

**Р.А. Кадиркулов, Г.К. Нурмуханбетова**

# ИНФОРМАТИКА

Учебник для учащихся  
9 класса общеобразовательной школы

Алматы кітап баспасы  
2019

## Условные обозначения

- |  |                  |  |                         |
|--|------------------|--|-------------------------|
|   | – Проблема урока |   | – Индивидуальная работа |
|   | – Подумай        |   | – Работа в паре         |
|   | – Новые знания   |   | – Работа в группе       |
|   | – Анализ         |   | – Практическая работа   |
|   | – Синтез         |   | – Примени на практике   |
|   | – Оценка         |   | – Вопросы               |
|  | – Задание        |  | – Домашнее задание      |

Кадиркулов Р.А.

Информатика. Учебник для учащихся 9 класса общеобразовательной школы./ Кадиркулов Р.А., Г.К. Нурмуханбетова. – Алматы: «Алматықітап баспасы», 2019. – 168 с.; илл.

© Кадиркулов Р.А., Нурмуханбетова Г.К., текст, 2019  
© ТОО «Алматықітап баспасы», 2019

## УВАЖАЕМЫЕ УЧЕНИКИ!

В наступающем учебном году вы закончите основную школу. В 9-м классе вы будете углублять свои знания, полученные в 5-8 классах по предмету информатика. Расширите свои знания, полученные при изучении предмета информатика в основной школе, и улучшите свои навыки работы с компьютерной техникой.

Во время работы с учебником вы сможете, получая новые знания и опыт, размышлять и отвечать на разные вопросы, обсуждать совместно в парах, в группах и приходить к общему мнению, при индивидуальном выполнении практических и творческих заданий формировать информационное образование и культуру, всесторонне повышать компьютерную грамотность.

Темы в учебнике основаны на критическом мышлении. В то же время задачи на развитие критического и логического мышления составлены от простого к сложному, чтобы можно было самостоятельно справиться и учиться друг у друга и направлены на тщательное освоение темы. В учебнике вы будете работать с 5 главами важного направления.

В разделе **«Работа с информацией»** учебника в первой четверти вы познакомитесь со свойствами информации, с облачными технологиями, сетевым этикетом.

В разделе **«Выбираем компьютер»** вы узнаете, как правильно подобрать компьютер, выбрать его конфигурацию и программное обеспечение.

Вы научитесь создавать **базы данных**, используя возможности электронной таблицы MSExcel при изучении тем учебника во II четверти. Узнаете, как сортировать и фильтровать данные в базе данных.

В разделе **Программирование алгоритмов на языке программирования Python** в III четверти для работы с массивами в учебнике представлен язык программирования Python. Python – это одновременно простой и мощный объектно-ориентированный язык программирования. Это отличный язык, который обеспечивает высокий уровень структуры данных и возможность работы на разных платформах в разных приложениях. Этот язык – не вызовет сложности у тех, кто только начинает изучать язык программирования сейчас. На этом языке написаны такие сайты, как Instagram, Pinterest и Rdio.

В разделе **Создание 2D игры на языке программирования Python** в IV четверти вы познакомитесь с модулем Pygame языка программирования Python. Pygame – это библиотека модулей для создания 2D игр на Python.

Учебник имеет электронное приложение. Электронная версия содержит открытый исходный код заданий и программы моделей, представленные в теоретической части учебника. Эти задачи были созданы для того, чтобы вы имели возможность не тратя время загрузить их на компьютер и одновременно выполнить, проанализировать и внести собственные изменения.

XXI век – это век информационного общества. Мы хотим, чтобы вы стали достойным членом общества благодаря знаниям, умениям и навыкам, полученным в области информатики.

*Желааем вам удачи!*

## Раздел I. Работа с информацией

1.1

## СВОЙСТВА ИНФОРМАЦИИ



Каковы способы определения актуальности, ценности и точности информации?



## Подумай

- Что такое информация?
- Какие существуют виды получения информации?
- Что можно сказать о значении информации в жизни человека?



## Новые знания

**Информация** (от лат. *informatio* разъяснение, изложение) – сведения, независимые от формы их представления. В целом, это понятие тесно связано с такими понятиями, как инструкция, связь, контроль, форма, обучение, знание, смысл, структура, воображение, чувства. Очень важно знать типы и свойства информации (*Схема 1*).

Вы ежедневно получаете какую-либо информацию. Получая информацию, вы принимаете ее, обрабатываете и используете. Вы можете отправлять информацию родителям, друзьям. У информации есть источник и приемник. Человек воспринимает информацию органами чувств.

**Информация** – это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии.

Вся деятельность человека связана с информацией. Для приготовления пищи нужно знать ее ингредиенты и способ приготовления. Для вождения автомобиля или построения дома надо владеть соответствующей информацией. Информация – это сведения об окружающем мире, которые являются объектом хранения, преобразования, передачи и использования.

При обмене информацией люди обращают особое внимание на ее понятность, определенность, актуальность, ценность и точность. Это в свою очередь поможет лучше понять друг друга, проанализировать различные ситуации и принять решение. Повседневная жизнь, как и трудовая деятельность людей, напрямую связана со следующими свойствами информации.



### Свойства информации

- 1 Надежность информации** определяется тем, что полученная информация соответствует реальной ситуации. К любым сигналам, которые записываются, всегда присоединяются посторонние шумы различного уровня (информационный шум). Если уровень информационного шума увеличивается, надежность информации будет уменьшена.
- 2 Полнота информации.** Информация считается полной, если ее объем достаточен для принятия решения. Недостаточность информации может привести к ошибкам. Неполная или избыточная информация может помешать принять верное решение.
- 3 Актуальность информации** определяется ее отношением ко времени. Правильная, но устаревшая и несвоевременная информация станет неактуальной. Например, программа телепередач на этой неделе будет неактуальной на следующей неделе.
- 4 Объективность информации** означает, что она должна быть независимой от мнения и суждения кого-либо.
- 5 Ценность информации** – это пригодность информации для использования в различных сферах человеческой деятельности.
- 6 Понятность информации.** Информация должна быть написана на понятном для пользователя языке. Например, трудно понять китайский язык в англоязычной среде.

$$2 \cdot 2 = 4$$



### 1. Надежность

Информация отражает истину дела



### 2. Полнота

Достаточность информации для понимания и принятия решений



### 3. Актуальность

Информация, ценная в данный момент времени



### 4. Объективность

Свойство, характеризующее независимость от мнения, суждения кого-либо

### 5. Ценность

Возможность использования информации в различных сферах человеческой деятельности

### 6. Понятность

Свойство, характеризующее изложение на понятном языке для получателя

Согласно исследованиям ученых, мы воспринимаем информацию в следующих долях: 81% – через зрение, 10% – на слух, 4% – обонянием, 3% – вкусовыми рецепторами и 2% – другими органами чувств.

В то же время у каждого есть свои индивидуальные особенности восприятия. Например, у дизайнера более развито зрительное восприятие, а у музыканта – слуховое. Некоторые из вас хорошо усваивают новый материал посредством чтения книг. Кому-то легче понять практическую задачу, которую необходимо решить.



### Выполнни

1. Перечислите, из каких источников вы получаете информацию.
2. Подготовьте перечень специальностей, которые связаны с обработкой большого объема информации.
3. Приведите примеры источника и приемника информации.
4. Вечером, когда вы возвращались после уроков, неожиданно позвонил одноклассник и сообщил об изменении расписания уроков на завтра. Какими свойствами обладает данная информация с вашей точки зрения (надежность, полнота, актуальность, ценность, понятность, объективность)?

**Анализ**

Какие виды информации могут воспринимать люди, животные и растения? С помощью диаграммы Венна сравните и проанализируйте сходства и различия принимаемой информации людьми и животными, людьми и растениями, а также животными и растениями соответственно.

**Синтез**

Обоснуйте приоритет охраны жизни и здоровья людей, животных, растений и окружающей среды, используя различные виды информации.

**Вопросы**

1. Какие бывают виды информации?
2. Какие вы знаете свойства информации?
3. Каким из органов чувств человек воспринимает больше всего информации?
4. Каковы характеристики исторической информации?
5. Какие примеры актуальности информации вы можете привести?

**Домашнее задание**

	<b>Темы исследований</b>	<b>Вопросы и направления исследований</b>
1	<b>Информационная деятельность человека</b>	<ul style="list-style-type: none"> <li>• Понятие информации</li> <li>• Виды и свойства информации</li> <li>• Значение информации для человека</li> </ul>
2	<b>Работа с информацией в различных отраслях</b>	<ul style="list-style-type: none"> <li>• Изучение проблем сбора, обработки и передачи информации, в различных отраслях</li> </ul>

## Раздел I. Работа с информацией

1.2

## СОВМЕСТНАЯ РАБОТА С ДОКУМЕНТАМИ



Используя сервисы сети Интернет для создания, редактирования и использование документов, ознакомьтесь с облачной технологией для выполнения совместной работы.



## Подумай

- Какие вы знаете надежные методы защиты документов от удаления?
- Каковы способы снижения стоимости приобретения прикладных программ и компьютеров последнего поколения?
- Каковы возможности обработки ваших собственных данных в любой части мира?



## Новые знания

Одной из последних технологий, которая недавно была разработана, является облачная технология. С английского языка «Cloud technology», или «Cloud computing» (cloud – облако; technology – технология; computing – расчеты) переводится как «облачные вычисления», или «облачная технология, служба». Облачные вычисления можно объяснить следующим образом: когда пользователь запускает на своем компьютере определенную программу, основные вычисления и их источники располагаются и хранятся на удаленных серверах в интернете; а результат работы отображается на компьютере пользователя в стандартном окне Веб-браузера (рис. 1).



Рис. 1. Визуальное изображение облачной технологии

Серверы, реализующие **Cloud computing**, называются «вычислительными облаками». Основное отличие облачной технологии от стандартного программного обеспечения заключается в том, что пользователи используют информационные ресурсы любого уровня и мощности удаленных вычислительных ресурсов, а не собственные компьютерные ресурсы. Другими словами, пользователь может иметь полный доступ к собственным ресурсам в любое время и в любом месте. Использование облачной технологии не только не влияет на быстродействие операционной системы, программное обеспечение пользователя, но и не может их изменить.

## Разделение облачных вычислений в зависимости от целей

- 1 Частное облако (private cloud)** – инфраструктура, предназначенная для использования одной организацией, включающей несколько потребителей, возможно также клиентами и подрядчиками данной организации. Частное облако может находиться в сервере самого потребителя.
- 2 Публичное облако (public cloud)** – инфраструктура, предназначенная для свободного использования широкой публикой. Публичное облако может находиться в собственности, управлении и эксплуатации коммерческих, научных и правительственные организаций. Публичное облако физически существует в юрисдикции владельца – поставщика услуг.
- 3 Гибридное облако (hybrid cloud)** – это комбинация из двух или более различных облачных инфраструктур (частных, публичных или общественных), связанных между собой технологиями передачи данных и приложений. Эта модель облачного типа может использоваться крупными компаниями с географически удаленными филиалами или со многими программными системами.
- 4 Облако сообщества (community cloud)** – инфраструктура для государственных потребителей с общими целями.

## Преимущества облачной технологии

- для решения сложных задач, требующих больших ресурсов, пользователь может использовать множество серверов и программ, которых нет у него в наличии;
- пользователь может работать с любым компьютерным устройством, которое подключено к интернету, в любое время, из любого места;
- пользователь не зависит от уязвимости, поломки или неисправности компьютерного устройства, а также от прерывания или сбоя программного обеспечения;
- пользователь может легко поделиться своими источниками с другими людьми и работать с ними вместе;
- по сравнению с программным обеспечением персонального компьютера облачные сервисы зачастую очень дешевы или часто бесплатны;
- некоторые из облачных проектов, особенно для крупных компаний, – это затраты на администрирование, обслуживание, обновление и лицензирование аппаратного и программного обеспечения оказываются очень экономичными;

- в рамках облачного проекта также решается проблема дефицита квалифицированных специалистов, сопровождающих программное обеспечение.

### Недостатки облачной технологии

Тем не менее у облачных технологий есть недостатки. Во многих отношениях они связаны с простыми пользователями, в меньшей степени, с провайдерами. Например, нельзя доверять все данные, относящиеся к проблемам безопасности – сторонним провайдерам.

Нельзя забывать, что из-за технических сбоев поставщиков облачных услуг возникает риск потери всех данных пользователей, или они могут быть использованы нечестными компаниями, предоставляющими услуги провайдера.



Рис. 2. Использование облачного хранилища

**Облачные приложения** (программное обеспечение как услуга) работают на удаленных компьютерах «в облаке», принадлежат другим компаниям и управляются ими (рис. 2).

Приведем примеры облачных сервисов, оказывающих данные услуги.

1. **Dropbox** – это сервис облачного хранения пользовательских файлов с возможностью синхронизации файлов между разными устройствами. Таким образом, пользователь имеет доступ к своим данным независимо от устройства, которым пользуется.

2. **Яндекс. Диск** – сервис создан для того, чтобы вы объединили медиатеку, документы и фотографии на ваших устройствах (телефонах и компьютерах, дома и на учебе), надежно их хранили и легко делились с близкими или друзьями.

3. **OneNote** – полнофункциональное приложение, входящее в стандартный пакет Microsoft Office. Вы можете встроить или просмотреть в OneNote любые файлы: снимки экрана, картинки, отсканированные документы, аудио и видеофайлы с любого устройства, в любом месте, где доступен интернет.

**4. Microsoft Office 365** – это стандартный пакет Microsoft Office, используемый в качестве приложения для доступа в интернет. С любого персонального компьютера вы можете войти в систему с вашим логином пользователя и паролем и работать с офисными программами. Из-за того, что интерфейс остался прежним, работать в браузере будет проще.

### Облачный сервис Google

Первым и наиболее доступным, удобным облачным сервисом является **Google**. Google-документы, Google-таблицы и Google-презентации являются ярким примером облачных сервисов. Google – облачный сервис бесплатного пользования (рис. 3).

Вы можете совместно использовать и редактировать документы с помощью **Google-документов** в интернете.

**Google-таблицы.** Учитель составляет таблицу и дает право ученикам проводить совместную работу. Работая в группе, ученики приступают к заполнению таблицы необходимой информацией, находя ее в интернете.

**Google-презентация.** Ученики группой делают презентацию в режиме доступа онлайн, составляют вопросы.

**Google-форма.** Учитель имеет возможность создавать вопросы и проводить анкетирование.

Пользователь облачных сервисов может поделиться своими ресурсами с другими людьми и совместно работать с ними. Эта возможность преимущество данной технологии. Например, вы можете добавить в список новых пользователей при работе с Google-документами, выполнив команду **Совместная работа** (рис. 3).

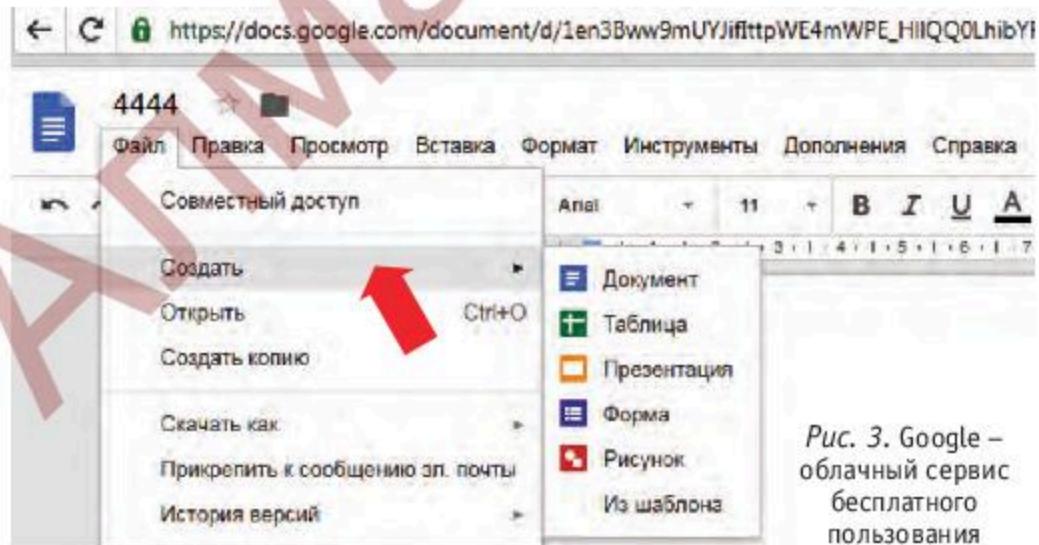


Рис. 3. Google – облачный сервис бесплатного пользования

**Анализ**

Сравните и проанализируйте доступы к компьютерным программам с доступом к облачным документам. Сравните преимущества и недостатки каждого из них.

**Оценка**

Оцените настоящее и будущее развитие облачной технологии.

**Вопросы**

1. Что такое облачная технология и для чего она используется?
2. Что можно сказать о преимуществах и недостатках облачной технологии?
3. Какие вы знаете облачные сервисы?
4. Как вы понимаете «облачные вычисления»?
5. Какие существуют виды «облачных вычислений»?
6. Какие вы можете перечислить виды облачных сервисов Google?

**Задания*****Задание 1. Google-документ. Выпуск газеты по предмету «Информатика»***

Проведите коллективную работу по данной теме, соберите материалы и подготовьте документы. Выберите название газеты, дату выпуска и номера, имя и адреса авторов. Каждая группа должна написать статью, вставить изображение, цитату и сформулировать основную идею статьи. Во время совместной работы проанализируйте идеи членов группы, поддерживайте творческие идеи, редактируйте статьи вместе.

***Задание 2. Google-презентация. Облачная технология***

В команде или небольшой группе создайте презентацию по облачной технологии (с общим доступом) с помощью Google Presentation. Определите тему слайда, выберите цвет текста и покажите презентацию в режиме **Показ слайдов** с использованием разных макетов слайдов.

## Раздел I. Работа с информацией

1.3

**СОВМЕСТНОЕ ИСПОЛЬЗОВАНИЕ ДОКУМЕНТОВ НА GOOGLE DRIVE**

Научитесь на практике выполнять совместную работу по созданию, изменению и анализу документов с использованием облачных сервисов.

**Подумай**

- Каковы преимущества совместной работы с документами на компьютере?
- Какие вы можете назвать пути организации совместного редактирования документов на компьютере, если у вас нет облачных сервисов?

**Новые знания**

Благодаря бурному развитию облачных технологий появилась возможность хранить документы, таблицы, фотографии, презентации и другие информационные объекты на сервере удаленного компьютера, а не на личных компьютерах. В этом случае вы не только сохраняете, но и можете обрабатывать эти документы. Ранее в сети Интернет использовалась служба Web 1.0. Сегодня эта технология устарела и взамен создана служба Web 2.0, в которой можно и загружать данные из интернета, и есть возможность для создания, редактирования, хранения информации и передачи другим пользователям.

Web 2.0 – интерактивные многопользовательские системы, контент (содержание) которых наполняется самими пользователями.

Особенностью сервиса Web 2.0 является возможность одновременного наполнения информационного материала пользователями. По сути, термин «Web 2.0» означает проекты и сервисы, активно развивающиеся и улучшаемые самими пользователями: блоги, wiki, социальные сети и т. д. Одна из таких служб – **Google Docs**. Через данную службу можно выполнить следующие действия:

- Создание документов, таблиц и презентаций в интернете, просмотр и редактирование созданного документа на любом компьютере, подключенном к интернету.
- Систематизация документов в папках хранилища информации **Google**.
- Сохранение созданных файлов на вашем компьютере.

- Загрузка готовых файлов с вашего компьютера и размещение их в хранилище Google документов.
- Публикация документа в виде веб-страницы или размещение в своем собственном блоге.
- Разрешение другим пользователям редактировать или просматривать документы.
- В режиме реального времени несколько человек могут редактировать документ одновременно.
- Поделиться с другими своими презентациями.
- Важными документами, графиками и презентациями сотрудники компании могут обмениваться с коллегами.

В настоящее время в сети Интернет имеется более 100 сайтов, предлагающих услуги Web 2.0. Некоторыми из них можно пользоваться бесплатно, но и в платных стоимость не слишком высока. Облачные сервисы дают возможность одновременно использовать основные функции пакета приложений Microsoft Office (Word, Excel, PowerPoint и другие) при обработке одного документа несколькими пользователями.

Логотип	Адрес	Примечание
	<a href="http://docs.google.com/">http://docs.google.com/</a>	Сайт с интерфейсом на русском языке. Вы можете создать таблицу и презентации из текста.
	<a href="http://writer.zoho.com/">http://writer.zoho.com/</a>	Сайт с русскоязычным интерфейсом и только с текстовым редактором, имеет готовые шаблоны для создания текстовых документов.

Для доступа к файлам, хранящимся на **Google Drive**, вам необходимо выполнить следующий шаг: перейти на **Google Drive** (<https://drive.google.com/>), создать аккаунт и зарегистрироваться. Документы, сохраненные на **Google Drive**, будут доступны, когда вы войдете в свою учетную запись (аккаунт).



### Практическая работа

1. Выберите один из документов, созданных в Word.
2. Загрузите этот документ на **Google Drive**.
3. Редактируйте загруженный документ в **Google Drive**.

### Пошаговое выполнение работы:

1. В сети Интернет, запустив сервис [google.kz](http://google.kz), активизируйте свою учетную запись (аккаунт).

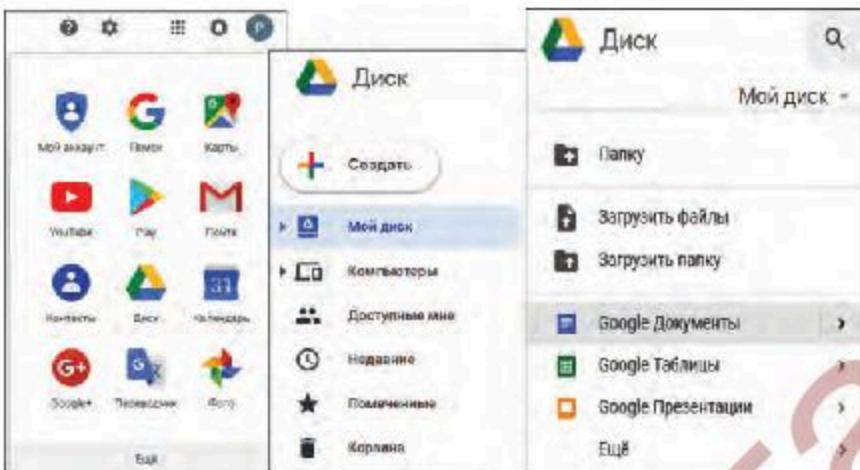


Рис. 1. Открытие службы Google Drive и построение текстового документа

- Набрав в адресной строке <https://drive.google.com/>, подключите сервер Google Диск.
- Нажмите на кнопку Приложения Google в правом верхнем углу окна сервера (рис. 1).
- Создайте текст «Программа родительского собрания» в текстовом редакторе. Для этого нажмите кнопку «Создать» (рис. 1). В открывшемся меню выберите Google-документ, а затем отредактируйте документ.
- Введите текст данного задания. Используйте инструменты, расположенные в окне редактора, чтобы отформатировать текст по своему желанию.
- Выполнив команду Файл – Переименовать, переименуйте имя созданного файла на «Собрание родителей 9-го класса».
- Файл, созданный по умолчанию, доступен только автору документа. Чтобы дать доступ к документу для всех пользователей интернета, в окне редактора выберите «Настройки доступа».
- В окне «Совместный доступ» введите адреса электронной почты пользователей, которым можно редактировать документ совместно, а затем нажмите кнопку «Отправить» (рис. 2).

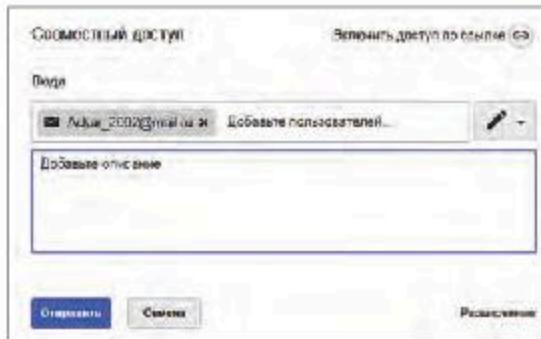


Рис. 2. Окно доступа для совместного редактирования

- Вы можете редактировать текст «Программа родительского собрания» совместно с одноклассниками.



### Синтез



Запланируйте совместную разработку документов со своими одноклассниками.

Выберите тему документа и разделите документы на части между собой для совместной обработки.



### Оценка



Оцените совместную работу редактирования в Google Drive текста «Программа родительского собрания» с одноклассниками. О каких эффективных сторонах такого редактирования документа вы можете сказать? Какие недостатки можно отметить? Приведите примеры.



### Вопросы

- В чем разница между Web 2.0 и Web 1.0?
- Какие облачные сервисы предлагает Google Docs?
- Каков алгоритм использования Google Drive?
- Какие команды должны быть доступны всем пользователям интернета?



### Задание

- Войдите в службу Google Docs в интернете.
- Создайте папку «Картина (облик) родного края» и сделайте ее доступной для своих одноклассников.
- Загрузите в папку «Пейзаж родного края» рисунки со своего компьютера или из сети Интернет.
- Создайте список использованных веб-ресурсов в отдельном документе и разместите его в общей папке.
- Сделайте презентацию трех слайдов о городе (селе), в котором вы живете, и сохраните ее в общей папке.
- Отправьте по электронной почте своим одноклассникам приглашение для совместной работы с презентацией в интернете.
- Примите приглашение одного из ваших одноклассников и добавьте один слайд к его презентации.
- Просмотрите презентацию, созданную совместно из хранилища документов.
- Закройте свой аккаунт. Выходите из сервиса Google.

## Раздел I. Работа с информацией

## 1.4 СЕТЕВОЙ ЭТИКЕТ



Какой вывод вы можете сделать о последствиях нарушения этических и правовых норм при работе в сети?



## Подумай

- Какие правила вы соблюдаете при работе в сети?
- Как вы понимаете слова – *этикет, сетевой этикет?*
- Что такое авторское право, плагиат?



## Новые знания

В результате быстрого развития интернета у людей появился доступ к виртуальной связи. Существуют различные формы онлайн-общения в сети (рис. 1). Междугородняя или международная связь по телефону уже не используется так активно, как в былые времена. Теперь никого не удивишь современными средствами связи, люди разговаривают друг с другом по мобильному телефону, обмениваются фотографиями, видео или общаются по видеосвязи. Благодаря динамичную развитию сети Интернет люди сделали эту сеть центром коммуникации. В процессе общения в сети Интернет сформировались своя культура и этикет, правила общения, обмена сообщениями по электронной почте, проведения телеконференций, тематических форумов и общения в чате. Хотя они не закреплены законом, каждый человек, считающий себя культурным, должен придерживаться свода этих правил.

Этикет (франц. *etiquette*) – набор правил поведения, принятых в определенном обществе, а также соблюдение этих правил и норм. В русском языке образовался неологизм **сетикет**, объединивший два понятия – «сеть» и «этикет». Сетевой этикет означает простейшие правила,



Рис. 1. Сетевое взаимодействие

традиции интернет-сообщества и культуру использования интернет-коммуникаций. Общение в сети должно быть одинаково доступно как опытному, так и начинающему пользователю. Подавляющее большинство правил сети основано на универсальных, а не на конкретных принципах. Следуя этим принципам, вы сможете при общении в сети повысить свою репутацию и привлечь внимание онлайн-сообщества к себе.

### **Основные термины, используемые в сети**

**Веб-форум** – платформа для общения между пользователями интернета на одну тему или на несколько тем. Сегодня существуют гипертекстовые, графические, звуковые и видеофорумы. Характерная черта форумов – это возможность общаться в интернете на определенную тему.

**Чат** – представляет собой службу текстовых сообщений в режиме реального времени, позволяющую многим пользователям одновременно общаться друг с другом.

**Флайм (flames)** – отправка смайликов позитивного посыла, не взирая на настроение участника форума. Обычно используется сетевыми пользователями для привлечения внимания.

**Флуд** – бессмысленный поток сообщений. Эти сообщения не несут смысловой нагрузки и могут быть удалены. Обычно во флуде ничего не говорится, но зато им пользуются для привлечения внимания большинства. Занимающиеся флудом проявляют быструю, но бессмысленную реакцию на многие новости. В сетях надо дистанцироваться от флуда, потому что флуды замедляют загрузку страниц и засоряют ненужной информацией.

**Смайлики** – это простые «картинки». С помощью смайликов пользователи выражают свои эмоции. В современном мессенджере эти значки представлены в виде пиктограмм, изображений. А при написании текстовых сообщений на мобильном телефоне вы можете выражать эмоции символами.

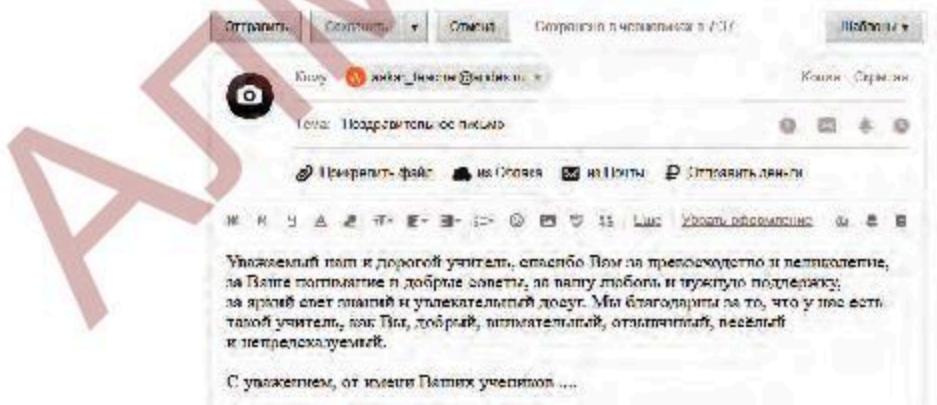
### **Правила сетевого этикета**

- 1** Не забывайте, что по другую сторону сети находится такой же человек, как и вы.
- 2** Следуйте правилам поведения в сети, как в реальной жизни.
- 3** Незабывайте, что вы работаете в кибер-пространстве (виртуальном мире).

- 4 Уважайте время и труд других людей.**
- 5 Сохраняйте свое человеческое достоинство при работе в сети.**
- 6 Если у вас есть такая возможность, помогайте другим.**
- 7 Ни с кем не конфликтуйте в сети, также не провоцируйте никого на ссоры.**
- 8 Не лезьте в личную жизнь человека, уважайте его право на личные письма.**
- 9 Не используйте службу сети в корыстных целях.**
- 10 Уважайте тех, кто в сети, всегда придерживайтесь этических норм.**

**Электронная почта – инструмент быстрой связи.** Обмен письмами по электронной почте должен осуществляться в определенных рамках правил вежливости. Эти правила следующие:

- Если у вас нет времени ответить на полученное письмо, то хотя бы выразите благодарность. «Спасибо, я получил ваше письмо. Я обязательно отвечу вам позже».
- Всегда помещайте заголовок темы в электронных письмах. Заголовок должен быть кратким и понятным, потому что он является ключом для поиска этого письма в сети, сортировки и других действий. Таким образом, вы экономите время получателей этого письма.
- Тема письма – суть всего сообщения. Если вы, отвечая на чье-то письмо, изменили основную идею беседы, то желательно изменить и тему.
- Обсуждение новой темы начните с написания нового письма.



*Рис. 2. Электронное письмо*

- Текст можно разделить на параграфы для удобства чтения письма.
- Избегайте жаргонных слов и грамматических ошибок.

### **Сетевой этикет в чате**

- Не забывайте здороваться, когда входите в чат, и прощаться, когда выходите.
- Четко излагайте свои мысли, на вопросы отвечайте кратко и ясно.
- Соблюдайте очередность диалога с участниками чата.
- Не обращайте внимание на агрессивное поведение какого-либо участника.

### **Простые культурные правила, которые применяются в текстовых сообщениях в сети**

Вид смайлика	Изменения в мимике
:-) или :)	Улыбка
:-(  или :(	Без настроения
; -)	Подмигивание
:-0	Удивление

При поиске необходимой информации в интернете важно сосредоточиться на правовой защите информации. Пользователи сети пишут свои доклады, рефераты и многое другое, используя возможности интернета: в полной мере скачивая информацию, публикуемую на веб-страницах, или загружая различные программы, изображения и анимации и многое другое. Не обращая при этом внимание на то, что эта информация в сети размещена кем-то, имеющим на нее право, и часто не знают о необходимости получить соответствующее разрешение на ее использование. В Казахстане авторское право защищено законом. Закон Республики Казахстан предоставляет авторам и другим правообладателям различные способы защиты своих прав. Защитными методами называют инструменты, предусмотренные законом. Они могут использоваться для недопущения нарушения прав, удаление или возмещение убытков. За нарушение Закона об авторском праве правонарушитель привлекается к гражданской, административной и уголовной ответственности. Поэтому пользователи сети должны учиться не нарушать правила защиты информации, изучать правила пользования информацией, правильно ссылаясь на источники информации и получать разрешение правообладателей на использование информационных ресурсов.

**Анализ**

Какие виды онлайн-связи используют ваши сверстники сегодня? Почему? Проанализируйте их. Приведите примеры.

**Синтез**

Что еще можно добавить к правилам, изложенным выше? Представьте свой свод правил работы в сети, обоснуйте.

**Оценка**

Напишите научную статью на тему «Сетевой этикет».

**Вопросы**

1. Каковы возможности обмена информацией и общения в интернете сегодня?
2. Что такое этикет?
3. С какими понятиями, используемыми в сети, вы познакомились?
4. Из каких правил состоит сетевой этикет?
5. Каковы правила написания писем в электронной почте, соблюдения вежливости в форуме и чате?
6. Каковы правовые нормы использования информации из сети?

**Домашнее задание**

Темы исследований	Вопросы и направления исследований
1 <b>Обмен информацией в социальных сетях</b>	<ul style="list-style-type: none"> <li>• Виды социальных сетей</li> <li>• Этика обмена информацией в социальных сетях</li> </ul>
2 <b>Что такое плагиат?</b>	<ul style="list-style-type: none"> <li>• Нормы защиты авторского права в мире</li> <li>• Нормы защиты авторского права в Казахстане</li> <li>• Современные методы распознавания плагиата</li> </ul>

## Раздел II. Выбираем компьютер

## 2.1 КОНФИГУРАЦИЯ КОМПЬЮТЕРА



Что такое конфигурация компьютера?



Подумай

- Какие вы знаете устройства персонального компьютера? Расскажите о них.
- Как можно определить технические свойства внешних и внутренних устройств компьютера?



Новые знания

## Основные устройства компьютера

Когда мы слышим слово компьютер, мы представляем себе либо персональный компьютер, либо ноутбук. Тем не менее, существуют другие виды компьютеров (рис. 1). Несмотря на их внешнее различие, принцип работы у них одинаковый. Среди них выберем для рассмотрения **настольный компьютер**. Мы используем такой компьютер на работе, в школе, дома. Они бывают небольшими, средними, большими, и обычно их располагают на столе. Такие компьютеры обычно называют **персональными компьютерами**. Эти компьютеры в основе состоят из системного блока, монитора, клавиатуры и мыши.

Аппаратные обеспечения персонального компьютера изменяют по мере необходимости. Компоненты компьютера называют его



Рис. 1. Виды компьютера

**конфигурацией.** Устройства, входящие в состав компьютера при его покупке, называют **основной конфигурацией**. В соответствии с принципом открытой архитектуры аппаратные компоненты компьютера могут быть различными. Но для любого персонального компьютера есть набор обязательных и дополнительных устройств. Вы можете подсоединять дополнительные устройства в зависимости от целей использования компьютера. В настоящее время компьютер с четырьмя устройствами, представленными на рис. 2, рассматривается как базовая конфигурация (рис. 2).

Главное устройство компьютера – **системный блок** (рис. 3). В нем расположены основные устройства компьютера. Системный блок включает в себя процессор, оперативную память, постоянную память, источник питания, входные, выходные порты и носители информации.

Устройства в системном блоке являются внутренними устройствами, а устройства вне блока называются **внешними устройствами**.

Системный блок персонального компьютера оснащен разъемами для подсоединения дополнительных устройств. Также в нем размещены основные устройства. Чтобы просмотреть внутренние устройства, надо открыть крышку корпуса.



Рис. 2. Основная конфигурация компьютера



Рис. 3. Состав системного блока



Рис. 4. Порты системного блока

Вид корпуса системного блока может быть различным. На передней панели системного блока расположены кнопка включения/выключения компьютера, дисковод для чтения компакт-дисков (DVD)-ROM. На некоторых компьютерах есть кнопка перезагрузки компьютера.

На задней панели системного блока расположены порты (монитор, мышь, клавиатура), разъемы (принтер, modem, сканер, микрофон) (рис. 4).

**Монитор** – устройство, предназначенное для вывода информации на экран. Внешне он похож на экран цветного телевизора.

Количество вертикальных и горизонтальных пикселей показывает разрешение монитора. Насколько большее количество пикселей на экране монитора, настолько же увеличивается разрешающая способность монитора, выше качество и цвет изображений.

Монитор является одним из основных устройств вывода. Его основными параметрами являются: размер экрана, глубина цвета, максимальная частота обновления, класс защиты. Размер экрана определяется его диагональю. В качестве единицы измерения выбран дюйм. Стандартные размеры экрана – 14', 15', 17', 19', 20', 21'. В последнее время распространены размеры в 15 и 17 дюйм. А для работы с графическими изображениями эффективно использовать мониторы с 19–20 дюймами.

### Внутреннее устройство персонального компьютера

- 1 Процессор
- 2 Жесткий диск
- 3 Память
- 4 Шины
- 5 Слоты

Материнская плата является основным устройством персонального компьютера. К ней подсоединяются следующие устройства.

**Процессор** – основная часть компьютера, другими словами, «мозг» компьютера. Он управляет компьютером и выполняет все арифметико-логические вычисления. Он распознает команды, выполняет их, извлекает результаты или записывает их обратно в память и даже выполняет оба действия одновременно.

Процессор работает совместно с памятью. Он получает необходимую информацию из микросхемы памяти и отправляет результат своей работы обратно в память. Процессор напрямую определяет класс компьютера. Процессор соединяется с другими устройствами компьютера шинами, в первую очередь, с оперативным запоминающим устройством (ОЗУ). Имеются **три основных вида шин: шина данных, адресная шина и командная шина**.

**Память компьютера** подразделяется на внутреннюю (основную память), регистры процессора и внешнюю память.

**Внешние устройства персонального компьютера.** Внешние устройства персонального компьютера подключаются к системному блоку и выполняют вспомогательные функции. Внешние устройства можно поделить на следующие четыре группы:

1

**Устройства ввода:** клавиатура, сканер, микрофон, веб-камера, манипуляторы

2

**Устройства вывода:** монитор, принтер, плоттер, динамики, наушники

3

**Устройства хранения:** различные флаш-накопители

4

**Устройства обмена:** модем и маршрутизатор

### Проверка конфигурации работающего компьютера

При работе компьютера вы можете узнать конфигурацию вашего компьютера, его устройства и программное обеспечение. Для этого надо выполнить команду: Пуск – Все программы – Стандартные – Служебные – Сведения о системе (рис. 5). В этом окне вы можете увидеть информацию об аппаратном обеспечении компьютера, технических характеристиках оборудования и установленном программном обеспечении.

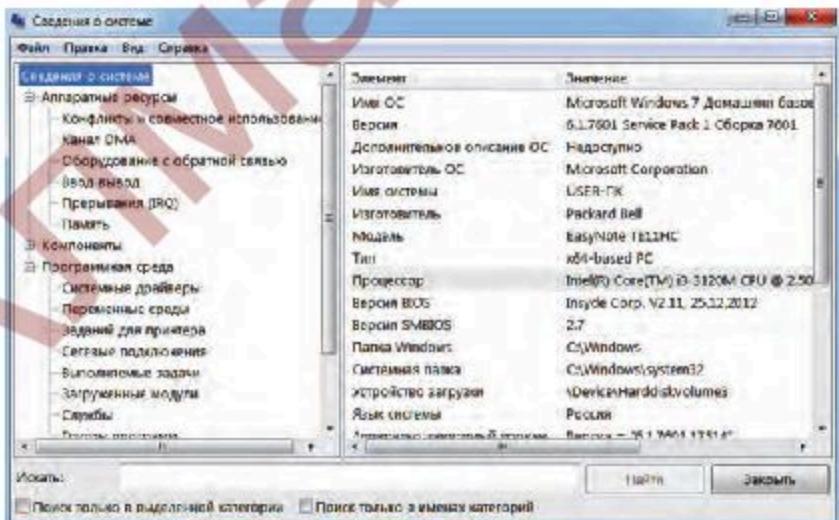


Рис. 5. Сведения о системе

Для выбора конфигурации компьютера надо знать все необходимые устройства и их технические характеристики. Только тогда вы выберете правильную конфигурацию своего компьютера в зависимости от ваших потребностей и рабочих целей.



### Анализ



В зависимости от целей использования проанализируйте конфигурацию компьютера. Определите технические характеристики компьютерных устройств в зависимости от целей назначения.



### Синтез



Выберите устройства для конфигурации компьютера специалисту по дизайну.



### Вопросы

1. Какие виды компьютера существуют? Дайте им характеристики.
2. Что такое персональный компьютер?
3. Что относится к внутренним устройствам компьютера? Опишите их.
4. Что относится к внешним устройствам компьютера?  
Охарактеризуйте их.
5. Как просмотреть описание программного обеспечения и устройств, установленных на компьютере?



### Домашнее задание

#### Задания для проектной работы

	Темы исследований	Вопросы и направления исследований
1	Конфигурация компьютера	<ul style="list-style-type: none"> <li>• Цели изменения конфигурации компьютера</li> <li>• Виды внутренних и внешних устройств современного компьютера</li> </ul>
2	Характеристики новейших процессоров	<ul style="list-style-type: none"> <li>• Каковы характеристики новейших процессоров?</li> <li>• Тенденции развития современных процессоров</li> </ul>
3	Мониторы компьютера	<ul style="list-style-type: none"> <li>• Жидкокристаллические мониторы</li> <li>• Плазменные мониторы</li> <li>• LED (светодиодные) мониторы</li> </ul>

## Раздел II. Выбираем компьютер

## 2.2

## ВЫБОР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



Выбор программного обеспечения, необходимого пользователю.



## Подумай

- Знаете ли вы, какие программы установлены на компьютере?
- Как их можно узнать?
- Какие вы знаете операционные системы и их версии?
- Как вы думаете, насколько важно знать характеристики программного обеспечения во время работы на компьютере?



## Новые знания

**Программа** – набор указаний на языке машины для выполнения и сохранения на магнитном носителе (диске) в виде файла, загружаемых по команде человека в память компьютера.

**Программное обеспечение** – совокупность программ и соответствующей технической документации, предназначенных для исполнения на компьютере при решении определенных задач. Комплекс всех программных средств, инструкций для компьютера называют программным обеспечением (Software). Любое аппаратное обеспечение (Hardware) без программы является просто набором элементов, оно не сможет быть использовано по назначению.

В настоящее время программное обеспечение по назначению подразделяется на три группы (*Схема 1*). С каждым годом разработчики программного обеспечения не просто обновляют интерфейс своего продукта, но и создают современное и сложное программное обеспечение с рядом полезных и интересных функций. Пользователи, ознакомившись с возможностями новой версии программного обеспечения, немедленно стремятся купить и установить ее на своих компьютерах. Конечно, лучше всего установить новейшую версию программы на свой компьютер. Но соответствует ли компьютер системным требованиям для установки этого программного обеспечения? Многие пользователи забывают об этой проблеме. Программное обеспечение, не соответствующее системным требованиям компьютера и установленное на нем, будет тормозить его работу. Вот почему покупатель каждый раз перед покупкой новой

## Виды аппаратного и программного обеспечения



Схема 1. Аппаратное и программное обеспечение

программы на компьютер должен знать технические характеристики и возможности своего компьютера.

Если ваша цель – быстродействие компьютера, оперативная загрузка операционной системы и ускоренный запуск программ, то вам нужно выбирать программное обеспечение, соответствующее техническим параметрам компьютера. Например, все игры и программное обеспечение, выпущенные в 2005 году, не смогут корректно работать на компьютере 2015 года выпуска. Это связано с высокими требованиями к программному обеспечению. Соответственно, должны улучшаться характеристики самого компьютера. Так как же выбрать программное обеспечение, чтобы оно не влияло отрицательно на быстродействие компьютера?

**1-й этап. Технические характеристики.** В первую очередь мы должны знать, какие ресурсы и возможности имеются на нашем компьютере. Если у вас нет документации по техническим характеристикам вашего компьютера, то вам нужно руководствоваться папкой **настройки** операционной системы. Например, в операционной системе Windows 7: выполнив команды **Пуск → Все программы – Стандартные – Выполнить**, в появившемся диалоговом окне надо набрать команду **DXDIAG** и нажать кнопку **OK** (рис. 1). В результате появляется окно «Средство диагностики DirectX», по ко-

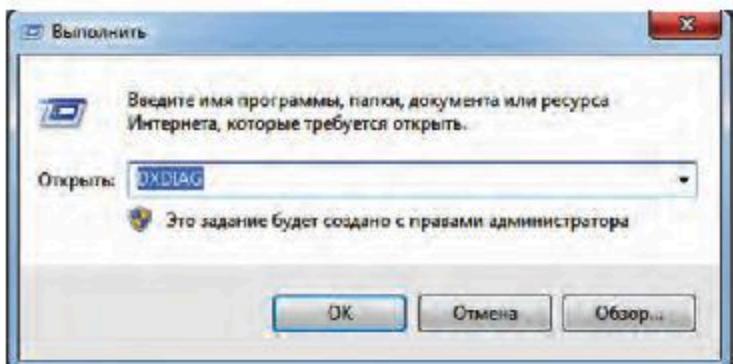


Рис. 1. Окно выполнения команд

торому вы сможете узнать о параметрах персонального компьютера. Это окно содержит характеристики операционной системы, процессора, памяти (ОЗУ); видеокарты, монитора, клавиатуры и звуковой карты (рис. 2).

**2-й этап. Системные требования.** После ознакомления с техническими характеристиками компьютера вы должны просмотреть установленное на нем программное обеспечение. Для этого выполните команды: Пуск → Панель управления → Программы и компоненты (рис. 3).

В открывшемся окне вы увидите список всего программного обеспечения, установленного на вашем компьютере. Вам необходимо ознакомиться с характеристиками этих программ в интернете и удалить те, которые не соответствуют параметрам вашего компьютера.

**3-й этап. Замена необходимых программ другими.** Что делать, если вам нужна программа, которая несовместима с техническими

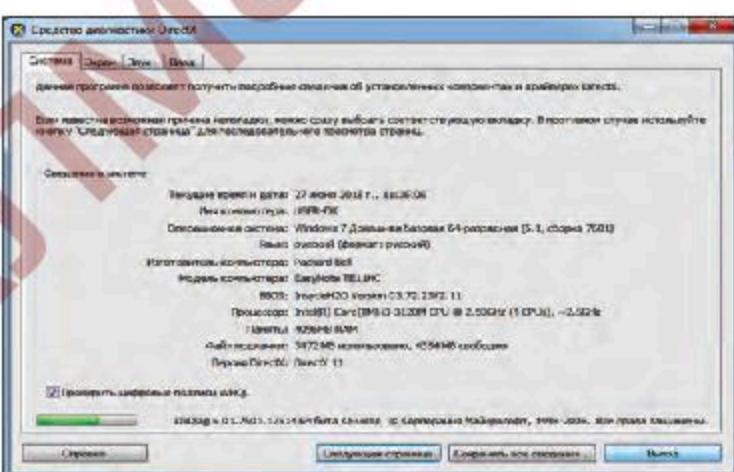


Рис. 2. Окно диагностики DirectX

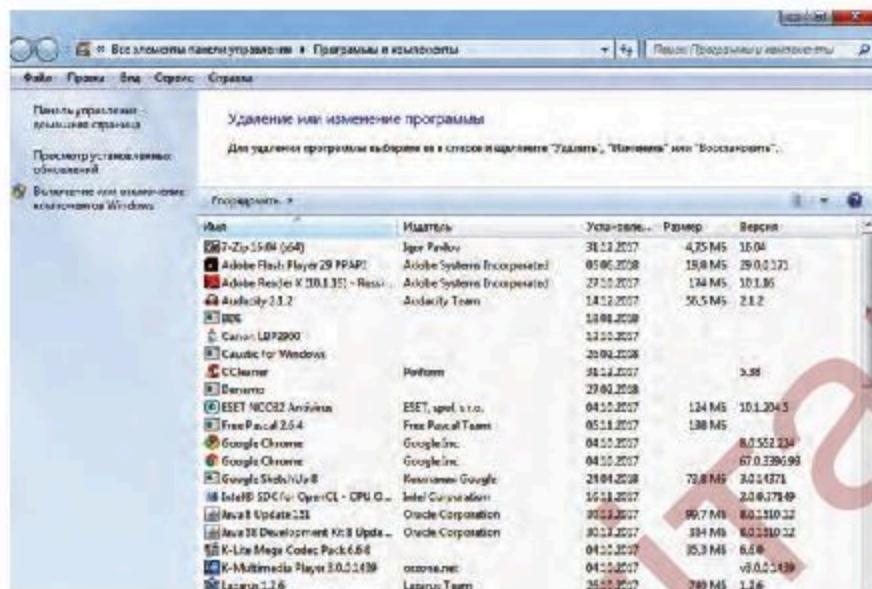


Рис. 3. Окно просмотра программ

требованиями вашего компьютера? В этом случае единственное, что нужно сделать, – это выбрать программу, выполняющую необходимые вам функции, но имеющую системные требования к компьютеру. На сегодняшний день существует множество таких программ. Например, Adobe Photoshop CS5 – очень мощный графический редактор. Он предъявляет высокие требования к техническим характеристикам компьютера. Но многие начинающие пользователи могут не использовать все функции этой программы. В таких случаях вы можете использовать бесплатный и простой аналоговый графический редактор GIMP. Он популярен среди отечественных и зарубежных пользователей.

Конечно, вы можете найти альтернативное программное обеспечение в интернете. Наиболее выгодным является использование сервиса SUGGESTUSE. Если вы напишете название программного обеспечения в этом сервисе, оно мгновенно покажет другое программное обеспечение, которым можно заменить нужное вам программное обеспечение.

### Классификация программного обеспечения по применению

**1. Программное обеспечение свободного пользования (Open Source).** Эти программы могут свободно использоваться, редактироваться и распространяться. Например,



Linux



OpenOffice



Firefox



Gimp

2. **Бесплатное программное обеспечение (Freeware).** Эти программы бесплатны не для коммерческого использования. Например,



Opera



Avast

3. **Условно-бесплатное программное обеспечение (Shareware).** При использовании этих программ не все службы функций выполняются в полном объеме. Срок использования ограничен (30 дней), ограниченное количество запусков, обязательно присутствует реклама. Например,



Nero BurningRom

**Коммерческое программное обеспечение.** При использовании этих программ каждое копирование платное, изменение кодов программ и данных запрещено.

**Анализ**

Проверьте со своими одноклассниками технические характеристики вашего компьютера в DXDIAG. Проанализируйте выбранное программное обеспечение.

**Синтез**

Какое прикладное программное обеспечение вы предложили бы на основе анализа технических характеристик компьютеров, имеющихся в наличии в классе? Обоснуйте список программ.

**Оценка**

Насколько важен правильный выбор программного обеспечения при использовании компьютера? Сделайте анализ.

**Вопросы**

- Что такое программное обеспечение?
- Какие существуют виды программного обеспечения?
- Что такое прикладная программа?
- Какие вы знаете распространенные прикладные программы?
- Какие имеются способы определения технических характеристик компьютера?
- Как классифицируется программное обеспечение по применению?
- Как выбрать программное обеспечение, чтобы оно отрицательно не влияло на быстродействие вашего компьютера?

**Домашнее задание**

- Определите технические характеристики персонального домашнего компьютера или ноутбука и заполните следующую таблицу.

Устройства	Технические характеристики
Процессор	
ОЗУ (оперативная память)	
Видеокарта	
Звуковая карта	
Монитор	
Клавиатура	

- Проанализируйте программное обеспечение персонального компьютера или ноутбука.

Название программ	Наименование	Вид применения (свободно, платно...)
Операционная система		
Текстовое ППО		
Графическое ППО		
Видео ППО		
Офисное ППО		
Браузеры		
Архиватор программ		

Примечание: ППО – прикладное программное обеспечение.

## Раздел II. Выбираем компьютер

## 2.3

## ВЫПОЛНЕНИЕ ПРОЕКТА «РАСЧЕТ СТОИМОСТИ КОМПЬЮТЕРА»



Как можно выбрать компьютерную конфигурацию и программное обеспечение в зависимости от ваших потребностей и возможностей?



## Подумай

- На что нужно обратить внимание при покупке компьютера?
- Как вы думаете, где и как лучше купить компьютер?
- Как выбрать конфигурацию компьютера и программное обеспечение в зависимости от ваших потребностей и возможностей?
- Что для этого нужно знать?



## Новые знания

В настоящее время можно собрать огромное количество компьютеров, так как имеется большое количество магазинов по продаже их комплектующих. Придя в магазин для покупки компьютера, вы не сразу определитесь из-за многообразия предлагаемых товаров. Если вы спросите совет у продавцов, они поспешат продать вам самый дорогой товар, не учитывая вашу цель покупки компьютера и покупательскую способность. Как при такой ситуации самому выбрать нужный компьютер? При выборе компьютера важно правильно подобрать внутренние устройства системного блока, монитор, клавиатуру и мышь. Также не будет особых проблем с выбором периферийных устройств – принтера, сканера, внешнего модема, динамиков и т.д. Поскольку эти устройства выбираются индивидуально и не зависят от работы друг друга.



Прежде чем ответить на вопрос: «**Как выбрать компьютер?**», нужно выяснить, для каких целей вы собираетесь использовать свой компьютер. Вы можете выбрать компьютер для работы в офисе, в сети Интернет, игр, прослушивания музыки, просмотра видео или для всех этих целей одновременно. При выборе вам нужно учитывать соотношение качество – цена. Допустим, мы выяснили, зачем компьютер нужен вам. Это даст возможность составить требования к компьютеру, необходимые для реализации

поставленных задач. Чтобы выбрать компьютер неопытному человеку необходимо знать следующие параметры:

**1. Тип процессора и тактовую частоту** (единица измерения гигагерц – ГГц). Чем выше тактовая частота, тем выше производительность компьютера. Также нужно обратить внимание на количество ядер.

**2. Объем оперативной памяти** (единица измерения гигагерц – ГГц). Сейчас не рекомендуется брать меньше 2 Гб. Предпочтительно более 4 Гб. Помните, что 32-битные операционные системы не поддерживают оперативную память свыше 4 Гб. Таким образом, если вы возьмете 8 Гб оперативной памяти для 32-битной операционной системы, все равно это не прибавит быстродействия компьютеру.

**3. Видеокарта.** Если вам нужен графический или игровой компьютер, видеокарта должна быть съемной, т.е. невстроенной. Единица объема – Мб (Мегабайт) или Гб (Гигабайт). Частота – Гц.

**4. Материнская плата.** Частота шины (Гц) является ее основным показателем.

**5. Объем и быстродействие жесткого диска.** Чем больше объем жесткого диска, тем больше информации в него помещается. Единица измерения – гигабайт (Гб), или терабайт (Тб). 1 Тб = 1024 Гб. При работе с большим объемом информации надо выбрать больший объем жесткого диска. Например, 1 видеофайл в среднем занимает 0,5 – 1 Гб объема. Если вы приобретете диск объемом 500 Гб, вы сможете хранить 300–400 таких видео и программ. Обычно хватает 320 Гб.

**6. Операционная система (ОС).** В продаже имеются операционные системы Windows, Mac OS, Linux. Большинство людей покупает компьютер без операционной системы и устанавливает ее после покупки. Цена компьютера без установленных программ будет дешевле. При необходимости можно удалить ОС и установить более подходящую.

**7. Оптический привод.** Вы должны выбрать оптический привод, способный читать и записывать большое количество форматов, например, super multi.

**8. Обратите внимание на наличие охлаждающих устройств и различных внешних аксессуаров.** В зависимости от портов PCI, DVD-приводов и многого другого могут быть разные карты расширения. USB более удобен, особенно в ноутбуках. Также нужно иметь картридер, он может легко открыть различные флешки. Это особенно важно для людей, которые работают с камерой или мобильным телефоном. Если ваш телевизор оснащен разъемом hdmi, то, используя специальный разъем на вашем компьютере, можно подключить видео к телевизору.

**9. Монитор.** Желательно выбрать качественный монитор. Электромагнитное излучение некоторых мониторов может превышать нормы, поэтому вы должны знать стандарты и разбираться в специальных знаках.

### Онлайн-расчет стоимости компьютерных устройств

Если вы знаете характеристики внутренней конфигурации ПК, описанные выше, продукты какой фирмы имеют высокое качество, то вы можете рассчитать стоимость необходимого вам компьютера на специальных сайтах в интернете (рис. 1). Таких сайтов достаточно много в сети Интернет.

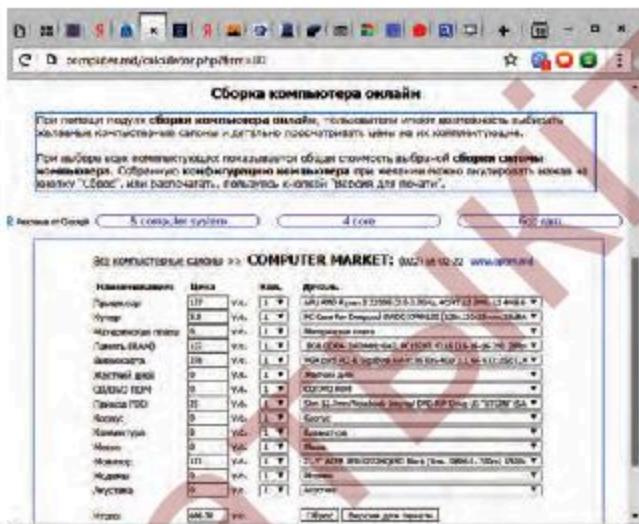


Рис. 1. Сайт онлайн-расчета стоимости компьютера



### Анализ



Определите параметры компьютера в соответствии с целью использования (дом, игра и т.д.). Для каждого компьютера должны быть даны технические характеристики. Составьте список необходимых устройств. Например, какие характеристики должны быть в игровом компьютере.



### Синтез



Смоделируйте таблицу «Калькулятор стоимости компьютера» в Excel, используя данные из сети Интернет и компьютерную литературу (рис. 1). Составьте таблицу, содержащую название компьютерных устройств, количество и формулы для вычисления стоимости. Компьютеры могут быть дешевые или дороже из-за изменений обменного курса доллара США на импорт комплектующих компьютеров в Казахстан (рис. 2).

Б9	Жесткий диск	Г.	Д.	Г.
1	РАСЧЕТ СТОИМОСТИ КОМПЬЮТЕРА	USD\$-	340	
4	№ Устройство компьютера	количество	USD \$	тенге
5	1 Процессор	1	100	34000
6	2 Кулер	2	24	16320
7	3 Материнская плата	3	60	61200
8	4 Память(RAM)	2	40	77200
9	5 Жесткий диск	2	0	0
10	6 Видеокарта	3	0	0
11	7 Звуковая карта	1	0	0
12	8 Монитор	1	0	0
13	9 Клавиатура		0	0
14	10 Мыши		0	0
15	11 Картридер		0	0
16	12 Дисковод		0	0
17	13 Модем		0	0
18	14 Операционная система(версии Windows)		0	0
19	<b>ОБЩАЯ СУММА</b>	<b>224</b>	<b>138720</b>	

Рис. 2. Таблица онлайн-расчета стоимости компьютера

**Оценка**

Считаете ли вы, что важно знать особенности покупки компьютера?

**Вопросы**

1. Как вы думаете, почему цена компьютера относительная?
2. Каковы преимущества расчета стоимости компьютера в интернете?
3. Какими должны быть параметры устройств при покупке компьютера для игр?
4. С какими характеристиками необходимо выбрать компьютер дизайнеру?

**Домашнее задание**

	Тема проекта	Основное направление проекта
1	<b>Программное обеспечение</b>	<ul style="list-style-type: none"> <li>• История разработки программного обеспечения</li> <li>• Индустрия (отрасль) программного обеспечения</li> </ul>
2	<b>Как выбрать для себя лучший и относительно дешевый компьютер?</b>	<ul style="list-style-type: none"> <li>• Параметры устройств современного компьютера</li> <li>• Основные проблемы выбора компьютера</li> </ul>

### 3.1 БАЗА ДАННЫХ



Как вы понимаете смысл терминов – *база данных, запись, поле?*  
Какое значение имеет в вашей жизни понятие *база данных*?



#### Подумай

- Как вы храните и обрабатываете объемную информацию?
- Что такое база данных?
- Какие области вы можете назвать, где база данных играет важную роль? Приведите соответствующие примеры.
- Какие вы знаете программы для создания базы данных?



#### Новые знания

База данных (БД) используется для хранения и обработки больших объемов систематизированной информации. В настоящее время термины «база данных» и «система управления базами данных» используются при работе на компьютере. **Данные** являются составной частью информации. Информация записывается, хранится и распространяется на разных носителях.

**База данных** – совокупность большого объема данных, систематизированных по определенным свойствам.

#### База данных:

- каталоги библиотек;
- телефонный справочник;
- данные клиентов банка;
- данные рабочих кадров;
- список пациентов больницы;
- школьные журналы.



Рис 1. Использование БД в повседневной жизни

## Типы баз данных

**По способу сохраненных данных:** фактографическая и документальная.

**По способу хранения данных:** усредненная, распространенная.

**По способу организации данных:** реляционная (табличная), иерархическая и сетевая (*Схема 1*).



Теория управления базами данных начала развиваться как самостоятельный предмет в начале 50-х годов XX века. За это время была сформирована система понятий баз данных. Например:

**Объектом** называется элемент информационной системы, хранящийся в базе данных.

**Атрибут** (латинский – *attributio*) – запись об определенных параметрах каждого объекта БД.

Атрибут является основным элементом БД.

### Реляционная база данных

Поле

Запись

Фамилия	Имя	Адрес	Телефон
Асанов	Канат	ул. Б. Момышулы, дом 32	325-41-41
Зиябек	Айдос	ул А. Молдагуловой, дом 25	212-35-21

**Поле** – это столбец таблицы, отражающий определенные свойства объекта.

**Запись** представляет собой строку таблицы, содержащую набор значений, описанных в полях базы данных.

**Ключевое поле** – поле, которое однозначно идентифицирует запись в таблице. В ключевом поле чаще всего располагаются порядковый номер, инвентарный номер, коды продуктов и другие.

Когда создавались первые базы данных, в них данные только хранились. Но многие современные системы управления базами данных включают не только хранение данных, они также способны решать самые разнообразные задачи. Современные базы данных являются основой многих информационных систем. Информация, которую они содержат, является ценным материалом. В настоящее время широко распространены способы обработки данных, которые тесно связаны с различными дополнительными способами работ с данными. В этой концепции база данных будет действовать как хранилище данных, и это направление называется «Хранилище данных» (Data Warehouse). Известно, что база данных является популярным понятием на сегодняшний день. В то же время информационные системы, использующие базы данных в области развития науки, здравоохранения, банков и производственных компаний, динамично развиваются.

### Правила создания реляционной базы данных

Существуют свои правила создания базы данных:

- Наименование полей (порядковый номер, фамилия, имя, адрес, домашний номер телефона) – размещается в первой строке.
- Заголовки столбцов форматируют (заголовок, цвет текста и т.д.).  
Например:
- Не допускается пустая строка между заголовком данных и записями.

	C2			Fx		
1	A	B	C	D	E	F
2		Фамилия	Имя	Адрес	телефон	...
3						
4						

- Между записями базы данных не должно быть пустых строк.
- В поле базы данных должны вводиться однотипные данные.
- Все записи БД должны располагаться на одном рабочем листе.

Сегодня существует множество программ для обработки баз данных. В зависимости от их использования различают однопользовательские и многопользовательские базы данных.

К программному обеспечению для создания однопользовательской базы данных относятся Visual FoxPro, Paradox, Clipper, dBBase, Access и многие другие. Программное обеспечение для создания многопользовательских баз данных разработано такими компаниями, как Oracle, Informix и другие.



Рис. 2. Программное обеспечение для создания баз данных



### Анализ

В каких сферах, по вашему мнению, важно работать с базами данных? Проведите анализ. Приведите примеры.



### Синтез

Создайте иерархическую и реляционную базы данных (данные об одноклассниках, взаимосвязь учебников и т.д.) в зависимости от ваших повседневных потребностей.



### Вопросы

- Что такое база данных?
- Какие типы баз данных существуют?
- Что такое запись и поле в базе данных?
- Какие правила существуют при создании реляционной базы данных?
- Какое программное обеспечение вы можете отметить для создания и обработки баз данных?



### Домашнее задание

Подготовка статьи «Значимость работы с базами данных в современном обществе» для научного журнала. Статья должна отражать важность базы данных в различных областях – в промышленности, в правительстве, на заводе, в банках, в библиотеке, в больнице и т.д.



### Оценка

## 3.2 СОЗДАНИЕ БАЗЫ ДАННЫХ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ MS EXCEL



Как можно использовать электронные таблицы MS Excel в качестве базы данных?



### Подумай

- Каковы особенности работы с ячейками в таблице Excel? Вспомните. Обсудите.
- Что может относиться к простейшей базе данных в электронной таблице?
- Из-за каких особенностей Excel считают удобным для использования в качестве базы данных?



### Новые знания

**Электронная таблица (spreadsheet)** – интерактивная система обработки данных, заданных в виде таблицы. В ячейки таблицы записываются формулы, содержащие адреса ячеек, функции; числа; текст. Пользователь имеет возможность записывать данные в ячейки, считывать данные из ячеек, изменять их, а также форматировать данные, осуществлять поиск и редактирование.

В большинстве случаев вам необходимо хранить и редактировать данные в виде таблицы. Информация, хранящаяся в таблицах, организована как строки и столбцы. Строки называются **Записями**, а столбцы – **Полями**. Если в **Записи** размещаются данные об объекте, то **Поле содержит какое-либо его свойство**. Таким образом, базой данных (БД) можно назвать набор разных записей с определенными свойствами. В первой строке любой базы данных должны отображаться имена полей. Максимальный размер базы данных в MS Excel определяется особенностями версии Excel (количество строк и столбцов на листе).

Вы также можете создать базу данных на одной странице. На одном листе может быть несколько баз данных, но только одна из них может быть активной при выполнении различных действий. Простейший тип базы данных состоит только из одной таблицы. Если вы имеете одну таблицу в Excel, то можете использовать ее как базу данных. Excel содержит набор функций, выполняющих все операции, специфичные для базы данных.

Чтобы рассматривать рабочий лист Excel как базу данных, должны выполняться следующие требования:

- каждому столбцу рабочего листа должно соответствовать только одно поле;
- поля базы данных должны располагаться одно за другим без пробелов;
- в первой строке каждого столбца должны отображаться заголовки полей, которые не должны повторяться;
- заголовок одного поля должен размещаться в одну ячейку;
- записи должны располагаться непосредственно под заголовком;
- записи ведутся последовательно одна за другой, не оставляя при этом пустых строк; если встречается пустая строка, то это понимается как конец базы данных.

В предыдущих классах вы научились работать с электронными таблицами Excel. Как теперь будете использовать Excel в качестве базы данных? Закрепим теоретический материал на практическом примере.

В 8-м классе тема *Электрический ток* в предмете «Физика» дается в виде таблицы (самая простая реляционная БД) зависимости тока от напряжения и сопротивления. Как вы обработаете эту таблицу в виде базы данных? Давайте рассмотрим подробнее (рис. 1).

Сила тока (А)	Напряжение (В)	Сопротивление (Ом)
11.0	220	20
7.0	210	30
5.0	200	40
3.8	190	50
3.0	180	60
2.4	170	70
2.0	160	80
1.7	150	90
1.4	140	100
1.2	130	110
1.0	120	120
0.8	110	130

Рис. 1. Простая база данных

В Excel инструменты для работы с базой данных представлены в разделе меню «Данные». **Форма** для работы с базой данных должна быть размещена на панели быстрого доступа (панель инструментов). Для этого нажмите «Панель быстрого доступа» в меню «Другие команды» – «Команды не на ленте». Если вы выберете «Форма» из списка команд, то они отобразятся в окне (рис. 1).

Теперь перейдем к рабочей области, в которой описывается зависимость тока от напряжения и сопротивления. Уникальная особенность Excel заключается в том, что у вас есть возможность создавать базу данных в рабочей области, не при-

бегая к использованию другого программного обеспечения. К таким базам данных можно отнести данные, которые отражают зависимость силы тока от напряжения и сопротивления в виде простой таблицы, описанной в рассмотренном примере. При заполнении ячеек вам необходимо взять за основу правила заполнения базы данных в Excel.

В Excel первоначальной базой данных может считаться **Список (Таблица 1)**, который состоит из простых таблиц. Мы также рассмотрим **Диапазон** базы данных как важное понятие.

**Диапазон** – это таблица, которая выполняет функцию базы данных и расположена в рабочей области листа Excel.

В качестве части списка служит строка заголовка, содержащая имена полей. В рассматриваемом примере диапазон базы данных представлен тремя столбцами с заголовками полей – **Сила тока**, **Напряжение**, **Сопротивление** (рис. 1). Таблица, содержащая эти три столбца, рассматривается как база данных.

Как мы работаем с таблицей Excel в качестве базы данных?

Для работы с таблицами Excel в качестве базы данных выполним команду **Панель быстрого доступа – Excel**, в результате в рабочей области появится диалоговое окно Excel, предназначенная для работы с базами данных. Современные реляционные базы данных состоят из связанных таблиц. База данных в примере, который мы рассматриваем, является самой простой однотабличной базой данных. Диалоговое окно **Форма** показывает номер листа, на котором расположена база данных. С помощью диалогового окна **Форма** управление базами данных осуществляется очень легко и удобно.

The screenshot shows the Microsoft Excel 'Form' dialog box. At the top, it displays the range A1:D6. Below this, there is a table with three columns: 'Сила тока (A)', 'Напряжение (B)', and 'Сопротивление (Cm)'. The data rows are: Row 2: 11,0; 220; 20; Row 3: 4,0; 211; 31; Row 4: 5,0; 200; 40; Row 5: 3,3; 190; 50; Row 6: 2,0; 180; 60. The dialog box has a title bar 'Лист1'. It contains three input fields: 'Сила тока (A):' with value 5,0, 'Напряжение (B):' with value 500, and 'Сопротивление (Cm):' with value 40. On the right side of the dialog box are several buttons: 'Закрыть' (Close), 'Удалить' (Delete), 'Выделить' (Select), 'Новая' (New), 'Далее' (Next), 'Следующий' (Next), and 'Предыдущий' (Previous).

Рис. 2. Окно Форма

## Диалоговое окно Форма

В диалоговом окне **Форма** вы можете узнать количество полей и записей. На рисунке 2 в окне **Форма** представлена база данных, содержащая 3 поля и 12 записей. Текущая запись в форме – 3. Она отражает данные третьей записи полей – **Напряжение (В)** и **Сопротивление (Ом)**. В окне диалога **Форма** имеется несколько командных кнопок. С помощью этих кнопок можно добавлять записи в базы данных (**Добавить**), удалять записи (**Удалить**) или вернуть удаленную запись (**Вернуть**) (рис. 2). Вы можете использовать кнопки **Далее** и **Назад** для перемещения вперед и назад по записям в базе данных. Эти же функции выполняет **Полоса прокрутки**. Кнопка **Критерии** используется для поиска данных по заданному условию. Тогда вы можете видеть данные в других полях, соответствующие найденным данным. Завершение работы с базами данных осуществляется кнопкой **Закрыть**.

В Excel вы можете импортировать данные из внешних источников, не ограничиваясь созданием базы данных. Для пользователя, который хорошо владеет Excel, это очень удобно. Чтобы импортировать базу данных, взятую извне, вы должны выполнить команду **Данные – Получение внешних данных** (рис. 3).

	3	7,0	210	30
	4	5,0	200	40
	5	3,8	190	50
	6	3,0	180	60
	7	2,4	170	70
	8	2,0	160	80
	9	1,7	150	90
	10	1,4	140	100
	11	1,2	130	110
	12	1,0	120	120
	13	0,8	110	130

Рис. 3. Получение внешних данных

Как вы, видите на рисунке, внешние данные могут быть получены из базы данных Access, интернета, текстового файла и других внешних источников.



## Анализ – Синтез



Создайте простую базу данных по дисциплинам «Математика», «Геометрия», «Биология» и «Физика» (рис. 2). Созданная вами база данных не должна превышать 4–6 полей и 10–15 записей.



## Вопросы

- Что такое **Запись**, **Поле**?
- Что такое **Диапазон**?
- Что такое **Список**?
- Какова функция диалогового окна **Форма**?
- Как перейти к окну **Форма** из Панели быстрого доступа?
- Какие функции выполняют кнопки диалогового окна **Форма**?
- Как можно импортировать данные из внешних источников?

№	Наименование	Автор	Год выпуска
1	Копии царя Соломона	Генри Хагард	2011

Рис. 4. Пример базы данных библиотеки



## Домашнее задание

Помогите школьному библиотекарю создать базу данных казахской литературы и мировой классической литературы, отсортированную в рамках проекта «100 замечательных книг». Запишите 10 книг, которые, по вашему мнению, заслуживают включения в данный список, используя форму (рис. 4), а также создайте и сохраните базу данных под названием «100 замечательных книг».

P/c	Наименование	Автор	Год издания	Издательство
1	«Путь Абая»	Мухтар Аuezов	2006	Жазушы
2	«Солдат из Казахстана»	Габит Мусрепов	2007	Жазушы
	...	...	...	...

**3.3****МЕТОДЫ ПОИСКА ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ  
MS EXCEL**

Какие имеются методы поиска информации в электронной таблице MS Excel?

**Подумай**

- Как вы ищете нужную информацию в повседневной жизни?
- Каковы пути поиска информации на компьютере?
- Как можно найти нужную информацию в Excel?

**Новые знания**

Информация является важной частью нашей жизни. Люди не смогут общаться, не владея некоторой информацией. Важным делом для каждого человека, работающего с информацией, является ее поиск. Конечно, существуют определенные способы поиска информации. Раньше люди читали газеты и журналы, ходили в музеи и библиотеки, чтобы найти нужную информацию. Сейчас в считанные секунды можно найти необходимую информацию в сети Интернет и скачать ее. Поиск информации вы осуществляете по определенному критерию. В компьютере есть свои возможности поиска программного обеспечения для обработки полученной информации. Например, в текстовом редакторе Word есть возможность поиска текстовых фрагментов, команда для замены одного текста другим. Такой же вид поиска, и даже более сложные формы поиска цифровой информации, имеются в программе Excel. Остановимся на них.

Трудно быстро найти нужную информацию из многих данных, имеющихся на рабочем листе. В таких случаях Excel облегчает поиск и замену с помощью возможностей диалогового окна для нахождения нужной информации. Наряду с ними, рассмотрим ряд полезных функций, которые многие пользователи не знают. Для использования службы поиска.

Выполните команду ► Главная ► Редактирование ► Найти и выделить ► Найти или Ctrl+F (рис. 1). Как вы можете видеть в этом диалоговом окне, наряду с поиском и заменой можно искать и находить формулы, заметки и выделенные ячейки.

В диалоговом окне «Найти» вводится слово, которое вы хотите

найти, или можно заменить слово (символ) командой «Заменить» (рис. 2).

Во многих случаях можно задавать не точный текст, который нужно найти, а приблизительно подходящий. Например, поиск информации об Ахмедовой Айгуль на листе Excel, введя точный текст, вы можете не найти ее данные. Это может произойти по той причине, что одна буква имени не соответствует, пропущена или написана иным образом. В таких случаях используются методы, облегчающие поиск нужного текста. Если вы введете в окно поиска правильный запрос, то возможность найти необходимый текст будет увеличена. Например, если вы введете Ax\*медова\* Ай\*гуль\*, вы сможете найти нужный текст. Причина в том, что теперь осуществляется поиск текста (фрагменты) по заданному запросу, а не точный текст. Вероятность нахождения нужного текста увеличивается.

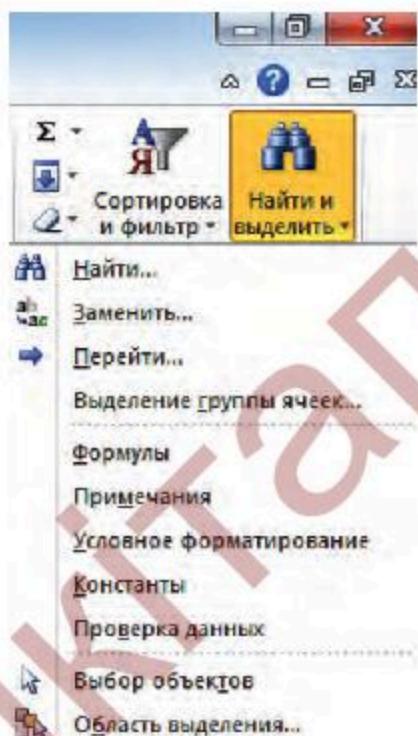


Рис. 1. Диалоговое окно «Найти и выделить»

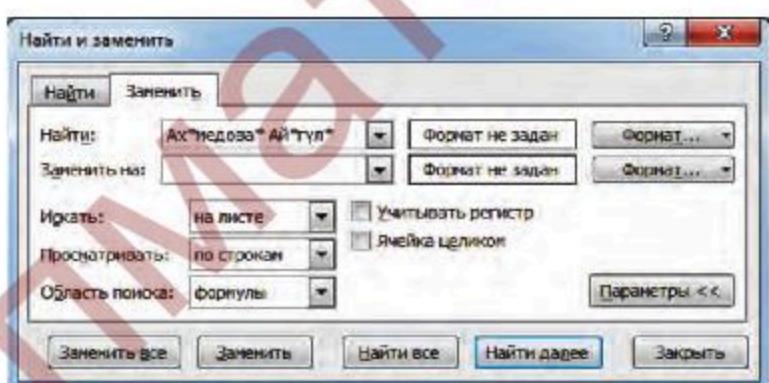


Рис. 2. Диалоговое окно «Найти и заменить»

Вы можете использовать два символа при поиске в диалоговом окне «Найти и заменить»:

- ? – заменяет один символ;
- \* – заменяет любое количество символов.

Вы также можете использовать эти символы при поиске числовых значений. Например, когда вы вводите 5\* в поле поиска, он находит все числа, начинающиеся с 5. Запись 3?7 находит все трехзначные числа, начинающиеся с цифры 3 и заканчивающиеся цифрой 7. Следует помнить, что поиск информации происходит только в пределах **выбранного диапазона ячеек**. Если вы хотите выполнить поиск по всему рабочему листу, то вам необходимо выбрать ячейку, с которой начнете поиск.

Вы не сможете найти числовые значения определенного формата, используя окно «Найти и заменить». Например, если вы введете \$5\* в поле поиска, то значение \$54.00 не будет найдено.

### Поиск данных в таблице

Существует несколько других способов поиска больших объемов данных в таблице. Эти же способы используются при поиске информации в программе. Рассмотрим следующий пример:

В разных городах страны цены на фрукты и овощи различаются в зависимости от географического положения и экономического потенциала (рис. 3). Используя данные из этой таблицы, определите цену определенного товара в конкретном городе, указав название города и название продукта. Например, найти помидоры в Шымкенте.

	А	Б	С	Д	Е	Г	Ж	И	К
1	Название товара	Астана	Алматы	Шымкент	Уральск	Наволодар	Караганда	Усть-Каменогорск	Актау
2	Огурцы	140	100	60	150	180	130	190	200
3	Помидоры	150	130	70	200	210	180	220	250
4	Абрикосы	100	250	150	320	400	320	450	500
5	Яблоки	350	300	350	400	300	300	450	500
6	Картофель	120	105	85	130	60	125	100	190
7	Лук	100	100	80	90	120	105	140	180
8	Морковь	120	400	70	120	130	115	140	150
9									
10	Область поиска	Расположение							
11	Название товара	Помидоры	2						
12	Название города	Огурцы	3						
13	Цена товара	Помидоры	70						
14		Абрикосы							
15		Яблоки							
16		Картофель							
		Лук							
		Морковь							

Рис. 3. Данные, отражающие стоимость товара

На рисунке 3 в рабочей области MS Excel в качестве образца представлена таблица, содержащая поля *наименование, название городов, стоимость товара*. Создайте данную таблицу и заполните ее своими данными. Отформатируйте ячейки таблицы.

В нижней части таблицы создайте «Окно поиска», содержащее *название продукта, название города и стоимость товара* в данном городе. Выполните команду **Данные – Работа с данными – Проверка данных**, чтобы проверить соответствие названия города и продуктов и увидеть их в виде списка в ячейке (рис. 4).

В области **Условие проверки** выберите в поле **Тип данных** свойство **Список**. В строке **Источник** вы можете ввести названия городов и продуктов вручную или выделить соответствующий диапазон с помощью мыши.

После создания окна поиска при выборе *названия города и товара из списка цена товара* в этом же городе автоматически появляется в окне поиска. По рассмотренному примеру цену помидоров в Шымкенте легко можно найти методом поиска (рис. 2). Как это выполнить? Для этого нам нужно знать назначение нижеследующих функций.

### Запись и назначение функции ПОИСКПОЗ

**ПОИСКПОЗ** (*искомое\_значение, просматриваемый\_массив, [тип\_сопоставления]*) – функция ищет заданное значение в строке или в столбце и возвращает ее порядковый номер (по строкам или столбцам). Отсчет номера начинается с первой позиции заданного диапазона.

**Искомое значение** – один из обязательных аргументов функции, представляет собой значение, которое запрашивается среди рассматриваемых элементов массива. Этим значением может быть текст, число, логическое значение.

**Просматриваемый массив** – один из обязательных аргументов функции, представляет собой диапазон области поиска данных.

**Тип\_сопоставления** – необязательный аргумент. Значением этого аргумента может быть число – 1, 0, 1. По умолчанию в качестве

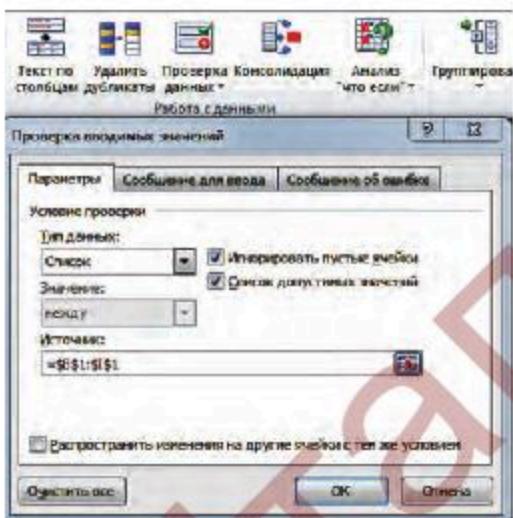


Рис. 4. Окно проверки вводимых значений

этого аргумента будет значение 1. Функция тип сопоставления указывает требования для поиска значений в массиве. Например, 0 – находит первоначальное соответствующее значение в массиве, 1 – любое наибольшее значение, -1 – соответствующее наименьшее значение исходного аргумента.

### Запись и назначение функции ИНДЕКС

**ИНДЕКС** (массив, номер\_строки, номер\_столбца) – функция, которая возвращает значение или ссылку на значение из таблицы или поименованного диапазона, заданного номером строки и номером столбца. В рассмотренном выше примере (рис. 3) для поиска **помидоров** в базе данных мы использовали формулу = ПОИСКПОЗ (B11; A2:A8; 0) для ячейки C11, а для поиска «Шымкент» использовали формулу = ПОИСКПОЗ (B12; B1:I1; 0) для ячейки C13. Формула = ИНДЕКС (B2:I8; C11; C12:E12) для ячейки C14 возвращает значение ячейки

Область поиска	Расположение
Название товара	Помидоры
Название города	Шымкент
Продукты	

Рис. 5. Окно «Проверка данных»

на пересечение исходной строки и столбца в диапазоне данных. Так вы можете быстро находить цену определенного товара в некотором городе в соответствии с имеющимися данными.



### Анализ – Синтез



Проанализируйте метод поиска данных в таблице. В таблице создайте проект для поиска данных из нашей повседневной жизни. Выполните задание на компьютере. Используйте функции ПОИСКПОЗ и ИНДЕКС.



### Вопросы

1. С какими методами поиска информации в электронной таблице Excel вы познакомились?
2. Каковы функции команды **Найти и выделить**?
3. Что такое заменяемый символ? Для чего он используется?
4. Каковы пути поиска табличных данных в Excel?
5. Для чего нужна функция ПОИСКПОЗ?
6. Какие аргументы есть у функции ПОИСКПОЗ?
7. В каких случаях используется функция ИНДЕКС?
8. Назовите аргументы функции ИНДЕКС.



## Задания

1. Для столбцов А, В, С в Листе 1 заполните строки от 1 до 20 произвольными числами от 1 до 999. Выполните нижеследующие операции:
- 1.1. Найдите все числа, начинающиеся с цифры 7.
  - 1.2. Найдите все трехзначные числа, которые начинаются с цифры 5 и заканчиваются цифрой 9.
  - 1.3. Замените все числа, начинающиеся цифрой 2, на 0.
2. В следующей таблице представлена зарплата сотрудников компании за 6 месяцев. Создайте таблицу с полями *список сотрудников*, названия месяцев. Определите зарплату работника №5 в апреле месяце.

А	Б	С	Д	Е	Ф	Г	Н	
1	№	Сотрудники	Январь	Февраль	Март	Апрель	Май	Июнь
2	1	сотрудник №1	100000	92000	102000	102700	109200	92000
3	2	сотрудник №2	120000	93500	103400	104000	110500	93500
4	3	сотрудник №1	110000	95000	104800	105300	111800	95000
5	4	сотрудник №2	150000	96500	106200	106600	113100	96500
6	5	сотрудник №1	130000	98000	107600	107900	114400	98000
7	6	сотрудник №2	80000	99500	109000	109200	115700	99500
8	7	сотрудник №1	105000	101000	110400	110500	117000	101000
9	8	сотрудник №2	80000	102500	111800	111800	118300	102500
10	9	сотрудник №1	67500	104000	113200	113100	119600	104000
11	10	сотрудник №2	55000	105500	114600	114400	120900	105500

Рис. 5. Таблица месячных зарплат

## Раздел III. База данных

## 3.4

## СОРТИРОВКА ДАННЫХ



Как выполняется поиск, сортировка и фильтрация данных в электронной таблице Excel?



## Подумай

- Как вы в повседневной жизни сортируете, упорядочиваете необходимую информацию?
- Как вы считаете, насколько важна сортировка при обработке больших объемов информации?
- С какой целью используется сортировка при работе с базами данных?



## Новые знания

В предыдущих параграфах данного раздела мы говорили, что Excel позволяет использовать списки как базу данных. Особенно Excel помогает работать с громоздкими и сложными таблицами в качестве баз данных. Вы можете быстро упорядочивать, сортировать, изменять, выполнять поиск и выбирать данные с помощью **встроенных инструментов** в Excel.

Основная работа в базе данных заключается в поиске информации в соответствии с заданными параметрами. По мере увеличения количества записей, поиск информации становится более сложным. Excel позволяет выполнять эту задачу с помощью сортировки и фильтрации данных.

Существует два типа сортировки базы данных в разделе **Данные – Сортировка и фильтр**. Для простейшего случая (рис. 1) показано выполнение команды Сортировка от А до Я и от Я до А.

## 1. Простая сортировка



Сортировка по возрастанию



Сортировка по убыванию

## 2. Сложная сортировка

К ней относится **сортировка**, которая выполняется по нескольким полям одновременно.

Одна из задач, выполняемых в базе данных, – отображение данных в определенном порядке, то есть сортировка данных. Сортировка



Рис. 1. Инструменты сортировки

данных может выполняться по одному или нескольким столбцам и может быть выполнена одним из двух способов: по возрастанию от **А** до **Я** или по убыванию от **Я** до **А**.

Например, диапазон данных сортируется по возрастанию от **А** до **Я** (рис. 2). При использовании этой команды появляется диалоговое окно сортировки (рис. 3). Если вы хотите сортировать базу данных по одному полю, вы должны выделить данные в этом столбце и выполнить одно из следующих двух действий (рис. 3):

**1) автоматически расширить выделенный диапазон**, то есть он автоматически расширяет выделенную область и выполняет полную сортировку по всем столбцам;

**2) сортировать в пределах указанного выделения.** Например, сортировка по фамилии учеников по возрастанию или убыванию приведет к тому, что отсортированные данные не будут согласованы, то есть фамилия ученика не совпадет с датой его рождения. В этом случае вы должны провести сортировку данных по действию 1.

	A	B
1	Начальные данные	Сортировка от А до Я
2	60	12
3	80	18
4	18	47
5	50	50
6	70	56
7	12	58
8	56	60
9	252	68
10	58	70
11	88	80
12	47	88
13	68	252

Рис. 2. Сортировка от А до Я

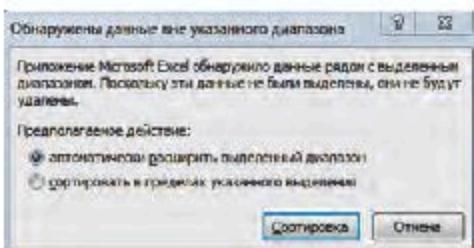


Рис. 3. Окно сортировки от А до Я

## Команда Сортировка

Чтобы отсортировать базу данных, выполните команды «Данные» – «Сортировка». Окно «Сортировка диапазона» позволяет вам выбирать, какие поля сортировать. Если диапазон состоит из полей, то вы должны выбрать одно из этих полей. Если в качестве названия полей определены имена столбцов, то вы должны выбрать соответствующее поле.

Диалоговое окно **сортировки** содержит три вкладки.

- Столбец Сортировать по полю, выбирается имя поля.
- Сортировка по значению, цвету ячейки, цвету шрифта.
- Порядок по возрастанию или убыванию (рис. 4).

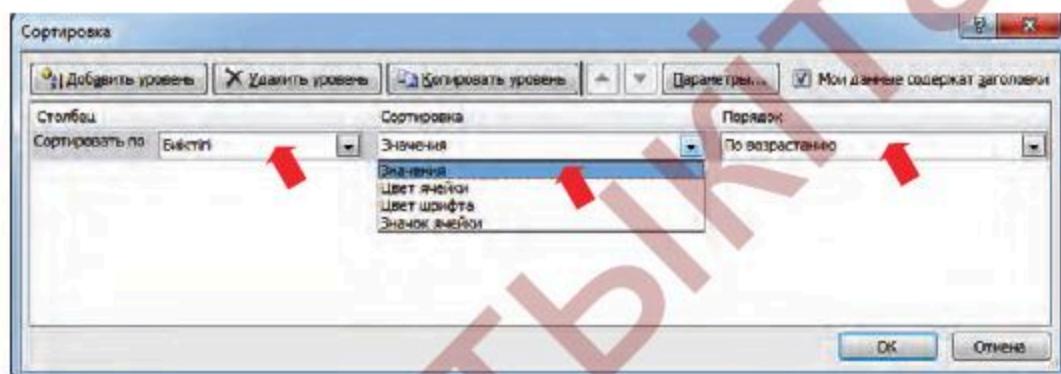


Рис. 4. Диалоговое окно Сортировка

Этим способом сортировки данные сортируются не только по значению, но и по цвету. Например, пусть даны *наименование* и *цена* ваших компьютерных устройств. И в зависимости от своих конкретных свойств они отличаются цветом (рис. 5.1).

	A	B	C
	Номер	Устройство	Цена (тенге)
1	5	Монитор	70000
2	6	DVD-ROM	15000
3	10	Сетевая карта	7000
4	11	Видеокарта	10000
5	1	Материнская плата	50000
6	2	Процессор	40000
7	8	Клавиатура	3600
8	9	Мышь	2500
9	3	Оперативная память	15000
10	4	Жесткий диск	15000

Рис. 5.1. Сортировка  
данных по цвету

A	B	C
	Устройство	Цена (тенге)
10	Сетевая карта	7000
11	Видеокарта	10000
5	Монитор	70000
6	DVD-ROM	15000
8	Клавиатура	3600
9	Мышь	2500
3	Оперативная память	15000
4	Жесткий диск	35000
1	Материнская плата	50000
2	Процессор	40000

Рис. 5.2. Данные, отсортированные  
по цвету

Результат сортировки по цвету показан в таблице на рис. 5.2. Вот как выполняется сортировка цвета в диалоговом окне «Сортировка» (рис. 6).

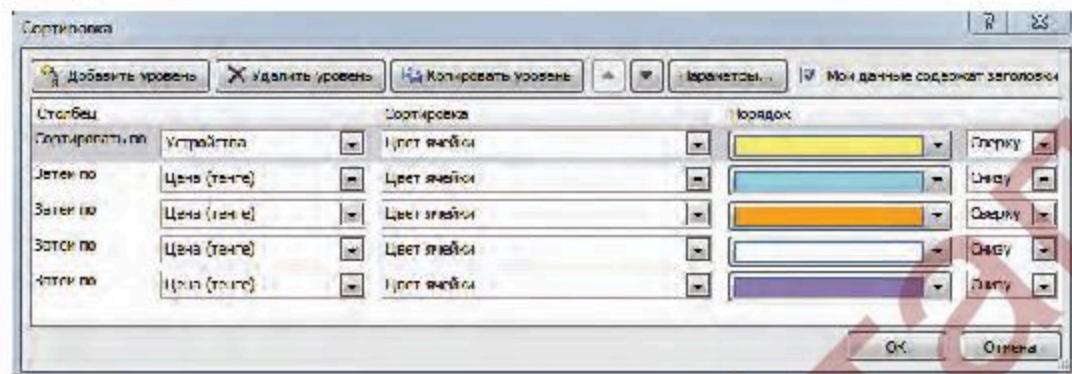


Рис. 6. Упорядочение в окне Сортировка

Теперь давайте рассмотрим конкретные практические примеры.



### Примени на практике

**Задание 1.** Ниже приведена реляционная база данных «Самые высокие здания и строения мира» (рис. 7). По этим данным надо найти:

- 1) самое высокое строение;
- 2) количество зданий и строений нашей страны;
- 3) отсортируйте строения по году возведения от старых к новым.

	A	B	C	D	E	F
1						
2						
Самые высокие здания и строения мира						
3	Название	Высота	Год	Тип	Страна	Город
4	Бурдж-Халифа	828	2010	Небоскрёб	ОАЭ	Дубай
5	Варшавская радиомачта	643,48	1974	Радиомачта	Польша	Копенгаген
6	Небесное дерево Токио	634	2012	Радиомачта	Япония	Токио
7	Телебашня Гуанчжоу	610	2009	Телебашня	КНР	Гуанчжоу
8	Останкинская башня	540,1	1967	Телебашня	Россия	Москва
9	Kingskey 100	439,8	2011	Небоскрёб	КНР	Шэньчжэнь
10	Бордж Машид	435	2003	Небоскрёб	Иран	Тегеран
11	Экибастузская ГРЭС-2	419,7	1987	Дымовая труба	Казахстан	Экибастуз
12	Абу-Даби Шиза	382	2015	Небоскрёб	Казахстан	Астана
13	Ташкентская телебашня	374,9	1985	Телебашня	Узбекистан	Ташкент
14	Атматипская телебашня	371,5	1983	Телебашня	Казахстан	Атмата
15	Рижская телебашня	366,5	1987	Телебашня	Латвия	Рига
16	Берлинская телебашня	368	1969	Телебашня	Германия	Берлин

Рис. 7. Диалоговое окно сортировки

**Задание 2.** Отсортируйте базу данных путем сортировки от А до Я и от Я до А.

Выполните сортировку:

- 1) Поле *Наименование* по алфавиту.
- 2) Поле *Высота строений* по возрастанию.
- 3) Поле *Год* по убыванию.



### Вопросы

1. Что такое сортировка данных?
2. В каком разделе меню расположена команда сортировки данных в Excel?
3. Как выполняются сортировки в таблицах Excel?
4. Какие отличительные особенности сортировок от А до Я и от Я до А?
5. С какой целью выполняется Сортировка?
6. Как выполняется сортировка в диалоговом окне «Сортировка»?



### Задания

На территории Казахстана насчитывается около 85 тысяч больших и малых рек. Общая протяженность только семи рек (Иртыш, Тобол, Есиль, Урал, Сырдария, Или и Чу) превышает 1000 км.

1.1. Введите данные в таблицу программы Excel. Отформатируйте по своему усмотрению. Постройте диаграмму длины рек.

1.2. Отсортируйте все реки, протекающие по территории Казахстана, по длине – от самой короткой до самой длинной.

1.3. Отсортируйте реки по их длине.

Река	Длина	В Казахстане
Тобол	1 591 км	800 км
Есиль	2 450 км	1 400 км
Иртыш	4 248 км	1 700 км
Чу	1 186 км	800 км
Сырдария	2 219 км	1 400 км
Урал	2 428 км	1 082 км
Или	1 439 км	815 км

## Раздел III. База данных

## 3.5

## ФИЛЬТРАЦИЯ ДАННЫХ



Как выполнять быстрый поиск данных с помощью фильтрации в электронной таблице Excel?



## Подумай

- Какие недостатки вы можете отметить в сортировке данных, с которыми познакомились при изучении предыдущей темы?
- При работе с какими данными выполнение сортировки вызывает затруднения?
- Какие можете привести примеры из повседневной жизни, когда можно быстро получить данные о себе в соответствии с определенными условиями?



## Новые знания

Суть работы с любой базой данных заключается в поиске определенной информации. Поиск становится все более сложным по мере наполнения базы данных. Фильтрацией таких больших объемов данных, вы можете облегчить поиск необходимой информации.

**Фильтрация** – один из самых удобных и быстрых приемов получения нужных данных из большого объема информации.

**Фильтрация** – это способ найти данные в списке в соответствии с конкретными условиями. В отличие от сортировки фильтрация не нарушает порядок записей в списке. В отфильтрованных данных отображаются только строки, соответствующие заданным условиям, а ненужные строки скрываются.

При фильтрации в Microsoft Excel выбранные строки могут быть отредактированы, отформатированы, распечатаны без изменения их порядка и на их основе можно создать диаграмму.

При использовании команды «Фильтр» в списке для фильтрации отображаются кнопки со стрелками в правой части наименования столбца. Microsoft Excel отражает отфильтрованные элементы в синем цвете (рис. 1).

В Microsoft Excel есть два вида фильтрации:

1. **Фильтр (автофильтр)** используется при простых условиях фильтрации.

2. **Расширенный фильтр.** Фильтрация по сложным условиям.

## Правила работы с инструментом **Фильтрация**:

**Фильтр (автофильтр)** работает только в непрерывном диапазоне. Различные таблицы на одной странице не фильтруются одновременно.

При фильтрации самая верхняя строка таблицы считается заголовком и к нему фильтрация не применяется.

В одной таблице можно выполнить несколько фильтров, но каждый предыдущий результат фильтрации может скрыть необходимые записи для следующего фильтра.

Рассмотрим нижеследующий пример практического применения фильтрации в Excel.

Дана база данных, содержащая данные о населении некоторых стран мира, наименования столиц, площадь территории и названия континентов. Рассмотрим фильтрацию данных в этой базе данных по определенным условиям (рис. 1).

Чтобы выполнить первый простой фильтр, мы зададим команду **Фильтр** в разделе **Данные – Сортировка и фильтр**.

	A	B	C	D	E
1	Название стран	Столица	Площадь, км <sup>2</sup>	Численность населения	Материк
2	Аргентина	Буэнос-Айрес	2 766 890	43 947 000	Южная Америка
3	Бразилия	Бразилия	8 511 965	206 081 432	Южная Америка
4	Германия	Берлин	357 021	82 155 684	Европа
5	Греция	Афины	1 001 450	11 701 723	Африка
6	Индия	Нью-Дели	3 287 590	1 258 561 000	Азия
7	Казахстан	Астана	2 734 900	17 751 200	Азия
8	Киргизстан	Бишкек	198 500	6 019 500	Азия
9	Китай	Пекин	9 640 821	1 410 549 741	Азия
10	Нигерия	Абуджа	923 768	186 988 000	Африка
11	Россия	Москва	17 102 345	146 541 700	Европа
12	США	Вашингтон	9 522 057	322 762 018	Северная Америка
13	Таджикистан	Душанбе	143 100	8 551 000	Азия
14	Туркменистан	Ашхабад	451 200	5 439 000	Азия
15	Узбекистан	Ташкент	417 100	31 807 000	Азия
16	Франция	Париж	517 030	64 513 212	Европа
17	Япония	Токио	377 944	127 103 388	Азия

Рис. 1. База данных

Если вам из списка на рис. 1 нужно выбрать страны, расположенные на азиатском континенте, то при нажатии на кнопку фильтр в заголовке «Континенты» (рис. 2) появится окно фильтра. Как видно из рисунка, здесь отражаются все имеющиеся континенты. Мы убираем галочки со всех надписей, кроме Азия, и тогда получим список стран только этого континента (рис. 3).

Во время фильтрации при работе с большими базами данных можно одним действием отобрать нужные числа.

В этом случае используем команду **Числовые фильтры** (рис. 4). Как вы видите на рисунке, есть несколько критериев для отбора чисел с помощью **Числовых фильтров**. Можно сделать

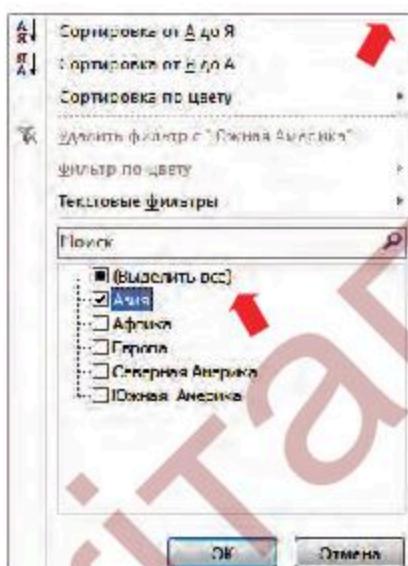


Рис. 2. Фильтрация континента Азия

A	B	C	D	E
	Название стран	Столица	Площадь, км <sup>2</sup>	Численность населения
6	Индия	Нью-Дели	3 287 590	1 258 961 000 Азия
7	Казахстан	Астана	2 724 900	17 753 200 Азия
8	Киргизстан	Бишкек	198 500	6 015 500 Азия
9	Китай	Пекин	9 640 821	1 410 545 741 Азия
13	Таджикистан	Душанбе		
14	Туркменистан	Ашхабад		
15	Узбекистан	Ташкент		
17	Япония	Токио		

Рис. 3. Результат фильтрации континента Азия

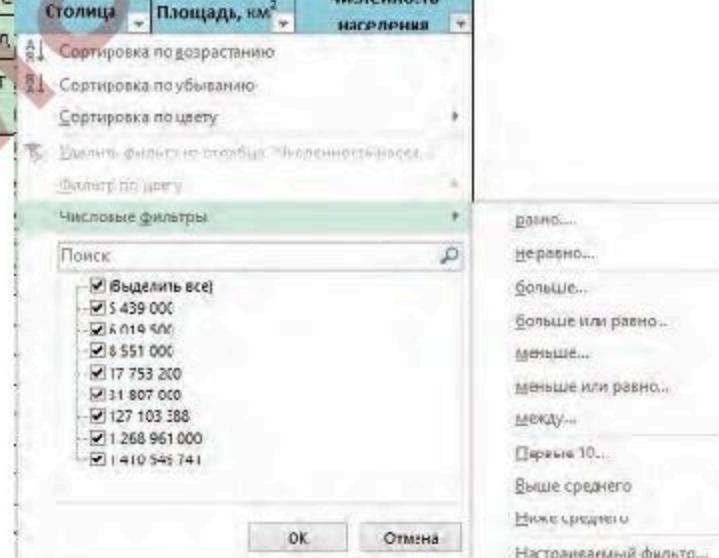


Рис. 4. Работа с фильтром чисел

выборку по следующим условиям – равное данному числу, неравное, больше, между, первые 10, выше среднего. В примере, который мы рассматриваем, какое условие нужно составить, если нам нужны страны численностью населения более 100 миллионов человек?

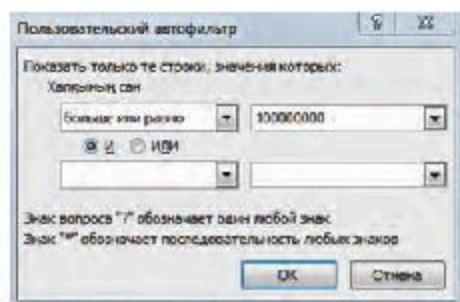


Рис. 5. Окно условий числового фильтра

Если вы выберете определенное Поле, выполните команду Числовые фильтры, то появится диалоговое окно (рис. 5). Составив условие, равное или превышающее 100 000 000, вы можете получить список стран с населением более 100 миллионов с использованием фильтра. Вы можете отфильтровать по одному или двум критериям списки в диалоговом окне. Для этого вы должны выбрать оператор

сравнения и следующее значение. Чтобы выполнить второе условие, вы устанавливаете переключатель в «И» или в «ИЛИ» и выбираете оператор сравнения и текстовое сравнение.

Чтобы отменить фильтр, нужно повторно выделить диапазон фильтрованных данных и нажать кнопку Фильтр.

	Название стран	Столица	Площадь км <sup>2</sup>	Численность населения	Материк
3	Бразилия	Бразилия	8 511 965	205 081 132	Южная Америка
5	Индия	Нью Дели	3 287 590	1 263 961 000	Азия
7	Китай	Пекин	9 640 821	1 410 549 741	Азия
8	Нигерия	Абуджа	923 768	185 983 000	Африка
9	Россия	Москва	17 102 315	145 541 700	Европа
10	США	Вашингтон	9 522 057	322 762 016	Северная Америка
11	Япония	Токио	377 944	127 103 386	Азия

Рис. 6. Результат фильтрации числовых данных



### Анализ – Синтез



Создайте свою базу данных по одной из тем предметов – физика, химия, биология. В созданной базе данных выберите данные для фильтрации и сформулируйте условия для фильтрации в группе.



Наберите разработанные базы данных в программе Microsoft Excel. Отфильтруйте по составленным условиям и сохраните полученную базу данных.



## Вопросы

1. В чем функция фильтра? При каких условиях он применяется?
2. В чем разница между сортировкой и фильтрацией?
3. Какие существуют типы фильтраций?
4. Как применить простейшую фильтрацию?
5. Как удалить фильтр?



## Задания

В нижеследующей таблице даны полугодовые доходы (в национальной валюте) магазинов крупных городов.

Задания:

1. Наберите таблицу в электронной таблице.
2. Отформатируйте таблицу.
3. Посчитайте ежемесячный общий и средний доход магазина.
4. Посчитайте полугодовой общий и средний доход магазина по каждому городу.
5. Используя фильтр, представьте данные за март месяц по магазинам с доходами свыше 1 млн тенге.
6. Определите города, где доход магазинов за июнь месяц не превышает 500 тыс. тенге.
7. Расположите, применяя сортировку, средний доход магазинов по результатам шести месяцев от наибольшего дохода к наименьшему.

A	B	C	D	E	F	G	H	
1	№	Города	Январь	Февраль	Март	Апрель	Май	Июнь
2	1	Астана	1150000	1250000	1350000	1450000	1550000	1650000
3	2	Алматы	1000000	1100000	1200000	1300000	1400000	1500000
4	3	Шымкент	1200000	1300000	1400000	1500000	1600000	1700000
5	4	Караганда	1090000	900000	710000	520000	330000	740000
6	5	Актобе	890000	900000	910000	920000	930000	940000
7	6	Актау	780000	980000	1180000	1380000	1580000	1780000
8	7	Атырау	952625	752620	552615	352610	152605	447400
9	8	Павлодар	748558	648558	548558	448558	348558	248558
10	9	Тараз	741100	541100	341100	241100	158900	258900
11	10	Усть-Каменогорск	400000	700000	1000000	900000	800000	700000

**3.6****ФИЛЬТРАЦИЯ ДАННЫХ РАСШИРЕНИМ ФИЛЬТРОМ**

Как выполнить фильтрацию данных расширенным фильтром в электронной таблице Excel?

**Подумай**

- Какие вы можете привести примеры необходимости выбора данных в повседневной жизни по нескольким условиям?
- Как можно выбирать данные из большого объема по нескольким условиям?

**Новые знания**

Мы познакомились с простым способом фильтрации в программе Excel. Теперь познакомимся с расширенной фильтрацией. **Расширенный фильтр** – фильтрация по составным условиям. Возможности **Расширенного фильтра** по сравнению с простым фильтром:

- ❖ При фильтрации данных есть возможность добавить несколько критериев.
- ❖ Отфильтрованные записи сохраняются в заданной таблице или можно переместить их в другую таблицу.
- ❖ **Расширенный фильтр** поможет легко найти данные, в таблицах с большим объемом данных.

Для подключения **Расширенного фильтра** выполняется команда **Данные > Фильтр > Дополнительный**. С помощью этой команды вы можете выполнить фильтрацию данных на местах расположения или отфильтровать записи отдельно на любом листе этой книги.

В диалоговом окне **Расширенный фильтр** (рис. 1) расположены следующие кнопки.

Можно выбрать местоположение отфильтрованных данных во вкладке **Обработка** диалогового окна.

- **Фильтровать список на месте.**
- **Скопировать результат в другое место.**

Во второй части окна есть строка **Исходный диапазон**, в которой выбирается диапазон фильтра.

Ниже расположена строка **Диапазон условий**.

В следующей строке расположена команда **Поместить результат в диапазон**.

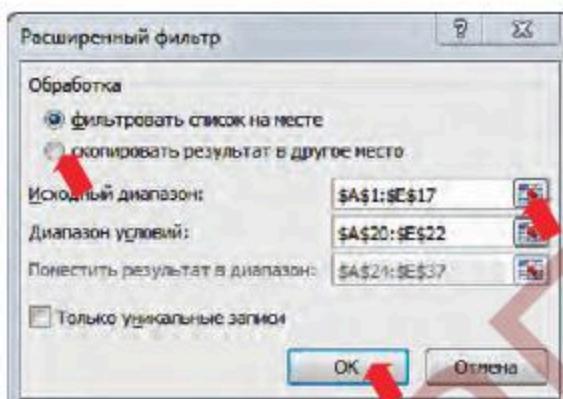


Рис. 1. Фильтрация по полю

Таблица 1. Мировые озера

к/с	Наименование	Территория страны	Площадь, км <sup>2</sup>	Длина, м	Глубина, км <sup>3</sup>	Объем бассейна, км <sup>3</sup>	Материк расположения
	Байкал	Россия	31500	636	1637	23600	Европа
	Большой медведь	Канада	31080	373	446	2236	Сев. Америка
	Онтарио	Канада – США	19477	311	244	1639	Сев. Америка
	Балхаш	Казахстан	18428	605	26	106	Азия
	Титикака	Боливия – Перу	8135	177	281	893	Южн. Америка
	Никарагуа	Никарагуа	8001	177	26	70	Южн. Америка
	Туркана	Кения – Эфиопия	6405	248	109	204	Африка
	Иссык-Куль	Кыргызстан	6200	182	668	1738	Азия
	Венери	Швеция					Европа
	Ханка	КНР – Россия					Азия

Рассмотрим некоторые практические примеры применения расширенного фильтра.



### Практическая работа

Давайте посмотрим базу некоторых данных озер всего мира (таблица 1 «Мировые озера»), чтобы научиться обрабатывать данные, выполнив расширенную фильтрацию в Excel, опираясь на полученные знания по предмету география.

**Задание №1.** Используйте расширенную фильтрацию для выбора озер, расположенных только на азиатском континенте. Расположите выбранные озера в другой таблице. Для выполнения данного задания следуйте следующему алгоритму.

1. Введите данные из таблицы 1 в программу Excel и отформатируйте их в соответствии с вашими предпочтениями.

Таблица 1

A	B	C	D	E	F	G	H
	Название	Страна расположения	Площадь, км <sup>2</sup>	Длина, км	Глубина, м	Объем воды, км <sup>3</sup>	Название материка
1	Байкал	Россия	31500	636	1637	23600	Европа
2	Большое Медвежье	Канада	31080	373	446	2236	Сев. Америка
3	Онтарий	Канада, США	19477	311	244	1639	Сев. Америка
4	Балхаш	Казахстан	18428	605	26	106	Азия
5	Титикака	Боливия - Перу	8135	177	281	893	Южн. Америка
6	Ниагарский	Никарагуа	8001	177	26	70	Южн. Америка
7	Туркана	Кения-Эфиопия	6405	248	109	204	Африка
8	Иссyk Куль	Киргизстан	6200	182	668	1738	Азия
9	Бенеци	Панама	5545	140	106	133	Европа
10	Ханка	КНР- Россия	4070	95	10,6	18,3	Азия

Таблица 2

A	B	C	D	E	F	G	H
	Название	Страна расположения	Площадь, км <sup>2</sup>	Длина, км	Глубина, м	Объем воды, км <sup>3</sup>	Название материка
13							Азия
14							

A	B	C	D	E	F	G	H
	Название	Страна расположения	Площадь, км <sup>2</sup>	Длина, км	Глубина, м	Объем воды, км <sup>3</sup>	Название материка
15							
16	Балхаш	Казахстан	18428	605	26	106	Азия
17	Иссyk Куль	Киргизстан	6200	182	668	1738	Азия
18	Ханка	КНР- Россия	4070	95	10,6	18,3	Азия

Рис. 2. Список озер азиатского континента, полученный путем применения расширенного фильтра

2. Для выбора озер, расположенных только на азиатском континенте, мы копируем и помещаем название таблицы и наименование выбранного континента в строку ниже основной таблицы (рис. 2, таблица 2).
3. Мышкой щелкаем наименование «Азия» в поле Название материка таблицы 1 (рис. 2).
4. При выполнении команд Данные – Фильтр – Дополнительный, появится диалоговое окно расширенного фильтра, как показано на рис. 1. Теперь достаточно показать соответствующий диапазон данных.
5. Устанавливаем метку Скопировать результат в другое место. Затем помещаем данные, полученные фильтрацией, в новую таблицу.
6. Устанавливаем отфильтрованные данные (рис. 2) из 1-й таблицы в диапазон данных, помещая его в соответствии со строкой Исходный диапазон.
7. Мы установим соответствие, показанное на рис. 2 фильтрованного поля с заголовком в таблице 2.
8. Результат располагается в выбранный диапазон командой «Поместить результат в диапазон».

**Задание №2.** Выберите из таблицы 1, в которой перечислены озера, расположенные в базе данных «Мировые озера», те из них, которые

длиной более 200 километров и глубиной свыше 300 метров. Расположите их в другой таблице. Для выполнения этого задания следуйте следующему алгоритму:

- Установите условия **Длина > 200 км** и **Глубина > 300 м** таблицы 2 рисунка 3, а затем выполните алгоритм по Заданию 1.  
Как вы можете видеть, условия помещаются непосредственно в соответствующее поле. Например,  $> =$ ,  $= <$ ,  $\text{<} \text{>}$ , и т.д.
- Получите таблицу 3, представленную на рисунке 3, отражающую фильтрацию по нескольким критериям.

Таблица 1

	A	B	C	D	E	F	G
	Название	Страна расположения	Площадь, км <sup>2</sup>	Длина, км	Глубина, м	Объем воды, км <sup>3</sup>	Название материки
1	Иссик-Куль	Киргизстан	6200	182	668	1738	Азия
2	Бечери	Швейцария	5515	110	106	153	Европа
3	Ханка	КНР-Россия	4070	95	10,6	18,5	Азия
4	Байкал	Красноярск	18428	605	26	106	Азия
5	Титикака	Боливия-Перу	8135	177	281	893	Южная Америка
6	Монголугу	Монголия	8001	177	26	70	Южная Америка
7	Туркана	Кения-Эфиопия	6405	248	109	204	Африка
8	Байсан	Россия	31500	636	1637	23600	Европа
9	Большое Медненское	Канада	31080	373	416	2236	Северная Америка
10	Онтарио	Канада-США	19177	311	244	1639	Северная Америка
11							
12							

Таблица 2

	A	B	C	D	E	F	G
	Название	Страна расположения	Площадь, км <sup>2</sup>	Длина, км	Глубина, м	Объем воды, км <sup>3</sup>	Название материки
14							
15				>200	>300		

Таблица 3

	A	B	C	D	E	F	G
	Название	Страна расположения	Площадь, км <sup>2</sup>	Длина, км	Глубина, м	Объем воды, км <sup>3</sup>	Название материки
16							
17	Байсан	Россия	31500	636	1637	23600	Европа
18	Большое Медненское	Канада	31080	373	416	2236	Северная Америка

Рис. 3. Базы данных с расширенным фильтром множественных условий



### Оценка



Оцените значение функций сортировка и фильтрация, сравнивая их при обработке базы данных. Каковы преимущества использования этих функций при обработке данных? Собрав аргументы, проанализируйте в группе.



### Вопросы

- Какова функция расширенного фильтра?
- В каких случаях применяется такой фильтр?
- Какие отличия существуют между простым фильтром и расширенным?
- Как выполняется расширенный фильтр?
- Каковы параметры расширенного фильтра?

## Химические элементы

В таблице 2 приведен список химических элементов, которые на слуху и часто используются в быту. Значения химических элементов даны с точностью до 0,1.



### Задания

Таблица 2

	Наименование	Атомная масса	Плотность, г/л	Температура плавления, С	Год открытия
1	Водород	1,0	0,09	259,1	1766
2	Литий	6,9	0,53	180,5	1817
3	Магний	24,3	1,74	648,8	1808
4	Алюминий	27,0	2,70	660,5	1825
5	Фосфор	31,0	1,82	44,0	1669
6	Уран	238,0	18,97	1132,4	1789
7	Никель	58,7	8,91	1453,0	1751
8	Платина	195,1	21,45	1772,0	1557
9	Азот	14,0	1,17	-209,9	1772
10	Бор	10,8	2,46	2300,0	1808
11	Кислород	16,0	1,33	-218,4	1774

**Задание №1.** Постройте базу данных списка химических элементов. Отформатируйте таблицу.

**Задание №2.** Определите элемент с наибольшей атомной массой, используя функцию MAX. Определите элемент с самой минимальной температурой плавления, используя функцию MIN.

**Задание №3.** Отсортируйте химические элементы по возрастанию атомной массы, используя функцию Сортировка.

**Задание №4.** Используя простой фильтр, определите химические элементы, открытые в период с 1800 до 1900 года.

**Задание №5.** Найдите химические элементы в полях по условиям, указанным ниже, используя расширенный фильтр.

	Наименование	Атомная масса	Плотность, г/л	Температура плавления, С	Год открытия
		>30	>1	>300	

## Раздел III. База данных

3.7

– 3.8

## РАБОТА С БАЗОЙ ДАННЫХ (МИНИ-ПРОЕКТ)



Как можно использовать в повседневной жизни знания, полученные при создании базы данных и ее редактировании в Excel?



## Подумай

- Как можно использовать в повседневной жизни (дома, в школе и т.д.) знания, полученные при работе с базами данных? Приведите соответствующее обоснование, доказательные примеры.



## Новые знания

**Проект** – это совокупность действий, при проведении исследования на определенную тему, которая в результате приводит к созданию реального объекта, предмета, разного рода теоретического продукта. Результат проекта представляется перед аудиторией, т.е. осуществляется его защита. Планирование является неотъемлемой частью подготовки проекта по теме. Можно подготовить также видеопроекты (ролики) с использованием цифровых устройств. Проекты могут быть в таких видах, как практический проект, исследовательский проект, информационный проект, творческий проект. Для нас представляют интерес практические проекты при работе в электронной таблице Excel. Причина в том, что с помощью этого программного обеспечения вы можете получить реальный результат, а также подготовить новый программный продукт.

**Практический проект** направлен на решение конкретных задач. Например, конкретным результатом практического проекта «Помощь птицам» является создание гнезд или кормушек.



## Образцовые темы для проектов

Вы можете выбрать одну из предложенных тем проектов и выполнить проект индивидуально или в паре. При выполнении проекта, кроме представления теоретического материала, отдельно создайте соответствующий программный продукт.

## Мини-проект «Наш класс»

Составьте базу данных 9-го класса, в котором вы учитесь, в электронной таблице Excel. Используйте диалоговое окно Форма при

К/с	ФИО ученика	Год рождения	Рост	Вес	Класс	Успеваемость		Хобби	Вид спорта
						отличника	отличательные		
1	Алдаутас Аслан	18.01.2018	168	60	9	...	...	...	...
2	Барыким Айдарек	...	...	...	...	...	...	...	...
3	...	...	...	...	...	...	...	...	...
4	...	...	...	...	...	...	...	...	...
5	...	...	...	...	...	...	...	...	...
6	...	...	...	...	...	...	...	...	...
7	...	...	...	...	...	...	...	...	...
8	...	...	...	...	...	...	...	...	...
9	...	...	...	...	...	...	...	...	...
10	...	...	...	...	...	...	...	...	...
11	...	...	...	...	...	...	...	...	...
12	...	...	...	...	...	...	...	...	...
13	...	...	...	...	...	...	...	...	...
14	...	...	...	...	...	...	...	...	...
15	...	...	...	...	...	...	...	...	...
16	...	...	...	...	...	...	...	...	...
17	...	...	...	...	...	...	...	...	...
18	...	...	...	...	...	...	...	...	...
19	...	...	...	...	...	...	...	...	...

Рис. 1. Пример продукта мини-проекта «Наш класс»

вводе в базу данных сведений об учащихся своего класса. Применяя возможности функции Среднее, рассчитайте средние значения роста и веса учащихся класса, используя данные полей Рост и Вес (рис. 1).

### Проект «Автотранспорт»

Группа компаний «Бипэк Авто – Азия Авто» является одной из крупнейших компаний в стране, занимающихся сбором и продажей автомобилей. На сайте компании [bipek.kz](#) вы можете найти описание и цены продажи автомобилей в стране и за рубежом на сегодняшний день. Создайте базу данных об автомобилях, используя информацию с этого сайта. Ниже даны технические характеристики и среднегодовые цены продаж на автомобили (рис. 2).

#### Задание:

- Составьте базу данных по сведениям из рисунка.
- В базе данных в полях отразите наименования автомобилей, объем цилиндра, мощность, расход топлива на каждые 100 км и среднегодовые цены соответствующего автотранспорта.
- С помощью расширенного фильтра выделите автотранспорт с показателями по мощности, равной 2.0 и 100 лошадиных сил и выше, а также с количеством проданных авто свыше 2000.
- Начертите диаграмму по количеству проданного автотранспорта.

### Проект «Расчет успеваемости класса» (в помощь классному руководителю)

У классного руководителя всегда очень большой объем воспитательной работы. Одной из обязанностей классного руководителя



«Chevrolet» – 980 штук  
Объем цилиндра – 3,0 л  
Мощность – 280  
Расход – 100 км/11 л



«KIA Sportage» – 2487 штук  
Объем цилиндра – 2 л  
Мощность – 155  
Расход – 100 км/10 л



LADA – 3800 штук  
Объем цилиндра – 1,6 л  
Мощность – 98  
Расход – 100 км/ 7 л



Renault – 1840 штук  
Объем цилиндра – 1,6 л  
Мощность – 105  
Расход – 100 км/7 л



«UAZ-ПРОФИ» – 1020 штук  
Объем цилиндра – 2,6 л  
Мощность – 110  
Расход – 100 км/10 л



«Skoda» – 1200 штук  
Объем цилиндра – 1,6 л  
Мощность – 110  
Расход – 100 км/6.5 л

Рис. 2. Характеристики автотранспорта

является расчет успеваемости учащихся класса по итогам каждой четверти. Такого рода расчеты классные руководители до сих пор осуществляют на бумаге с помощью калькулятора или в текстовом редакторе MS Word. Разумеется, это занимает много времени.

**Задание.** Выполните работу по созданию практического проекта по расчету успеваемости класса. В проекте должны быть учтены следующие параметры:

1. При создании базы данных проекта используйте кнопку **Форма**. Первая строка базы должна содержать поля **Фамилия, Имя, Названия предметов**, изученных в данной четверти. Записи должны содержать данные о конкретном ученике: фамилия, имя, оценки по каждому предмету.
2. Дизайн проекта должен быть привлекательным, данные – удобными для чтения, важные моменты выделены цветом.
3. В последней строке должны быть отражены средние баллы по предметам и успеваемость в процентах, а в последнем столбце – средний балл каждого учащегося в четверти и успеваемость в процентах.
4. С помощью сортировки расставить учащихся по уменьшению или увеличению процента успеваемости.
5. С помощью простого фильтра выделить отлично успеваемых по каждому предмету.
6. С помощью расширенного фильтра выделить учащихся только с оценками «4» и «5» либо по естественно-математическим предметам, либо по гуманитарным, а также список учащихся по всем предметам, успевающих на «отлично».

A	B	C	D	E	F	G	H	I
Сводная ведомость успеваемости учащихся 9 класса								
№/с	ФИО ученика	Казахский язык	Алгебра	...	...	Рейтинг	Средняя оценка	Качество знаний %
1	Алжуман Айдан	5	5	...	...	...	...	...
2	Наражан Айдан	4	5	...	...	...	...	...
	Средняя оценка	...	...	...	...	...	...	...
7	Качество знаний	...	...	...	...	...	...	...

**Обратите внимание на последовательность защиты проекта!**

- Ознакомление с выбранной темой.
- Объяснение причины выбора той или иной темы.
- Определите свои цели и задачи.
- Перечисление использованных устройств.
- Выполнение краткой аннотации по теме.
- Поделиться мнением о пользе данного проекта.
- Расскажите о новых знаниях и умениях, приобретенных в процессе создания проекта.
- Ознакомление с выводами по проекту.
- Выслушайте мнение и оценки слушателей, сумейте их воспринять правильно.

**Чтобы проект был успешным, необходимо:**

- Индивидуальное выполнение проекта.
- Актуальность темы.
- Тема должна быть раскрыта оригинально и творчески.
- Способ подачи рассматриваемой проблемы.
- Умение проявить артистичность при защите проекта.
- Умение использовать технические устройства и приспособления.
- Правильно и аргументировано отвечать на поставленные вопросы.

**В конце проекта напишите отчет!**

- Почему я выбрал эту тему?
- Что узнал нового, чему научился?
- Как нашел нужную информацию?
- Как обработал информацию?
- Как совместил теорию с практикой?
- Что получилось рациональным?
- Что не смог сделать?
- Что удалось легко?
- Что вызвало затруднения?
- Если бы пришлось создать проект еще раз, что бы изменил?
- Мое впечатление от реализации проекта.

## 4.1 ОДНОМЕРНЫЙ МАССИВ



Как обработать большое количество однотипных данных на языке программирования Python?



### Подумай

- Как вы понимаете понятие массив?
- Какие вы можете привести примеры из повседневной жизни работы с однотипными данными больших объемов?



### Новые знания

#### Массив

**Массив** (с франц. *massif* – мощный, сплошной) – совокупность множества однородных предметов или объектов, составляющих единое целое. Например, большое, однородное по некоторому признаку пространство (лесной массив, жилой массив). **Массив** – последовательность идентичных данных, элементы которой разными индексами объединены в общее имя.



Рис 1. Жилой массив

В математике массив (набор) рассматривается как последовательность чисел. В программировании массивом является последовательность чисел, символов, строк. Например, последовательность четных чисел – **2, 4, 6, ...**. В математике и в программировании массивы выражаются формулами типа  $a_1, a_2, \dots, a_{n-1}, a_n$ . В программировании такая последовательность называется **одномерным массивом**. Роль массива велика при обработке большого количества однотипных данных. Если этот тип данных используется для обозначения последовательности чисел, то все разряды будут сгруппированы под одним именем (часто одной буквой). Вместе с тем составляющие массива называются элементами массива, а места элементов в массиве – индексами элемента. Массив широко используется при обработке однотипных данных.

очень большого объема на компьютере. Например, номера абонентов городской телефонной сети, полученные учащимися на экзамене и т.п.

#### Виды массивов

- одномерный;
- двумерный;
- многомерный

#### Типы массивов

- числовой;
- текстовый;
- символьный;
- логический

#### Отличительная особенность массивов

- все элементы одного типа;
- элементы под одним именем;
- элементы расположены в памяти последовательно.

#### Одномерные массивы

Массив имеет такие параметры, как имя, члены (элемент), количество элементов (длина массива) и полученное значение. Например, в приведенной ниже схеме 1 имя массива (списка) равно A, а длина массива равна  $n = 7$ .



Схема 1.

В большинстве языков программирования данные одного и того же типа обрабатываются массивом. Существует массив и в языке программирования **Python**. В **Python**, как и в других языках программирования, элементы массива должны быть одного типа, а для списков такого ограничения нет. Элемент списка может быть числом, символом или строкой одновременно. В отличие от других языков программирования индексы массивов нумеруются от 0, а не с 1. При изучении массивов в **Python** мы будем рассматривать понятия **массив** и **список** в одном и том же смысле.

## Одномерный массив



Например, в приведенном программном коде массив `A` состоит из 5 элементов. Чтобы вывести массив на экран, достаточно нажать кнопку «Выполнить» (код 1) и указать название массива `A`.

Если вы зададите массив `A` как `A[X:Y:Z]`, то он отобразит массив элементов от `X` до `Y` с шагом `Z` (код 2). Если шаг равен `Z = 1`, то его необязательно указывать, массив от `X` до `Y` выведется на экран с шагом 1 по умолчанию.

Элементы массива в Python можно считывать с конца. Тогда нумерация индекса начинается с `-1`. Для заданного массива `A` имеем `-1, -2, -3, -4, -5` соответственно. То есть `A[0] = A[-5]`. Оба указывают на один и тот же элемент (код 2).

Можно вывести элементы массива от начала или с конца до определенной позиции (код 3).

Для вывода массива от начала до заданной позиции надо указать `A[:K]`, где `K` – индекс позиции.

Чтобы вывести массив с определенной позиции до конца, необходимо указать `A[K:]`, где `K` – индекс позиции (код 3).

### Код 1

```
>>>A = [45, 25, 30, 20,
-40]
>>>A
[45, 25, 30, 20, -40]
```

### Код 2

```
>>>A = [45, 25, 30, 20,
-40]
>>>A[0:5:2]
[45, 30, -40]
>>>A[2:4]
[25, 30, 20]
>>>A[-1], A[-5]
[-40], [45]
>>>A[0], A[-5]
[45], [45]
```

### Код 3

```
A=[45, 25, 30, 20, -40]
A[:3]
[45, 25, 30]
A=[45, 25, 30, 20, -40]
A[3:]
[20, -40]
```

## Ввод элементов одномерного массива

Для ввода элементов одномерного массива необходимо произвести импорт библиотеки, отвечающей за работу с массивом. Для этого потребуется добавить строку `from array import array` в файл программы.

`N = int(input())` – вводит длину массива.

Данные, введенные оператором `input()`, будут строковыми. Мы используем `int` для преобразования их в числовое значение.

`A = [0]*N` резервирует в памяти компьютера место длиной `N` для размещения элементов массива. Например, в случае  $N = 10$ ,  $A = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$

`for i in range(0,N):` Значение переменной цикла `i` изменяется от 0 до 9.

`A[i] = int(input())` позволяет ввести элементы массива по одному с клавиатуры.

```
from array import array
A=array("i")
N=int(input())
A=[0]*N
for i in range(0,N):
    A[i]= int(input())
print(A[4:])
```



### Практическая работа

**Сумма.** Задан массив  $A[N]$ . Найти сумму элементов заданного массива.

Например	Результат
5	
1	
-2	
6	
-9	
10	17

```
from array import array
A=array("i")
N=int(input())
S=0
A=[0]*N
for i in range(0,N):
    A[i]= int(input())
for i in range(0,N):
    if A[i]> 0:
        S+=A[i]
print(S)
```



### Анализ



Сделайте анализ функций оператора цикла при вводе элементов массива. Как вводятся элементы массива с помощью оператора цикла? Объясните.

**Синтез**

Предложите и обоснуйте два массива однотипных данных, известных вам из повседневной жизни. Например, определение веса 20 учащихся класса.

**Оценка**

Оцените значение массива в математике и программировании и поделитесь мнением.

**Вопросы**

1. Что такое массив?
2. Что такое одномерный массив?
3. Каковы различия между массивом и списком в Python?
4. Как в Python ввести массив?

**Задания****Решение задач на применение одномерного массива**

1. Дан одномерный массив. Найдите количество положительных элементов массива.
2. Дан одномерный массив. Выведите на экран элементы массива, делящиеся на число Р без остатка.
3. Дан одномерный массив. В этом массиве имеется только один элемент, равный нулю. Выведите на экран все элементы до указанного элемента.
4. Данна последовательность чисел  $A_1, A_2, \dots, A_n$ . Выведите на печать числа из диапазона  $[c; d]$  данной последовательности.
5. Дан одномерный массив целых чисел. Замените в этом массиве числа больше Х числами М.
6. Дан одномерный массив. Выведите на печать удвоенные значения элементов массива меньшие, чем К.

## Раздел IV. Программирование алгоритмов на языке программирования Python

4.2

## ВВОД И ВЫВОД ДАННЫХ В ОДНОМЕРНОМ МАССИВЕ



Как можно организовать ввод и вывод однотипных данных на языке программирования Python?



**Подумай**

- Какие способы ввода данных в компьютер вам известны?



**Новые знания**

В зависимости от поставленной задачи имеется три способа ввода данных в массив (*схема 1*). Если количество данных, вводимых в массив, не превышает 20-30, то эффективен ввод с клавиатуры. Для обработки больших объемов данных с использованием массива удобнее их считывать из файла, в который они предварительно записаны. В большинстве случаев необходимо протестировать производительность нового программного обеспечения с максимальными и минимальными пороговыми значениями. В этом случае выгодно вводить данные с помощью оператора случайных чисел. Например, если вам нужно проверить работоспособность программы с 2 миллионами различных целых чисел, то ввод этих данных займет много времени.

С помощью оператора случайных чисел **Range** эта база данных может быть создана за 1-2 секунды.

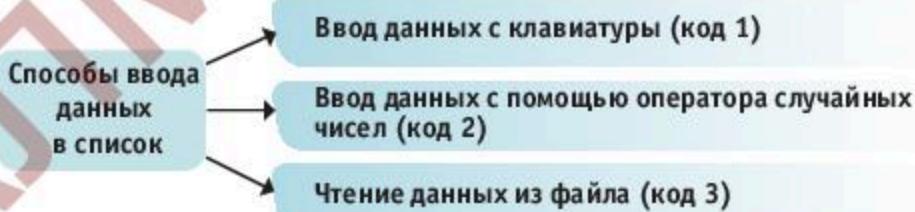


Схема 1

Генерация любых случайных чисел с помощью модуля **Random**. Функция **Randint** модуля **Random** возвращает случайные числа в заданном интервале. Например, от 1 до 100 или от 300 до 770.

```

from random import randint (подключает модуль random в
программу)
N = 10
A =[0]*N
for i in range(N):
    A[i] = randint(1, 100) (выбирает случайное число в промежутке от
1 до 100)
print(A)
результат: [69, 34, 50, 95, 57, 52, 43, 57, 46, 27]
Вместе с тем есть следующие способы заполнения таблицы:
A = [ i for i in range (10) ] результат: A = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
A = [ i*i for i in range (10) ] результат: A = [0, 1, 4, 9, 16, 25, 36, 49,
64, 81]
A = [ randint (1,10) for x in range (10)] результат: A = [5, 1, 3, 4, 9,
8, 1, 7, 1, 2]

```

Давайте рассмотрим ввод данных в массив всеми тремя способами. Оператор, необходимый для ввода данных в массив, – это оператор цикла. В Python вы можете вводить и выводить массивы, обрабатывать данные, используя 2 типа циклов (*for, while*), с которыми вы ознакомились в 8-м классе. Теперь давайте рассмотрим примеры того, как вводить, редактировать данные и выводить результат на экран.

Задан список A[N], длина которого равна N (N<10<sup>2</sup>). Выведите на экран все четные числа из этого списка.

#### Код 1

```

N=10 (длина массива равна 10)
A=[0]*N (выделяет место для 10 элементов массива )
for i in range(0,N):(значение i изменяется от 0 до 9 )
    A[i]= int( input() ) (ввод элементов массива с клавиатуры)
for i in range(0,N):
    if A[i] % 2==0 : (определение четных элементов массива)
        print ('A[',i,']=',A[i])

```

#### Код 2

```

from random import randint (подключает модуль случайных
чисел)
N=10
A =[0]*N
for i in range(0,N):
    A[i] = randint(1,100)(загружает в массив случайные числа из
промежутка от 1 до 100)
for i in range(0,N):
    if A[i] % 2==0 :
        print ('A[',i,']=',A[i])

```

Хотя данные вводятся тремя различными способами, результат выполнения 3-х кодов программ одинаков. Введя с клавиатуры:

```

A[1] = 8;
A[2] = 58;
A[3] = 83;
A[4] = 17;
A[5] = 2;

```

И выполнив команду RUN, мы получим следующие результаты:

```

A[1] = 8;
A[2] = 58;
A[5] = 2;

```

**Код 3**

```

f = open('input.txt',"r")(открывает файл для чтения)
N=int(f.readline())(читает значение длины списка из файла)
h = open('output.txt',"w")(создает и открывает файл для записи)
S=0
A=[0]*N
t=f.readline().split()(читает элементы, записанные в строке, как строку)
for i in range(0,N):
    A[i] = int(t[i])(удаляя из строки t пробелы, загружает список целых чисел)
for i in range(0,N):
    if A[i] % 2==0:
        h.write(str(A[i])+"\\n")(записывает четные числа в виде строки)
h.close()(закрывает файл, если файл не закрыт, то результат не сохранится в файле)

```

**Практическая работа****Успеваемость**

В школе N параллельных 9-х классов. Итоги 3-й четверти этих параллельных классов даны в виде массива A[N], элементы которого отражают процент успеваемости каждого класса ( $N < 102$ ). Необходимо вывести на экран среднюю успеваемость всех классов, а также самую низкую среди них. Создайте проект, в котором должно быть рассчитано, на сколько процентов нужно повысить успеваемость, чтобы достичь среднего показателя для класса с наименьшей успеваемостью. Дробные числа должны быть округлены с точностью до 0,01.

Ввод	Результаты
Количество классов 8	Средняя успеваемость = 67.375% Самое низкое качество знаний = 40% Процент повышения = 25.375%
Успеваемость класса (%) 80	
90	
40	
65	
70	
68	
60	
50	

**Алгоритм выполнения задания:**

Успеваемость учащихся считываем как элемент массива  $A_i$ . Среднее арифметическое массива рассчитываем по формуле:

$$S = \frac{A_1 + A_2 + \dots + A_{n-1} + A_n}{n}$$

Наряду с этим определяем минимальный элемент массива **min** и вычисляем процент повышения (**S - min**).

**end=""** – не дает перевести курсор на новую строку.

```
print (Количество классов =', end="")
N = int( input() )
S = 0
A= [0]*N
print (Введите успеваемость классов =')
for i in range(0,N):
    A[i] = int( input() )
min =A[0]
for i in range(0,N):
    S+=A[i]
if A[i]<min:
    min=A[i]
S=S/N
print (Средняя успеваемость=', S,'%')
print (Самое низкое качество знаний =',min)
print (Процент повышения=', S-min)
```



### Анализ



Сделайте сравнительный анализ трех способов ввода массива. Какой из этих трех способов может наиболее часто применяться? Объясните причину соответствующими примерами.



### Задания

#### Решение задач на использование одномерных массивов

- Считав номера телефонов пяти одноклассников из файла **telefon.txt**, запишите в файл **telout.txt** те из них, которые заканчиваются на цифру 5.
- Дан массив **B[20]**, состоящий из случайных чисел из промежутка от 20 до 50. Выведите на экран четные элементы, расположенные на нечетных местах этого массива.
- Нужно выбрать членов школьной баскетбольной команды из списка учеников. Дан рост всех **K** (**K>13**), желающих попасть в команду. В команду отбираются только 12 учеников с ростом выше, чем **P**. Если не набирается количество учащихся, соответствующее вышеуказанному условию, то результатом будет «**Набор команды невозможен**». Если же выполняется условие набора команды, то должны быть показаны порядковые места баскетболистов в списке учащихся. Вычислите средний рост набранной команды, самого высокого и самого низкого члена команды и выведите результат.

## Раздел IV. Программирование алгоритмов на языке программирования Python

4.3

## ПОИСК ЭЛЕМЕНТА С ЗАДАННЫМИ СВОЙСТВАМИ



Как можно среди однотипных данных найти элемент с заданными свойствами на языке программирования Python?



**Подумай**

- Как вы осуществляете поиск необходимого предмета среди большого количества предметов в повседневной жизни?



**Новые знания**

В процессе обработки информации необходимо постоянно осуществлять поиск данных с определенными свойствами из большого их количества. Например, телефон друга, адрес нужного человека, правильный ответ на тест и т. д. При работе с одномерными массивами наиболее распространенным типом задач является поиск данных.

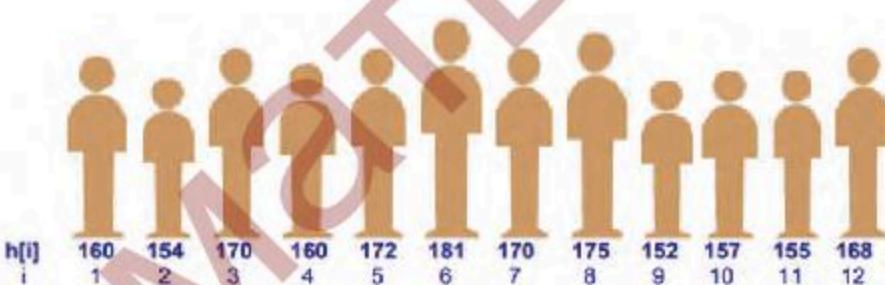


Рис. 1. Произвольное построение учащихся

Рассмотрим следующий пример: дан массив  $X[12]$ . Элементами массива будет рост учащихся 9 класса. В этом массиве мы можем осуществить поиски разного вида. Например:

- Есть ли среди учащихся хотя бы один, у которого рост равен 170 см?
- У скольких учащихся рост превышает среднее значение?
- Чей рост превышает 160 см? Выведите на экран соответствующие индексы учащихся.
- Определите самого высокого учащегося класса.

- Определите самого низкого учащегося класса.
- Какова разница в росте между самым высоким и самым низким учащимся в классе?

Как видно из этого простого примера, вы можете осуществлять поиск среди элементов массива по широкому спектру параметров.

В программировании существует много различных способов (алгоритмов) поиска элементов массива по какому-либо параметру.

Самый простой вид поиска среди них – это нахождение элемента простым перебором по порядку всех элементов массива. Такой алгоритм называют **линейным или последовательным**.

**Пример 1.** Дан рост учеников 9-го класса в виде массива X [12]. Найдется ли хотя бы один ученик с ростом, равным 170 см? Если найдется такой ученик, определите его место (индекс) в списке. Если найдется несколько таких учеников, то выведите на экран порядковый номер (место) последнего из них.

Этот простой пример выполним с помощью линейного алгоритма. Сравниваем каждый элемент массива с цифрой 170. Место элемента массива, удовлетворяющего условию, сохраним как переменную P. Если значение P будет равно нулю, то выведем сообщение «**В массиве такого элемента нет**» или «**Нет учащегося ростом ниже 170 см**».

#### Код программы для примера 1

```
from array import array
X=array("i")
N=int( input() )
S=0
p=0
X=[0]*N
for i in range(0,N):
    X[i] = int( input() )
for i in range(0,N):
    if X[i]==170:
        p=i
if p!=0:
    print(p)
else: print('Нет учащегося ростом
ниже 170 см')
```

#### Код программы для примера 2

```
from array import array
X=array("i")
N=int( input() )
X=[0]*N
for i in range(0,N):
    X[i] = int( input() )
min=X[0]
max=X[0]
for i in range(0,N):
    if X[i]< min:
        min=X[i]
    if X[i]> max:
        max=X[i]
print(max-min)
```

**Пример 2.** Дан рост учеников 9-го класса в виде массива X [12]. Определите разницу в росте между самым высоким учеником и самым низким.

Таких примеров можно привести множество. Например, определить количество учеников, сдавших экзамены на «отлично» из экзаменационной ведомости, найти определенное слово из текстового массива и т.д.

Рассмотрим еще один пример.

**Пример 3.** Дан одномерный массив В[N], (0<N<100). Определите сумму всех нечетных чисел, расположенных на четных местах массива.

При решении этой задачи мы должны найти элементы массива, которые должны быть нечетными и располагаться на четных местах списка.

#### Код программы для примера 3

```
from array import array
x = array("i")
N = int( input() )
S = 0
X = [0]*N
for i in range(0,N):
    X[i] = int( input() )
for i in range(0,N):
    if X[i] % 2 ==0 and i % 2 == 1:
        S+=X[i]
```

Среди условий поиска в одномерном массиве есть те, которые используются чаще, чем другие. Например, для массива A

можно выделить следующие условия:  
 $A[i] > 0$  нахождение положительных чисел;  
 $A[i] < 0$  нахождение отрицательных чисел;  
 $A[i] == 0$  нахождение чисел, не равных нулю;  
 $A[i] != 0$  нахождение чисел, равных нулю;  
 $A[i] \% 2 = 0$  нахождение четных цифр;  
 $A[i] \% 2 = 1$  нахождение нечетных цифр;  
 $A[i] \% 10 = 0$  отбор последней цифры;  
 $A[i] > \max, A[i] < \min$  нахождение чисел, удовлетворяющих заданному условию;  
 $A[i-1] < A[i] \text{ and } A[i] > A[i+1]$  нахождение среднего элемента (промежутка).



#### Практическая работа

#### Разработка проекта поиска элементов массива на языке программирования Python

На языке программирования Python имеется возможность для создания множества интересных и полезных проектов, отражающих поиск по определенным условиям в одномерном массиве. Ознакомимся с выполнением таких проектов.

**Проект «Погода»**

Однодневная температура в крупных городах нашей страны показана на карте (рис. 2). Используя эти данные, сделайте анализ изменения погоды за один день в крупных городах страны.

1. Определите самый холодный город.
2. Определите самый теплый город.
3. Уточните количество городов, где температура ниже нуля.



Рис. 2. Карта крупных городов Казахстана

```
From array import array(подключение(импортирование)библиотеки array)
A= array("i") (описывает целочисленный массив A)
S=[] (подготовка первоначально пустого списка S)
N=int( input() ) (ввод длины массива)
A=[0]*N; S=[0]*N(организация в памяти компьютера N мест для элементов массива)
t=0;z=0; p=0;
print('Введите название городов')
for i in range(0,N):
    S[i] = input() (осуществляет ввод названия городов)
    A[i] = int(input())(ввод температуры воздуха в указанном городе)
    min=A[0]; max=A[0]
    for i in range(0,N):
        if A[i]<min:(нахождение номера самого холодного города)
            min=A[i]; p=i
        if A[i]> max:
            max=A[i]; z=i(нахождение номера самого теплого города)
            if A[i]<0:(нахождение количества городов с температурой ниже нуля градусов)
                t+=1
print('Самый холодный город->', S[p])
print('Самый теплый город-->', S[t])
print('Количество городов с температурой ниже нуля градусов=',z)
```

**Анализ**

Проанализируйте представленный код проекта поиска элементов массива по заданным условиям.

**Синтез**

**Задание.** На рынке по продаже животных выставлены 10 лошадей. Запишите стоимость каждой лошади. Составьте одномерный массив стоимости лошадей. Какие бы вы предложили задания для выполнения с элементами данного массива? Составьте и представьте несколько задач по поиску в одномерном массиве.

**Оценка**

Оцените значимость выполнения поиска элементов массива по заданным условиям. Насколько важно выполнение поиска для исследования и анализа однотипных данных?

**Вопросы**

1. Что такое линейный алгоритм?
2. Какое действие выполняется чаще всего в одномерной таблице?
3. Какие условия поиска осуществляются чаще всего в одномерном массиве?
4. Сравните и определите различия проектов поиска данных в модульной среде и визуальной среде. Какая из них наиболее удобна? Обоснуйте.

**Задания****Задание №1. Кратные числа\***

Дан массив однотипных натуральных чисел  $B[N]$ ,  $N(1 \leq N \leq 100000)$ . Из заданного массива выведите на экран числа, кратные  $P$ .

№	Пример ввода	Результат
1	10 3 11 12 96 32 69 14 15 7 115 207	12 96 69 15 207

**Задание №2. Соседи\*\***

Дан массив однотипных целых чисел  $A[N]$ ,  $N(1 \leq N \leq 1000)$ . Найдите сумму двух соседних чисел  $M$  (справа и слева) заданного массива.

№	Пример ввода	Результат
1	8 6 11 25 3 6 90 8 10 15	100

**Задание №3. Чтение с файла\*\*\***

Задан массив однотипных целых чисел  $A[N]$ ,  $N(1 \leq N \leq 1000)$  в файле **Input.txt**. Найдите сумму элементов массива, расположенных в этом файле. В первой строке файла **Input.txt** задана длина массива. На следующей строке записаны соответствующие значения  $N$  элементов массива. Результат в виде суммы запишите в файл **Output.txt**.

№	Input.txt	Output.txt
1	10 15 -6 3 9 17 6 25 -42 0 1	28

**Задание №4. Поиск**

Дан массив однотипных целых чисел  $C[N]$ ,  $N(1 \leq N \leq 30)$ . Заполните этот массив случайными числами в интервале  $-100 \dots 100$ . Выведите на экран индексы положительных чисел.

№	Input.txt	Output.txt
1	8 -75 60 32 -97 -9 12 -3 97	2 3 6 8

**Домашнее задание**

Составьте одну задачу из часто встречающихся примеров повседневной жизни для разработки проекта поиска элементов массива. Запишите алгоритмы реализации данной задачи.

## 4.4

## РЕШЕНИЕ ЗАДАЧ НА ОДНОМЕРНЫЕ МАССИВЫ.

### ПРАКТИЧЕСКАЯ РАБОТА



Какие задачи можно решать, используя одномерные массивы?



#### Подумай

- Как можно использовать одномерный массив в математике?
- Как обрабатываются физические данные с помощью одномерного массива?



#### Новые знания

В математике часто применяется обработка числовых последовательностей, представленных в виде одномерного массива, с помощью компьютера. Сохранив в памяти компьютера числовые последовательности в виде массива, в любой момент можно обработать и исследовать последовательность с помощью программирования. Особенно удобно использование одномерных массивов при обработке в компьютере результатов физических экспериментов. С помощью компьютера любые последовательности могут обрабатываться в виде числовых, текстовых и других однотипных массивов. Рассмотрим выполнение нескольких заданий из повседневной практики.



#### Практическая работа

##### Практическое задание №1.

###### Исследование числовой последовательности (математика).

Дано числовая последовательность  $a_1, a_2, \dots, a_{n-1}, a_n$ , где ( $n < 10^3$ ) и ( $0 < a_i < 10^3$ ).

Выполните следующие задания по данной последовательности.

а) Определите количество четных чисел в последовательности.

б) Вычислите количество простых чисел в последовательности.

в) Выведите на экран количество чисел, меньших среднего.

№	Пример ввода	Результат
1	9 11 511 17 803 53 538 140 505	Количество четных чисел = 3 Количество простых чисел = 3 Среднее арифметическое значение = 286.4 Количество чисел, меньших среднего арифметического значения: 11 17 53 140

**Ход выполнения задания:** введите математическую последовательность в диапазоне [2...1000] с помощью оператора случайных чисел `random` в одномерный массив.

При исследовании последовательности рассмотрите выполнение в программе нижеследующих условий:

**Определение четных чисел в последовательности:** проверка выполнения условия  $a_i \% 2 = 0$ .

Определение простых чисел в последовательности: определить количество делителей  $a_i$  на числа из промежутка  $2 \dots \sqrt{a_i}$ . Если количество делителей равно 0, то это число является простым.

**Определение чисел в последовательности, меньших среднего арифметического значения:** вывести на экран члены последовательности, удовлетворяющие условию  $\sum \frac{a_i}{n} > a_i$ .

### Код проекта «Исследование чисел»

```
from array import array
from random import randint
import math
A=array("i")
N=int( input())
A=[0]*N; t=0; prost=0;S=0;
chetn=0
A =[randint(2,1000) for x in
range(N)]
#Определение четных чисел
for i in range(0,N):
    if A[i]% 2==0:
        chetn+=1
#Определение простых чисел
for i in range(0,N):
    t=0;
    for j in range(2, round(math.
sqrt(A[i]))+1):
        if A[i]% j==0:
            t=1
    if t== 0:
        prost+=1
```

```
#Определение количества чисел,
меньших среднего арифметического'
for i in range(0,N):
    S+=A[i]
S=S/N
print ('Количество четных
чисел = ',chetn)
print ('Количество простых
чисел = ',prost)
print('Среднее арифметическое = ', S)
print('Меньшие
среднего арифметического')
for i in range(0,N):
    if A[i]<S:
        print(A[i],end = " ")
```

Для использования в программе `sqrt` (квадратный корень) подключаем модуль `Math` (математика).

`A =[ randint (2,1000)for x in range (N)]` этой программной строкой заносим в массив `A` `N` случайных чисел из промежутка от 2 до 1000.



### Практическая работа

#### Практическое задание №2. Исследование зависимости пройденного пути от времени при прямолинейном движении с постоянной скоростью (физический практикум).

В стеклянную трубку, заполненную водой, опускается небольшой полый шарик (рис. 1). Открытый конец трубы закупоривается. Затем трубка переворачивается на 180 градусов. Трубку держат вертикально. Шарик, оказавшийся внизу, всплывает вверх. Засекается время, когда шарик достигнет самой верхней точки. С помощью линейки делают засечки на отметках, равные половине, одной трети, одной четвертой, одной пятой пути. В этих отметках засекается время и заносится в таблицу (Таблица 1).

Таблица 1. Данные практических изменений

№	Части пути	Путь $s, \text{ м}$	Время движения $t, \text{ с}$	Скорость $v, \text{ м/с}$
1	1	0,6	18	0,033
2	$\frac{1}{2}$	0,30	9,2	0,0326
3	$\frac{1}{3}$	0,20	6,1	0,0327
4	$\frac{1}{4}$	0,15	4,45	0,0337
5	$\frac{1}{5}$	0,12	3,68	0,0326

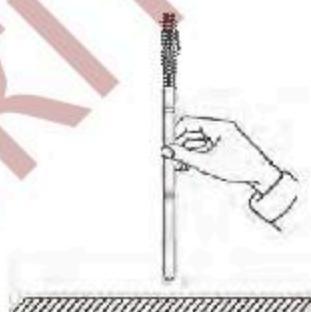


Рис. 1.



#### Анализ



Проанализируйте практические задания использования однотипных данных в математике и физике. Какова эффективность таких использований?



#### Синтез



Разработайте практическое задание по обработке однотипных данных из математики, физики, химии, биологии или географии.



#### Оценка



**Письмо другу.** Напишите письмо другу, в котором поделитесь мыслями о значимости решения практических заданий и выполнения их в виде проекта на языке программирования Python.

**Задания****Задание 1. Замена\***

Дана последовательность чисел длиной  $N(2 \leq N \leq 999)$ . В этой последовательности члены, большие  $M$ , замените числом  $P$ . Выведите на экран результат замены и количество замен.

№	Пример ввода	Результат
1	6 10 30 11 25 3 6 90 8	30 30 3 6 30 8 Количество замен = 4

**Задание 2. Наибольший элемент\*\***

Дан одномерный массив  $X[N]$ ,  $N(1 \leq N \leq 1000)$ . Найдите наибольший элемент этого массива и определите количество таких элементов в массиве, если их несколько.

№	Пример ввода	Результат
1	8 10 14 12 69 32 69 14 69	69 Количество максимальных элементов = 3

**Задание 3. Ожидание в очереди\*\*\***

Герой фильма «Меня зовут Кожа» стоит в очереди в книжном магазине. Всего в очереди  $N$  покупателей-учеников. Известно, что для обслуживания каждого клиента-ученика продавцу требуется  $t$  минут. Кожа на этот раз не хочет нарушать очередь и показаться невоспитанным. Сколько минут придется ждать своей очереди Коже, если он стоит в очереди на  $P$ -м месте? Помогите Коже вычислить количество минут ожидания.

№	Пример ввода	Результат	Пояснение
1	10 7 5 6 3 7 8 4 2 6 9 7	33	Очередь Кожи дойдет через 33 минут.

## Раздел IV. Программирование алгоритмов на языке программирования Python

4.5

## ПЕРЕСТАНОВКА ЭЛЕМЕНТОВ



Как можно изменить места элементов или переставить элементы в одномерном массиве на языке программирования Python?



## Подумай

- С какой целью меняют местами элементы массива?
- Какие примеры вы можете привести из повседневной жизни по теме данного урока?



## Новые знания

В процессе обработки информации часто возникают ситуации, когда значение или места элементов однотипных данных должны быть изменены. Например, изменить в телефоне номера некоторых ваших друзей из-за замены их номеров; изменение занимаемых мест за партами в классе; изменение очередности уроков в расписании и т.д. Очень важно представлять повседневные задачи в математической форме, решать эти задачи на компьютере при помощи методов программирования. Например, можно, представив расположение солдат по росту в виде массива, с помощью методов программирования определить порядок перестановки солдат местами в строю (рис. 1).

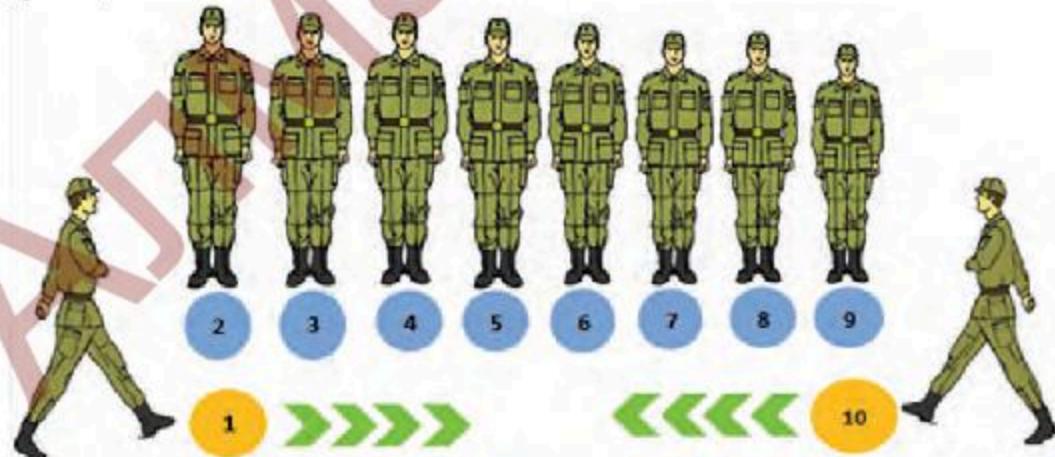


Рис. 1

## Перестановка двух элементов в массиве

Рассмотрим перестановку двух любых элементов в одномерном массиве.

**Пример 1.** Дан одномерный массив  $B[N]$ , ( $0 < N < 100$ ). В этом массиве поменяйте местами  $m$ -ый элемент с  $k$ -ым элементом и выведите на печать полученный массив.

Для перестановок любых двух элементов массива используют введение новой переменной; например  $t$ :

**1 способ:**  $t=b[m]$ ;  $b[m]=b[k]$ ;  $b[k]=t$ ; например, если  $m=1$ ,  $k=4$ , тогда замена будет выполнена следующим образом:  $t=b[1]$ ,  $b[1]=b[4]$ ,  $b[4]=t$  (код 1).

**2 способ:**  $B[m], B[k] = B[k], B[m]$ ; например, если  $m=1$ ,  $k=4$ , тогда замена осуществляется напрямую  $B[1], B[4] = B[4], B[1]$  (код 2).

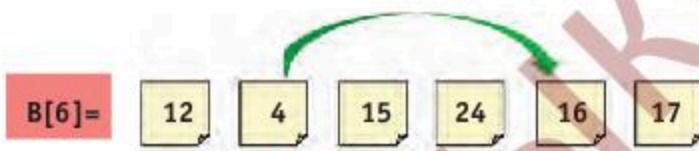


Рис. 2

### Код 1

```
from array import array
B=array("i")
N, m, k=map(int, input().split())
B=[0]*N; t=0
for i in range (0,N):
    B[i]=int( input())
t=B[m]; B[m]=B[k]; B[k]=t
print(B)
```

### Код 2

```
from array import array
B=array("i")
N, m, k=map(int, input().split())
B=[0]*N; t=0
for i in range (0, N):
    B[i]=int(input())
B[m], B[k]=B[k], B[m]
print(B)
```

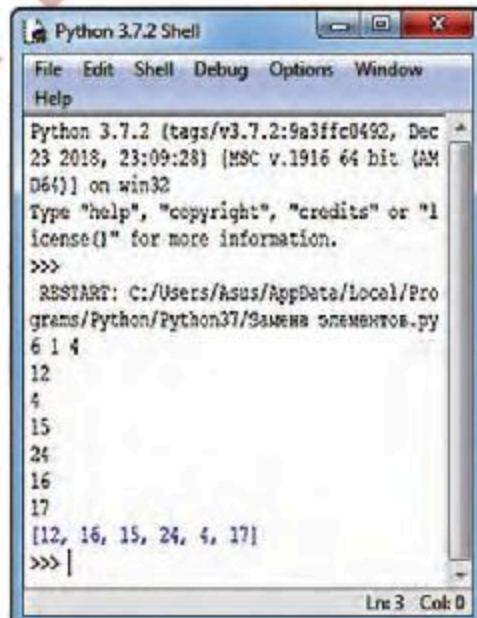


Рис. 3. Процесс выполнения программы

## Замена элементов одного массива на элементы другого

При работе с массивами можно полностью заменить элементы одного массива элементами другого массива. При этом расположение элементов в массиве остается прежним.

Элементы одного массива могут быть записаны во второй массив в соответствии с конкретным условием. Например, элементы массива А можно переписать в точно такой же массив В (код 3).

### Код 3

```
from array import array
A=array("i")
B=array("i")
N=int(input())
A=[0]*N; B=[0]*N;
for i in range (0,N):
    A[i]=int( input())
B=A
print(A)
print(B)
```

### Результат:

```
N=5
10
20
30
40
50
A= [10, 20, 30, 40, 50]
B= [10, 20, 30, 40, 50]
>>>
```

**Пример 2.** Дан одномерный массив D[N], (0<N<100). В данном массиве элементы, меньшие числа Z, замените на Z(Z<1000). Выведите на печать количество замен и элементы нового массива.

### Код программы

```
From array import array
from random import randint
D=array("i")
N=int(input('N='))
Z=int(input('Z='))
D=[0]*N; t=0;
D =[randint(1,100) for x in
range(N)]
print('D=',D)
for i in range(0,N):
    if D[i]< Z:
        D[i]=Z; t+=1
print('Количество замен=', t)
for i in range(0,N):
    print('D=[',i,',']=',D[i])
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 3 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)
] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
>>> runfile('C:/Users/Asus/AppData/Local/Programs/Python/Python37/Zamena_elementov.py'
N=10
Z=50
D= [20, 99, 66, 3, 65, 44, 95, 22, 51, 53]
Количество замен = 4
D[ 0 ]= 50
D[ 1 ]= 99
D[ 2 ]= 66
D[ 3 ]= 50
D[ 4 ]= 65
D[ 5 ]= 50
D[ 6 ]= 95
D[ 7 ]= 50
D[ 8 ]= 51
D[ 9 ]= 53
>>> |
```

Рис. 3. Этап выполнения программы

Рассмотрим перестановку элементов одномерного массива на языке программирования Python.



### Практическая работа

**Пример 3.** Дан одномерный массив  $A[N]$ , ( $1 < N < 100$ ). Создайте проект «Сдвиг вправо» циклического смещения элементов этого массива вправо на одну позицию.

Рассмотрим схему выполнения циклического сдвига вправо. Например, циклический сдвиг вправо массива  $A[5]$ , представлен на схеме 1.

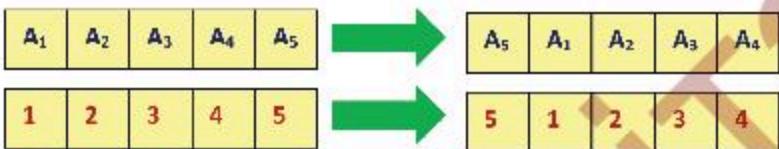


Схема 1. Схема циклического сдвига вправо на одну позицию

```
from array import array
A=array("i")
print('длина массива')
N=int(input('N='))# {длина массива}
print('Количество сдвигов')
P=int(input('P=')) # {количество сдвигов}
A=[0]*N;
print('введите элементы массива')
for i in range(0, N):
    A[i]=int(input())
for i in range(1, P): # {цикл сдвига}
    m=N; k=A[N-1]
    for j in range(0, N):
        m=1
        A[m]=A[m-1]#{сдвиг массива}
        A[0]=k#{присвоение последнего элемента}
print('A=',A)
```

Рис. 5. Циклический сдвиг вправо



### Анализ



Проведите анализ проекта по циклическому сдвигу элементов массива вправо вместе с одноклассником. Проанализируйте функции всех операторов, использованных в программе, и организацию каждого из циклов в проекте, определите их назначение.

**Синтез**

Подготовьте в группе схему сдвига элементов массива влево. Запишите программу сдвига элементов массива влево, изменив программу сдвига элементов массива вправо. Представьте ее.

**Вопросы**

- Можно ли поменять элементы массива на их противоположные значения, не используя условный оператор (например, если положительное число, то заменить его на отрицательное и наоборот)?
- Как можно поменять местами два любых элемента массива?
- Какие проекты из курса математики, физики и других предметов можно создать с использованием способа замены элементов массива на языке программирования Python?
- Как вы думаете, можно ли автоматически загрузить массив числовыми элементами от 1-го до 20-ти по возрастанию без использования клавиатуры? Приведите свои обоснования.

**Задания****Задание №1. Перемещение\***

Дан одномерный массив целых чисел  $A[N]$ ,  $N(1 \leq N \leq 1000)$ . Замените элементы массива, равные нулю, индексом этого элемента. Выведите на экран полученный массив, а также количество замененных чисел.

<b>№</b>	<b>Пример ввода</b>	<b>Результат</b>
1	10 4 0 5 17 0 6 3 0 11 0	4 2 5 17 5 6 3 8 11 10 Количество замененных чисел = 4

**Задание №2. Выбор\***

Дана успеваемость  $K$  учащихся 9-го класса  $K(1 \leq K \leq 200)$ . Отлично успевающие учащиеся на весенних каникулах поедут в путешествие в Астану. В классе 5 отличников, 10 хорошистов и 3 с удовлетворительными оценками. Создайте программу, определяющую список индексов учащихся, которые поедут путешествовать. Созданный список запишите в новый массив.

№	Пример ввода	Результат
1	10 3 5 4 5 3 5 5 3 5 4	2 4 6 7 10



## Домашнее задание

## Задание №1. Помогите физикам

Составьте программу перевода значений температуры в районе вашего проживания, заданного в виде массива  $T[N]$ , из шкалы температур по Цельсию в шкалу температур по Кельвину:  $N(1 \leq N \leq 31)$ .

№	Пример ввода	Результат
1	8 5 -2 7 12 0 -17 -5 6	278 271 280 285 273 256 268 279

## 4.6

## СОРТИРОВКА



Как можно отсортировать элементы одномерного массива в порядке возрастания или убывания?



## Подумай

- Что такое сортировка?
- Как вы понимаете сортировку элементов массива?
- Какие примеры из повседневной жизни вы можете привести по данной теме?



## Новые знания

При изучении прошлого параграфа мы познакомились с поиском элемента одномерного массива по некоторому заданному условию с помощью линейного алгоритма. Например, даны четыре яблока, различающихся по весу (рис. 1). Какое количество перестановок нужно сделать для размещения их по увеличению веса?

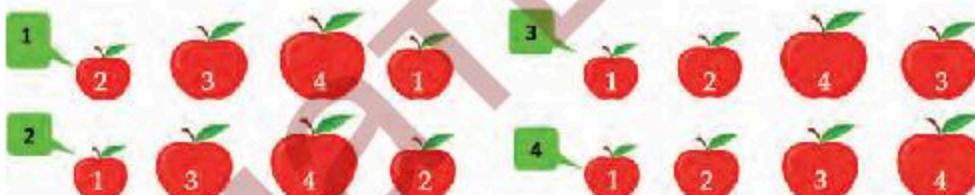


Рис. 1. Сортировка яблок по увеличению веса

В рассматриваемом случае яблок переставляются 4 раза для упорядочивания их по увеличению веса. Хотя перестановок было 4, но каждые два элемента сравниваются 6 раз. При сортировке элементов массива размерностью равным  $n$ , в большинстве случаев на них приходится  $n * n$  сравнений. Такая сортировка относится к типу сортировки *пузырьком*.

**Сортировки в программировании** – это процесс расстановки элементов массива в порядке их возрастания или убывания.

В программировании существует несколько широко используемых алгоритмов сортировки одномерных массивов. Эти алгоритмы сортировки подразделяются по эффективности процесса выполнения (Таблица 1).

Таблица 1

Алгоритм сортировки	Процесс выполнения	Количество всех перестановок при сортировке N=100 элементов
Пузырек – Bubble	Попарно сравнивает все элементы снизу вверх.	10 000 раз
Вставка – Insertion	Элементы массива делятся на две части – упорядоченные и неупорядоченные. Элементы из неупорядоченной части вставляются в часть упорядоченных, сортируя сравнением.	10 000 раз
Выбор – Selection	Нахождение наибольшего (наименьшего) элемента массива и замена первого элемента неотсортированного массива.	10 000 раз
Быстрый – Quicksort	Среди элементов массива выбирается опорный элемент. Элементы со значением меньше опорного упорядочиваются слева, с большим значением – справа (сортировка по возрастанию).	700 раз

**Quick Sort (Быстрая сортировка)** считается наиболее быстрым алгоритмом сортировки среди всех типов сортировок. С помощью Quick Sort числовые элементы массива сортируются по возрастанию или убыванию, текстовые элементы массива сортируются по алфавиту. Quick Sort был разработан англичанином Чарльзом Хоаром, когда он был студентом Московского государственного университета, в 1960 году. Хоар представил свою сортировку в виде усовершенствованной модели пузырьковой сортировки.

Основным требованием для алгоритма сортировки массивов является скорость сортировки и рациональное использование компьютерной памяти.

При создании программ реализации задач целесообразно запоминать коды алгоритмов сортировки для дальнейшего применения. Существует большое количество видов сортировок. Так, например, в таблице 1 представлены только четыре вида. С другими видами сортировок можно ознакомиться в сети Интернет. Теперь давайте ознакомимся со способами использования алгоритмов сортировки для обработки одномерных массивов на языке программирования Python. Можно сортировать не только числа, но и слова с помощью команд

сортировки, представленных в таблице ниже. Например, имена Раушан, Андрей, Магжан и Анастасия могут быть отсортированы в алфавитном порядке, то есть Анастасия, Андрей, Магжан, Раушан. Для этого в программном коде нужно указать, что будут вводиться элементы массива  $A[i] = \text{int}(\text{input}())$ , а также записать  $A[i] = \text{input}()$  для ввода самих элементов массива.

Вид сортировки	Программный код	Пояснение
Сортировка по возрастанию	$B = \text{sorted}(A)$	Отсортирует массив $A$ по возрастанию, записывает его в массив $B$ .
	$A.\text{sort}()$	Производит сортировку массива $A$ по возрастанию и в него же записывает отсортированный массив.
Сортировка по убыванию	$B = \text{sorted}(A, \text{reverse} = \text{True})$	Отсортирует массив $A$ по убыванию, записывает его в массив $B$ .
	$A.\text{sort}(\text{reverse} = \text{True})$	Производит сортировку массива $A$ по убыванию и в него же записывает отсортированный массив.

**Пример 1.** Задан одномерный массив  $A[N]$ . Выведите на печать элементы данного массива, отсортировав их в порядке возрастания, в порядке убывания. Код программы и процесс выполнения программы представлены на рисунке 2.

### Код 1

```
from random import randint
from array import array
print('Длина массива')
A=array("i")
B=array("i")
C=array("i")
N=int(input())
A=[0]*N;
print('Заданный массив')
for i in range(0,N):
    A[i] = randint(1,100)
print('A=',A)
print('Сортировка заданного массива по возрастанию')
B = sorted( A )#(сортировка по возрастанию)
print('B=',B)
print('Сортировка заданного массива по убыванию')
C=sorted( A, reverse = True ) #(сортировка по убыванию)
print('C=',C)
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Wi
Python 3.7.2 (tags/v3.7.2:9a3ffc049b, Dec 3 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\Asus\AppData\Local\Programs\Python\Python37\Сортировка.py
Длина массива
5
Заданный массив
A= [2, 16, 57, 5, 30]
Сортировка заданного массива по возрастанию
B= [2, 5, 16, 30, 57]
Сортировка заданного массива по убыванию
C= [57, 30, 16, 5, 2]
>>>
```

Рис. 2. Результат сортировки массива

**Пример 2.** Дан одномерный массив  $B[N]$ , в котором нет двух одинаковых элементов, ( $0 < N < 100$ ). Найдите в этом массиве  $K$ -й максимальный элемент. Например, если  $N=10$  и  $K=2$ , то в массиве из 10 элементов (8 14 52 14 25 35 67 41 25 30) 2-й максимальный элемент равен 52.

Существует несколько способов решения данной задачи. Самый простой способ – сначала найти максимальный и минимальный элементы, затем заменить максимальный на минимальный. После этого, осуществляется поиск второго максимального элемента в массиве. Конечно, когда количество элементов в массиве мало, скорость выполнения будет удовлетворительной. Что получится, если количество элементов массива не 100, а 1000000 и нужно находить не 2-й максимальный элемент, а 99-й? В таком случае, сортируя массив по убыванию, надо представить 99-й элемент как результат (А. Шень. Программирование: теоремы и задачи. 2004 г.). Если скорость работы компьютера среднего класса 1000000 действий в секунду, то нахождение 99-го максимального элемента займет 1 секунду. Код программы данной задачи показан ниже, а результат представлен на рисунке 3.

### Код 2

```
from array import array
A=array("i")
B=array("i")
N=int(input('Длина массива='))
K=int(input('Значение K ='))
A=[0]*N; B=[0]*N;
for i in range(0,N):
    A[i] = int(input())
B=sorted(A, reverse = True )
print(B)
print(K, 'maximum=',B[K-1])
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffe0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" () for more information.
>>> RESTART: C:/Users/Azua/AppData/Local/Programs/Python/Python37/Поиск второго максимального.py
Длина массива=10
Значение K=2
8
14
52
41
35
30
25
25
14
67
maximum = 52
>>>
```

Рис. 3

Рассмотрим пути разработки проектов для решения задач, встречающихся в повседневной жизни, с использованием сортировки элементов одномерного массива на языке программирования Python.



## Практическая работа

### Проект «Яблоки в дар»

У Алии имеется N штук яблок ( $0 < N < 100$ ). Вес яблок образует массив  $b_i$ , ( $b_i < 1000$ ). Алия хочет подарить своей сестренке Гаухар K самых больших яблок, ( $0 < K < N$ ). Подготовьте проект на языке программирования Python, чтобы помочь Алие выбрать необходимое количество самых больших яблок. Отразите в проекте сортировку яблок по уменьшению их веса. Вместе с этим определите вес K яблок, которые будут подарены сестренке Гаухар, и выведите результат на экран.

#### Код 3

```
from random import randint
from array import array
B=array("i")
# процедура Qsort
def qSort ( A, p, q):
    if p >= q: return
    L = p; R = q
    X = A[(L+R)//2]
    while L <= R:
        while A[L] > X: L += 1
        while A[R] < X: R -= 1
        if L <= R:
            A[L], A[R] = A[R], A[L]
            L += 1; R -= 1
```

#### Основная программа

```
N=int(input('Введите количество яблок ='))
K=int(input('Количество подаренных яблок='))
B=[0]*N
for i in range (0,N):
    B[i] = randint(1,100)
print('Вес яблок')
print(B)
qSort ( B, 0, N-1)
print('Расположение яблок по уменьшению веса')
print(B)
print('Подаренные яблоки, весом ')
print(B[:K])
```

**Пояснение:** При выполнении проекта массив веса яблок образуется с помощью оператора случайных чисел. Поскольку в данной задаче количество веса яблок велико, для быстрой сортировки используем метод Qsort. В программном коде проекта (код 3) мы размещаем яблоки в порядке убывания их веса. В результате мы выдаем первые K яблоко в подарок (рис. 4).

```
Python 3.7.2 |Anaconda v3.7.2-9ae3fc0492, Dec 2 2019, 23:09:26| 64bit |IPython 7.14.0|
Type "help", "copyright", "credits" or "license()" for more information.
>>>
Имя проекта: C:/Users/Aliya/Desktop/
Проект Яблоки в дар.py
Введите количество яблок10
Количество подаренных яблок3
Вес яблок
[78, 82, 28, 14, 13, 83, 63, 54, 9, 46]
Расположение яблок по уменьшению веса
[83, 82, 78, 63, 54, 46, 28, 14, 13, 9]
Подаренные яблоки, весом
[83, 82, 78]
```

Рис. 4. Процесс выполнения проекта «Яблоки в дар»

**Анализ**

Проанализируйте код программы проекта «Яблоки в дар». Выполните также анализ работы каждого использованного оператора и каждого организованного цикла программы проекта, выявите особенности функциональных возможностей.

**Синтез**

С целью пригласить своих одноклассников на чтение художественной литературы создайте проект «100 замечательных книг», в котором должны быть два массива из названий прочитанных вами 10 книг и года их изданий. Составив план выполнения проекта, отражающий нижеследующие условия, представьте его:

1. Упорядочите названия книг по алфавиту.
2. Отсортируйте книги по году издания от новых к старым.

**Вопросы**

1. Что такое сортировка массива?
2. Что можно сказать об основных способах сортировки массивов?
3. Какие требования предъявляются к алгоритмам сортировок?
4. Что можете сказать о самом быстром способе сортировки?
5. Как выполняется сортировка пузырьковым способом?

**Задания****Задание №1. Р-й минимум\***

Задан одномерный массив  $X[N]$  целых чисел,  $N(1 \leq N \leq 1000)$ . Найдите Р-ый минимальный элемент, используя сортировку.

№	Пример ввода	Результат
1	10 4 4 9 5 17 27 6 3 15 11 0	5

**Задание №2. Сумма максимума и минимума\*\***

Задан рост  $K$  учащихся 9-го класса,  $K(1 \leq K \leq 30)$ . Найдите сумму самого высокого и самого низкого роста учащихся.

№	Пример ввода	Результат
1	8 150 181 164 172 171 146 174 158	327

**Задание №3. Сортировка словаря\*\*\***

Сегодня на уроке информатики Канат выучил произношение  $K$  новых терминов на английском языке ( $1 \leq K \leq 30$ ). Помогите Канату отсортировать новые термины по алфавиту. **Примечание:** Текстовые массивы сортируются по тому же принципу, что и числовые. Используй метод сортировки пузырьком.

№	Пример ввода	Результат
1	5 Sort Insertion Quick Bubble Selection	Bubble Insertion Quick Sort Selection

## 4.7

## УДАЛЕНИЕ И ВСТАВКА ЭЛЕМЕНТА



Как осуществляется удаление или вставка нового элемента в одномерном массиве?



## Подумай

- Как можно удалить элементы из массива?
- Как можно вставить новые элементы в массив?



## Новые знания

Познакомимся со способами удаления и вставки элементов одномерного массива. Для этого возьмем рассмотренный в предыдущем параграфе пример со строем солдат на плацу. Как показано на рис. 1.1, 4-й в строю солдат выходит из строя, а 5-й солдат становится на его место, и, в свою очередь, 6-й солдат становится на место 5-го, таким образом заполняется строй (рис. 1.2). При удалении элемента одномерного массива другие элементы перемещаются, заполняя образовавшиеся свободные места.



Рис. 1. Выход солдата из строя

На рисунке 2.1 показано, как рядом с 6-м солдатом в строй становится 7-й солдат (рис. 2.2). А что будет, если этому солдату надо встать не в конце строя, а на  $K$ -е место? В этом случае, начиная с последующего места  $N$ , все солдаты вместе с  $K$ -м солдатом должны сделать шаг вправо, освободив таким образом  $K$ -е место. Точно так же, как добавляется новый солдат в строй, выполняется вставка нового элемента в одномерный массив.

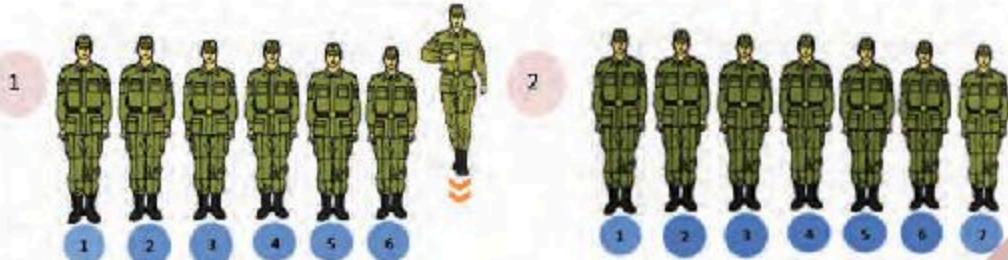


Рис. 2. Добавление в строй солдата

Давайте рассмотрим, как вставлять и удалять элементы в одномерном массиве в **Python** (Таблица 1). Эти действия можно легко реализовать в **Python** по сравнению с другими языками программирования.

Таблица 1

Программный код	Действия, выполняемые с одномерным массивом
<code>A.append(X)</code>	Вставка в конец массива А элемента Х. Например, если $A = [2, 9, 15]$ , тогда $A.append(10) \Rightarrow A = [2, 9, 15, 10]$
<code>A.pop(i)</code>	Удаляет $i$ -ый элемент массива А. Например, если $A = [2, 9, 15, 10]$ , тогда $A.pop(2) \Rightarrow A = [2, 9, 10]$
<code>A.remove(X)</code>	Удаляет элемент массива А, который равен Х. Например, если $A = [2, 9, 15, 10]$ , тогда $A.remove(9) \Rightarrow A = [2, 15, 10]$
<code>A.insert(i, X)</code>	Вставляет элемент на $i$ -ое место в массив А. Например, если $A = [2, 9, 15]$ , тогда $A.insert(1, 7), \Rightarrow A = [2, 7, 9, 15]$
<code>A.count(x)</code>	Определяет количество элементов массива А, равных Х. Например, если $A = [2, 9, 15, 9]$ , тогда $A.count(9) = 2$

Рассмотрим пример на удаление элементов массива, чтобы научиться выполнять данное действие в **Python**.

**Пример 1.** Дан одномерный массив  $A[N]$ , ( $0 < N < 100$ ). Только один элемент этого массива равен нулю. Сократите размер массива, удалив элемент, равный нулю. Например, при удалении элемента, равного нулю, из массива с 10 элементами (80 11 25 14 5 43 5 0 41 75) образуется новый массив (80 11 25 14 5 43 5 41 75) с 9 элементами.

Рассмотрим пример, где способом линейного поиска нужно найти место элемента, равного нулю, удалить его командой `A.pop(i)` (код1) и начиная с этого места перемещать элементы вправо. Выполнение программы представлено на рисунке 3.

**Код 1**

```
from array import array
A=array("i")
N=int(input())
A=[0]*N; k=0;
for i in range (0,N):
    A[i]= int(input())
print('A=',A)
for i in range (0,N):
    if A[i]==0:#(определение
        позиции элемента, равного
        нулю)
        k=i
A.pop(k)#(удаление к -го
        элемента)
print('A=',A)
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffe0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:\Users\User\Desktop\array.py =====
10
80
11
25
14
5
43
5
0
41
75
A= [80, 11, 25, 14, 5, 43, 5, 0, 41, 75]
A= [80, 11, 25, 14, 5, 43, 5, 41, 75]
>>>
```

Рис. 3. Удаление нулей

Теперь давайте рассмотрим программу вставки нужного элемента в массив (пример 2). Для этого мы используем команду `A.insert(i, X)` (код 2). Результат выполнения кода программы примера 2 приведен на рисунке 4.

**Пример 2.** Дан одномерный массив  $B[N]$ , ( $0 < N < 100$ ). Введите в этот массив новый элемент на  $K$ -е место. Например, если в массив с 10 элементами введем в 6-е место число 55 (80 11 25 14 5 43 35 17 41 75), то размер массива увеличится на 1 элемент (80 11 25 14 5 55 43 35 17 41 75).

**Программный код примера 2**

```
from array import array
B=array("i")
N=int(input('Длина
        массива='))
K=int(input('Значение K-ой
        позиции='))
X=int(input('Вставленный
        элемент='))
B=[0]*N; k=0;
print('Введите элементы
        заданного массива')
for i in range (0,N):
    B[i]= int(input())
print('B=',B)
B.insert(K-1,X)
print('B=',B)
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffe0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: C:/Users/Asus/AppData/Local/Programs/Python/Python37/вставка элемента на k-ую по
       зицию.py =====
Длина массива=10
Значение K-ой позиции=6
вставленный элемент=55
Введите элементы заданного массива
12
45
47
13
56
89
78
10
3
5
B= [12, 45, 47, 13, 56, 89, 78, 10, 3, 51]
>>> 1
```

Рис. 4

## Создание проекта перестановки элементов массива местами

Познакомимся с выполнением проекта по удалению и вставке элемента в определенную позицию одномерного массива на языке программирования Python.



### Практическая работа

#### Проект «Удаление и вставка»

Задан массив длиной N, ( $0 < N < 100$ ). Создайте проект, в котором будут выполнены следующие действия:

1. Удаление минимального элемента в массиве.
2. Удаление наибольшего элемента в массиве.
3. Вставка минимального и максимального элементов массива в конец массива. При вводе элементов массива используйте оператор случайных чисел.

#### Код проекта

```
from array import array
from random import randint
A=array("i")
N = int(input('Длина массива='))
A=[0]*N;k=p=0
for i in range(0,N):
    A[i] = randint(1,100)
print('A=',A)
min=A[0]; max=A[0]
for i in range(0,N):
    if A[i]> max :
        max=A[i]; p=i
```

```
for i in range (0,N):
if A[i]< min:
    min=A[i];k=i
A.pop(k)
print('A=',A)
print('Минимальный элемент=',min)
A.pop(p)
print('A=',A)
print('Максимальный элемент=',max)
A.append(min)
A.append(max)
print('A=',A)
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:26) [MSC v.1916 64 bit (AMD64)] on
win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/Asus/AppData/Local/Programs
/Python/Python37/Вставка макс к кон.py
Длина массива=10
A= [41, 17, 12, 86, 59, 69, 17, 86, 1, 10]
A= [41, 17, 12, 86, 59, 69, 17, 86, 10]
Минимальный элемент= 1
A= [41, 17, 12, 59, 69, 17, 86, 10]
Максимальный элемент= 86
A= [41, 17, 12, 59, 69, 17, 86, 10, 1, 86]
>>>
```

Рис. 5. Результат выполнения проекта



## Синтез



Проанализируйте в группе проекты с примерами удаления или вставки элементов массива в определенную позицию. Спроектируйте и представьте проект с использованием процедур

**A.remove(X), A.insert ( i ,X), A.count(x)**

из таблицы 1 для удаления или вставки элементов одномерного массива в примерах, взятых из математики или из повседневной жизни.



## Вопросы

1. Как удаляются элементы из массива?
2. Как выполняется вставка нового элемента в массив?
3. Как можно вставить новый элемент на первую позицию в массив?
4. В чем заключается функция процедуры `A.remove(X)`?
5. С какой целью используется процедура `A .count(x)`?



## Задания

### Задание №1. Очистка массива\*

Дан одномерный массив из целых чисел  $A[N]$ ,  $N(1 \leq N \leq 1000)$ . Удалив в этом массиве числа, превышающие  $P$ , образуйте новый массив. Значения элементов массива не должны превышать 10000.

№	Пример ввода	Результат
1	10 9 4 9 5 17 27 6 3 15 11 0	4 9 5 6 3 0

### Задание №2. Элементы больше среднего арифметического\*\*

Дан одномерный массив из действительных чисел  $K[N]$ ,  $N(1 \leq N \leq 1000)$ . Удалите в этом массиве все числа, превышающие среднее арифметическое. Выведите на экран среднее арифметическое элементов массива. Округление значений элементов в массиве осуществляется с точностью 0,1. Значения элементов массива не превышают 500.

№	Пример ввода	Результат
1	8 152.5 14.6 24 29.5 80 26 10.2 40.6	41.2 14.6 24 29.5 26 10.2 40.6

**Задание 3. Удаление нескольких элементов\*\*\***

Дан одномерный массив из целых чисел A[N], N( $1 \leq N \leq 1000$ ). Удалив в этом массиве элементы от P до K, образуйте новый массив ( $1 \leq P, K \leq 1000$ ). Значения элементов массива не превышают 10000.

№	Пример ввода	Результат
1	10 3 6 58 62 44 478 2 32 6 88 64 55	58 62 6 88 64 55

**Задание 4. Вставка нескольких элементов\*\***

Дан одномерный массив из целых чисел A[N], N( $1 \leq N \leq 1000$ ). Вставив в этот массив число Т в начало, а число К в конец, выведите на экран новый массив. ( $1 \leq P, K \leq 1000$ ). Значения элементов массива не превышают 10000.

№	Пример ввода	Результат
1	10 35 43 14 15 86 68 74 65 89 32 41 65	35 14 15 86 68 74 65 89 32 41 65 43

## 4.8

## ДВУМЕРНЫЙ МАССИВ



Как можно обработать двумерные массивы на языке программирования Python?



## Подумай

- Как вы понимаете понятие **матрица**? В каких отраслях техники встречаются матрицы?
- Какие вы можете привести примеры использования данных, представленных в виде таблицы, из повседневной жизни?



## Новые знания

## Матрица

**Матрица** в математике представляет собой прямоугольную таблицу из каких-либо элементов (чисел, математических выражений), состоящую из  $m$  строк и  $n$  столбцов. Объекты, составляющие матрицу, называются ее **элементами** (рис. 1). В математике и аналитической геометрии многие задачи могут быть решены с помощью матриц (двумерная таблица, массив). Многочисленные задачи, возникающие в различных отраслях, решаются в программировании с помощью матриц.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

Рис. 1. Матрица в математике

$$A = \begin{pmatrix} 2 & 4 \\ -1 & 3 \end{pmatrix}$$

$$a_{11} = 2, a_{12} = 4,$$

$$a_{21} = -1, a_{22} = 3.$$

Рис. 2. Элементы матрицы

**Двумерным массивом** называется набор однотипных данных, имеющий общее имя, каждый элемент которого определяется пересечением номера строки и номера столбца (рис. 3).

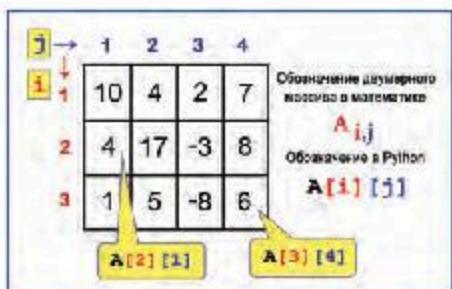


Рис. 3. Элементы матрицы

**Пример 1.** Данна двумерная квадратичная матрица A[N][N]. Запишите в данный массив случайные числа из промежутка 1..9.

Программный код примера (код 1) и результат после выполнения показаны на рисунке 4.

```
from random import randint
N=int(input())
a = [ [0 for j in range(N)] for i in
range(N)]
for i in range(0, N):
    for j in range(0,N):
        a[i][j]=randint(1,9)
    for i in range(N):
        for j in range(N):
            print(a[i][j], end = ' ')
        print()
```

Следующая запись показывает, что определен двумерный массив с нулевыми элементами.

```
a = [ [0 for j in range(N)] for i in
range(N)]
```

Заполнение массива значениями элементов выполняется с помощью вложенных циклов, которые знакомы вам из 8-го класса

Двумерный массив называется прямоугольной таблицей. Двумерный массив A [n] [m] содержит n \* m элементов. Например, массив A [5] [7] имеет 35 элементов.

Давайте рассмотрим использование двумерного массива (матрицы) на языке программирования Python.

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window
Python 3.7.2 (tags/v3.7.2:9a3ffc
(AMD64) on win32
Type "help", "copyright", "credi
>>> ===== RESTART: C:/Users
10
1 3 6 3 9 9 3 6 7 4
7 5 9 4 2 8 1 4 8 8
7 8 5 7 1 6 7 3 4 7
5 5 4 4 6 8 9 9 7 2
9 5 2 3 7 1 5 8 2 1
3 2 8 7 1 1 6 5 6 6
2 3 9 2 7 5 7 6 9 7
3 1 8 1 1 3 4 7 3 8
1 7 3 1 6 0 2 1 7 6
7 3 1 8 4 5 7 5 8 2
>>>
```

Рис. 4. Элементы матрицы

### Главная и побочная диагонали двумерного массива

Различные действия могут быть выполнены с элементами главной диагонали и побочной диагонали двумерного массива. В основном задачи, задаваемые на двумерный массив, выполняются с элементами, расположенными сверху и снизу этих двух диагоналей.

Например, на рисунках 5 и 6 показана структура главной и побочной диагоналей двумерного массива A [4] [4].



Рис. 5. Элементы главной диагонали



Рис. 6. Элементы побочной диагонали

Давайте рассмотрим практическое использование главной и побочной диагоналей двумерного массива на языке программирования Python.



### Практическая работа

**Пример 2.** Задан двумерный массив A [N] [N] ( $1 < N < 20$ ). Найдите сумму элементов, расположенных на главной диагонали этого массива.

```
from random import randint
N=int(input())
A = [[0 for j in range(N)] for i
in range(N)]
for i in range(0, N):
    for j in range(0,N):
        A[i][j]=int(input())
S=0; i=0
for i in range(N):
    for j in range(N):
        if i == j:      #(условие
            #определения главной
            #диагонали)
            S+=A[i][j]
print(S)
```

```
from random import randint
N=int(input())
A = [[0 for j in range(N)] for i
in range(N)]
for i in range(0, N):
    for j in range(0,N):
        A[i][j]=randint(0, 1)
    print(A[i][j], end=' ')
print()
S=0;
for i in range(0,N):
    for j in range(0,N):
        if i>N-j-1 and A[j][i]==0:
            S+=1
print(S)
```

**Пример 3.** Задан двумерный массив A [N] [N] ( $1 < N < 20$ ). Найдите количество нулей среди элементов, расположенных в нижней побочной диагонали массива. Найдите сумму четных элементов массива.

При решении примеров 2 и 3 используйте условия, представленные на рисунках 5 и 6.



### Анализ



Проанализируйте функцию двумерных массивов. Как реализовать ввод элементов массива с помощью цикла на языке программирования Python? Объясните.



### Синтез



Создайте двумерный массив из табличных данных, которые вам встречаются ежедневно. Представьте его. Например, одним из примеров является таблица умножения Пифагора.



### Оценка



Обменяйтесь мнением о важности двумерного массива в математике, программировании и технике.



### Вопросы

1. Что такое матрица?
2. Что такое двумерный массив?
3. С помощью каких условий определяют главную диагональ и побочную диагональ двумерных массивов?



### Задания

#### Задание №1. Наибольший (максимальный)\*

Задан двумерный массив C [n] [m] ( $1 < n, m < 20$ ). Найдите наибольший элемент в этом массиве.

№	Пример ввода	Результат
1	3 4 4 5 6 2 7 8 90 11 20 3 14 68	max=90

**Задание №2. Количество нулей\***

Задан двумерный массив  $A[n][m]$  ( $1 \leq n, m \leq 20$ ). Найдите количество элементов, равных нулю в этом массиве ( $-20 \leq A_{i,j} \leq 20$ ).

№	Пример ввода	Результат
1	4 -5 0 4 11 12 -7 0 8 -9 0 0 7 15 -8 0 0	6

**Задание №3. Экзамен\*\***

Выпускные экзамены по 5 предметам проводятся для выпускников основной школы (9 класс). N учеников приняли участие в этих экзаменах. Сколько учеников сдали все экзамены на **отлично**? Экзаменационная ведомость  $X[N, 5]$  представлена в виде двумерного массива N ( $1 \leq N \leq 20$ ).

№	Пример ввода	Результат
1	4 3 4 5 5 3 4 4 5 5 3 5 5 5 5 5 5 5 5 5 4	1

## 4.9

## ОСНОВНЫЕ ПАРАМЕТРЫ ДВУМЕРНОГО МАССИВА



Как можно выполнить сортировку, удаление и вставку элементов в двумерные массивы?



## Подумай

- Какие параметры имеются у двумерного массива?
- Каким образом можно использовать элементы двумерного массива?



## Новые знания

## Вставка элемента в двумерный массив

Существует несколько видов ссылок на элемент двумерного массива на языке программирования Python. Давайте познакомимся с одним из них.

В скобках можно перейти к следующей строке.

```
A=[[4, 9, 1],  
 [6, 7, 1],  
 [9, 1, 3]  
 [8, 0, 8]]
```

Можно записать в одну строку.

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

Выполнение ссылок на элемент массива  
`print(A[1][2])` значение 7  
`print(len(A))` значение 3  
`print(len(A[1]))` значение 4  
`print(A[2])` значение [5,6,7,8]

**Пример 1.** Задан двумерный массив A [N] [N] ( $1 < N < 20$ ). Найдите сумму элементов и наибольший элемент этого массива.

Существует несколько способов ввода элементов в двумерные массивы. На этот раз мы будем использовать функцию `append(x)` для ввода элементов в массив (**код 1**).

**Код 1**

```

N=int(input(длина массива=))
a=[]; s=x=0
for i in range(N):
    a.append([]) # определяет пустую строку
    for j in range(N):
        x =int(input())#вводится значение x
        a[i].append(x) #размещение x в массив
    for i in a:
        print(i)
    print();
max=a[0][0];
for i in range(0,N):
    for j in range(0,N):
        s+=a[i][j]# вычисление значения суммы
        if a[i][j]> max:
            max=a[i][j]#нахождение наибольшего среди
элементов
    print('Sum=' s)
    print('Max=' s)

```

Длина массива=3  
[1, 2, 3]  
[5, 9, 12]  
[36, 25, 22]  
Sum=115  
Max=36

**Сортировка двумерных массивов**

Под сортировкой двумерного массива надо понимать сортировку по строкам или столбцам. При выполнении сортировки двумерного массива будем опираться на сортировку одномерных массивов. Например, давайте рассмотрим двумерный массив размером 5x5 (рис. 1). Как видим, элементы каждой строки этого массива отсортированы в порядке возрастания. Мы используем функцию сортировки (код 2) для упорядочения каждой строки двумерного массива. Результат выполнения программы показан на рисунке 2.

**Заданный массив**

1	23	4	5	6
25	14	78	35	18
8	9	4	0	16
7	11	58	3	11
25	74	78	75	38

**Упорядоченный массив**

1	4	5	6	23
14	18	25	35	78
0	4	8	9	16
3	7	11	11	58
25	38	74	75	78

Рис. 1

**Код 2**

```
from random import randint
N=int(input())
a = [ [0 for j in range(N)] for i in
range(N)]
b = [ [0 for j in range(N)] for i in
range(N)]
for i in range(0, N):
    for j in range(0,N):
        a[i][j]=randint(1,30)
for i in a:
    print(i)
print()
for i in range(0, N)
    b=sorted(a[i])#(сортировка
элементов массива )
print(b)
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options
Python 3.7.2 (tags/v3.7-
(AMD64) ) on win32
Type "help", "copyright"
>>>
== RESTART: C:/Users/t
5
[29, 8, 7, 23, 15]
[3, 25, 21, 25, 2]
[23, 22, 30, 25, 15]
[1, 11, 1, 4, 21]
[13, 21, 28, 11, 8]
[7, 8, 16, 23, 29]
[2, 3, 24, 25, 25]
[15, 22, 23, 25, 30]
[1, 1, 4, 11, 21]
[8, 11, 13, 21, 28]
>>>
```

Рис. 2

**Удаление K-ой строки в двумерном массиве**

Можно удалить строку в двумерном массиве. Возьмем двумерный массив размером 5x5 (рис. 3). Давайте рассмотрим удаление K-ой строки этого массива. Например, как видим на рисунке 2, при K = 0 первая строка массива полностью удалена (нумерация строки в массиве начинается с нуля). Мы используем процедуру удаления `remove(X)`(код 3), чтобы удалить выбранную строку заданного массива. Результат реализации программы показан на рисунке 4.

**Заданный массив**

1	23	4	5	6
25	14	78	35	18
8	9	4	0	16
7	11	58	3	11
25	74	78	75	38

**Массив с удаленной строкой**

25	14	78	35	18
8	9	4	0	16
7	11	58	3	11
25	74	78	75	38

Рис. 3

**З-код**

```
N=int(input('длина массива= '))
a=[]; s=x=0
for i in range(N):
    a.append([])
    for j in range(N):
        x = int(input())
        a[i].append(x)
    print(' массив A ')
for i in (a):
    print(i)
k=int(input ('Изменяемая строка=' ))
a.remove(a[k])
for i in (a):
    print(i)
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window
Python 3.7.2 (tags/v3.7.2:9a34be7, Dec 23 2018, 23:09:12) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/Asus/AppData/Local/Programs/Python/Python37/Измененная строка.py
длина массива = 3
12
45
74
10
3
51
9
62
36
массив A
[12, 45, 74]
[10, 3, 51]
[9, 62, 36]
Удалляемая строка=1
[12, 45, 74]
[9, 62, 36]
>>>
```

Рис. 4

**Практическая работа**

**Пример 2.** Задан двумерный массив С [N] [N] ( $1 < N < 20$ ). В этом массиве удалите строку с максимальным количеством нечетных элементов. Значение элементов массива не должно превышать 100.

Чтобы удалить строку с наибольшим количеством нечетных элементов в массиве, мы можем сохранить позиции нечетных чисел. Используйте эти позиции, чтобы стереть строку (код 4). Результат программы показан на рисунке 5.

**Код 4**

```
from random import randint
N=int(input()); max=k=0
a = [[0 for j in range(N)] for i in range(N)]
for i in range(0, N):
    for j in range(0,N):
        a[i][j]=randint(1,50)
    for i in (a):
```

```

print(i)
for i in range(0, N):
    k=0
    for j in range(0,N):
        if a[i][j] % 2 == 1:
            k+=1#(подсчет нечетных
            элементов)
        if k> max:
            max=k;p=i;#(номер строки с
            наибольшим
            print() количеством нечетных
            элементов
            сохраняется в p)
a.remove(a[p])#(удаление строки p)
for i in (a):#(вывод массива на экран)
    print(i)

```

```

Python 3.7.2 Shell
File Edit Shell Debug Options Window
Python 3.7.2 (tags/v3.7.2:9a34
(AMD64) ) on win32
Type "help", "copyright", "c
>>>
===== RESTART: C:/P
6
[23, 43, 24, 48, 13, 34]
[13, 31, 4, 44, 14, 1]
[27, 15, 29, 32, 7, 34]
[3, 31, 13, 36, 10, 19]
[22, 39, 21, 9, 30, 26]
[23, 12, 33, 29, 1, 37]

[23, 19, 21, 18, 13, 34]
[13, 31, 4, 44, 14, 1]
[27, 15, 29, 32, 7, 34]
[3, 31, 43, 36, 40, 49]
[22, 39, 21, 9, 30, 26]
>>>

```

Рис. 5



### Анализ



Проанализируйте программы сортировки и удаления элементов двумерных массивов. Определите, что будет выполнять каждая из строк программы.



### Синтез



Составьте и представьте программный код, в котором будут найдены и удалены строки, содержащие наибольший и наименьший элементы двумерного массива. Если наибольший и наименьший элементы находятся в одной строке, то тогда должна быть удалена одна эта строка.



### Оценка



Подготовьте небольшую научную статью на тему: «Пути использования двумерных массивов при решении математических и физических задач».



### Вопросы

- Какие способы ввода данных в двумерный массив вы можете назвать?

2. Что вы думаете о практической значимости сортировки двумерных массивов?
3. Как выполняется удаление строки массива?
4. Какой фрагмент программы нужно написать, чтобы элементы двумерного массива были выведены в четырехугольной форме?



## Задания

### Задание №1. Количество четных элементов\*

Задан двумерный массив  $A[N][m]$  – двумерный массив ( $1 < n, m < 20$ ). Найдите количество четных элементов в этом массиве.

№	Пример ввода	Результат
1	3 4 40 5 6 2 70 8 9 11 20 3 14 68	8

### Задание №2. Удаление\*

Задан двумерный массив  $A[N][m]$  – двумерный массив ( $1 < n < 20$ ). Удалите из этого массива ( $-20 \leq A_{i,j} \leq 20$ ) строку с максимальной суммой элементов.

№	Пример ввода	Результат
1	3 -15 4 11 12 -7 8 -9 7 6	3 -15 4 11 -9 7 6

## Раздел IV. Программирование алгоритмов на языке программирования Python

**4.10****ТВОРЧЕСКО-ПРАКТИЧЕСКАЯ РАБОТА ПО ИСПОЛЬЗОВАНИЮ  
ОДНОМЕРНЫХ И ДВУМЕРНЫХ МАССИВОВ**

Как эффективно использовать одномерные и двумерные массивы для решения разных задач?

**Подумай**

- Как вы думаете, какие задачи, встречающиеся в повседневной жизни и науке, могут быть решены с помощью массивов?

**Практическая работа****Задачи для использования циклов с параметром****Задача №1. Двухзначные числа \***

Задан одномерный массив  $A[N](1 \leq N \leq 99, a_i < 1000)$ . Найдите количество всех двухзначных чисел в этом массиве.

№	Пример ввода	Результат
1	8 485 14 586 2 540 58 27 896	3

**Задача №2. Сортировка\*\*\***

Достижения в боксе нашей страны признаны во всем мире. В этом виде спорта среди наших боксеров есть несколько победителей Олимпийских игр. В любительском боксе имеется 10 весовых категорий от 49 кг до 91 кг. Интерес к боксу среди учеников 9-х классов очень высок. Для записи в спортивную секцию вес учеников должен находиться в промежутке  $47 \leq m_i \leq 92$  кг. Среди учеников 9-го класса  $N$  учеников измерили свой вес, чтобы записаться в спортивную секцию по боксу ( $2 \leq N \leq 100, 30 \leq m_i \leq 125$ ). По измеренным данным определите, сколько учеников имеют право записаться в спортивную секцию по боксу. Каковы номера этих учеников в списке?

№	Пример ввода	Результат
1	10 80 32 78 98 47 85 65 110 34 119	4 1 3 5 6 7

**Задача №3. Чтение из файла\*\*\***

Файл **Input.txt** содержит одномерный целочисленный массив  $A[N]$  ( $1 \leq N \leq 1000, 0 \leq a_i \leq 10^6$ ). Найдите в массиве, размещенном в этом файле, нечетные числа на четных позициях и вычислите их сумму. В первой строке файла **Input.txt** указывается длина массива  $N$ . Следующая строка содержит значение этих  $N$  элементов. Запишите полученную сумму в файл **Output.txt**.

№	Input.txt	Output.txt	Пояснение
1	12 21 26 46 87 41 5 16 10 15 3 8 70	95	Нечетные числа в четной позиции $87+5+3=95$

**Задача №4. Одинаковые цифры\*\***

Задан одномерный массив трехзначных натуральных чисел  $X [N]$  ( $1 \leq N \leq 99, 0 \leq a_i < 10000$ ). Определите среди элементов этого массива трехзначные числа, имеющие равные между собой первую и последнюю цифры, и выведите их на печать.

№	Пример ввода	Результат
1	10 141 605 786 177 989 666 909 545 100 101	141 989 666 909 545 101

**Задача №5. Равный К \***

Задан двумерный массив  $B [N][N]$  ( $1 < N < 20$ ) и число  $K$ . Необходимо среди элементов этого массива найти те числа, последняя цифра которых равна  $K$ , и вычислить сумму этих чисел.

№	Пример ввода	Результат	Пояснение
1	3 7 12 47 38 17 15 69 70 11 7	71	Для $K=7$ это будут числа 47, 17, 7

**Задача №6. Количество нечетных чисел\*\***

Задан двумерный массив  $A [N][N]$  ( $1 < N < 20$ ). Найдите количество нечетных чисел среди элементов главной диагонали этого массива.

№	Пример ввода	Результат	Пояснение
1	5 1 2 3 6 5 2 5 6 9 8 1 7 8 9 6 4 8 9 5 3 1 5 3 6 7	4	Нечетные числа главной диагонали 1, 5, 5, 7

**Задача №7. Количество хорошистов в классе \*\*\***

В 9 «А» классе учатся N учеников ( $1 < N < 20$ ). Имеется сводная ведомость успеваемости учащихся класса по K предметам. Сколько учеников класса являются *хорошистами* по результатам четверти? Хорошистом следует считать только того ученика, у которого все оценки «4» и «5».

№	Пример ввода	Результат
1	5 6 3 4 5 4 5 5 4 5 4 4 4 4 5 5 5 5 5 5 3 3 3 3 4 4 4 4 5 4 4 4	2

**Задача №8. Замена максимумов\*\*\***

Задана числовая последовательность длиной N ( $1 < N < 1000$ ,  $1 \leq d_i \leq 10^6$ ). В этой последовательности имеется несколько максимальных элементов. Замените все максимальные элементы минимальным элементом заданной последовательности. Выведите на печать количество замен и полученную новую последовательность.

№	Пример ввода	Результат
1	10 15 18 98 36 98 7 13 87 65 98	Количество замен = 3 15 18 7 36 7 7 13 87 65 7

**Задача №9. Помощник архитектора\*\*\***

С сегодняшнего дня Тимур устроился работать помощником выдающегося архитектора. Архитектор Тимуру в своем первом задании задал обработку данных, хранящихся в файле. Имя файла архитектора *architect.txt*. Этот файл содержит сведения о длине, ширине и высоте N ( $1 < N < 1000$ ,  $1 \leq a_i, b_i, c_i \leq 200$ ) домов, которые будут строиться. Тимур

должен использовать эти данные, чтобы найти площадь, объем и периметр домов. Тимур также обязан определить самый большой дом по площади. Учитывая, что это первый рабочий день Тимура, как близкий друг, помогите ему написать программу. Запишите результат в файл `result.txt`.

№	architect.txt	result.txt
1	5 12 8 4 14 11 7 42 36 15 25 24 23 30 10 14	96 384 40 154 1078 50 1512 22680 156 600 13800 98 300 4200 80 Номер самого большого дома по площади = 3

### Задача №10. Сумма\*\*

Задан массив одномерных целых чисел  $C [N]$  – ( $1 \leq N \leq 99$ ,  $0 \leq a_i < 1000$ ). Найдите сумму первых  $K$  наибольших чисел в этом массиве.

№	Пример ввода	Результат	Пояснение
1	10 4 17 70 20 30 49 78 20 60 10 50	258	$78 + 70 + 60 + 50 = 258$ .

### Задача №11. Удаление\*\*\*

Жулдыз очень любит математику, но не любит простые числа. Арай хорошо это знает. Арай решила подарить Жулдыз на день рождения последовательность чисел, но чтобы она не содержала простые числа. Тем не менее Арай не находит одно простое число в последовательности. Помогите Арай в качестве друга найти простое число и удалите его из последовательности. Выведите на печать последовательность после удаления простого числа. Длина последовательности  $N$  ( $1 \leq N \leq 1000$ ,  $1 \leq b_i \leq 200$ ).

№	Пример ввода	Результат
1	10 40 18 62 57 39 19 33 15 37 85	40 18 62 57 39 33 15 37 85

#### Дескрипторы для выполнения задач:

- ❖ Внимательно прочтайте задание и поймите заданное условие.
- ❖ Правильно составьте математическую модель задачи.
- ❖ Напишите код программы.

*Примечание: \* легкий, \*\* средний, \*\*\* сложный, \*\*\*\* очень сложный уровень заданий.*

**5.1****PYGAME. PYGAME БИБЛИОТЕКА**

Как запустить библиотеку PyGame?

**Подумай**

- В какие компьютерные игры вы любите играть?
- Знаете ли вы, с помощью какой программы были созданы эти игры?
- Хотите ли вы создать игру на своем компьютере по собственному сценарию?
- Как вы думаете, какие нужны знания, чтобы разработать компьютерную игру?

**Новые знания**

Игра является одним из лучших изобретений, придуманных людьми для развития логического мышления. Благодаря игре мы можем стать героическими персонажами в образе солдата, короля, спасателя, мастера и т.д. В последние 25-30 лет наряду с интенсивным развитием компьютерных технологий появилось множество разнообразных видов компьютерных игр. Но ничто не может сравниться с игрой, разработанной по своему сценарию и своим правилам.

Создание графики на компьютере (добавление звука к нему) для разработки игр требует определенных знаний. Модуль Pygame языка программирования Python является удобной средой, которую можно использовать для создания игры. Теперь давайте узнаем, как создать игру, используя модуль Pygame.

Модуль Pygame устанавливается отдельно от языка программирования Python (требуемую версию можно загрузить по адресу <https://www.pygame.org/download.shtml>). Чтобы установить библиотеку Pygame для программы Python, необходимо выполнить команду Пуск – Все программы – Стандартные – Командная строка (пример для Windows 7). В командной строке введите команду `pip install pygame` и нажмите клавишу Enter (*рис. 1*).

**Pip** – система управления пакетами, используемая для установки и управления программными пакетами, написанными на Python.

Как видно из рисунка 1, библиотека Pygame установлена на 100% и готова к работе. Если Pygame не установлен, вам нужно установить pip в ОС Windows. В версии 3.4 Python или более новых версиях pip будет установлен вместе с Python.

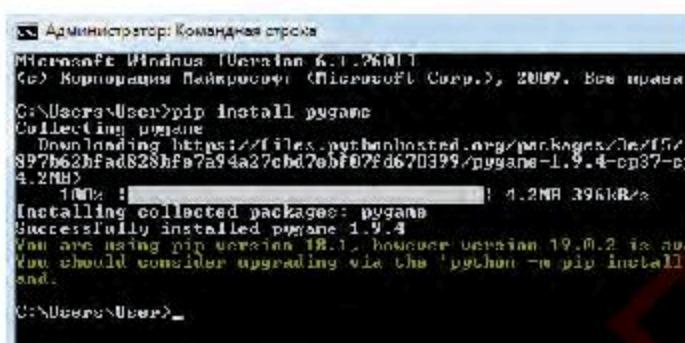


Рис. 1

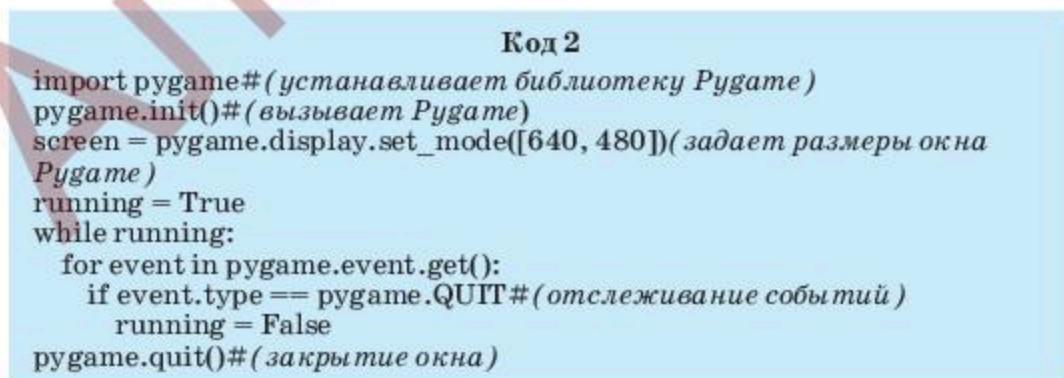
### Окно Pygame

После установки Pygame нам нужно создать игровое окно (код 1). Когда вы запускаете программный код 1, на экране появляется черный фоновый экран размером 640 x 480 пикселей (рис. 2).



Рис. 2

Любая игровая программа должна быть на связи с игроком (пользователем). Поэтому в программах с модулем Pygame используется команда **event loop** – цикл событий, постоянно проверяющий, выполняет ли пользователь какое-либо действие или нет. Это окно не отвечает щелчкам мыши. В Pygame-программах цикл событий всегда должен работать непрерывно. В нашей же первой программе этот цикл вообще не был запущен, из-за отсутствия этого цикла программа работала некорректно.



Теперь давайте познакомимся с командами Pygame.draw, которые дают возможность построить геометрические фигуры, принадлежащие библиотеке PyGame, такие как круги, прямые и прямоугольники.

### Рисование круга

**pygame.draw.circle (screen, color, pos, radius, width)**

screen – поверхность, на которой будет рисоваться круг

color – цвет круга [R,G,B],

pos – позиция центра круга, pos=[x,y]

radius – радиус круга R

width – это толщина для рисования внешнего края. Если ширина равна нулю width=0, то круг будет заполнен (код 3, рис. 3).

Запись в программе: pygame.draw.circle(screen, [93,68,255],[170,60], 50, 0)

### Рисование прямоугольника

**pygame.draw.rect (screen, color, rect, width)**

rect[x<sub>1</sub>, y<sub>1</sub>, x<sub>2</sub>, y<sub>2</sub>] – рисует прямоугольную фигуру по координатам левого верхнего и правого нижнего угла (код 3, рис. 3)

Запись в программе:

pygame.draw.rect (screen, [255,0,0], [80, 130, 180, 180], 0)

### Рисование

**pygame.draw.line (screen, color, start, end, width )**

start=[x<sub>1</sub>, y<sub>1</sub>] – координата точки, с которой начинается прямая

end=[x<sub>2</sub>, y<sub>2</sub>] – координата точки, с которой заканчивается прямая

Запись в программе:

pygame.draw.line (screen, [0,255,33], [50, 100], [50, 300],5)

### Код 3

```
import pygame
pygame.init()
screen = pygame.display.set_mode ([360, 320])
screen.fill([255,255,255])
pygame.draw.circle(screen, [93,68,255],[170,60], 50, 0)
pygame.draw.rect(screen, [255,0,0], [80, 130, 180, 180], 0)
pygame.draw.line (screen, [0,255,33], [50, 100], [50, 300],5)
pygame.display.flip()
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
pygame.quit()
```

## Поверхность в модуле Pygame

В модуле Pygame поверхность (**surface**) – это то, на чем вы рисуете. Поверхность отображения (**display surface**) – то, что вы видите на экране. Именно она в коде 3 получила имя **screen**. Но в Pygame-программе при разработке игр может быть множество поверхностей.

### Цвета в модуле Pygame

В модуле Pygame используется распространенная цветовая модель RGB. Буквы R,G и B означают красный (red), зеленый (green) и синий (blue) цвета.

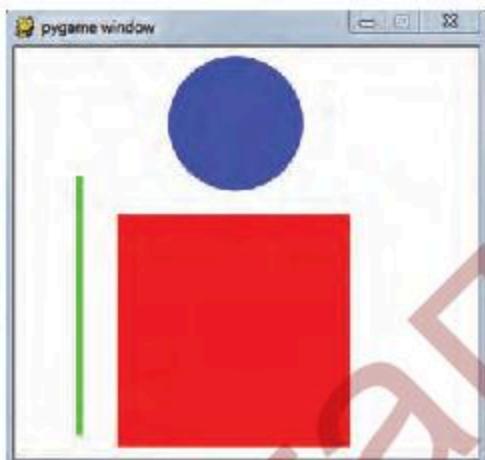


Рис. 3



### Анализ



Проанализируйте команды, с помощью которых строятся геометрические фигуры. Сравните формы фигур, таких как круг, прямая и прямоугольник в геометрии. Поясните сходства и различия.



### Синтез



Представьте программу рисования фигур, показанных на рисунке 4, с помощью команд pygame.draw.

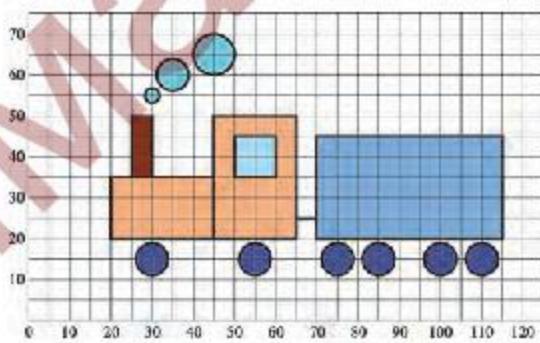


Рис. 4



### Вопросы

1. В чем заключается функция модуля Pygame?
2. Как можно запустить модуль Pygame?
3. С какими командами вы познакомились для создания геометрических фигур?
4. Как выбираются цвета в Pygame?

**5.2****ЗАДНИЙ ФОН И ПЕРСОНАЖИ ИГРЫ**

Как в модуле PyGame можно загрузить задний фон и персонажей игры?

**Подумай**

- Что такое фон игры?
- Как выбрать фон при создании игры?
- Насколько важно правильно выбрать фон для создания интересной игры? Поделитесь своими мыслями.
- Какая игра вам нравится? На персонажа, какой игры хотелось быть похожими? Объясните, почему.

**Новые знания**

Имеется основное различие между чтением про какое-то событие, просмотром фильма про это событие и созданием компьютерной игры об этом событии и игры в нее. В процессе игры игроку разрешается принять участие в развитии событий. Как правило, развитие логического мышления игрока в компьютерных играх происходит вместе с развитием персонажа. В игре фон является одним из ключевых факторов, определяющих историю игры и игровую атмосферу. В предыдущей теме мы узнали, как подготовить игровое окно Pygame для игровых площадок. Это окно изначально белое. Поэтому важно установить фон в окне игры для выбранной вами темы игры.

**Фон** – основной цвет, образующий задний план, на котором размещается изображение или текст.

Во время игры фон, видимый за персонажем, можно представить в двух вариантах:

1. Игровое поле раскрасить одним выбранным цветом.
2. Вставить на задний план изображение, которое соответствует теме игры.

Как видно на рисунке 1, с помощью команды `screen.fill ((150, 150, 100)` полностью закрасили игровое окно цветом RGB = (150, 150, 100). Разумеется, раскраска фона одним цветом в игровом окне не делает

игру привлекательной. Поэтому для каждой игры вставка заднего фона в соответствии с тематикой и ее содержанием, делает игру увлекательной (*рис. 2*).

Теперь давайте посмотрим, как вставить задний фон в игровое окно (код 1). Вместе с задним фоном мы вставляем изображение главного героя на игровую площадку. Теперь давайте познакомимся с командами, которые нам нужны для получения игрового процесса, представленного на рисунке 2 (*таблица 1*).



Рис. 1

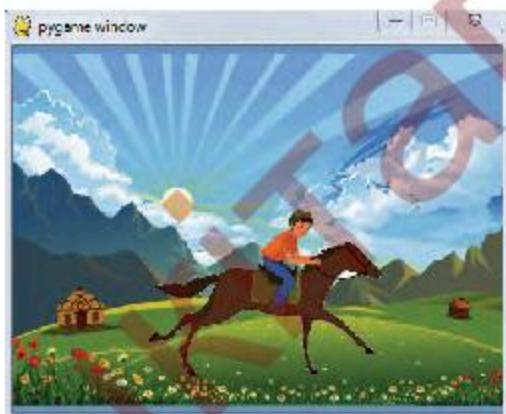


Рис. 2

### Загрузка рисунка

Функция `pygame.image.load()` загружает изображение с диска. Графический формат, который pygame распознает всегда, – это BMP(Paint), однако, если функция `pygame.image.get_extended()` принимает значение «True», то можно загружать ряд других форматов: PNG, GIF, JPG и др.

### Вывод на экран

`screen.blit(image,(x,y))` – изображение копируется из одного места на другое, начиная с координаты (x, y) верхнего левого угла. `pygame.surface.blit` помещает одно изображение поверх другого. Метод `blit()` применяется к той поверхности, на которую «накладывается», т.е. на которой «отрисовывается», другая.

### Изменение размера изображения

`new_image = pygame.transform.scale (image, (width, height))`  
`image, (width, height)` отображает название, длину и ширину изображения, которое вводится.

`Rect` используется для управления прямоугольными областями и сохранения их.

`pygame.Surface.convert` изменяет формат пикселя изображения.

`pygame.Surface.set_colorkey` устанавливает прозрачную поверхность.

**Пояснение кода программы**

`jaulay_surf = pygame.image.load('001.bmp').convert()` – загрузка фонового изображения.

`jaulay_rect = jaulay_surf.get_rect(center=(200, 150))` – изображение жайляу размещается в центре экрана.

`screen.blit(jaulay_surf, jaulay_rect)` – изображение жайляу помещается на место для рисунка.

**Код 1**

```
import pygame
pygame.init()
screen = pygame.display.set_mode((400, 300))
jaulay_surf = pygame.image.load('001.bmp').convert()
jaulay_surf.set_colorkey((255, 255, 255))
jaulay_rect = jaulay_surf.get_rect(center=(200, 150))
screen.blit(jaulay_surf, jaulay_rect)
bala_surf = pygame.image.load('bala.png')
bala_rect = bala_surf.get_rect(bottomright=(300, 270))
screen.blit(bala_surf, bala_rect)
pygame.display.update()
pygame.display.update()
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
pygame.quit()
```

**Анализ**

Проанализируйте программу вставки изображений на задний фон. Прокомментируйте действие, выполняемое каждой строкой программы.

**Синтез**

Предложите сценарий интересной игры. Выберите персонажа для этой игры и поместите задний фон и персонажа в поле с игрой.

**Вопросы**

1. Что такое фон?
2. Какие вы можете назвать пути размещения заднего фона игры?
3. Какую команду мы используем для загрузки фонового изображения в Pygame?
4. Какая команда используется для размещения одного изображения поверх другого?

**Задания**

1. Какие действия выполняют фрагменты программы, представленные в таблице? Заполните таблицу.

№	Код команды	Выполняемое действие
1	screen = pygame.display.set_mode((500, 400))	
2	horse_surf = pygame.image.load('horse.bmp')	
3	for event in pygame.event.get():	
4	ball_rect = ball_surf.get_rect(center = (300, 270)) screen.blit(ball_surf, ball_rect)	

2. Запишите полный код или фрагмент программного кода размещения двух изображений (персонажа) на заданном фоновом рисунке игры.

1. Название изображения фона игры – фон 2.bmp (*рис. 3*)
2. 1 персонаж: Мальчик-наездник – всадник.png (*рис. 4*)
3. 2 персонаж: Жеребец-жеребец.png (*рис. 5*)
4. Создание сюжета (*рис. 6*)

*Рис. 3. Фоновый рисунок**Рис. 4. Мальчик наездник**Рис. 5. Жеребец-жеребец**Рис. 6. Создание сюжета*

**5.3****ВЫБОР ПЕРСОНАЖЕЙ ДЛЯ ИГРЫ**

Как можно выбрать и загрузить игровых персонажей в PyGame?

**Подумай**

- Каких персонажей вы бы выбрали, чтобы сделать их героями создаваемой вами игры?
- Насколько важны выбор игровых персонажей и правильная настройка параметров игры?

**Новые знания**

В предыдущей теме мы узнали пути вставки изображений на задний фон в игровое поле Pygame. Помимо вставки фона в игровое поле, мы научились размещать персонажей. Теперь давайте более подробно остановимся на проблеме размещения персонажей в игре.

Хотя данная программа считается одним из видов игровой графики, дающей возможность рисовать и изменять фигуры, линии и отдельные пиксели, во многих случаях персонажи игры не рисуют в самой программе. Причина в том, что играть с такими персонажами непривлекательно и неинтересно. Чтобы сделать игру привлекательной, лучше персонажей для создаваемой игры изображать с помощью графических редакторов.

Например, давайте персонажем нашей игры станет легковая автомашинка (рис. 1).



Рис. 1

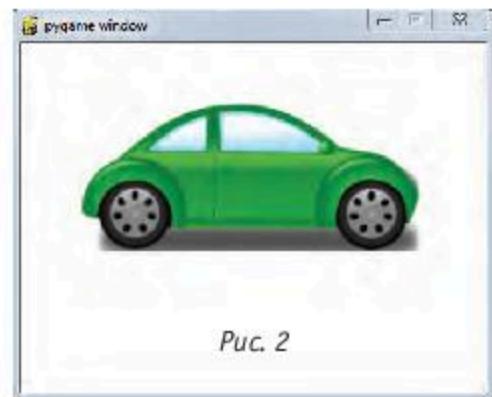


Рис. 2

Скопируем и сохраним изображение автомобиля (персонаж) из сети Интернет на свой компьютер. Для использования этого изображения в программе желательно хранить изображение и программный код в одной папке. Если вы загружаете изображение персонажа в другую папку, то при использовании данного изображения необходимо указать папку, в которой оно хранится. Например, если изображение хранится в папке **Рисунок**, то запишем `pygame.image.load("Рисунок/авто.png")`, чтобы загрузить изображение в программу. Когда вы запустите код (код 1), персонаж игры появится на игровом поле, как показано на рисунке 2.

### Код 1

```
import pygame
pygame.init()
screen = pygame.display.set_mode([640,480])
screen.fill([255, 255, 255])#(закраска экрана белым цветом)
avto = pygame.image.load("авто.png")#(загрузка персонажа)
screen.blit(avto, [50, 50])#(показание рисунка, начиная с
позиции[50,50])
pygame.display.flip()
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    pygame.quit()
```

В Pygame игровые персонажи также называются спрайтами. **Спрайт** (англ. *sprite* – фея; эльф) в компьютерной графике считается объектом.

**Спрайт (sprite)** – графический объект, который перемещается по экрану и взаимодействует с другими графическими объектами. Другими словами, спрайт можно считать персонажем игры. В Pygame для работы с ними существует специальный модуль **sprite**.

При загрузке в программу спрайта (персонажа) мы должны обратить внимание на следующее. Когда вы вставляете спрайт в программу, он сохраняет свои графические размеры. И в зависимости от окна

игрового поля, если вы захотите уменьшить или увеличить размеры персонажа игры, вам нужно будет использовать команду `pygame.transform.scale`.

```
picture = pygame.transform.scale(picture, (40, 50))
```

После выполнения данной команды, на игровой площадке появится спрайт (игровой персонаж) шириной 40 и длиной 50.

Чтобы написать заголовок в окне игры, выполняется следующая команда:

```
pygame.display.set_caption('Название игры')
```



### Практическая работа

**Пример 1.** Создайте проект «Гонки» и поместите 3 автомобиля в окно одновременно.

С помощью команды `pygame.display.set_caption ('автомобиль')` устанавливаем заголовок окна.

Функцию `pygame.transform.scale` размещаем после загрузки изображения. Ниже приведен фрагмент программы.

```
avto = pygame.image.load('avto.png')
```

`avto = pygame.transform.scale(avto, (120, 80))` – загруженный спрайт имеет длину 120, ширину 80 (код 2). Когда программа запускается, на игровом поле появляются 3 машины (рис. 3).

#### Код 2

```
import pygame, sys
pygame.init()
screen = pygame.display.set_mode([300,300])
pygame.display.set_caption('автомобиль')
screen.fill([255, 255, 255])
avto = pygame.image.load('avto.png')
avto = pygame.transform.scale(avto, (160, 80))
screen.blit(avto,[50, 20])
screen.blit(avto,[50, 100])
screen.blit(avto,[50, 180])
pygame.display.flip()
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
pygame.quit()
```



Рис. 3



## Анализ



Проанализируйте фоновое изображение, спрайт автомобиля и программный код (код 3), который отражает сюжет игры проекта «Путешествие по дороге». Запишите пояснение к каждой из команд в программе.

- Игровое окно, появляющееся при запуске программы (*рис. 4*).
- Фон игры – doroga.png (*рис. 5*).
- Изображение автомобиля – avto2.png (*рис. 6*).



Рис. 4

## Код 3

```
import pygame; pygame.init()
screen = pygame.display.set_mode([400,200])
pygame.display.set_caption('Путешествие по
дороге')
screen.fill([255, 255, 255])
avto = pygame.image.load('avto2.png')
jol = pygame.image.load(doroga.png')
jol = pygame.transform.scale(jol, (400, 200))
screen.blit(jol,[0, 0])
avto = pygame.transform.scale(avto, (200,
100))
screen.blit(avto,[100, 80]); pygame.display.
flip()
running = True
while running:
for event in pygame.event.get():
if event.type == pygame.QUIT:
running = False
pygame.quit()
```



Рис. 5. doroga.png



Рис. 6. avto2.png

**Вопросы**

1. Что такое спрайт?
2. Какую команду вы должны использовать, чтобы задать заголовок окну игры?
3. Какую команду нужно использовать, чтобы задать размер изображения в игровом окне?

**Домашнее задание**

Выполните анализ программного кода проекта «Гонки», приведенного в примере 2. Какие изменения необходимо внести в программный код, чтобы выполнить следующие изменения в проекте:

Если все 3 автомобиля будут разного цвета в этом проекте;

Если увеличить размер второй машины по сравнению с другими вдвое.

## 5.4

## АНИМИРОВАНИЕ ПЕРСОНАЖЕЙ



Как можно создать движение персонажей в игровом поле PyGame?



## Подумай

- Что такое анимация?
- Как оживляли героев мультфильмов до появления компьютера?
- В каких программах вы создавали анимации?
- Как создать движение героев компьютерной игры?



## Новые знания

В Pygame мы научились создавать игровое окно, задний фон игры, вставлять игровых персонажей. Теперь остановимся на основной задаче, стоящей при разработке игры, – научить персонажей двигаться (*рис. 1*).



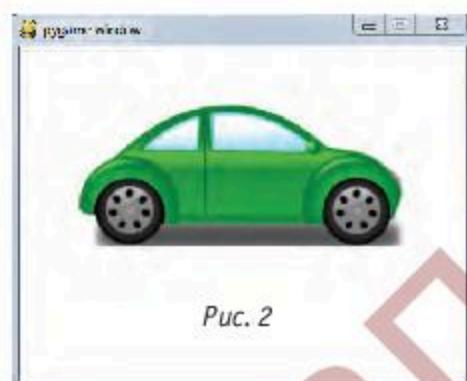
*Рис. 1*

**Анимация(Animation)** – это произведение последовательности картинок, создающее впечатление движущегося изображения.

Под компьютерной анимацией понимается перемещение изображений (групп пикселов) с одного места на другое.

Давайте рассмотрим перемещение легкового автомобиля (*рис. 2*), размещенного в окне игры в предыдущей теме, чтобы анимировать (оживить) игрового персонажа в Pygame. Чтобы создать движение автомобиля, представленного на рисунке 2, переместите автомобиль

на 200 пикселей вправо от его исходного положения. Для этого нужно изменить координату X, которая описывает положение автомобиля по горизонтали, на 200. Программный код (код 1) и результат выполнения передвижения автомобиля показаны на рисунке 3. Используемая в программе функция `pygame.time.delay(K)` останавливает выполнение программы на K миллисекунд.



### Код 1

```
import pygame; pygame.init()
screen = pygame.display.set_mode([400,360])
screen.fill([255, 255, 255])
#Задание название игры
pygame.display.set_caption('Анимация')
#Загрузка изображений
avto = pygame.image.load('avto.png')
#Задает размеры изображения в игре
avto=pygame.transform.scale(avto,(180, 100))
#Отображает изображение на экране
screen.blit(avto,[10, 50]); pygame.display.flip()
#Останавливает программу на 1000 миллисекунд pygame.
time.delay(1000)
# Снова отображает изображение на экране на новом месте
screen.blit(avto,[210, 50]); pygame.display.flip()
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
pygame.quit()
```

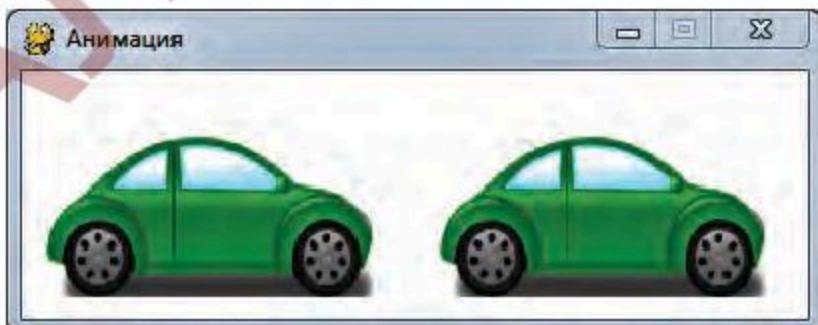


Рис. 3

Как видно на рисунке 3, автомобиль сменил свое место на новое положение через 1000 миллисекунд (1 секунду) после запуска программы. В окне игры появилось одновременно два автомобиля. В компьютерной графике анимация персонажа (графический объект) осуществляется в два этапа:

1. Перемещение персонажа на новую позицию.
2. Удаление персонажа с исходного места.

Рисунок, нарисованный на бумаге можно легко стереть с помощью ластика. Представьте, что вы на бумаге масляными красками нарисовали голубое небо, а потом пририсовали темно-синие облака. Что делать, если нужно стереть облака? В этом случае единственное, что можно сделать, это нарисовать на этом месте небо. В окне компьютерных игр вы можете удалять объекты точно так же, то есть место расположения объекта закрасить цветом заднего фона. Так как фон игрового окна на рисунке 3 белый, то зарисуем первый автомобиль белым цветом. Отредактируем программу 1, добавив следующую строку:

```
pygame.draw.rect(screen, [255,255,255], [10, 30, 170, 110], 0)
```



Рис. 4



Рис. 5

### Изменение кода программы 1

```
import pygame; pygame.init()
screen = pygame.display.set_mode([400,360])
...
#Останавливает программу на 1000 миллисекунд pygame.time.
delay(1000)
# Местоположение первого автомобиля закрашивает в белый цвет
pygame.draw.rect(screen, [255,255,255], [10, 30, 170, 110], 0)
...
pygame.quit()
```



### Анализ

 Какие компьютерные игры сегодня наиболее популярны среди подростков? Проведите анализ. Приведите примеры.



### Синтез



Сделайте вывод о пользе компьютерных игр для развития логического мышления и вреде для человеческого организма чрезмерной игры. Для своих друзей, сверстников, которые любят играть в компьютерные игры, подготовьте и представьте правила игры, при соблюдении которых не будет причинен вред здоровью.



### Оценка

Как, на ваш взгляд, компьютерные игры будут развиваться в ближайшем будущем? Сделайте прогноз. Обоснуйте свое предположение.



### Вопросы

1. Что такое компьютерная анимация?
2. Как выполнить движение персонажа в Pygame? Объясните.
3. Каково назначение функции pygame.time.delay?
4. Какие анимационные мультфильмы, произведенные в Казахстане, вам известны?
5. Какова цель стирания персонажей в окне игры? Как работает команда удаления? Объясните.



### Домашнее задание

Выберите игрового персонажа на интересующую вас тему и создайте программу его движения.



### Задание

#### Проект «Казахский танец»

На рисунке 1 показаны четыре позиции движения мальчика, танцующего казахский танец. Создайте программу в Pygame, как мальчик танцует казахский танец.

**Пояснение к задаче:** при разработке программы необходимо взять за основу команды движения автомобиля (код 1) в рассмотренном проекте. Эти команды должны повторяться по одному разу для каждого из 4 состояний движения. Чтобы повторить движение несколько раз, набор команд, которые выполняют перемещение, должен быть помещен в цикл *for*.



Рис. 6

**5.5****УПРАВЛЕНИЕ ПЕРСОНАЖЕМ С КЛАВИАТУРЫ**

Как управлять действиями персонажа с клавиатуры во время игры?

**Подумай**

- Какие способы управления компьютерными играми вам известны?
- Как вы думаете, какой из этих способов наиболее удобный?
- Как эффективнее управлять персонажами в игре (мышь, джойстик, клавиатура)?
- Что такое событие в повседневной жизни? Что мы относим к событиям?

**Новые знания**

Мы научились перемещать, анимировать персонажей в игре. Теперь у нас возникает вопрос: как управлять движением персонажа? Конечно, есть джойстик – специальное устройство для управления игрой на компьютере. Но это не означает, что это устройство имеется на всех компьютерах, его можно приобрести, хотя это необязательно. Игра может управляться мышью и клавиатурой.

Рассмотрим несколько способов использования клавиатуры в управлении игрой.

**Событие****Что такое событие?**

Под событием мы понимаем текущую ситуацию, то, что имеет место, происходит. То есть когда мы идем, подскользнувшись падаем, разбивается предмет, который мы несли в руках, и так далее. Все это относится к жизненным событиям. Точно так же имеется понятие «событие» в программировании. В качестве события в компьютерных играх можно считать: движение персонажей, столкновение одного героя с другим и т. д. В игре нам нужно реагировать на ситуацию (событие) с помощью мыши или клавиатуры. Программа, которую мы создали во время разработки игры, должна отвечать трем приведенным ниже ситуациям.

**1. Перемещение мыши, ее нажатие – модуль MOUSEMOTION.****2. Нажатие клавиши на клавиатуре – модуль KEYDOWN.****3. Завершение игры в течение указанного периода времени.**

Управление событием программы с помощью клавиатуры.

Давайте рассмотрим примеры управления событием с помощью клавиатуры. Например, когда вы нажимаете какую-либо клавишу, должно происходить некоторое событие. В Pygame за это отвечает модуль KEYDOWN. Давайте рассмотрим пример ниже, чтобы познакомиться с модулем KEYDOWN.

**Пример 1.** Анимация в проекте перемещения автомобиля, рассмотренном в предыдущем разделе, выполнялась автоматически через 1000 миллисекунд. Давайте сделаем то же самое движение, нажав клавишу «а» (код 1).

В библиотеке Pygame имена присвоены каждой клавише. При использовании клавиш в программе они должны начинаться с буквы K\_, за которой следует имя клавиши. Например, в рассматриваемом нами примере нужно использовать K\_a. Все клавиши используются по данному принципу, например:

- K\_b, K\_p, K\_h и т.д. (для клавиш с буквами);
- K\_SPACE – клавиша пробел;
- K\_ESCAPE (ескейп) клавиша;
- K\_UP и K\_DOWN – перемещение вверх и вниз.

**Код 1**

```
Import pygame; pygame.init()
screen = pygame.display.set_mode([380,120])
screen.fill ([255, 255, 255])
pygame.display.set_caption('Нажатие клавиши ')
avto = pygame.image.load('avto.png')
avto=pygame.transform.scale (avto, (160, 80))
screen.blit (avto,[10, 30])
pygame.display.flip()
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.KEYDOWN: #(ожидает нажатие клавиши)
            if event.key == pygame.K_a: #(условие нажатия клавиши a)
                screen.blit(avto,[200, 30])
                pygame.display.flip()
pygame.quit()
```



### Практическая работа

**Пример 2.** Запишите программу (код 2) перемещения автомобиля с помощью клавиши SPACE (ПРОБЕЛ) и возвращения автомобиля назад, когда он упрется в стену (рис. 1).

#### Код 2. Код программы и пояснение

```

Import pygame; pygame.init()
screen = pygame.display.set_mode([380,120]); screen.fill([255, 255,
255])
pygame.display.set_caption('Проект "Движение по нажатию
клавиши" ')
avto = pygame.image.load('avto.png')
x = 20; #(показывает координаты x, у положения автомобиля, с
которого начинается движение)
y = 40;
x_speed = 10#(скорость автомобиля в пикселях)
avto=pygame.transform.scale(avto,(160, 80))
pygame.display.flip()
running = True
while running:
# Осуществляет движение автомобиля без остановки
for event in pygame.event.get()
ifevent.type == pygame.QUIT:
running = False
elif event.type == pygame.KEYDOWN:
ifevent.key == pygame.K_SPACE:#(ждет нажатия клавиши SPACE )
pygame.time.delay(50)
#Стирает первоначальное изображение автомобиля, в случае, если
оно не стирается, будет отображаться рядом или накладываться
pygame.draw.rect(screen, [255,255,255], [x, y, 180, 120], 0)
x = x + x_speed #(увеличивает скорость автомобиля по x на 10)
#(Определяет, когда автомобиль упрется в стену)
if x > screen.get_width() - 170 or x < 0:
#Когда автомобиль упрется в стену, скорость меняется на
противоположное значение
x_speed = - x_speed
pygame.draw.rect(screen, [255,255,255], [x, y, 90, 90], 0)
screen.blit(avto, [x, y])
pygame.display.flip()
pygame.quit()

```

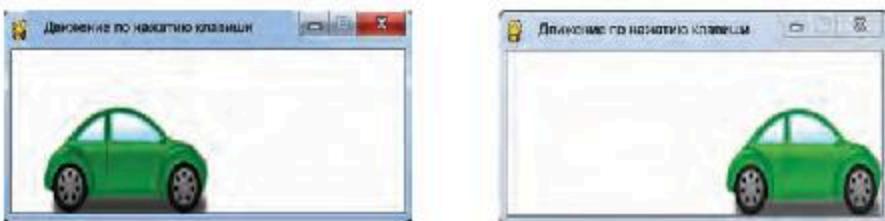


Рис. 6. Исходное местоположение автомобиля и упирание в стену

**Анализ – Синтез**

Проанализируйте программный код в примере 2. Какие изменения необходимо внести в программу, чтобы добавить или изменить некоторые из следующих событий?

1. Вставить мяч вместо автомобиля.
2. Перемещать мяч прямо вверх и вниз по вертикали.
3. Перемещать мяч по диагонали.

**Оценка**

Как вы думаете, насколько клавиатура удобна для управления компьютерными играми? Дайте оценку.

**Вопросы**

1. Что такое событие? Приведите примеры.
2. Какие примеры событий вы можете привести из компьютерных игр?
3. Для чего используется модуль KEYDOWN?
4. Каким условиям должна отвечать игровая компьютерная программа во время игры?

**Домашнее задание**

Выберите игрового персонажа на интересующую вас тему и создайте программу его движения.

**Задания**

**Задание 1.** Какие изменения нужно внести в программу в первом примере для управления автомобилем стрелками вправо или влево?

**Задание 2.** Какие изменения должны быть сделаны в первом примере в программе, чтобы узнать количество упираний автомобиля в стену?

## 5.6

## ОПРЕДЕЛЕНИЕ (РАСПОЗНАВАНИЕ) СТОЛКНОВЕНИЙ СПРАЙТОВ



Как определить столкновение спрайтов для расчета результатов игры в PyGame?



## Подумай

- Какими параметрами наделяются персонажи компьютерных игр? Приведите примеры таких параметров из игр, которые вы знаете.
- Как бы вы предложили определять воздействие одного персонажа на другого или их столкновение на игровой площадке, основываясь на знания, полученные вами на предыдущих занятиях?



## Новые знания

Для игры, разработанной по определенному сценарию, выбор персонажа, его перемещение и вычисление результатов игры – ключевой вопрос для любой игры. В конечной счете в любой игре один из игроков (пользователь и компьютер, пользователь и пользователь) должен выиграть, другой – проиграть, или они должны сыграть вничью, то есть игра должна достичь определенного результата. Если это не так, то нельзя рассматривать эту программу как игру. В предыдущей теме мы познакомились с движением графических объектов (персонажей) на игровой площадке. Теперь давайте рассмотрим пути определения взаимодействий героя с другим персонажем в соответствии с конкретной целью игры. В предыдущих темах мы познакомились с понятием **спрайт** в Pygame. Давайте остановимся более подробно на спрайте. Спрайт обладает двумя базовыми свойствами:

`image` – выводимое спрайтом изображение;

`rect` – прямоугольная область, заключающая в себя спрайт.

В качестве изображения может выступать рисунок, созданный при помощи Pygame – функций или взятый из файла. Модуль `sprite` библиотеки Pygame предоставляет базовый класс `Sprite`. С базовыми классами обычно не работают напрямую. Мы тоже не будем работать напрямую с классом `pygame.sprite.Sprite`, на его основе создадим подкласс `Robots` (код 1).

**Код 1 (фрагмент). Создание подкласса Robots**

```
class Robots(pygame.sprite.Sprite):
    def __init__(self, image_file, location):
        #Инициализирует спрайт.
        pygame.sprite.Sprite.__init__(self)
        #Загружает файл с изображением
        self.image = pygame.image.load(image_file)
        #Формирует прямоугольник, определяющий границы изображения
        self.rect = self.image.get_rect()
        #Задает исходное положение персонажа
        self.rect.left, self.rect.top = location
```

В данном коде переменная **location** представляет собой координаты [x, y], то есть список из двух элементов. В данном случае мы присвоили атрибуты **left** и **top** прямоугольнику, в который заключен наш спрайт.

Мы научились создавать подкласс **Robots**. При необходимости создание нескольких копий этого подкласса больше не вызовет у нас затруднений. Для этого мы будем использовать вложенный цикл **for** (код 2).

**Код 2 (фрагмент). Вывод роботов на экран с помощью циклов**

```
img_file = "robot.png"
robi = []
for row in range(0, 2):
    for column in range(0, 2):
        #Каждому персонажу присваивается новая
        #координата
        location = [column * 180 + 10, row * 180 + 10]
        robot= Robots(img_file, location)
        #Добавляет персонажей в список
        robi.append(robot)
```

Теперь, объединив фрагменты программы, получим полный программный код (код 3). Когда код 3 будет запущен, тогда изображение 4 роботов появится в два ряда (*рис. 1*). То есть можно использовать циклы в программе для отображения нужного количества роботов на экране.

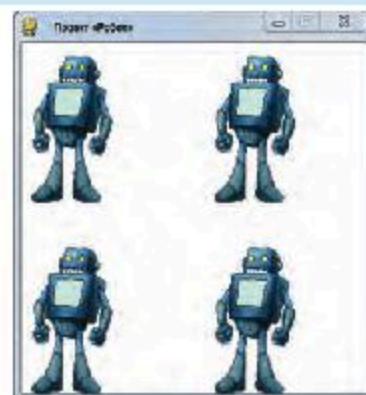


Рис.6

**Код 3**

```

import pygame
class Robots(pygame.sprite.Sprite):
    def __init__(self, image_file, location):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.image.load('robot.png')
        self.image = pygame.transform.scale(self.image, (80, 140))
        self.rect = self.image.get_rect()
        self.rect.left, self.rect.top = location
size = width, height = 640, 480
screen = pygame.display.set_mode(size)
pygame.display.set_caption('Проект «Робот»')
screen.fill([255, 255, 255]); robi = []; img_file = "robot.png"
for row in range(0, 2):
    for column in range(0, 2):
        location = [column * 180 + 10, row * 180 + 10]
        robot = Robots(img_file, location)
        robi.append(robot)
for robot in robi:
    screen.blit(robot.image, robot.rect)
pygame.display.flip()
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
pygame.quit()

```

Программа, представленная в коде 3, показывает 4 робота на экране без движения. Что нужно сделать, чтобы все роботы начали двигаться? Для этого воспользуемся методом `move()`. Мы создали роботов, как копии класса `Robots`. Давайте переместим роботов с помощью того же класса и посмотрим на пример фрагмента кода ниже.

**Код 4 (фрагмент). Создание методом `move()` класса**

```

def move(self):
    self.rect = self.rect.move(self.speed)
    if self.rect.left < 0 or self.rect.right > width:
        self.speed[0] = -self.speed[0]
    if self.rect.top < 0 or self.rect.bottom > height:
        self.speed[1] = -self.speed[1]

```

В библиотеке Pygame `sprite` (спрайты, точнее, связанные с ними прямоугольники) обладают встроенным методом `move()`. В этот метод

передается параметр **speed**, определяющий скорость перемещения объекта. Так как мы собираемся создавать двухмерную графику, параметр **speed** будет представлен списком из двух чисел – скоростями по координатам **x** и **y**. Кроме того, требуется проверка столкновений робота с границами окна. Добавим параметр **speed** методом **move()** в класс **Robots** кода 3.

### Код 5. Проект взаимодействия роботов (рис. 2)

```
from random import*
class Robots(pygame.sprite.Sprite):
    def __init__(self, image_file, location, speed):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.image.load('robot.png')
        self.image=pygame.transform.scale(self.image, (80, 140))
        self.rect = self.image.get_rect()
        self.rect.left, self.rect.top = location
        self.speed = speed
    def move(self):
        self.rect = self.rect.move(self.speed)
        if self.rect.left < 0 or self.rect.right > width:
            self.speed[0] = -self.speed[0]
        if self.rect.top < 0 or self.rect.bottom > height:
            self.speed[1] = -self.speed[1]
def animate(group):
    screen.fill([255,255,255])
    for robot in group:
        robot.move()
    for robot in group:
        group.remove(robot)
        if pygame.sprite.spritecollide(robot, group, False):
            robot.speed[0] = -robot.speed[0]
            robot.speed[1] = -robot.speed[1]
            group.add(robot)
        screen.blit(robot.image, robot.rect)
    pygame.display.flip()
size = width, height = 460, 380
screen = pygame.display.set_mode(size);
pygame.display.set_caption('Проект «Столкновение роботов»')
screen.fill([255, 255, 255])
img_file = "robot.png"
clock = pygame.time.Clock()
group = pygame.sprite.Group()
```

**Атрибут скорости был добавлен в класс Робот.**

**Инициализация метода move( ).**

**animate(group).** В библиотеке Pygame можно разбить спрайты на группы. С помощью этого метода вы можете определить столкновение одной группы спрайтов со второй группой спрайтов. Робот меняет скорость на обратное значение при взаимодействии со вторым роботом. Столкновение спрайтов определяется функцией *animate( )*.

**Загружается класс Clock.**

```

for row in range(0, 2):
    for column in range(0, 2):
        location = [column * 180 + 10, row * 180 + 10]
        speed = [choice([-4, 4]), choice([-4, 4])]
        robot = Robots(img_file, location, speed)
        group.add(robot)
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
            frame_rate = clock.get_fps()
            print("frame rate = "), frame_rate
animate(group)
clock.tick(30)
pygame.quit()

```

*Задает место и скорость работы, добавляет в группу.*

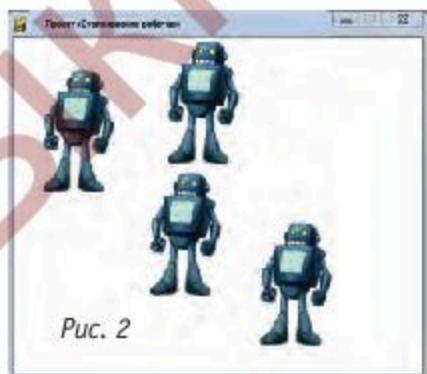
*Функция clock.tick() управляет скоростью движения кадров. Чем больше это число, тем выше скорость движения роботов на экране.*

В строке `def __init__(self, image_file, location, speed)` в отличие от предыдущего программного кода добавлена функция скорости `speed`. Ниже представлена строка программного кода, в которой вместе с исходной позицией робота задается и его скорость

```

speed = [choice([-2, 2]), choice([-2, 2])]
robot = Robots(img_file, location,
speed)

```



Pygame – модуль `sprite` обладает функцией `spritecollide()`, которая определяет столкновение одного спрайта с любым спрайтом из группы. Проверка столкновения спрайтов внутри группы происходит в три этапа.

1. Удаляем спрайт из группы.
2. Проверяем, не сталкивается ли этот спрайт с остальными спра-тами группы.
3. Возвращаем спрайт в группу.

Цикл `for` во фрагменте `def animate(group)` кода 5 выполняет эту задачу.



### Анализ



Проанализируйте и обсудите проект «Столкновение роботов». Определите функциональность каждого блока программы и запишите их.

**Вопросы**

1. Какими свойствами обладает спрайт?
2. Как создается класс Robots в коде 1? Объясните.
3. Каково назначение функции animate ()?
4. Из скольки шагов состоит алгоритм определения столкновения спрайтов?

**Задания**

1. Какие изменения должны быть внесены в программный код 5, чтобы заменить робота на воздушного змея?
2. Определите назначение функций и параметров, использованных в коде 5. Для уточнения воспользуйтесь сетью Интернет. Запишите их назначение в таблицу 1.

<b>№</b>	<b>Название функции или параметра</b>	<b>Назначение</b>
1	pygame.time.Clock	
2		
3		

**5.7****ПРОГРАММИРОВАНИЕ УСЛОВИЙ. ИГРА «ФУТБОЛИСТ»**

Как можно по готовому сценарию разработать игру в PyGame и подсчитать результаты игры?

**Подумай**

- Как вы думаете, каким образом выполняется подсчет очков, накопленных в игре персонажем, и как одушевляется персонаж?
- Как можно определить переход от нижнего этапа игры к высшему этапу, конец игры и подсчет очков?

**Новые знания**

Давайте создадим простую игру для систематизации знаний, которые мы получили при изучении предыдущих тем, и назовем ее «Футболист».

**Игра «Футболист»****Цель игры:**

На игровом поле игрок должен не дать мячу упасть и вернуть мяч головой, по возможности собрать максимальное количество очков.

**«Контроль над жизнью» игрока:**

Продолжительность игры. Это количество попыток, при которых игроку давалась возможность вернуть мяч.

**Подсчет количества очков:**

За каждый возвращенный мяч игрок получает одно очко.

**Завершение игры:**

Игра заканчивается, когда игрок допускает падение последнего мяча.

**Сообщение о результатах игры:**

Сообщите об окончании игры и количестве очков, полученных игроком.

**Основные этапы создания игры «Футболист».**

Нам нужны спрайты игрока и мяча, чтобы создать игру. Мы выбираем изображение футболиста для спрайта футболиста, изображение футбольного мяча для спрайта мяча (Рис. 1 и 2). При создании

копии мяча, нужно определить первоначальное положение мяча и скорость полета мяча. Кроме того, нужно создать группу, которая поможет определять взаимодействие игрока с мячом.



Рис. 1. Футболист



Рис. 2. Футбольный мяч

Во время игры мы должны отслеживать две вещи: количество жизней футболиста и количество очков. Мы будем начислять одно очко за каждое касание мячом верхней границы окна при каждой игре. Футболисту дается три «жизни». Это означает, что у футболиста есть три возможности набрать максимальное количество очков. Нам также нужно показывать количество очков на экране. В библиотеке Pygame за это отвечает модуль font. Для использования модуля font нужно:

- 1) в модуле font создать объект, указав гарнитуру и размер шрифта;
- 2) в объект, созданный модулем font, размещаем текст;
- 3) скопируйте текстовый объект в игровое окно.



### Практическая работа

На рисунках 3 и 4 показано игровое окно игры «Футболист». В программном коде игры показано назначение каждого блока (код 1).



Рис. 3. Процесс игры



Рис. 4. Конец игры

**Код 1. Программный код игры «Футболист»**

```

import pygame, sys
#Определяется класс для мяча, создается спрайт для мяча
class MyBallClass(pygame.sprite.Sprite):
    def __init__(self, image_file, speed, location):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.image.load('football.png')
#Задаются размеры мяча (30 x 30 пиксель)
        self.image=pygame.transform.scale(self.image, (30, 30))
        self.rect = self.image.get_rect()
        self.rect.left, self.rect.top = location
        self.speed = speed
# Блок движения мяча
    def move(self):
        global score, score_surf, score_font
        self.rect = self.rect.move(self.speed)
#Мяч меняет скорость на обратное значение, когда сталкивается со
стенами.
        if self.rect.left < 0 or self.rect.right > screen.get_width():
            self.speed[0] = -self.speed[0]
#Каждый раз, когда мяч касается верхней границы стены,
добавляется 1 очко
        if self.rect.top <= 0 :
            self.speed[1] = -self.speed[1]
            score = score + 1
#Записывает изменение значения количества очков с помощью
строковой функции str.
        score_surf = score_font.render(str(score), 1, (0, 0, 0))
#Представляет спрайт и класс футболиста
class MyPaddleClass(pygame.sprite.Sprite):
    def __init__(self, location = [0,0]):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.image.load('footballer1.png')
#Задает размеры игрока на игровой площадке (150 x 220 пиксель)
        self.image=pygame.transform.scale(self.image, (150, 220))
        self.rect = self.image.get_rect()
        self.rect.left, self.rect.top = location
pygame.init()
screen = pygame.display.set_mode([460,320])
pygame.display.set_caption('Проект «Футболист»')
clock = pygame.time.Clock()
myBall = MyBallClass('footballer.png', [10,20], [20, 20])
ballGroup = pygame.sprite.Group(myBall)

```

```

paddle = MyPaddleClass([270, 150])
#Дает игроку «жизнь» (выдается 3 мяча за одну игру)
lives = 3; score = 0
score_font = pygame.font.Font(None, 50)
#Задается текстовый объект
score_surf = score_font.render(str(score), 1, (0, 0, 0))
score_pos = [10, 10]; done = False; running = True
#Цикл, запускающий основную часть программы
while running:
    clock.tick(30)
    #В качестве заднего фона игры загружается футбольное поле pole_
    surf = pygame.image.load('pole.png')
    pole_rect = pole_surf.get_rect(center=(230, 160))
    screen.blit(pole_surf, pole_rect)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
    #Игра управляется с помощью мыши (можно заменить на клавиши)
    elif event.type == pygame.MOUSEMOTION:
        paddle.rect.centerx = event.pos[0]
    #Определяет взаимодействие футболиста и мяча
    if pygame.sprite.spritecollide(paddle, ballGroup, False):
        myBall.speed[1] = -myBall.speed[1]
    #Задается движение мяча
    myBall.move()
    if not done:
        screen.blit(myBall.image, myBall.rect)
        screen.blit(paddle.image, paddle.rect)
        screen.blit(score_surf, score_pos)
        for i in range(lives):
            width = screen.get_width()
            screen.blit(myBall.image, [width - 40 * i, 20])
    pygame.display.flip()
    #Если мяч упадет на поле, то удаляется один мяч («жизнь»)
    if myBall.rect.top >= screen.get_rect().bottom:
        lives = lives - 1
    if lives == 0:
        #Записывает результат игры в окно игры
        final_text1 = "Конец игры"
        final_text2 = "Количество очков: " + str(score)
        ft1_font = pygame.font.Font(None, 70)

```

```

ft1_surf = ft1_font.render(final_text1, 1, (255,0,0))
ft2_font = pygame.font.Font(None, 50)
ft2_surf = ft2_font.render(final_text2, 1, (255,0,0))
screen.blit(ft1_surf, [screen.get_width()/2 - \
                      ft1_surf.get_width()/2, 100])
screen.blit(ft2_surf, [screen.get_width()/2 - \
                      ft2_surf.get_width()/2, 200])
pygame.display.flip()
done = True
else:
#1 секунду начинается следующая «жизнь» футболиста
    pygame.time.delay(1000)
myBall.rect.topleft = [50, 50]
pygame.quit()

```



### Анализ



Проанализируйте программный код игры «Футболист». Объясните функцию каждого блока программы.



### Синтез



Какую игру можно создать на основе игры «Футболист»? Какие изменения для этого нужно внести в игровых персонажей и код программы и какая игра может получиться?



### Оценка

Оцените возможности создания игры в Pygame. Что вам понравилось в Pygame? Что было легко? Что было сложно? С помощью этих вопросов обобщите свои мысли.



### Домашнее задание

Выберите одну из ваших любимых тем и создайте простой игровой проект. В качестве примера можно взять программный код игры «Футболист».

<b>ТЕРМИНОЛОГИЯ – TERMINOLOGY</b>		
<b>Свойства информации – ақпараттың қасиеттері – the properties of information</b>		
информация	ақпарат	information
Источник информации	Ақпарат көзі	The source of information
Принятие информации	Ақпаратты қабылдау	Acceptance of information
Видеть	Көрү	To see
чувствовать	сезу	feeling
слышать	есту	hearing
надежный	сенімді	reliable
полный	толық	complete
фактический	әзекті	actual
ценность	құндылық	value
объективно	объективті	objectively
понятность	түсініктілік	understanding
<b>Совместная работа с документами – құжаттармен бірлескен жұмыс – collaborative work with documents</b>		
документ	құжат	document
Облачная технология	Бұлтты технология	Cloud technology
расчет	есептеу	computing
Частные облака	Жеке бұлттар	Private clouds
Общие облака	Жалпы бұлттар	Public clouds
Смешанные облака	Аралас бұлттар	Mixed clouds
Облака сообщества	Қоғамдық бұлттар	Public clouds
Облачное хранилище	Бұлттық қойма	Cloud storage
<b>Использование службы обработки документов для совместной работы на Google Drive – Google Drive-та құжатты бірлесе өндөу қызметін пайдалану – use document processing service collaboration in google drive</b>		
создание информации	ақпаратты құру	make the information
Изменить информацию	Ақпаратты өзгерту	Change information
Сохранить информацию	Ақпаратты сақтау	Save information
сервис	сервис	service
блог	блог	blog
Социальная сеть	Әлеуметтік желі	Social network

Хранение документов	Құжаттар қоймасы	Document storage
публиковать	жариялау	publication
адрес	мекен-жай	address
аккаунт	аккаунт	account

**Сетевой этикет – желілік этикет – network etiquette**

Виртуальная связь	Виртуалды қарым-қатынас	Virtual communication
Сетевое общение	Желілік қарым-қатынас	Network communication
мессенджер	мессенджер	messenger
чат	чат	chat
форум	форум	forum
электронная почта	электронды пошта	email
телеконференция	телеконференция	teleconference
флайм	флайм	flame
флуд	флуд	flooding
смайлик	смайлик	smiley

**Конфигурация компьютера – компьютердің конфигурациясы – computer configuration**

Системный блок	Жүйелік блок	System blog
монитор	монитор	monitor
клавиатура	пернетақта	keyboard
мышка	тінтуір	mouse
Размер экрана	Экранның мөлшері	Screen size
дюйм	дюйм	inch
Процессор	Процессор	CPU
Оперативная память	Жедел жад	Emergency memory
Постоянная память	Тұрақты жад	Constant memory
шина	шина	tire
слот	слот	slot
сканер	сканер	scanner
микрофон	микрофон	microphone
веб-камера	веб-камера	webcam
Принтер	Принтер	Printer

Плоттер	Плоттер	Plotter
динамик	динамик	loudspeaker
наушники	құлаққап	headphones

Дисковод	Дискжетек	Disk drive
Жесткий диск	Қатқыл диск	Hard drive
модем	модем	modem

**Выбор программного обеспечения – программалық қамтамасыз етуді таңдау – software selection**

программа	программа	program
программное обеспечение	программалық қамтамасыздандыру	software
Аппаратное программное обеспечение	Аппараттық программалық жабдықтама	Hardware software
системное программное обеспечение	Жүйелік программалық жабдықтама	System software
Прикладное программное обеспечение	Қолданбалы программалық жабдықтама	Applied software
Бесплатное программное обеспечение	Еркін қолданыстағы программалық жабдықтар	Free software
Условно-бесплатное программное обеспечение	Шектеулі-тегін программалық жабдықтар	Limited-free software
Коммерческое программное обеспечение	Коммерциялық программалық жабдықтар	Commercial software equipment

**Отчет об оценке стоимости компьютера – компьютер бағасының есебі – computer costing evaluation report**

Тактовая частота	Тактілік жиілік	Tactical frequency
Объем оперативной памяти	Оперативті жады көлемі	Volume operational memory
видеокарта	видеокарта	videocard
материнская плата	аналық тақша	motherboard
Объем и скорость жесткого диска	Қатқыл дисктің көлемі және жылдамдығы	Volume and speed of hard disk
Операционная система	Операциялық жүйе	Operating system
Оптический привод	Оптикалық диск жетегі	Optical drive

<b>База данных – мәліметтер базасы – Data basa</b>		
Поиск информации	Ақпаратты іздеу	Search for information
Фактография	Фактография	Factography
централизованная	орталықтандырылған	centralized
распределенный	таратылған	distributed
Таблица	Кесте	Table
реляционный	реляциалық	relational
Иерархия	Иерархия	Hierarchy
Объект	Нысан	Object
атрибут	атрибут	attribute
запись	жазба	recording
поле	өріс	field
Список	Тізім	List
диапазон	ауқым	range
База данных	Мәліметтер қоры	Data basa
электронная таблица	электронды кесте	spreadsheet
сортировка	сұрыптау	sorting
Порядок по возрастанию	Өсу реті	Growth order
Порядок по убыванию	Кему реті	Reduction order
упорядочивание	реттеу	regulation
фильтрация	сүзгілеу	filtering
Расширенный фильтр	Кеңейтілген сүзгі	Extended filter
Числовые фильтры	Сандық сүзгі	Numeric filters

**Массив данных – деректер ауқымы – data range**

массив	массив	array
индекс	индекс	index
тип	тип	type
код	код	code
ввод	енгізу	introduction
вывод	шығару	output
Однотипные данные	Біртипті деректер	Individual data
компонент	компонент	component
свойства	қасиет	property
программирование	программау	programming

проект	жоба	project
Элемент массива	Массив элементі	Array element
измерение	өлшем	measurement
перестановка	орын ауыстыру	replacement
форма	пішін	form

цикл	цикл	cycle
Сортировка пузырьком	Көбік сұрыптау	Foam sorting
выбор	тандай	selection
Выстрая сортировка	Жылдам сұрыптау	Quick sorting
Выключить	Өшіру	Switch off
удалить	жою	delete

## Глоссарий

**Алгоритм** – пошаговое описание решения задачи, ведущее к получению верного однозначного результата, выполненное на одном из алгоритмических языков.

**Аргументами** называются значения, которые при использовании функций указываются в скобках.

**Атрибут (латинский – attributio)** – запись об определенных параметрах каждого объекта базы данных.

**База данных** – совокупность большого объема данных, систематизированных по определенным свойствам.

**Биоритмология** – наука, которая изучает активность и пассивность процессов человеческого организма в периодической зависимости.

**Блоком (block) кода** называют объединенные друг с другом строки.

**Ввод и вывод в программировании** – это процесс обмена информацией между оперативной памятью компьютера и внешними источниками (файл, клавиатура, сеть, выходные данные из другой программы).

**Веб-форум** – платформа для общения между пользователями интернета на одну тему или на несколько тем.

**Видеокарта** – плата, через которую монитор подключается к компьютеру.

**Двоичный (бинарный) поиск** – классический алгоритм поиска элемента в отсортированном массиве.

**Дескриптор**(лат. describere описывать) служит для описания основного смыслового содержания документа (текста).

**Диапазон** – это таблица, которая выполняет функцию базы данных и расположена в рабочей области листа Excel.

**Жесткий диск** – внешняя память компьютера, предназначенная для постоянного хранения данных, программ операционной системы и часто используемых пакетов программ.

**Запись** представляет собой строку таблицы, содержащую набор значений, описанных в полях базы данных.

**Запрос** – это специальные структуры, предназначенные для обработки данных базы.

**Индекс** – функция, которая возвращает значение или ссылку на значение из таблицы или поименованного диапазона, заданного номером строки и номером столбца.

**Инициализация** (от англ. initialization – инициирование) – создание, активация, подготовка к работе, определение параметров.

**Инициализация игры** – настройка Ригаме-окна, загрузка изображений и присвоение переменным начальных значений.

**Интернет** – это глобальная компьютерная сеть, в которой локальные и региональные соединены между собой многочисленными каналами передачи информации с высокой пропускной способностью.

**Интерфейс** – система связей, посредством которых устройства компьютера взаимодействуют между собой, а также набор средств диалога, взаимодействия между пользователем и компьютером.

**Информационный проект** – посвящен сбору информации об определенном объекте или процессе.

**Информация** – сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии.

**Исследовательский проект** – предложение идей, определив актуальность выбранной темы, выполнение проекта и проведение анализа полученных результатов.

**Ключевое поле** – поле, которое однозначно идентифицирует запись в таблице.

**Команда** – приказ исполнителю на выполнение действий из указанного конечного набора.

**Компьютерная анимация** – перемещение изображений (групп пикселов) с одного места на другое.

**Конфигурация** – компоновка системы с четким определением характера, количества, взаимосвязей и основных характеристик ее функциональных элементов; совокупность аппаратных средств и соединений между ними.

**Линейный алгоритм (line aralgorithm)** – алгоритм, команды которого выполняются строго одна за другой (линейно) и каждая команда выполняется только один раз.

**Массив** (с франц. *massif* – мощный, сплошной) – последовательность идентичных данных, элементы которой разными индексами объединены в общее имя.

**Материнская плата** (от англ. *motherboard*) – сложная многослойная печатная плата, являющаяся основой построения вычислительной системы (компьютера).

**Моделирование** (лат. *simulation*) – система создания аналогов для изучения и анализа свойств и характеристик явлений, процессов или объектов.

**Моделированием** (*simulation*) называется любая компьютерная имитация явлений реального мира.

**Модель** (лат. *modulus* – измерение) – в общем виде физический или информационный аналог, соответствующий по определенным параметрам фактическому объекту.

**Монитор** – устройство визуального представления данных.

**Монитор** – устройство, предназначенное для вывода информации на экран. Внешне он похож на экран цветного телевизора.

**Облачная технология** – современное средство для реализации множества задач бизнеса от хранения до обработки информации.

**Облачные вычисления** – предоставление вычислительных служб (серверов, хранилища, баз данных, сетевого оборудования, программного обеспечения и т.д.) через Интернет (так называемое «облако»).

**Облачный сервис** – это услуга интернет-компаний по предоставлению в распоряжение пользователя облачного хранилища данных.

**Объект** – определенная часть окружающей нас реальной действительности (предмет, процесс, явление) или это любая информация, которая с помощью специальных средств может быть включена в документ. Это могут быть тексты, графические изображения, таблицы и многое другое.

**Одномерный массив** – это фиксированное количество элементов одного и того же типа, объединенных одним именем, где каждый элемент имеет свой номер.

**Окно** – ограниченная рамкой часть экрана, с помощью которой обеспечивается взаимодействие программы с пользователем.

**Оператор (operator)** – законченное выражение на языке высокого уровня, предписывающее процессору выполнение некоторого действия.

**Операционная система** – совокупность программных средств, обеспечивающая управление аппаратным и программным обеспечением.

**Оптический привод** – устройство для считывания данных с оптического носителя.

**Память** – физическая система с большим числом возможных устойчивых состояний, служащая для хранения данных. Память ЭВМ можно разделить на внутреннюю (оперативную) память, регистры процессора и внешнюю память.

**Персональный компьютер** – это настольная или переносная ЭВМ, удовлетворяющая требованиям общедоступности и универсальности применения.

**Поле** – это столбец таблицы, отражающий определенные свойства объекта.

**Программа** – набор указаний на языке машины для выполнения и сохраненная на магнитном носителе (диске) в виде файла, загружаемая по команде человека в память компьютера.

**Программное обеспечение** – совокупность программы соответствующей технической документации, предназначенных для исполнения на компьютере при решении определенных задач.

**Проект** – это совокупность действий, при проведении исследования на определенную тему, которая в результате приводит к созданию реального объекта, разного рода теоретического продукта.

**Процессор** – основное устройство ЭВМ, предназначено для выполнения вычислений по хранящейся в запоминающем устройстве программе и обеспечения общего управления ЭВМ.

**Распознавание столкновений** – получение информации о соприкосновении или перекрывании двух спрайтов.

**Редактирование** – изменение существующего документа.

**Сетевой этикет** – это система правил, созданная людьми для общения друг с другом в сети Интернет.

**Система счисления** – это совокупность правил для обозначения и наименования чисел.

**Системный блок** – основной узел компьютера, внутри которого установлены наиболее важные компоненты: материнская плата с процессором, жесткий диск, дисковод гибких дисков, дисковод компакт-дисков.

**Смайлики** – это простые «картинки». С помощью смайликов пользователи выражают свои эмоции.

**Сортировка** – это процесс расстановки элементов массива в порядке их возрастания, или убывания.

**Список** – стандартный элемент диалоговых окон, который позволяет выбирать из приведенного перечня один или несколько вариантов.

**Спрайт (графический объект)** – это группа пикселов, которые двигаются и выводятся на экран как единое целое.

**Столкновение** – момент, когда два движущихся объекта натыкаются друг на друга.

**Счетные циклы (countingloops)** – циклы, повторяющиеся определенное число раз.

**Таймер** генерирует событие через определенные интервалы времени.

**Творческий проект** включает в себя такие проекты как литературные вечера, спектакли, наблюдения по конкретной теме, организация воздухоплавания, выставка шедевров ремесленников, фотоколлажей.

**Телеконференция** – общение группы людей по объединяющей их теме.

**Условные циклы (conditionalloops)** – циклы, повторяющиеся до наступления определенного события.

**Файл** – логически связанная последовательность данных одного типа, имеющая имя.

**Фильтрация** – это способ найти данные в списке в соответствии с конкретными условиями.

**Флэйм (flames)** – отправка смайликов, позитивного посыла, невзирая на настроение участника форума.

**Флуд** – бессмысленный поток сообщений. Эти сообщения не несут смысловой нагрузки и могут быть удалены.

**Форма** – это объект, с помощью которого в базу вводят новые данные или просматривают имеющиеся.

**Чат** представляет собой службу текстовых сообщений в режиме реального времени, позволяющую многим пользователям одновременно общаться друг с другом.

**Электронная почта** – инструмент быстрой связи. Обмен письмами по электронной почте должен осуществляться в определенных рамках правил вежливости.

**Электронная таблица (spreadsheet)** – интерактивная система обработки данных, заданных в виде таблицы.

**Ячейка** – основная единица хранения данных.

### Список литературы

1. Атре, Ш. Структурный подход к организации баз данных / Ш. Атре. – М.: Финансы и статистика, 2010. – 317 с.
2. Булавин, Л.А. Компьютерное моделирование физических систем: Учебное пособие / Л.А. Булавин, Н.В. Выгорницкий, Н.И. Лебовка.: Интеллект, 2011. – 352 с.
3. Гартман Т.Н. Практическое руководство по решению некоторых вычислительных задач с использованием Microsoft Excel. Учебное пособие. – М.: РХТУ, 2006. – 68 с.
4. Глушаков, С.В. Базы данных / С.В. Глушаков, Д.В. Ломотько. – М.: Харьков: Фолио, 2000. – 504 с.
5. Грацианова Т.Ю. Программирование в примерах и задачах. М.: Лаборатория знаний, 2016. – 373 с.
6. Грэй, П. Логика, алгебра и базы данных / П. Грэй. – М.: Машиностроение, 2015. – 368 с.
7. Златопольский Д. М. Сборник задач по программированию. – 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2011. – 304 с.
8. Леонтьев В. Знакомство с компьютером. Устройство, выбор, конфигурация, Олма Медиа Групп, 2009
9. Седжвик, Роберт, Уайн, Кевин, Дондеро, Роберт. C28 Программирование на языке Python; учебный курс. : Пер. с англ. - СПб.: ООО "Альфа-книга": 2017. – 736 с. : ил. - Парал. тит. англ. ISBN978-5-9908462-1-0 (рус.)
- 10.Хахаев И. А. Практикум по алгоритмизации и программированию на Python: / И. А. Хахаев \_М.:АльтЛинукс, 2010. \_ 126 с.:ил.\_ (Библиотека ALT Linux).
11. У. Сэнд, К. Сэнд. Hello World! Занимательное программирование. – СПб.: Питер, 2016. – 400 с.:ил. – (Серия «Вы и ваш ребенок»).

## СОДЕРЖАНИЕ

Предисловие .....	3
-------------------	---

### **Раздел I. Работа с информацией**

1.1. Свойства информации .....	4
1.2. Совместная работа с документами .....	8
1.3. Совместное использование документов на Google Drive .....	13
1.4. Сетевой этикет .....	17

### **Раздел II. Выбираем компьютер**

2.1. Конфигурация компьютера .....	22
2.2. Выбор программного обеспечения.....	27
2.3. Выполнение проекта «Расчет стоимости компьютера» .....	33

### **Раздел III. База данных**

3.1. База данных .....	37
3.2. Создание базы данных в электронных таблицах MS Excel .....	41
3.3. Методы поиска информации в электронных таблицах MS Excel.....	46
3.4. Сортировка данных.....	52
3.5. Фильтрация данных .....	57
3.6. Фильтрация данных расширенным фильтром .....	62
3.7–3.8. Работа с базой данных (мини-проект) .....	67

### **Раздел IV. Массив данных**

4.1. Одномерный массив .....	71
4.2. Ввод и вывод данных в одномерном массиве .....	76
4.3. Поиск элемента с заданными свойствами .....	80
4.4. Решение задач на одномерные массивы. Практическая работа .....	86
4.5. Перестановка элементов .....	90
4.6. Сортировка .....	96
4.7. Удаление и вставка элемента .....	103

4.8. Двумерный массив .....	109
4.9. Основные параметры двумерного массива .....	114
4.10. Творческо-практическая работа по использованию одномерных и двумерных массивов .....	120

## Раздел V. Создание 2D игры на языке программирования Python

5.1. PyGame. PyGame Библиотека .....	124
5.2. Задний фон и персонажи игры .....	128
5.3. Выбор персонажей для игры .....	132
5.4. Анимирование персонажей .....	137
5.5. Управление персонажем с клавиатуры .....	141
5.6. Определение (распознавание) столкновений спрайтов .....	145
5.7. Программирование условий. Игра футболист .....	151
Терминология .....	156
Глоссарий .....	161
Список литературы .....	165

Алматык

*Оқулық басылым Учебное издание*

**Кадиркулов Роман Алауович  
Нурмуханбетова Гулира Кенжеевна**

**ИНФОРМАТИКА**

Жалпы білім беретін мектептің  
9-сынып оқушыларына арналған оқулық

**ИНФОРМАТИКА**

Учебник для учащихся 9 класса  
общеобразовательной школы

Спецпредакторы *Б.А. Айтуллина*  
Редакторы *А.А. Альмурсина*

Керкемдеуші редакторы *Т.В. Толыбекова*  
Компьютерде беттеген *Л.В. Макарова*

Спецпредактор *Б.А. Айтуллина*  
Редактор *А.А. Альмурсина*  
Художественный редактор *Т.В. Толыбекова*  
Компьютерная верстка *Л.В. Макаровой*

Басыу 14.03.2018 ж. қол қойылды.  
Пішімі 70x100  $\frac{1}{10}$ . Офсеттік басылым.  
Әріп түрі «DS SchoolBook». Офсеттік қағаз.  
Тапсырыс № 2136.

Подписано в печать 14.03.2018 г.  
Формат 70x100  $\frac{1}{10}$ . Печать офсетная.  
Гарнитура «DS SchoolBook». Бумага офсетная.  
Заказ № 2136

Казақстан Республикасы,  
«Алматықітап баспасы» ЖШС,  
050012, Алматы қаласы, Жамбыл көшесі, 111-үй,  
тел. (727) 250 29 58, факс: (727) 292 81 10.  
e-mail: alkitap@intelsoft.kz

Республика Казахстан,  
ТОО «Алматықітап баспасы»  
050012, г. Алматы, ул. Жамбыла, 111,  
тел. (727) 250 29 58; факс: (727) 292 81 10.  
e-mail: alkitap@intelsoft.kz