерные сети.

Информационная безопасность — состояние сохранности информационных ресурсов и защищенности законных прав личности и общества в информационной сфере.

Информационная модель — совокупность информации об объекте или процессе.

Компьютер — устройство для ввода, обработки и отображения всевозможной информации.

Компьютерная этика — занимается исследованием поведения людей, использующих компьютер.

Кроссплатформенная программа — это программа, у которой есть версии для различных операционных систем (Windows и Linux).

Массив — это последовательность однотипных элементов, имеющих общее имя, причем каждый элемент этой последовательности определяется порядковым номером (индексом) элемента.

Одномерный массив — это фиксированное количество элементов одного типа, объединенных одним именем, причем каждый элемент имеет свой уникальный номер, и номера элементов идут подряд.

Размерность массива — это количество индексов, необходимое для однозначной индексации элемента в рамках массива.

Модель — упрощенное представление о реальном объекте.

Монитор — устройство отображения информации на экране.

Мультимедиа — это специальная технология, которая позволяет с помощью программного обеспечения объединить на твоём компьютере обычную информацию (текст или графику) со звуком и движущимися изображениями (можно создавать свои видеофильмы).

Носитель информации — материальный объект, предназначенный для хранения информации.

Облачные технологии (вычисления) — это технологии распределенной обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователю как Интернет-сервис.

Объект — некоторая часть окружающего нас мира, которая может быть рассмотрена как единое целое.

Объем оперативной памяти — важная характеристика компьютера, влияющая на его производительность. Измеряется в мегабайтах и гигабайтах.

Оперативная память — совокупность специальных электронных ячеек, каждая из которых может хранить конкретную комбинацию из нулей и единиц — один байт.

Операционная система — это базовая и необходимая программа, которая обеспечивает пользователю удобный интерфейс (способ обмена информацией) с различными программами и электронными устройствами.

Память внешняя — энергонезависимая. Магнитные и оптические диски, флэш-память. Используются для долговременного хранения информации и переноса данных с одного компьютера на другой.

Поле базы данных — это столбец таблицы, содержащий значения определенного свойства.

Программа — набор инструкций, составляемый программистом и исполняемый компьютером.

Программирование — профессиональная деятельность по разработке программного обеспечения компьютеров.

Программное обеспечение (ПО) — совокупность всех программ, которые используются в компьютере. Без программного обеспечения компьютер — просто набор железа.

Провайдер — фирма, предоставляющая пользователям выход в Интернет.

Процессор — это устройство, предназначенное для автоматического считывания команд программы, их расшифровки и выполнения.

Разрядность — это максимальное количество двоичных разрядов, которые процессор способен обработать за одну команду.

Символьная строка — это последовательность символов, которая заканчивается нулевым символом (`'\0'`).

Свойство объекта — совокупность признаков объекта, по которым его можно отличить от других объектов.

Сетикет, нетикет — правила поведения, общения в Сети, традиции и культура интернет-сообщества, которых придерживается большинство.

Сортировка — (англ. *sorting* — классификация, упорядочение) — последовательное расположение или разбиение на группы чего-либо в зависимости от выбранного критерия.

Состояние объекта — результат выполнения объектом активного или пассивного действия.

Утилиты — это служебные программы, которые используются для расширения или улучшения функций системных программ, основное их назначение состоит в автоматизации работ по проверке, наладке и настройке компьютерной системы (это специальные служебные программы, предназначенные для улучшения и других возможностей операционной системы).

Фильтрация данных — вывод нужных пользователю данных, в результате созданного им запроса.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Велихов А. С.* Основы информатики и компьютерной техники: учебное пособие. М.: СОЛОН-Пресс, 2017. 539 с.
2. *Дейкстра Э.* Дисциплина программирования. М.: Мир, 1978. 276 с.
3. *Джонстон Г.* Учитесь программировать. М.: Финансы и статистика, 1989. 368 с.
4. *Кнут Д.* Искусство программирования для ЭВМ. Т.1: Основные алгоритмы. М.: Мир, 1976. 736 с.
5. *Кнут Д.* Искусство программирования для ЭВМ. Т.3: Сортировка и поиск. М.: Мир, 1978. 844 с.
6. *Першиков В. И., Савинков В. М.* Толковый словарь по информатике. М.: Финансы и статистика, 2016.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Базы данных	37
Библиотека PyGame	84
Двумерный массив	68
Запись базы данных	38
Индекс	54
Компьютерная этика	16
Конфигурация компьютера	23
Массив	54
Облачные технологии	15
Одномерный массив	56
Перестановка элементов массива	64
Поиск данных	49
Поле базы данных	38
Программное обеспечение	26
Размерность массива	68
Свойства информации	5
— актуальность	5
— достоверность	6
— объективность	8
— понятность	6
— ценность	5
Сеткет	17
Сортировка	47
Сортировка пузырьком	75
Фильтрация	48
Функция <code>sort()</code>	76

СОДЕРЖАНИЕ

Введение	3
Раздел 1. РАБОТА С ИНФОРМАЦИЕЙ	
§ 1. Свойства информации	4
§ 2–3. Совместная работа с документами	11
§ 4. Сетевой этикет	16
Раздел 2. ВЫБИРАЕМ КОМПЬЮТЕР	
§ 5. Конфигурация компьютера	22
§ 6. Выбор программного обеспечения	25
§ 7. Выбираем компьютер	29
Раздел 3. БАЗЫ ДАННЫХ	
§ 8. Базы данных	36
§ 9. Создание базы данных в электронных таблицах	40
§ 10. Методы поиска информации	43
§ 11–12. Сортировка и фильтрация данных	46
§ 13. Работа с базой данных	51
Раздел 4. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON	
§ 14. Одномерный массив	54
§ 15–16. Поиск элемента с заданными свойствами	59
§ 17. Перестановка элементов	64
§ 18–19. Двумерный массив	68
§ 20–21. Сортировка	75
§ 22–23. Удаление и вставка элемента	80
Раздел 5. СОЗДАНИЕ 2D ИГРЫ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON	
§ 24–25. Библиотека PyGame	84
§ 26–27. Задний фон и персонажи игры. Анимирование персонажей	88
§ 28–29. Программирование условий	91
Глоссарий	104
Список использованных источников	108
Предметный указатель	109

Учебное издание

**Кольева Наталья Станиславовна
Шевчук Елена Владимировна
Ержанов Эльзар Бауржанович**

ИНФОРМАТИКА

Учебник для 9 классов общеобразовательных школ

Редактор *К. Амирова*
Дизайн макета, иллюстрации *Е. Мельник*
Техн. редактор *И. Тарануец*
Корректор *Е. Дремкова*
Компьютерная верстка *С. Жумагелдиевой*

Государственная лицензия № 0000001 выдана издательству
Министерством образования и науки Республики Казахстан
7 июля 2003 года



ИБ № 5893

Подписано в печать 11.06.19. Формат 70x100 ¹/₃₂. Бумага офсетная.
Гарнитура "ММ Мектептік". Печать офсетная. Усл.-печ. л. 9,03. Усл. кр.-отг. 36,77.
Уч.-изд. л. 3,65. Тираж 15 000 экз. Заказ №

Издательство "Мектеп", 050009, Алматы, пр. Абая, 143.

Факс: 8(727) 394-42-30, 394-37-58

Тел.: 8(727) 394-41-76, 394-42-34

E-mail: mektep@mail.ru

Web-site: www.mektep.kz

