



Рамин МАХМУДЗАДЕ
Исмаил САДЫГОВ
Наида ИСАЕВА

ИНФОРМАТИКА

Учебник для 7 класса
общеобразовательной
школы

Утверждено приказом
Министерства образования
Азербайджанской
Республики №1192 от
04.07.2011 г.

B

A

K

I



N

Ə

Ş

R

2011

Информатика – 7. Учебник для общеобразовательной школы.
Махмудзаде Р.А., Садыгов И.Дж., Исаева Н.Р. Баку, “Bakınəşr”, 2011, 128 с.

www.bakineshr.az
ISBN 978-9952-430-06-8

© Министерство образования Азербайджанской Республики, 2011
© “Bakınəşr”, 2011

Формат 70×100¹/₁₆. Офсетная бумага №1. Физический печатный лист 8.
Сдано в печать 10.08.2011. Тираж 8 400. Бесплатно.

О Г Л А В Л Е Н И Е

Введение	4
----------------	---

1. Информационные технологии

1.1.	СФЕРЫ ПРИМЕНЕНИЯ КОМПЬЮТЕРОВ	8
1.2.	КЛАССИФИКАЦИЯ КОМПЬЮТЕРОВ	11
1.3.	ИНФОРМАЦИОННАЯ КУЛЬТУРА	14
1.4.	ВИДЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	17

2. Моделирование

2.1.	МОДЕЛЬ И МОДЕЛИРОВАНИЕ	26
2.2.	ГРАФИЧЕСКИЙ РЕДАКТОР GOOGLE SKETCHUP	29
2.3.	ТЕХНОЛОГИЯ РАБОТЫ С МОДЕЛЯМИ. РЕБРА И ГРАНИ	32
2.4.	СОЗДАНИЕ ТРЕХМЕРНЫХ МОДЕЛЕЙ	36
2.5.	СТИЛИ И ТЕНИ	41
2.6.	ИСПОЛЬЗОВАНИЕ ГОТОВЫХ МОДЕЛЕЙ	44

3. Настольные издательские системы

3.1.	НАСТОЛЬНЫЕ ИЗДАТЕЛЬСКИЕ СИСТЕМЫ	54
3.2.	ИНФОРМАЦИОННЫЙ БЮЛЛЕТЕНЬ	58
3.3.	РАБОТА С ТЕКСТОВЫМИ РАМКАМИ	62
3.4.	ИНФОРМАЦИОННЫЙ БУКЛЕТ	64

4. Алгоритмизация

4.1.	ПОНЯТИЕ АЛГОРИТМА	72
4.2.	ОСНОВНЫЕ АЛГОРИТМИЧЕСКИЕ КОНСТРУКЦИИ. ЛИНЕЙНЫЕ АЛГОРИТМЫ	76
4.3.	АЛГОРИТМЫ С ВЕТВЛЕНИЕМ. НАХОЖДЕНИЕ МАКСИМАЛЬНОГО И МИНИМАЛЬНОГО ЭЛЕМЕНТА	80
4.4.	ЦИКЛИЧЕСКИЕ АЛГОРИТМЫ. СОРТИРОВКА	84

5. Программирование

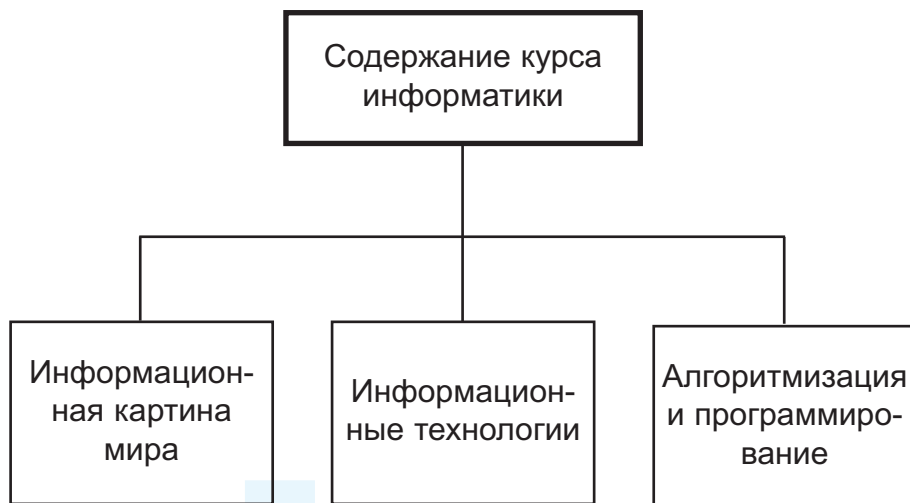
5.1.	ОБЪЕКТЫ И ИХ СВОЙСТВА	94
5.2.	ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ	97
5.3.	СРЕДА ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ LAZARUS	99
5.4.	СОЗДАНИЕ ПЕРВОЙ ПРОГРАММЫ	103
5.5.	СОБЫТИЯ И ИХ ОБРАБОТКА	107
5.6.	ЯЗЫК ПРОГРАММИРОВАНИЯ PASCAL	111
5.7.	ОЧЕНЬ ПОЛЕЗНАЯ ПРОГРАММА	115

Программа	128
-----------------	-----

ВВЕДЕНИЕ

Период, в котором мы живем, считается переходным к информационному обществу. Как и в любом обществе, в информационном обществе каждый человек должен обладать соответствующими знаниями и умениями, которые определяются такими понятиями, как «информационная культура», «компьютерная грамотность». Одной из основных задач школьного курса информатики является формирование информационной культуры и алгоритмического мышления.

Изменения, постоянно происходящие в мире и обществе, требуют усовершенствования преподавания предмета информатики. В международной практике преподавание информатики ведется в основном по трем направлениям:



В предложенном учебнике наряду с разделами, касающихся «компьютерной грамотности», особое место отведено традиционным разделам информатики – алгоритмизации и программированию.

Основы программирования изучаются на примере среды программирования Lazarus, в основе которой лежит язык Pascal. Этот выбор обусловлен тем, что в системах образования многих стран язык Pascal считается основным языком программирования, преподаваемым в учебных заведениях. Еще одна причина в том, что Lazarus является свободно распространяемым бесплатным программным продуктом и авторами проведена его локализация на азербайджанский язык.

Данный учебник составлен в соответствии с современными технологиями обучения на основе программы, утвержденной в 2010 году. Все уроки начинаются с подготовительных вопросов, что способствует активизации учащихся и подготовке их к усвоению нового материала.

В учебнике используются следующие условные обозначения:



– тезисы основных положений урока;



– интересная информация, на которую стоит обратить внимание;



– вопросы, адресованные учащимся при объяснении нового урока.

В конце каждого раздела даны

- вопросы и задания, охватывающие все темы раздела;
- примечательные исторические материалы по теме раздела;
- проверочные тестовые задания;
- проекты исследовательского характера по разным темам;
- проблемные вопросы под рубрикой «Критический взгляд» для формирования у учащихся критического мышления в ходе анализа сути раздела.

Принимая во внимание то, что во время работы на компьютере большое значение имеют понятия «файл» и «папка», информация о них помещена в конце книги как дополнительный материал.

На первых страницах разделов помещены последовательно фрагменты картины «Метаморфозы III» известного голландского художника XX века Маурица Корнелиса Эшера.

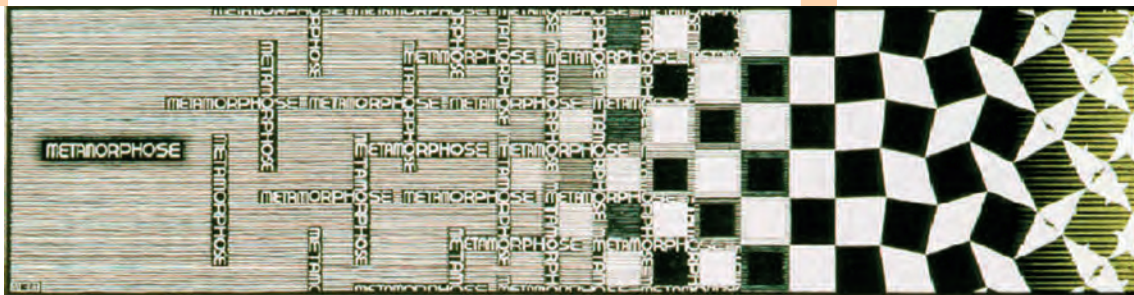
В конце учебника размещен указатель терминов и понятий, которые использованы в этом учебнике.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

1



- Сферы применения компьютеров
- Классификация компьютеров
- Виды программного обеспечения
- Этические нормы, которые следует соблюдать при электронной переписке
- Графический интерфейс пользователя и его преимущества



1.1. СФЕРЫ ПРИМЕНЕНИЯ КОМПЬЮТЕРОВ

Для чего нужны компьютеры и где их используют?

В какой области деятельности человека нет нужды в компьютерах?



Компьютеры стремительно входят во все сферы нашей жизни. И если 15-20 лет тому назад их можно было увидеть только на рабочих местах, сейчас компьютер имеется почти в каждом доме. Сегодня очень сложно представить себе науку, образование, здравоохранение, делопроизводство и другие сферы без компьютеров.

Компьютеры в учреждениях. На первых порах компьютеры стоили очень дорого, и поэтому они устанавливались только в университетах, банках, на крупных предприятиях. По мере того, как стала снижаться стоимость компьютеров, расширился ареал их распространения. В настоящее время даже на мелких предприятиях для написания писем, подготовки докладов, финансовых документов, отчетов пользуются компьютерами.

Процесс производства на крупных заводах также управляется посредством компьютеров. Они контролируют деятельность различных устройств, следят за поддержанием температуры и давления.

Выполнение некоторых работ для человека сопряжено с опасностью, а зачастую и вовсе не представляется возможным. Компьютеры (роботы) эту работу могут выполнять как и люди, а иногда и лучше их. Они могут быть просто незаменимы, например, в условиях сильной жары или лютого мороза. Они могут изготавливать опасные химические препараты, работать в сильно загрязненном воздухе и полнейшей темноте. Кроме того, компьютерам поручают работы, включающие многократно повторяющиеся действия. Они выполняют их аккуратно, не уставая, без тени неудовольствия или признаков усталости.

Компьютеры в торговле. Наверное, все вы видели на товарах в магазине этикетку из черных

Компьютер — это электронное устройство, которое принимает, хранит, обрабатывает информацию различного вида (текстовую, цифровую, графическую, звуковую) и выдает результаты в форме, понятной пользователю.

Существуют разные виды компьютеров. Наиболее широко распространены из них *персональные компьютеры*.

Они состоят из системного блока, монитора, клавиатуры и мыши. Кроме этого к компьютеру могут быть подсоединены принтер, сканер, звуковая система, модем и другие устройства. Устройства, подсоединенные к компьютерной системе извне, называются *периферийными устройствами*.



полос разной толщины, так называемый бар-код. *Бар-код (штрих-код)* содержит в себе информацию о товаре и его производителе. Во время оплаты бар-код пропускается через оптический считыватель (сканер). Компьютер, подключенный к сканеру, находит из общего списка цену товара соответствующего бар-кода и отправляет результат в кассовый аппарат. Эта технология позволяет быстро и точно рассчитывать с покупателями. Одновременно каждый проданный товар регистрируется в базе данных. Поэтому в любой момент можно получить информацию о проданной и имеющейся в наличии продукции.

В крупных супермаркетах компьютеры освещают помещения и регулируют температуру воздуха.

Компьютер в медицине. До недавнего времени компьютеры использовали в больницах только для ведения учета и регистрации. Сегодня они играют большую роль и в уходе за больными. Сенсоры, прикрепленные к телу пациента, могут показывать изменения, происходящие в его организме.

В настоящее время многие врачи используют компьютер в качестве помощника при постановке диагноза, а именно для уточнения того, что именно болит у пациента. Для этого больной тщательно обследуется, результаты обследования передаются компьютеру. Через некоторое время компьютер сообщает, какой из сделанных анализов дал аномальный результат. При этом он может назвать возможный диагноз. Конечно, окончательное решение принимает врач, но компьютер ускоряет процесс принятия решения.

Компьютер в образовании. По сравнению с другими сферами компьютеры в образовании выполняют больше функций.

В информатике компьютер в первую очередь – это *объект изучения*. На уроках информатики изучаются строение, принцип работы, виды, сферы применения, история, программное обеспечение компьютера. В то же время компьютеры незаменимы при изучении таких составляющих основу информатики тем, как алгоритмизация и программирование.

Используя специальные программы (текстовые, графические и музыкальные редакторы, электронные таблицы, базы данных), а также соответствующие мультимедийные пособия, можно значительно повысить уровень преподава-

Белые и черные полосы по-разному отражают свет лазерных лучей, проходящих через бар-код. На основе этой информации компьютер «узнает» товар и определяет его стоимость в своей базе данных.



Сейчас в некоторых супермаркетах в тележки для продуктов вмонтированы калькуляторы, чтобы покупатель мог быстро ориентироваться в том, на какую сумму он отобрал продукты.





Какую работу вы выполняете на домашнем компьютере? С какой целью используются компьютеры на работе у ваших родителей?

ния многих предметов. В настоящее время компьютеры успешно используют для обучения людей с ограниченными физическими возможностями.

И, наконец, компьютеры и соответствующее программное обеспечение широко применяются при оценивании знаний обучающихся (учеников, студентов) и в управлении образовательными учреждениями.

Компьютеры на страже закона. В правоохранительных органах создаются информационные банки преступников, разыскиваемых и пропавших людей, украденных вещей, автомобилей, оружия и т.д. Кроме этого компьютер является хорошим помощником во время расследования преступления. Например, криминалисты используют возможности компьютера при анализе материальных улик, взятых с места события.



- ✓ Компьютеры широко используются в науке, образовании, медицине, делопроизводстве и других сферах.
- ✓ На крупных предприятиях управление производственными процессами осуществляется посредством компьютеров.
- ✓ Компьютеры (роботы) могут выполнять работу, которая для людей оказывается слишком тяжелой или вообще невозможной, порой даже лучше них.
- ✓ В огромных супермаркетах компьютеры освещают помещения и регулируют температуру воздуха.
- ✓ Многие врачи используют компьютер в качестве помощника при постановке больным диагноза. Компьютерные технологии позволяют ухаживать за пациентами.
- ✓ В сфере образования компьютеры выполняют несколько функций. В первую очередь компьютеры – это объекты изучения в преподавании информатики.
- ✓ Компьютеры и соответствующее программное обеспечение успешно применяются при оценивании знаний обучающихся (учеников, студентов).
- ✓ Компьютеры используют при преподавании учебных дисциплин и в управлении образовательными учреждениями.

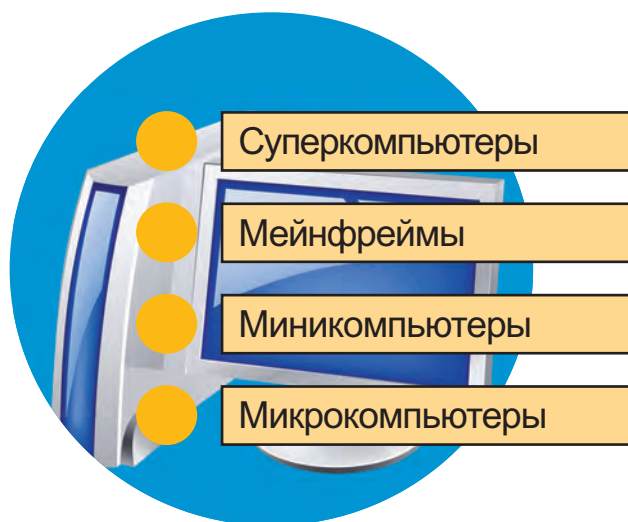
1.2. КЛАССИФИКАЦИЯ КОМПЬЮТЕРОВ

Можно ли мобильный телефон считать компьютером?

Если да, то почему?

Где вы встречали компьютеры, внешне не похожие на обычные компьютеры?

Используемые дома, в школе персональные компьютеры – это только один из их видов. Существуют и другие виды компьютеров, предназначенные для решения разнообразных задач (1-1).



1-1. Виды компьютеров

Суперкомпьютеры – это самые большие и производительные компьютеры (1-2). Они используются в крупных организациях для обработки данных значительного объема. В отличие от других компьютеров у них высокое быстродействие, а их стоимость исчисляется миллионами долларов.

По сравнению с суперкомпьютерами **мейнфреймы** меньше по размеру и менее производительные. По размеру они относительно больше персональных компьютеров, которые мы видим каждый день (1-3). Эти компьютеры обрабатывают задания многочисленных пользователей, а их стоимость часто достигает сотен тысяч долларов. Мейнфреймы используют в крупных организациях и государственных структурах для централизованного хранения, обработки и управления данными очень большого объема.

Миникомпьютеры по размеру больше персональных микрокомпьютеров, но имеют почти одинаковые с ними возможности, однако стоимость их намно-

го выше обычных компьютеров. Организации, имеющие многочисленных пользователей и данные большого объема, отдают предпочтение миникомпьютерам.

Микрокомпьютеры нередко называют *персональными*, или *настольными компьютерами*. Эта разновидность компьютеров предназначена для использования дома или в учреждении. Их размеры и форма позволяют устанавливать их на столе или под ним. Обычно микрокомпьютеры классифицируют по процессорам, которые они используют. Наиболее распространены разнообразные версии компьютеров PC (образовано от английских слов «Personal Computer», произносится как «пи-си») с операционными системами Microsoft Windows и различные версии Apple Macintosh с операционными системами Mac OS (1-4).

Ноутбуки также относятся к классу микрокомпьютеров. Возможности *ноутбуков*, или *лэптопов* (англ. laptop, lap – «колени сидящего человека», top – «верх») (1-5), схожи с возможностями настольных компьютеров. Однако по размерам они меньше микрокомпьютеров и стоят дороже. Их легко переносить из-за небольших размеров, и работают они как от электрической сети, так и от батареи. Для подключения к Интернету ноутбуки снабжены устройствами беспроводной связи.

Другие виды компьютеров. *Планшетный компьютер* очень похож на ноутбук. Однако экран планшета может вращаться вокруг своей оси так, что клавиатура остается под ним. В таком виде планшет напоминает записную книжку. Для ввода данных и выделения объектов используют *цифровую «ручку»* или *стилус*.

Карманный компьютер по размеру меньше ноутбука. Его возможности ограничены и у него могут отсутствовать присущие компьютерам традиционные ком-



1-2. Суперкомпьютер

поненты (например, клавиатура). Поэтому его сенсорный экран принимает символы, которые вводят пальцем или специальным устройством.

На сегодняшний день и *мобильные телефоны* имеют некоторые возможности, присущие микрокомпьютерам (например, подсоединение к Интернету, электронная почта). Большинство мобильных телефонов снабжены цифровой камерой.

Примером новых технологий, облегчающих решение повседневных проблем, могут быть калькуляторы, интерактивные книги, цифровые камеры. Микропроцессорами и компьютерными чипсетами снабжены и используемые в быту посудомоечные машины, холодильники, CD и DVD плееры.

Почему в ноутбуках в основном пользуются не мышью, а трекболом или тачпадом?



1-3. Мейнфрейм

- ✓ Компьютеры классифицируют по размерам, быстродействию и сферам применения.
- ✓ Основные четыре класса компьютеров: суперкомпьютеры, мейнфреймы, мини-компьютеры и микрокомпьютеры.
- ✓ Суперкомпьютеры – это самые большие и самые быстродействующие компьютеры.
- ✓ Мейнфреймы в сравнении с суперкомпьютерами менее производительны и имеют меньшие размеры, но они немного крупнее персональных компьютеров, с которыми мы встречаемся каждый день.
- ✓ Миникомпьютеры по размерам больше персональных микрокомпьютеров, но имеют почти одинаковые с ними возможности, стоимость же их намного выше.
- ✓ Микрокомпьютеры часто называют персональными, или настольными, компьютерами.
- ✓ Мобильные телефоны имеют некоторые возможности, присущие компьютерам.



PC



Mac

1-4. Микрокомпьютеры



Ноутбук



Планшетный компьютер

1-5. Портативные компьютеры

1.3. ИНФОРМАЦИОННАЯ КУЛЬТУРА

Как вы передадите имеющуюся на вашем компьютере информацию своему другу?

Какие средства сетевого общения, кроме электронной почты, вы знаете?

Работа большинства людей в наше время так или иначе связана с информацией, поэтому век, в который мы живем, называют веком *информационного общества*. Это означает, что большинство людей заняты ее производством, хранением, обработкой и использованием. Как и в каждом обществе, имеющем свое представление о культуре, в информационном обществе также существует понятие «*информационная культура*».



Одинаковы ли этические нормы, которые необходимо соблюдать при переписке с другом и при письменном общении с незнакомым вам человеком?

Информационная культура – это знания и навыки, необходимые для жизнедеятельности человека в информационном обществе. В настоящее время это понятие в первую очередь подразумевает потребность в информационных технологиях и умение пользоваться ими. Другими словами, в информационном обществе «культурным» считается тот человек, который хорошо владеет информационными технологиями и использует их по назначению.



В формировании информационной культуры большую роль играют *компьютерные сети*.

Самая большая сеть компьютеров в мире – *Интернет*. В наши дни ежедневно Интернетом пользуются миллионы людей. Среди предлагаемых в Интернете многочисленных услуг наиболее используемая – *электронная почта (e-почта)*. С ее помощью посредством сети можно отправлять информацию с одного компьютера на другой. Развитие электронной почты и других услуг сетевого общения (например, телеконференция, чат) привело к необходимости формирования определенных *этических норм*.

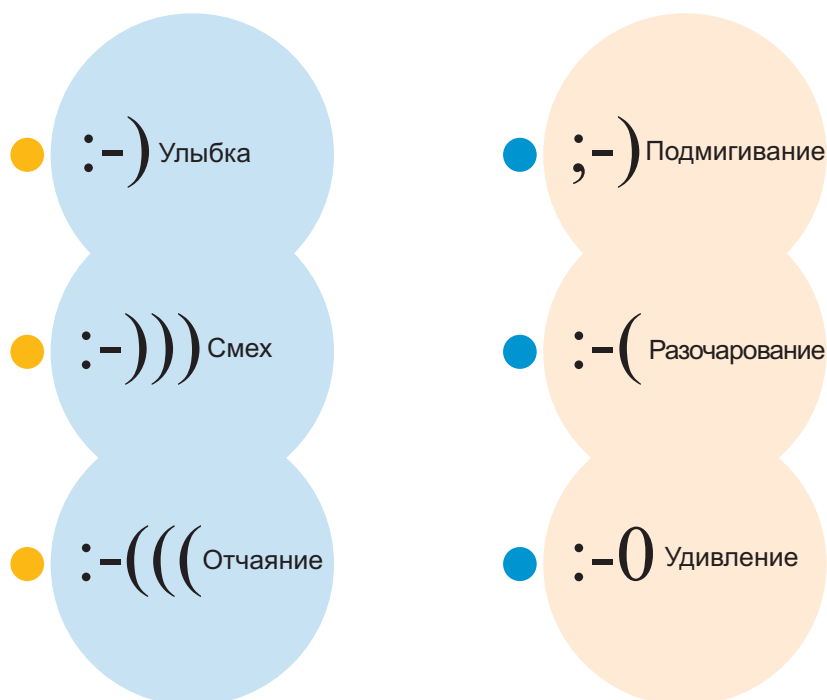
Правилам поведения в общественных местах обучают с детства. Но виртуальное общение отличается от реального. Собеседники вас не видят, они не знают ваших преимуществ и ваших недостатков, никого не интересует ваше общественное положение. Во время виртуального общения кто-то из собеседников может надоедать другим, раздражать их, а зная о своей безнаказанности, может и оскорбить. Поэтому во время общения в сети необходимо соблюдать общепринятые правила – *этику общения*.

Основные правила сетевого этикета при общении в сети следующие:

- **Придерживайтесь традиционного стиля.** В тексте электронного сообщения без необходимости не используйте заглавные буквы. Не выделяйте весь текст или большую его часть заглавными буквами. ТАКОЙ ТЕКСТ ТЯЖЕЛО ЧИТАТЬ, И ОН СЧИТАЕТСЯ НЕКОРРЕКТНЫМ.
- **Четко указывайте тему письма.** Занятые люди, прежде чем решить читать или не читать полученное письмо, обращают внимание на его тему. Особенно это важно, если автор письма неизвестен. Поэтому в теме письма не стоит использовать неопределенные или предельно обобщенные (например, такие как «информация», «полезная информация») выражения. Тема должна быть настолько краткой и информативной, чтобы сразу можно было понять содержание письма.
- **Своевременно отвечайте на письма.** Если просят вовремя ответить на полученное письмо, задерживать ответ нетактично. На очень срочные или важные письма следует отвечать в течение нескольких минут (хотя бы в течение часа). На остальные письма можно ответить в течение дня.
- **Не обращайтесь на письма, которые могут стать причиной конфликта.** Конечно же, получив бессмысленное или оскорбительное письмо, вы захотите сразу же ответить на него соответственно. Обычно такие ответы еще более обостряют ситуацию. Не стоит моментально отве-



чать на такие письма. Тщательно взвесьте свой ответ, а лучше бессмысленные, оскорбительные письма оставлять без ответа. Берегите свое время и нервы!



Смайлики

При общении по электронной почте есть и другие общепринятые «золотые» правила. Знать и следовать им очень важно.

- Выражайте мысли кратко, передавая только суть. Не прикрепляйте к письму очень большие файлы. Если у получателя вашего письма нет скоростного Интернета, загрузка прикрепленного файла на его компьютере займет очень много времени.
- Для выражения эмоций в электронной переписке часто вместо слов используют последовательность символов (*смайликов*). Каждый символ, используемый в смайлике, имеет свое значение: *двоеточие* означает глаза, *дефис* – нос, *скобка* – рот.
- В конце письма указывайте свои контактные данные.

- 
- ✓ В информационном обществе большинство людей заняты производством, хранением, обработкой и использованием информации.
 - ✓ Информационная культура – это знания и навыки, необходимые для жизнедеятельности человека в информационном обществе. В наше время это понятие в первую очередь подразумевает потребность в информационных технологиях и умение пользоваться ими.
 - ✓ В формировании информационной культуры велика роль компьютерных сетей, в частности Интернета.
 - ✓ Идея соединения электронных вычислительных машин между собой возникла почти одновременно с их созданием. Это было обусловлено необходимостью более плодотворной обработки информации, использования программ и аппаратных ресурсов.
 - ✓ При общении в сети необходимо соблюдать общепринятые правила – этику общения.

1.4. ВИДЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Какие компьютерные программы вам известны?

Как вы уже знаете, компьютерная система состоит из различных устройств – клавиатуры, мыши, монитора, принтера, микросхем, дисководов, устройств CD. Сами по себе эти устройства ничего делать не способны, поэтому для их работы требуются специальные программы.

Все программы, используемые на компьютере, составляют его *программное обеспечение*. По аналогии все устройства компьютерной системы называют его *аппаратным обеспечением*. При обработке данных компьютером его аппаратное и программное обеспечение взаимодействуют друг с другом. Данные поступают в компьютер посредством устройств ввода. Затем специальные программы «рекомендуют» компьютеру, как обработать эти данные. Другими словами, программное обеспечение дает указание аппаратной части компьютера как и что делать. Представление данных в нужной пользователю форме и вывод их на экран монитора или на печать также осуществляется посредством программ.

Компьютерные программы можно разделить на три класса: *системные программы*, *прикладные программы* и *инструментарии программирования* (1-6).



1-6. Виды программного обеспечения

Системные программы обеспечивают правильную работу всех частей компьютера. Особое место в системных программах занимают *операционные системы*. Операционная система налаживает диалог между компьютером и пользователем.

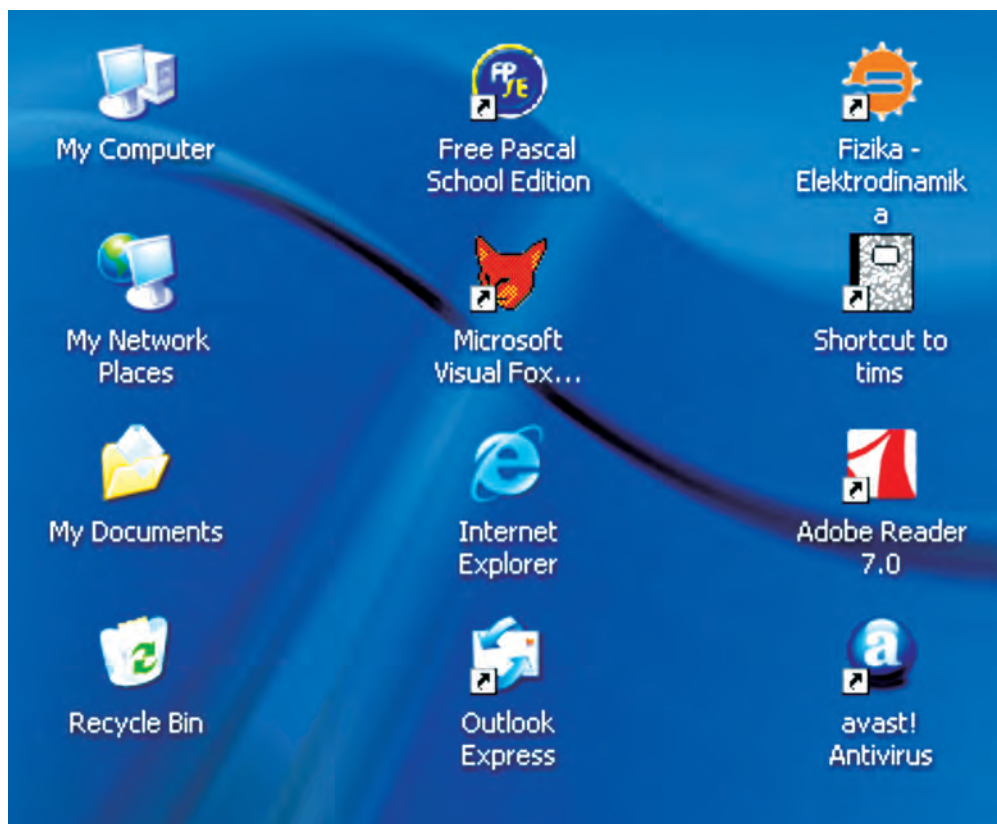
Вы уже знакомы с такими программами, как графический редактор Paint, текстовый редактор WordPad, программой создания презентаций PowerPoint. Наверняка каждый из вас играл в компьютерные игры. Такие программы называют **прикладными программами**. На сегодня для персональных компьютеров разработаны сотни тысяч прикладных программ.

Как же создаются все эти программы? Для этой цели используют программы, которые называют **инструментарии программирования**. Для создания прикладных программ широко используются такие языки программирования как Basic, Pascal, C++.

Как уже отмечалось, процесс взаимосвязи компьютера и человека регулируется операционной системой. Удобство работы на компьютере зависит именно от того, как эта взаимосвязь осуществляется. Совокупность методов и средств взаимодействия человека и компьютера называют *пользовательским интерфейсом*. Пользовательский интерфейс является основной чертой, характеризующей операционную систему.



Какой тип программы используют для регистрации оплаты покупок в кассах супермаркетов?



1-7. Значки в графическом интерфейсе пользователя

Долгое время наиболее распространенной операционной системой была **DOS**. Сейчас встретить ее можно только на устаревших и маломощных компьютерах.

На сегодняшний день наиболее распространенной операционной системой является **Windows**. Причиной огромной популярности стало наличие у нее графического интерфейса, а точнее, *графического интерфейса пользователя*. В таком интерфейсе общение с компьютером происходит при помощи графических символов, то есть специальных рисунков. При запуске компьютера на экране открывается рабочий стол, на котором расположены разные объекты, или *значки* (1-7). Эти графические символы обозначают файлы, диски, программы и другие объекты.

В большинстве современных компьютеров имеется один из типов графического интерфейса пользователя. Хорошо спроектированный графический интерфейс пользователя облегчает работу на компьютере, освобождает пользователя от запоминания текстовых команд. Для активизации программ или

команд пользователю достаточно установить указатель мыши на соответствующем объекте и щелкнуть по нему.



- ✓ Компьютерные программы можно разделить на три класса: системные программы, прикладные программы и инструментarii программирования.
- ✓ Системные программы обеспечивают правильную работу всех частей компьютера.
- ✓ Особое место в системных программах занимают операционные системы.
- ✓ Текстовые редакторы, электронные таблицы, издательские системы, системы управления базами данных, графические редакторы, компьютерные игры, обучающие программы и т.д относят к классу прикладных программ.
- ✓ Программы, относящиеся к классу инструментариев программирования, предусмотрены для создания системных и прикладных программ. Для разработки прикладных программ используют такие языки программирования, как *Basic*, *Pascal*, *C++*.
- ✓ Существуют два широко распространенных вида интерфейса пользователя: интерфейс командной строки (*DOS*) и графический интерфейс пользователя (*Windows* и *Mac OS*). А в операционной системе *Linux* эти два интерфейса почти равноправны.
- ✓ В графическом интерфейсе пользователя общение с компьютером осуществляется посредством графических символов (значков), рисунков.

Вопросы и задания

1. Назовите несколько областей, где широко применяются компьютеры.
2. В каких сферах компьютеры действуют лучше людей?
3. Как можно использовать компьютер в сельском хозяйстве?
4. К какому виду компьютеров относят ноутбуки?
5. В чем отличие планшетных компьютеров от ноутбуков?
6. Что такое стилус?
7. Как вы представляете дальнейшее развитие компьютерной техники?
8. Что такое информационное общество и информационная культура?
9. С какой целью были созданы компьютерные сети?
10. Какие особенности имеет общение в виртуальном мире в отличие от реального общения?
11. Какие нормы важно соблюдать во время электронной переписки?
12. Что такое смайлик и с какой целью его используют?
13. Придумайте новые смайлики для общения с друзьями.
14. На какие классы делят программное обеспечение?
15. Какова функция системных программ?
16. Для чего нужны прикладные программы?
17. Что имеют в виду, говоря об инструментариях программирования?
18. К какому классу программного обеспечения относят программы Microsoft Office?
19. Что такое графический интерфейс пользователя?
20. Сгруппируйте известные вам программы по классам.

ИСТОРИЧЕСКАЯ СПРАВКА

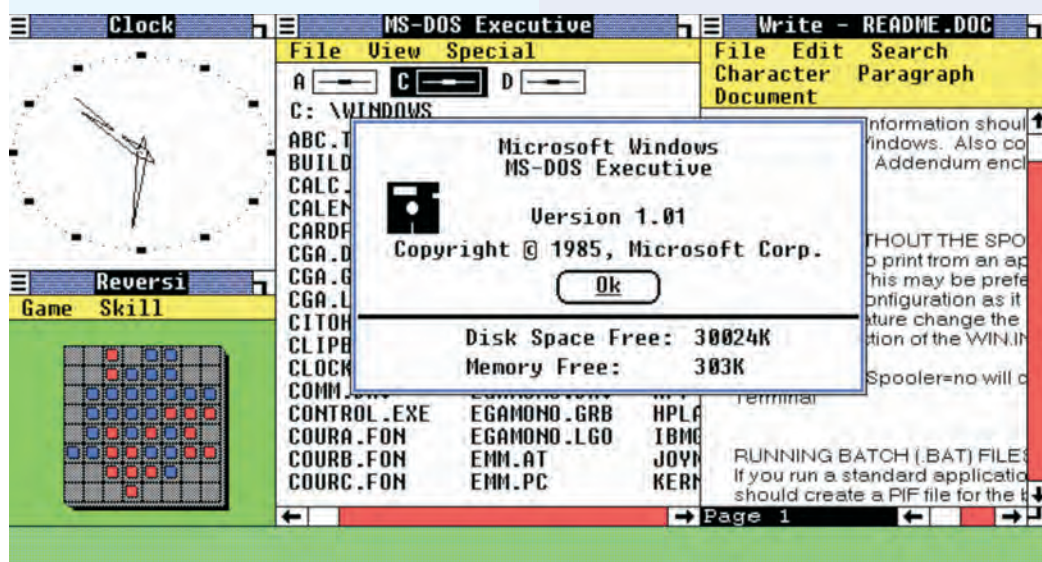
20 ноября 1985 года компания Microsoft приступила к продаже разработанной для персональных компьютеров операционной системы **Windows 1.01**. Версия Windows 1.0 этой системы не была выпущена на продажу из-за серьезных ошибок.



Билл Гейтс (1985)

Компания Microsoft, разрабатывающая программное обеспечение для персональных компьютеров, была основана **Биллом Гейтсом** и его другом **Полем Алленом**. Билл Гейтс начал заниматься программированием в возрасте 13 лет. В 1973 году он поступил в Гарвардский университет. Во время учебы он разработал для первых компьютеров Altair 8800 язык программирования **BASIC**. В 1975 году, после создания компании Microsoft, Билл Гейтс, не окончив обучение, ушел с третьего курса университета.

В 1981 году Microsoft разработала операционную систему **MS-DOS**.



ПРОВЕРЬ СЕБЯ

Выбери правильный ответ

1. Основные устройства компьютера:

- A. монитор, клавиатура, мышь, модем
- B. системный блок, монитор, клавиатура, принтер
- C. системный блок, монитор, клавиатура, мышь
- D. системный блок, модем, клавиатура, мышь

2. Программное обеспечение называют также:

- A. аппаратным обеспечением
- B. программами
- C. алгоритмом
- D. интерфейсом

3. Набор программ, управляющих работой компьютера, называют:

- A. системными программами
- B. прикладными программами
- C. инструментарием программирования
- D. утилитами

4. К какому классу компьютеров относится ноутбук?

- A. суперкомпьютеры
- B. мейнфреймы
- C. миникомпьютеры
- D. микрокомпьютеры

5. Самая большая сеть компьютеров в мире?

- A. Арпанет
- B. Интернет
- C. Ethernet
- D. Интранет

ПРОЕКТЫ

ПРОЕКТ 1-1

Компьютеры называют иногда «электронно-вычислительными машинами». Как по-вашему, правильно ли считать компьютеры машинами? Имеют ли они свойства, присущие машинам? Чтобы обосновать свое мнение, соберите информацию о машинах и их свойствах.

ПРОЕКТ 1-2

Выберите сферу ваших интересов (например, спорт, музыка, наука и т.п.). Соберите информацию об использовании компьютеров в этой сфере и составьте небольшой текст об этом.

ПРОЕКТ ДЛЯ ВЫПОЛНЕНИЯ В ГРУППАХ

Проанализируйте с точки зрения сетевого этикета письмо, полученное от вашего друга или отправленное ему по электронной почте. Укажите, какие нормы в нем не соблюдены. Обсудите результаты в группе.

КРИТИЧЕСКИЙ ВЗГЛЯД

Представьте себе, что все компьютеры в мире перестали на какое-то время работать. Какие события могут произойти в этом случае? Опишите вероятное развитие событий в небольшом сочинении, а также добавьте свои размышления о вредном влиянии компьютеров на нашу жизнь.

МОДЕЛИРОВАНИЕ

2



- Модели и цели их создания
- Разные модели одного и того же объекта
- Построение моделей в программе SketchUp
- Изменение стиля модели
- Придание моделям тени
- Использование имеющихся в Интернете готовых моделей



2.1. МОДЕЛЬ И МОДЕЛИРОВАНИЕ

Что такое глобус?

Как вы понимаете слово «модель»?

Окружающий нас мир очень многообразен. Объекты вокруг нас имеют сложное строение и обладают присущими им отличительными свойствами. В процессе познания мира невозможно учесть все свойства объектов. И поэтому нередко в процессе изучения объектов используют их *модели*. Другими словами, намного удобнее изучать не сам реальный объект, а его модель, которая отражает определенные его свойства.



Термин «модель» происходит от латинского слова «modulus», что означает «образец».

Для примера рассмотрим в качестве объекта человека. Для маленьких детей первой моделью человека является кукла. С возрастом эта модель становится им неинтересной. Поэтому кукла должна быть заменена другой моделью.

Как и сам человек, его модели разнообразны. То есть конкретная модель будет изменяться в зависимости от цели. Например, портному в работе пригодится манекен, так как его в первую очередь интересуют размеры человека – рост, объем груди, ширина спины и т.д.

А для учителя, преподающего анатомию, или медицинского работника, изучающего структуру костей человека, ничто не заменит скелета человека. В моделях, созданных скульптором или художником, большое значение имеет внешний вид человека. Таким образом, для одного и того же объекта можно создать разные модели. Кукла, манекен, скульптура, картина – это модели, отражающие разные свойства человека.



2-1. Разные модели человека (кукла, рисунок, скульптура)

В зависимости от формы реализации модели делят на два вида: *материальные* и *информационные*.

Материальную модель можно потрогать руками. Приведенные выше кукла, скульптура (2-1) – материальные модели человека.

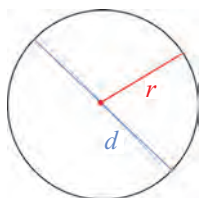
Информационная модель – это представленная в определенном виде информация, описывающая существенные свойства объекта или события. Информационные модели широко распространены. Например, описание любого события или объекта в литературном произведении, изображение материального объекта на бумаге (рисунок, схема) тоже являются информационными моделями.

Таким образом, когда говорят о модели объекта, имеют в виду образ, отражающий его определенные свойства. Сам объект, по которому создают модель, называют **объектом-оригиналом**. Процесс создания и изучения моделей называют **моделированием**. На протяжении жизни человек постоянно сталкивается с разными задачами: как создавать новые машины и механизмы, как строить новое, комфортабельное жилье, как уберечься от болезней и т.д. Моделирование дает возможность решать такие задачи намного быстрее и с меньшими затратами.

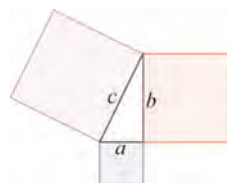
Для информатики, естественно, интересны именно информационные модели. Для представления информационных моделей пользуются как естественными, так и формальными языками. Естественные языки предпочтительнее при создании *словесных, изобразительных моделей*. Например, между литературным произведением и понятием модели есть непосредственная связь, так как произведение обращает внимание читателя на разные стороны жизни. Наиболее ярко это видно в баснях.

При создании *математических, логических и специальных моделей* используют формальные языки. Примером математической модели может быть любая формула, знакомая вам с урока математики.

Какие модели Земли вам известны?
Для какой цели были созданы эти модели? Могут ли разные объекты иметь одинаковые модели?



$$S = \frac{1}{2} ah$$



2-2. Математические модели

Одной из простейших форм информационной модели является *графическая модель*. Графические модели отражают внешние свойства объектов – размеры, форму, цвет. В отличие от словесных моделей графические модели более информационны. На уроках биологии, географии, военного дела, истории

широко используют карты. Прежде чем приступить к созданию технических устройств, строительству зданий, готовят их чертежи.

Для того чтобы наглядно показать цифровые и статистические данные, используют графики и диаграммы.

После того, как появились компьютеры, их стали широко применять для создания моделей. Современные компьютеры позволяют создавать очень сложные *компьютерные модели*.

Компьютерные модели могут заменить как материальные, так и нематериальные (информационные) модели. Когда-то, чтобы заранее изучить, как будут работать машины и механизмы, создавали их материальные модели. Сегодня, в некоторых случаях, нет необходимости в таких моделях. С этой задачей гораздо успешнее справляются современные компьютерные программы.



2-3. Компьютерная модель



- ✓ Когда говорят о модели объекта, имеют в виду образ, который отражает его определенные свойства. Объект, по которому создают модель, называют объектом-оригиналом.
- ✓ Модели разделяют на два вида: материальные и информационные.
- ✓ Процесс создания и изучения модели называют моделированием.
- ✓ Кукла, манекен, скульптура, портрет – это разнообразные модели человека.
- ✓ Моделирование дает возможность решать задачи, с которыми сталкивается человек, намного быстрее и с меньшими затратами.
- ✓ Для представления информационной модели в той или иной форме используют как естественные, так и формальные языки.
- ✓ Компьютерные модели могут заменить материальные и информационные модели.

2.2. ГРАФИЧЕСКИЙ РЕДАКТОР GOOGLE SKETCHUP

Что такое графический редактор?

Какие графические редакторы вам известны?

Долгое время графические модели создавали на бумаге. С появлением компьютерной техники графические модели создают на компьютере. Для этого используется соответствующее программное обеспечение – *графический редактор*. Созданную в графическом редакторе модель можно качественно распечатать на принтере.

С графическим редактором Paint вы знакомы с младших классов. Рисунки, созданные в этой программе, двумерные. С усовершенствованием программного обеспечения появилась возможность создавать трехмерные графические объекты. Современные компьютерные игры и мультфильмы состоят именно из таких объектов.

В отличие от двумерной графики объемные объекты можно просматривать со всех сторон, вращая их на экране монитора.

Плоские графические изображения на компьютере и технология их создания называется *двумерной графикой (2D-графика)*. А объемные графические объекты и технологию их создания называют *трехмерной графикой (3D-графика)*.

Для создания трехмерных графических объектов и работы с ними были разработаны различные графические редакторы. Научиться работать на предусмотренном для профессионалов 3D-редакторе не так-то легко. Но, несмотря на это, существуют более простые для освоения 3D-редакторы. Одной из таких программ является свободно распространяемая и имеющая очень удобный интерфейс программа **Google SketchUp**. Есть и платная версия этой программы – **SketchUp Pro**. При появлении новой версии этой программы более ранняя версия распространяется бесплатно.

Как и в большинстве программ, при запуске этой программы на экране открывается главное окно (2-4). Оно состоит из четырех основных частей: строки меню, панели инструментов, окна моделирования и панели состояния.

Рабочее поле редактора называют *окном моделирования*. На нем цветными стрелками (красной, зеленой и синей) изображена трехмерная система координат.

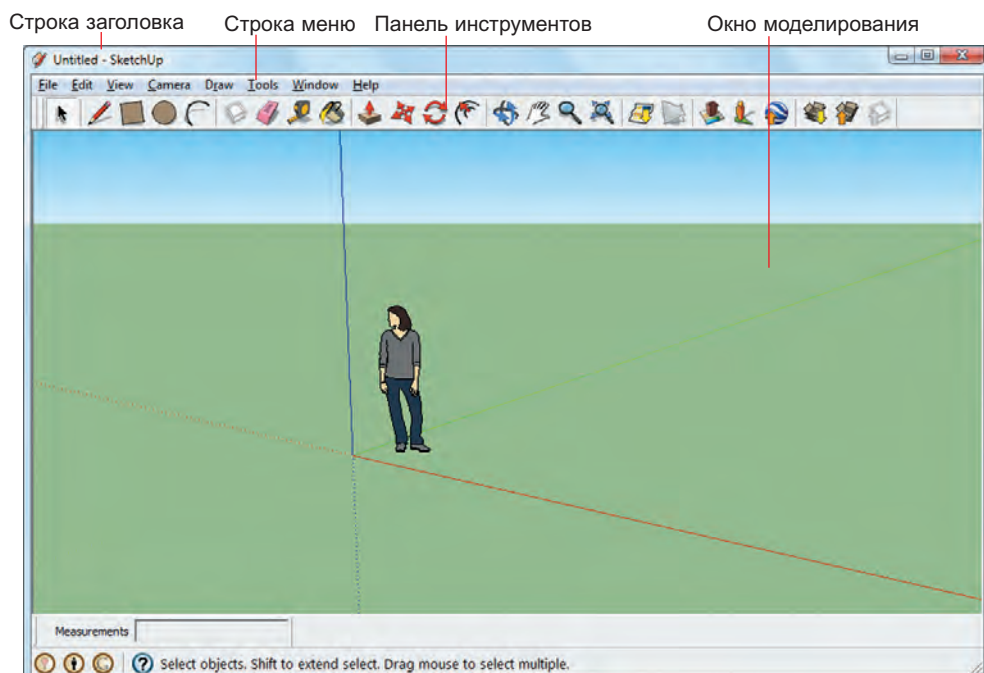
В результате развития технологий появляются все новые и новые термины. Один из них термин **трехмерная**, или **3D-графика** для обозначения объемных объектов. Соответственно плоские объекты называют **двумерными**, или **2D- графикой**.



Как по-вашему, что означает буква D в термине «3D-графика»?



Виртуальные плоскости, проходящие через систему координат, условно делят пространство на 8 частей. Для определения расстояния и восприятия масштаба у начала координат изображена фигурка человека.



2-4. Главное окно программы Google SketchUp

Когда в компьютерной графике говорят «перетаскивание», имеют в виду, что надо «схватить» изображение или объект и «перетащить» его по экрану с одного места на другое. При работе с мышью перетаскивание производится следующим образом: (1) указатель мыши наводится на объект; (2) удерживая левую кнопку мыши, объект перемещается на новое место; (3) кнопка мыши отпускается.



Для просмотра объемных моделей в этой программе используют следующие принципы: объекты как будто снимаются на виртуальную камеру и передаются в окно редактору. Камера может двигаться в виртуальном пространстве. В этот момент изображение в окне меняется. Для того чтобы открыть режим изменения местоположения камеры, на панели инструментов предусмотрены специальные кнопки.



Кнопка Орбита дает возможность камере двигаться по любой орбите. Для изменения местоположения в этом режиме используют метод *перетаскивания мыши*.



Кнопка Панорама предусмотрена для активизации режима изменения панорамы камеры. В этом режиме для изменения местоположения также используют метод перетаскивания мыши.



Кнопка Масштаб приближает и удаляет объектив камеры в направлении стрелки. И здесь изменение местоположения связано с перетаскиванием мыши.



Щелчок по кнопке **В размер окна** размещает камеру так, чтобы модели заняли весь экран.

Задание 1. Работа с кнопками изменения положения камеры

Приведите в движение изображение на экране, используя кнопки изменения положения камеры. Чтобы потренироваться, повторите одни и те же действия несколько раз.

Создание изображений в графическом редакторе обычно связано с использованием стандартных геометрических фигур – *графических примитивов*. К ним относятся следующие геометрические фигуры: прямая, кривая, прямоугольник, овал (эллипс), многоугольник, скругленный прямоугольник.

В редакторе Google SketchUp графических примитивов всего четыре:



– линия;



– прямоугольник;



– окружность;



– дуга.

Инструмент «Линия» служит для рисования объектов типа «линия». А инструменты «Прямоугольник» и «Окружность» рисуют фигуры, соответствующие их названию. При помощи инструмента «Дуга» можно создать дугообразный объект.

Задание 2. Работа с кнопками Прямоугольник и Окружность

1. Выберите инструмент **Прямоугольник**. Указатель мыши примет форму карандаша.
2. Переместите указатель на горизонтальную плоскость и, перетаскивая мышью, нарисуйте прямоугольник.
3. Аналогично нарисуйте окружность. Обратите внимание на то, что построение окружности начинается с центра.

На следующем уроке вы познакомитесь с технологией построения с помощью плоских фигур объемных.



- ✓ *Плоские графические изображения и технология их создания называется двумерной графикой (2D-графика).*
- ✓ *Для создания трехмерных графических объектов и работы с ними разработаны различные графические редакторы. Одной из таких графических программ является программа Google SketchUp.*
- ✓ *Главное окно программы состоит из четырех основных частей: строки меню, панели инструментов, окна моделирования и панели состояния.*
- ✓ *Рисование в графическом редакторе обычно связано с использованием стандартных геометрических фигур – графических примитивов. Основные графические примитивы: прямая, кривая, прямоугольник, овал (эллипс), многоугольник, скругленный прямоугольник.*
- ✓ *В редакторе Google SketchUp используют четыре графических примитива: линия, прямоугольник, окружность и дуга.*

2.3. ТЕХНОЛОГИЯ РАБОТЫ С МОДЕЛЯМИ. РЕБРА И ГРАНИ

Сколько ребер у куба?

Сколько граней у четырехугольной пирамиды?

Написать на языке программирования программу для получения трехмерных объектов на экране компьютера достаточно сложно. Но в программе SketchUp создавать такие объекты просто: для этого предусмотрен инструмент

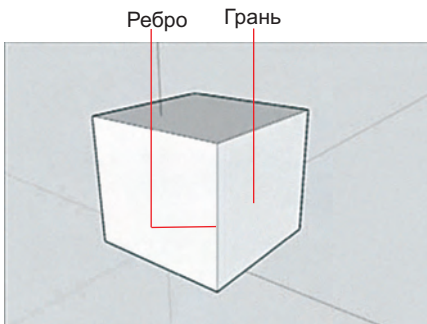


Тяни/Толкай (Push/Pull). При выборе этого инструмента указатель мыши принимает соответствующую форму. Затем при перемещении указателя мыши на любую плоскую фигуру, она выделяется. При этом фигура покрывается мелкой сеткой из точек. Для того чтобы эту фигуру превратить в трехмерную, достаточно перетащить мышью в соответствующем направлении.

Задание 3. Работа с инструментом Push/Pull

1. Запустите программу Google SketchUp.
2. Используя инструмент  Прямоугольник, нарисуйте прямоугольник на поле моделирования.
3. Выберите инструмент  Тяни/Толкай на панели инструментов. Указатель примет соответствующую форму.
4. Установите указатель на прямоугольнике и перетащите мышь вверх. Вслед за указателем будет увеличиваться объем фигуры. Получится параллелепипед.
5. Установите курсор на боковой грани параллелепипеда. Она выделится (покроется мелкой сеткой из точек).
6. Измените размеры параллелепипеда, перетащив мышь в соответствующем направлении.

Основу всех моделей программы SketchUp составляют *ребра* и *грани*. То есть ребра и грани – это основные блоки любой создаваемой модели. На рисунке (2-5) слева показан куб, созданный в программе SketchUp. Он состоит из 12 ребер и 6 граней. Модель справа, несмотря на сложность, также состоит из ребер и граней.

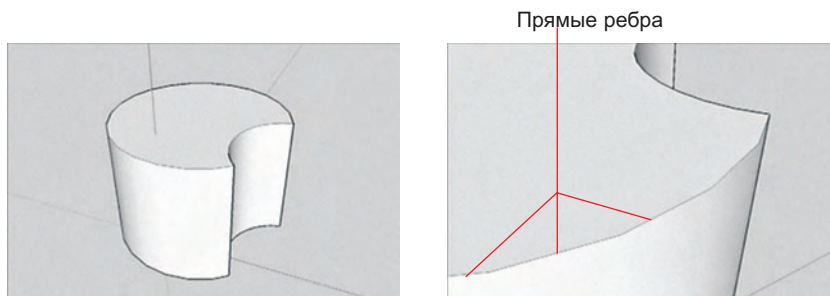


2-5. Модели SketchUp строятся из ребер и граней

При работе в SketchUp необходимо знать некоторые данные, касающиеся ребер и граней.

- **Ребра бывают только прямые.** Несмотря на то, что в программе SketchUp можно построить модель любой формы, ее ребра на самом деле всегда прямые. Даже дуги и окружности состоят из небольших прямых отрезков (2-6).
- **Ребра не имеют толщины.** Независимо от того, как они выглядят на экране, ребра не имеют никакой толщины.

- **Невидимость ребер не означает их отсутствия.** Если ребра не видны на экране, это не значит, что их нет, – просто они скрыты. Этот популярный способ используют при создании определенных форм.



2-6. Кривые линии также состоят из прямых ребер

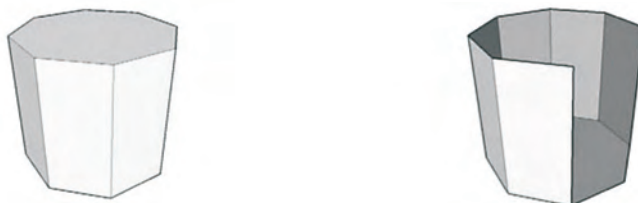
- **Не бывает граней без ребер.** Для существования на плоскости грани необходимо минимум три ребра, образующих замкнутый контур. То есть



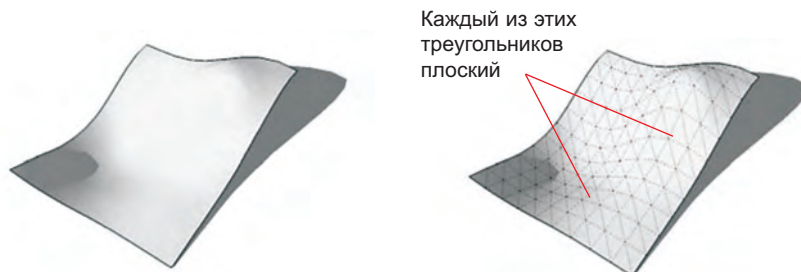
Как вы представляете себе шар в программе SketchUp, если основу каждой модели составляют ребра и грани?

три ребра ограничивают ее. А так как для получения замкнутого контура необходимо три прямых линии, у граней также будет три стороны. В программе SketchUp нет ограничений для количества сторон у граней. На рисунке (2-7) можно заметить, что происходит при удалении ребер, ограничивающих одну или несколько граней.

- **Грани всегда плоские.** В программе SketchUp поверхности, которые кажутся кривыми, состоят из множества плоских граней (2-8).
- **Как и ребра, грани не имеют толщины.** Хотя грани и похожи на лист бумаги, они не имеют толщины. Чтобы создать поверхность, обладающую толщиной, необходимо использовать две поверхности.



2-7. При удалении одного из ребер, образующих грань, стирается и сама грань



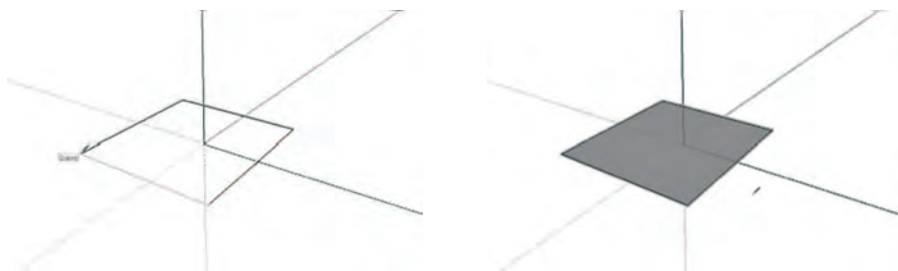
2-8. Кривые поверхности состоят из небольших плоских граней

Ознакомившись с некоторыми основными свойствами ребер и граней, рассмотрим их взаимосвязь. В программе SketchUp нет специального инструмента для создания граней. Однако, если на плоскости нарисовать замкнутую фигуру из трех или более ребер, SketchUp автоматически сформирует грань. Это показано на рисунке (2-9): как только последнее ребро соединяется с первым, цепочка ребер замыкается и SketchUp создает грань.

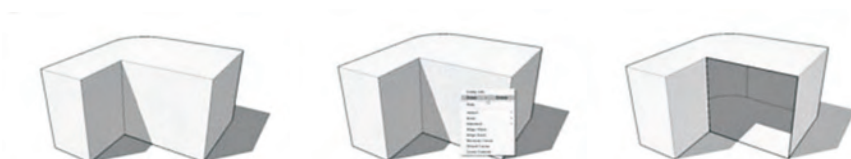
Если какая-то созданная грань вам не нужна и ее надо удалить, необходимо выполнить следующие действия.

Задание 4. Удаление грани

1. Установите указатель мыши на грани, которую необходимо удалить, и щелкните правой кнопкой мыши.
2. В открывшемся контекстном меню выберите пункт Удалить (Erase).
Грань будет удалена, но ребра, составляющие ее, останутся.



2-9. Как только на плоскости из ребер создается замкнутая цепь, SketchUp автоматически формирует грань



2-10. Грань можно стереть, не удаляя формирующее ее ребро

Стертую грань можно восстановить. Для этого ребра, расположенные на плоскости и образующие замкнутую область, необходимо заново нарисовать при помощи инструмента **Линия (Line)**.



При удалении одного ребра, составляющего грань, стирается и сама грань. Например, если удалить одно ребро куба, то две грани куба, примыкающие к этому ребру, станут невидимыми. То есть существование грани при отсутствии ребра, которое является ее частью, невозможно.



- ✓ При помощи инструмента **Тяни/Толкай (Push/Pull)** можно плоскую фигуру превратить в объемную (трехмерную).
- ✓ В программе **SketchUp** все модели состоят из ребер и граней.
- ✓ Необходимо знать некоторые важные факты, связанные с ребрами: 1) ребра всегда прямые; 2) ребра не имеют толщины; 3) если ребра не видны, это не означает их отсутствия.
- ✓ Грани тоже имеют несколько важных свойств: 1) не бывает грани без ребер; 2) грани всегда плоские; 3) как и ребра, грани не имеют толщины.
- ✓ В программе **SketchUp** нет специального инструмента для создания граней. Однако, если на плоскости нарисовать замкнутую фигуру из трех или более ребер, **SketchUp** автоматически сформирует грань.

2.4. СОЗДАНИЕ ТРЕХМЕРНЫХ МОДЕЛЕЙ

Составьте алгоритм изготовления стула из дерева.

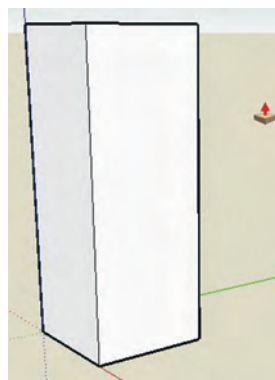
Для того чтобы научиться хорошо работать в программе **SketchUp**, как и в любой новой программе, необходимо время. Выполняя следующее задание, вы освоите основные инструменты и понятия этой программы.

Задание 5. Создание модели стула

1. Запустите программу Google SketchUp.
2. Выберите на панели инструментов инструмент Прямоугольник и нарисуйте прямоугольник с вершиной в начале системы координат (2-11 а).



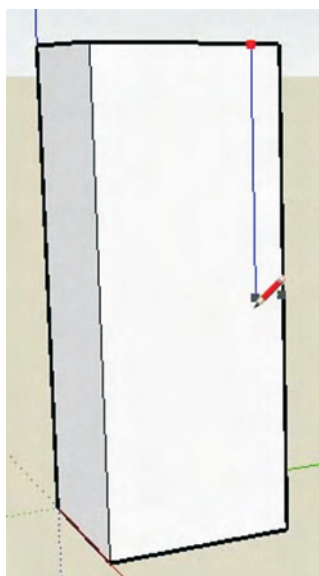
а



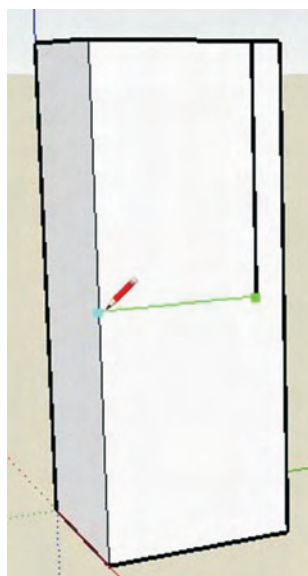
б

2-11.

3. Для определения размеров прямоугольника в строке состояния в поле Указатели размеров (Measurements) введите с клавиатуры «1800; 1800» и нажмите клавишу <Enter>. Если этого поля нет на панели состояния, активизируйте его при помощи команды меню Вид⇨Панели инструментов⇨Указатели размеров.
4. Используя кнопки изменения местоположения камеры (Орбита, Масштаб, Панорама), расположите прямоугольник так, чтобы с ним было легко работать.
5. Выберите инструмент  Тяни/Толкай на панели инструментов.
6. Установите указатель на прямоугольнике и перетащите мышью вверх.
7. Введите число «4» с клавиатуры и нажмите клавишу <Enter> (2-11 б).
8. Выберите инструмент  Линия (Line) и, начиная сверху боковой грани до ее середины, нарисуйте прямой вертикальный отрезок (2-12 а).
9. От конца отрезка до края грани нарисуйте горизонтальную линию (2-12 б).



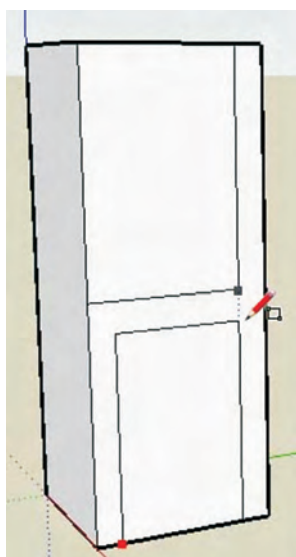
а



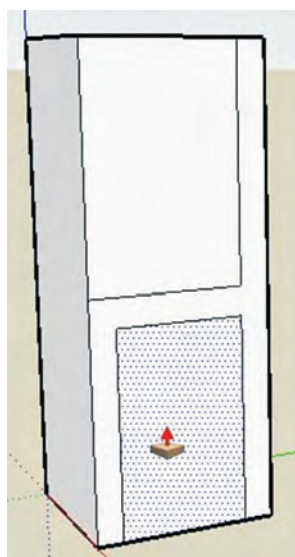
б

2-12.

10. При помощи инструмента  **Линия**, начиная сверху боковой грани до ее середины, нарисуйте прямой вертикальный отрезок (2-13 а).
11. Выберите инструмент  **Тяни/Толкай** и «вытолкните» только что нарисованный между ножками стула прямоугольник (2-13 б). Когда указатель достигнет задней грани модели, на экране появится надпись «На грани» («On face») (2-14 а).
12. Щелкните кнопкой мыши. Материал, находящийся под сиденьем стула, исчезнет.
13. Таким же образом удалите материал над сиденьем стула. После этого модель будет похожа на трехмерную модель стула (2-14 б).
14. Используя кнопки изменения местоположения камеры, поверните модель так, чтобы были видны задние ножки стула (2-15 а).

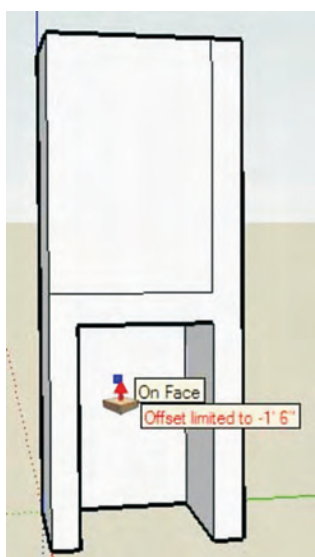


a

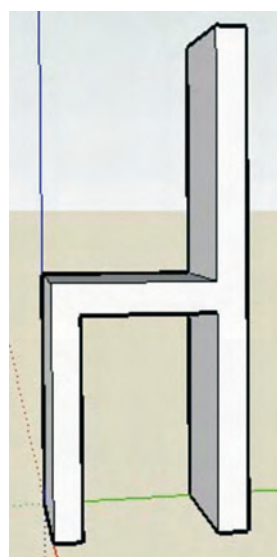


б

2-13.



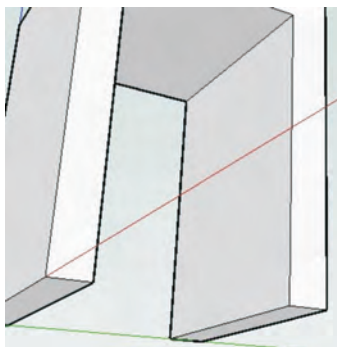
a



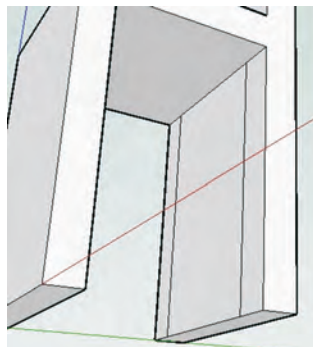
б

2-14.

15. При помощи инструмента  Прямоугольник нарисуйте прямоугольник, соответствующий пространству между задними ножками стула (2-15 б).



а



б

2-15.

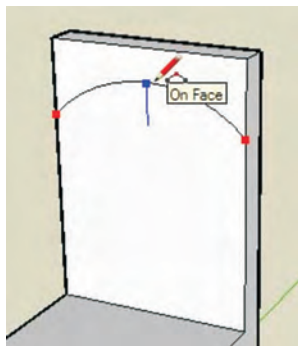
16. Используя инструмент  Тяни/Толкай, удалите пространство между задними ножками стула.

17. Точно так же удалите пространство между передними ножками стула.

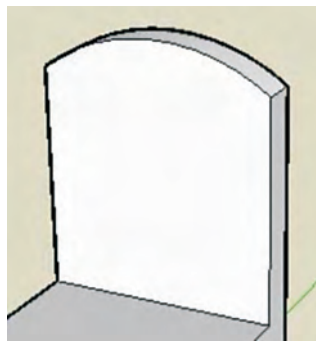
18. При помощи инструмента  Дуга нарисуйте сверху спинки стула дугу (2-16 а).

19. Используя инструмент  Тяни/Толкай, удалите пространство, возникшее сверху спинки стула.

20. Используя кнопку изменения местоположения камеры, просмотрите модель со всех сторон.



а



б

2-16.

2.5. СТИЛИ И ТЕНИ

Можно ли по тени определить объект?

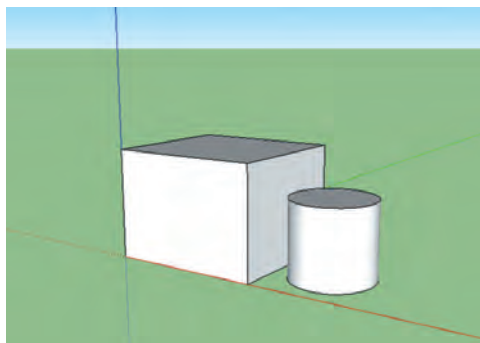
Для большей наглядности и правильного восприятия модели используют *стили*. Стил ь может состоять из определенного вида ребер, граней, фоновых оттенков и других изобразительных эффектов. В программе SketchUp множество стилей. Некоторые из них лучше использовать в процессе создания модели, а другие во время ее печати и представления.

Самый простой способ работы со стилями – использование готовых стилей, имеющихся в SketchUp. Применение стилей в SketchUp включает четыре этапа:

1. Выбрав команду меню **Окно⇒Стили (Window⇒Styles)**, откройте диалоговое окно **Стили**.
2. Перейдите на вкладку **Выбрать (Select)**.
3. В открывшемся списке **Стили (Styles Libraries)** выберите нужную вам библиотеку стилей.
4. Выбрав в окне понравившийся стиль, примените его к модели.

Задание 6. Применение стиля к модели

1. Запустите программу Google SketchUp.
2. Нарисуйте параллелепипед, используя инструменты  **Прямоугольник** и  **Тяни/Толкай**.
3. Аналогично, используя инструменты  **Окружность** и  **Тяни/Толкай**, нарисуйте рядом с ним цилиндр (2-17).



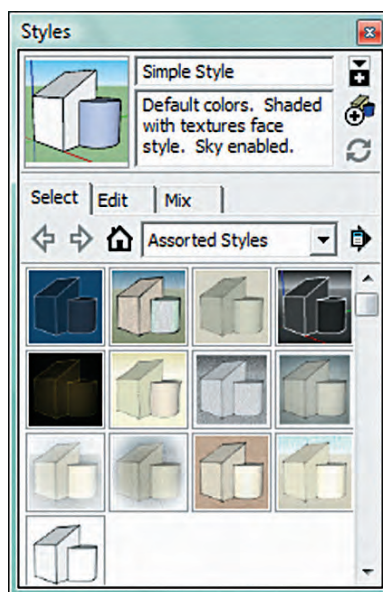
2-17.

4. Выберите команду **Стили** в меню **Окно (Window)**.

Откроется диалоговое окно **Стили (Styles)**.



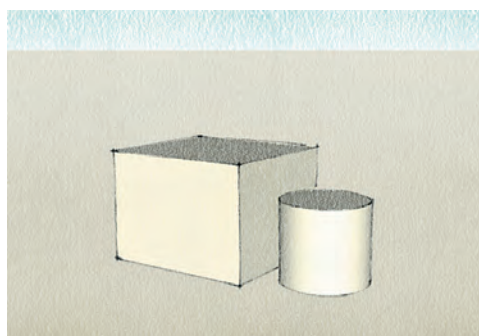
В древнем Риме палочку, которую использовали для письма, называли «stil». В дальнейшем почерк пишущего этой палочкой стали называть стилем. Позже определенную манеру письменной и устной речи также стали именовать стилем. Сейчас наряду с литературой, термин «стиль» существует в изобразительном искусстве, скульптуре, архитектуре, музыке и других сферах.



2-18.

5. Из открывшегося во вкладке **Выбрать (Select)** списка **Стили** выберите любую библиотеку стилей.

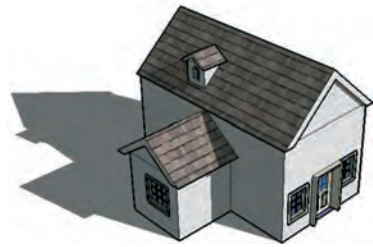
6. Выбрав в окне **Стили (Styles)** произвольный стиль, примените его к модели (2-19).



2-19.

7. Повторите шаги 5 и 6 для других стилей и обратите внимание на то, как меняются эффекты.

В программе SketchUp есть возможность добавления тени к моделям. Это помогает воспринимать объекты такими, какие они существуют в реальности. Тени могут меняться в зависимости от времени суток и расположения модели. Например, вы можете задать тень так, чтобы просмотреть, как будет выглядеть модель дома в городе Гянджа 10 декабря в 10²⁰ утра. Функция придания тени в SketchUp позволяет ощутить реальность и глубину объекта.



Что такое солнечное затмение и как оно происходит?



Чтобы вывести на экран панель инструментов для работы с тенью, в меню Вид (View) надо выбрать пункт Панели инструментов (Toolbars), а затем пункт Тени (Shadows). В результате на экране появится окно Тени (Shadows) (2-20). Нет необходимости для специального пояснения, как использовать имеющиеся здесь инструменты. Все просто и понятно: нужно применить каждый инструмент один раз, и станет ясно, как с ним работать.

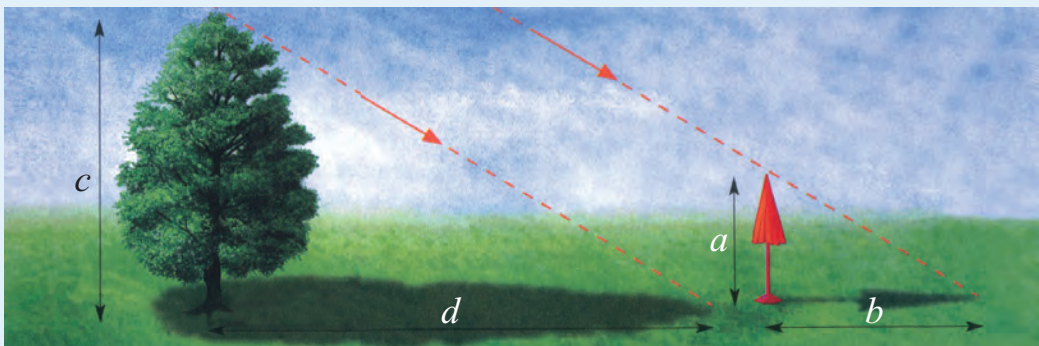


2-20. Инструменты работы с тенью



С помощью тени можно определить высоту объекта. Например, в солнечный день можно вычислить высоту дерева в зависимости от длины его тени. Для этого измерьте высоту любого предмета, например, сложенного зонта.

На рисунке эта величина отмечена буквой a . Потом измерьте длину тени зонта b и дерева d . Теперь можно определить высоту дерева по формуле $c = (a \times d) : b$.



2.6. ИСПОЛЬЗОВАНИЕ ГОТОВЫХ МОДЕЛЕЙ

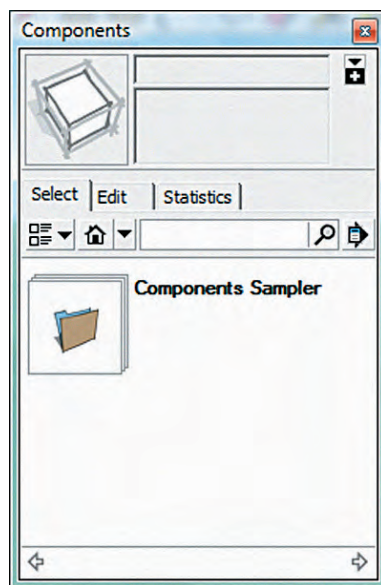
В чем смысл выражения «начать с нуля»?

Часто гораздо удобнее создавать модель не с нуля, а используя уже готовую модель. Если же готовая модель полностью не соответствует тому, что нужно вам, ее можно переделать.

Программа SketchUp предоставляет пользователям много различных готовых моделей. В предлагаемом задании указано, как использовать эти модели. Имейте в виду, что для использования готовых моделей компьютер должен быть временно подсоединен к Интернету.

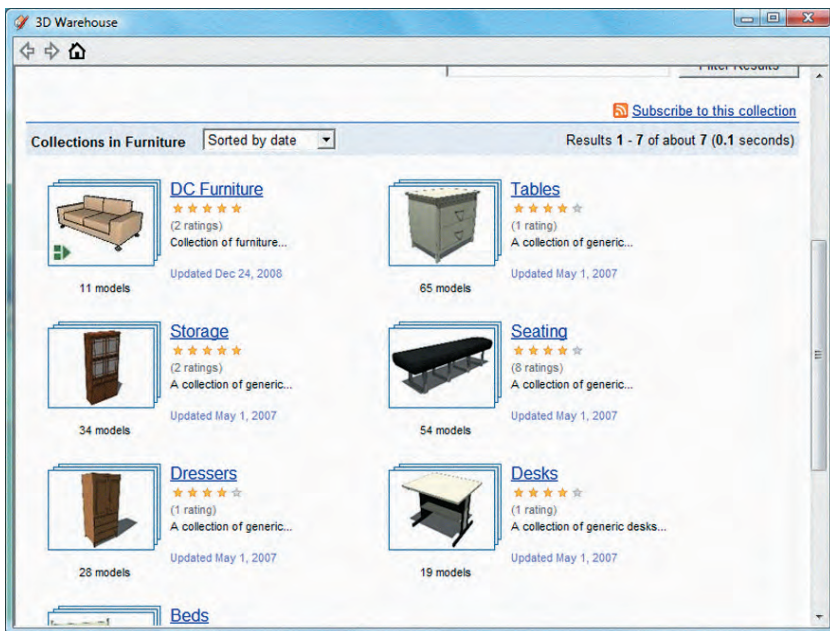
Задание 7. Использование готовой модели

1. Запустите программу Google SketchUp.
2. Выберите в меню **Окно (Window)** команду **Компоненты (Components)**.
Откроется диалоговое окно **Компоненты (Components)** (2-21).



2-21. Диалоговое окно
Компоненты (Components)

3. Щелкните по кнопке **Navigation** со знаком стрелки вниз (▼).
4. Из открывшегося списка выберите пункт **Architecture**. На экране появится архитектурная коллекция.
5. Из списка выберите, например, коллекцию мебели **Furniture**. Откроется набор трехмерных моделей (2-22).



2-22. Коллекция мебели

6. Выберите одну из этих коллекций, например **Beds** (Кровати). Откроется несколько готовых моделей.
7. Для того чтобы выбрать одну модель, например, модель **Twin size bunk bed** (Двухъярусная кровать), щелкните мышью по ее названию. На экране появится модель двухъярусной кровати (2-23).
8. Щелкните по кнопке **Download Model**. Откроется диалоговое окно **Load Into Model?**.
9. Щелкните по кнопке **Yes** (Да). Модель появится в окне SketchUp.
10. Перенесите модель в любое место поля и щелкните левой кнопкой мыши. Модель разместится на поле, и вы сможете работать с ней.



2-23. Двухъярусная кровать

Среди готовых моделей в коллекции вы можете увидеть очень сложные. Может возникнуть вопрос: как созданы эти модели? Для создания их с нуля необходимо много времени, терпения и умений.

В программе SketchUp можно создавать модели и на основе фотографий. Каждый из вас, вероятно, когда-то фотографировал, если не цифровой камерой, то хотя бы мобильным телефоном. SketchUp имеет возможность импортировать изображения в формате JPG, TIFF, PNG и PDF. Фотографии можно использовать в программе SketchUp несколькими способами. «Приклеив» импортированные фотографии на плоскости, можно создать реальные фотомодели зданий и других объектов. Фотографии незаменимы при создании моделей очень сложных объектов.



Дословный перевод слова «фотография» с греческого означает «пишу светом». Датой создания фотографии считается 19 августа 1839 года.

Фотография модели, которую вы хотите создать, импортируется на поле рисования программы. Потом прорисовывают контуры объекта на рисунке.

Этот метод удобен еще и тем, здесь сохраняются пропорции между элементами объекта.

В конце раздела познакомимся с технологией создания модели фигуры, которая часто используется для создания различных моделей. Для создания конуса или трехмерной модели пирамиды в программе SketchUp необходимо уметь работать со следующими инструментами:

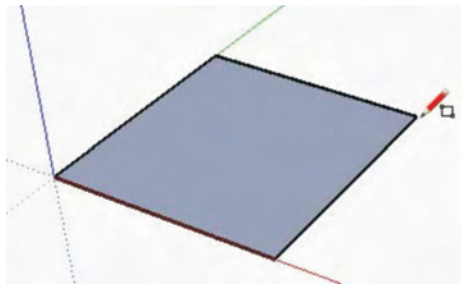
- Rectangle (Прямоугольник)
- Move (Переместить/Копировать)
- Line (Линия)

Что такое пирамида? Какие еще трехмерные геометрические фигуры вам известны?

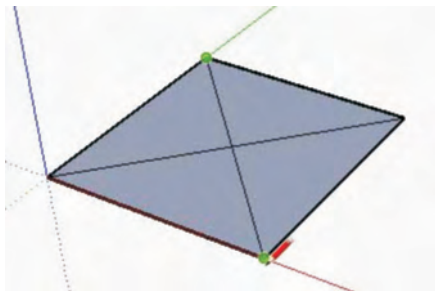


Задание 8. Создание пирамиды

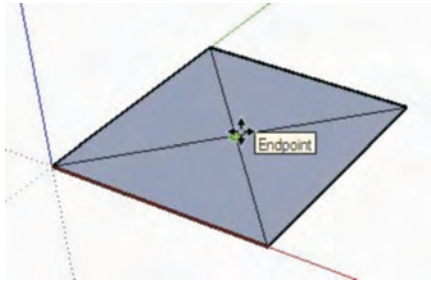
1. Используя инструмент  Прямоугольник, нарисуйте квадрат с вершиной в начале системы координат. Если прямоугольник примет форму квадрата, его диагональ будет показана пунктирной линией, а на экране появится слово «квадрат».



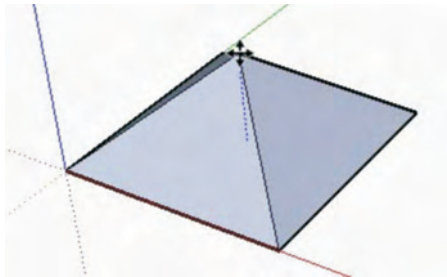
2. Используя инструмент  Линия, нарисуйте диагонали квадрата (другими словами, соедините его вершины).



3. Выберите инструмент  Переместить/Копировать.
4. Установите указатель на пересечении диагоналей. На экране появится надпись «конечная точка» .



5. Щелкните левой кнопкой мыши.
6. Перетащите мышь вверх вдоль синей стрелки. Получив необходимую высоту, отпустите кнопку мыши.



Вопросы и задания

1. Что такое модель и моделирование?
2. Почему иногда создают несколько моделей одного реального объекта?
3. Можно ли басни считать моделями? Если да, то к какому типу моделей их можно отнести?
4. Приведите примеры различных моделей по предметам, которые вы проходите в школе.
5. Что такое трехмерная графика?
6. Какие графические примитивы используют в программе Paint?
7. Как в графическом редакторе осуществляется перетаскивание мыши?
8. На основе какого принципа просматривают модели, созданные в графическом редакторе Google SketchUp?
9. Какие свойства имеют ребра?
10. Что такое грань и какими свойствами она обладает?
11. Если грани не имеют толщины, то как создаются утолщенные поверхности?
12. Что происходит при стирании одной грани? Объясните причину.
13. Метод создания стула, с которым вы познакомились на уроке, называется «методом вычитания». Как по-вашему, почему этот метод построения назвали так? В какой сфере искусства используется этот метод?
14. Создайте модель стула «методом сложения» (соединения разных частей).
15. Как вы понимаете слово «стиль»?
16. Примените разные стили к одной модели и сравните их с известными вам художественными жанрами.
17. Как меняется ваша тень, когда вы проходите по улице под фонарем?
18. Придайте тень любой модели, потом, изменив дату и время, обратите внимание на изменение тени.
19. В коллекции готовых моделей выберите любую интересующую вас модель и измените ее по своему желанию.

ИСТОРИЧЕСКАЯ СПРАВКА

Глобус – трехмерная модель Земли

По историческим источникам, первый глобус был изготовлен Кратетом Малосским во II веке до н.э. (город Маллос находился в провинции Киликия, область Чукурова современной Турции).

Первые глобусы, изображающие Земной шар, были изготовлены в период «золотого века» ислама (VIII – XIII вв). Наиболее известен глобус, созданный мусульманским географом **Аль-Мямуном** в IX веке. В 1267 году глобус, изготовленный **Джамал ад-Дин Мухаммед ибн Тахир ибн Мухаммед ал-Зайди ал-Бухари**, позднее был вывезен в Пекин.

Глобус, созданный в **1492** году немецким географом и путешественником **Мartiном Бехаймом** (1459-1507), сохранился до наших дней. Диаметр этой модели, названной «Земным яблоком», – 54 см. В настоящее время этот глобус хранится в городе Нюрнберг в Германии.

Самый большой глобус в мире был создан в 1998 году в США. Он имеет диаметр 12,5 метра и весит 25 тонн.



«Земное яблоко» Мартина Бехайма

ПРОВЕРЬ СЕБЯ**Какие из приведенных высказываний истинны?**

1. Графические модели – это один из видов информационных моделей.
2. Если в модели SketchUp не видно ребра, значит его не существует.
3. В программе SketchUp нет специального инструмента для создания грани.
4. В редакторе SketchUp пользуются четырьмя графическими примитивами.
5. Для того чтобы образовать грань, достаточно двух ребер.

Выбери правильный ответ**1. Манекен – это ... модель человека.**

- A. информационная
- B. материальная
- C. графическая
- D. математическая

2. Программа SketchUp является продуктом компании

- A. Microsoft
- B. Adobe
- C. Google
- D. Mozilla

3. В программе SketchUp ... не считается графическим примитивом.

- A. точка
- B. линия
- C. прямоугольник
- D. круг

4. Все модели в редакторе SketchUp созданы на основе

- A. линий и кругов
- B. линий и дуг
- C. ребер и граней
- D. ребер и прямоугольников

ПРОЕКТЫ

ПРОЕКТ 2-1

Найдите в Интернете трехмерные модели для редактора Google SketchUp. Сохраните на своем компьютере предусмотренные для общего пользования модели и просмотрите их в редакторе.

ПРОЕКТ 2-2

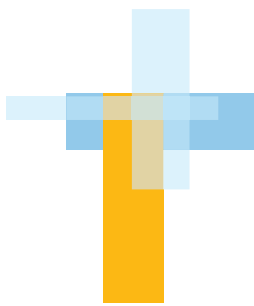
Создайте в редакторе Google SketchUp трехмерную модель вашей школы.

СОВМЕСТНЫЙ ПРОЕКТ

Найдите памятники из мрамора (гранита) или бронзы в том районе, где вы живете. Подготовьте информацию об использовании «метода вычитания» или «метода сложения» при их создании. Включите в доклад известную вам информацию о соответствующем жанре скульптуры.

КРИТИЧЕСКИЙ ВЗГЛЯД

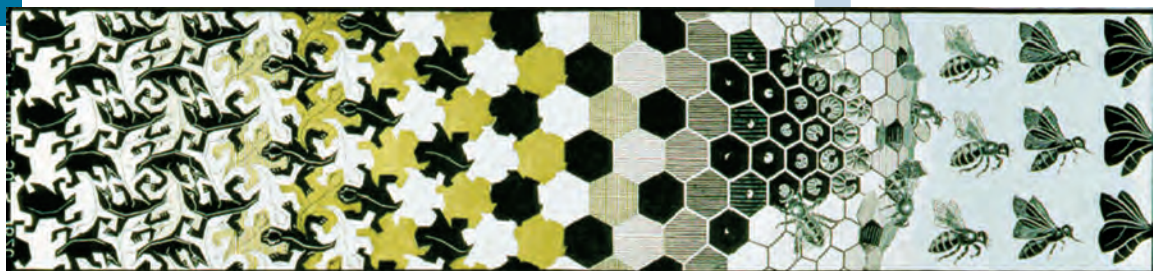
Сравните географическую карту Земли и глобус. Подготовьте в текстовом редакторе небольшой доклад о преимуществах глобуса перед картами. Добавьте в него ваши суждения о недостатках глобуса.



Настольные издательские системы



- Понятие «публикация»
- Как быстро подготовить стенгазету
- Виды публикаций
- Информационный бюллетень и его создание
- Изменение цвета и шрифта публикаций
- Объединение различных текстовых рамок
- Создание буклета



3.1. НАСТОЛЬНЫЕ ИЗДАТЕЛЬСКИЕ СИСТЕМЫ

Можно ли этот учебник назвать публикацией? Если да, то почему?

Когда говорят **издание**, имеют в виду изданную, напечатанную книгу, журнал и т.п. Существует два основных вида издательской продукции – листовое издание и книжное издание.

Буклет



Бланк



Листовка



Листовые издания

Брошюра



Книжное издание

Листовое издание – печатный продукт, состоящий из одного или нескольких не скрепленных друг с другом листов (например, афиша, буклет, бланк, газета, карта, плакат, календарь).

Книжное издание – печатный продукт, состоящий из нескольких скрепленных друг с другом и переплетенных листов (например, брошюра, книга, журнал).

Еще 20-30 лет тому назад профессиональный бюллетень или отчеты могли быть изготовлены только в результате совместной работы нескольких людей. Художники создавали графики и иллюстрации, наборщики набирали тексты, дизайнеры подготавливали макеты листов к печати. Нередко эта технология обходилась очень дорого. Сегодня при помощи персональных компьютеров и недорогого программного обеспечения можно создать хорошо оформленные, профессиональные документы.

Для этой цели используют специальные программы, называемые **настольными издательскими системами**. Они дают возможность на профессиональном уровне подготовить и распечатать издательскую продукцию. Используя эти программы, можно напечатать рекламные буклеты, брошюры, почтовые открытки для поздравления, каталоги, информационные бюллетени и фирменные бланки. На сегодняшний день наиболее распространенными настольными издательскими программами являются **Microsoft Publisher**, **Adobe PageMaker** и **QuarkXPress**. На следующем уроке вы познакомитесь с одной из них – программой **Microsoft Publisher**. Сейчас же рассмотрим некоторые понятия издательского дела.

Для более привлекательного, профессионального вида печатной продукции очень важен правильный выбор шрифта. **Шрифт** – это набор символов одинако-

вого стиля, начертания, толщины и размера. Вероятно, во время работы в текстовом редакторе вы пользовались шрифтами Times New Roman, Arial.

По *стилю* шрифты делят на две группы: с засечками и без засечек.

AaBbCc

Шрифт с засечками

AaBbCc

Шрифт без засечек

Шрифт, верхние или нижние концы символов которого имеют короткие штрихи или узоры, называют *шрифтом с засечками*. *Шрифт без засечек* таких штрихов или узоров не имеет.

Это предложение написано шрифтом с засечками.

Это предложение написано шрифтом без засечек.

Шрифты с засечками обычно используют в больших текстовых блоках, а шрифты без засечек – в заголовках.

Начертания символов в тексте тоже имеет большое значение. В основном используют четыре вида начертания: Нормальный, *Курсив*, **Жирный** и **Жирный курсив**. Курсив применяют в тексте, если хотят на что-то обратить внимание читателя или показать к чему-то личное отношение. **Жирный шрифт** используют для выделения слов и фраз в тексте, а также для написания заголовков в документах.

Наглядность тексту придают и *размеры* символов. В издательстве для указания размера шрифта используют понятие «кегель». Кегль измеряется в *пунктах* (pt). Один пункт соответствует 0,35 миллиметра. Обычно в документах размер используемых шрифтов составляет 12 или 14 pt.

При работе с программами настольных издательских систем необходимо иметь определенные знания о размерах печатного листа.

Комплекс компьютерных аппаратных и программных средств, служащих для предпечатной подготовки документа, состоящего из комбинации текстовых и графических материалов, называют **настольной издательской системой**.

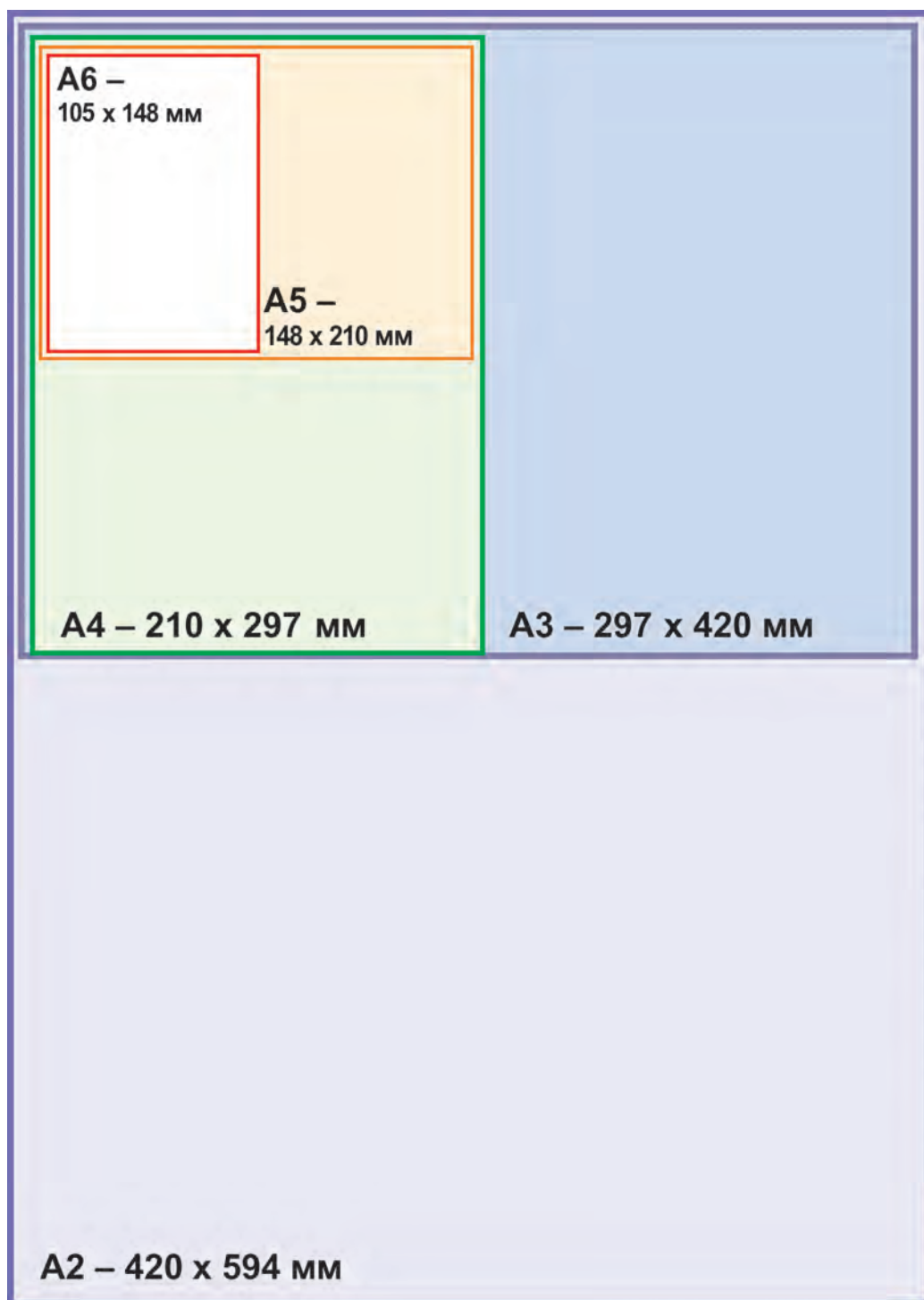


Какие текстовые редакторы вам известны? А какому графическому редактору для работы с рисунками вы отдадите предпочтение?



Arial, Tahoma – шрифты без засечек.
Times, Palatino Linotype – шрифты с засечками.





3-1. Размеры печатного листа

Документы печатают на бумаге стандартного формата. В качестве стандартного принят лист бумаги размером 841×1189 мм и площадью 1 м². Этот формат назван **A0**. Лист меньшего формата получается делением текущего листа на две равные части. То есть площадь каждого последующего листа меньше предыдущей в два раза. Площадь имеющего наиболее широкое применение листа **A4**, составляет $\frac{1}{16}$ м², а размеры – 210×297 мм.

A6 – 105×148 мм

A5 – 148×210 мм

A4 – 210×297 мм

A3 – 297×420 мм

A2 – 420×594 мм

A1 – 594×841 мм

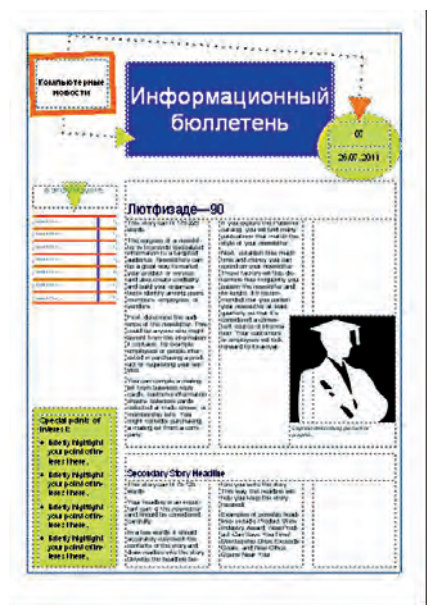
A0 – 841×1189 мм

- 
- ✓ Когда говорят об издании, имеют в виду изданную, напечатанную книгу, журнал и т.п. Существует два основных вида издательской продукции – листовое издание и книжное издание.
 - ✓ Настольные издательские системы дают возможность на профессиональном уровне подготовить и распечатать издательскую продукцию.
 - ✓ Для привлекательного, профессионального вида печатной продукции важен правильный выбор шрифта. Шрифт – это набор символов одинакового стиля, начертания, толщины и размера.
 - ✓ По стилю шрифты делят на две группы: с засечками и без засечек. Шрифты с засечками обычно используют в больших текстовых блоках, а шрифты без засечек – в заголовках.
 - ✓ В тексте в основном используют четыре вида начертания шрифта: нормальный, курсив, жирный и жирный курсив.
 - ✓ Для указания размера шрифта используют понятие «кегель». Кегль измеряется в пунктах (pt). Один пункт соответствует 0,35 миллиметра.
 - ✓ Площадь широко используемого в канцелярии листа A4 составляет $\frac{1}{16}$ м², а размеры – 210×297 мм.

3.2. ИНФОРМАЦИОННЫЙ БЮЛЛЕТЕНЬ

Какого типа публикации имеются в вашей классной комнате?

Любая публикация при изготовлении проходит несколько этапов. Сначала выбирается ее тип. В большинстве издательских систем имеется возможность создания публикаций нескольких видов. В таких программах используются готовые элементы дизайна.



Следующим после выбора типа публикации шагом является выбор дизайна. К дизайну публикации относят форматирование текста, нумерацию страниц, особенности графических элементов. Нет необходимости все эти действия проделывать самостоятельно.

В настольных издательских системах существуют готовые рамки (поля) для текстов. В эти поля помещается текст. Например, в рамке может быть надпись «Имя адресата» или же «Название фирмы». Если вам понадобятся дополнительные поля, вы можете создать их и разместить в них свой текст. Исходные тексты, как правило, создаются при помощи текстовых редакторов.

Во всех настольных издательских программах есть возможность добавлять к тексту графические изображения. После набора текста и выбора графического материала, тексты и рисунки необходимо оптимально разместить на листе для наглядности и привлекательности публикации. Так как каждый текстовый блок и изображение являются отдельными объектами, их очень легко перемещать в нужное место.



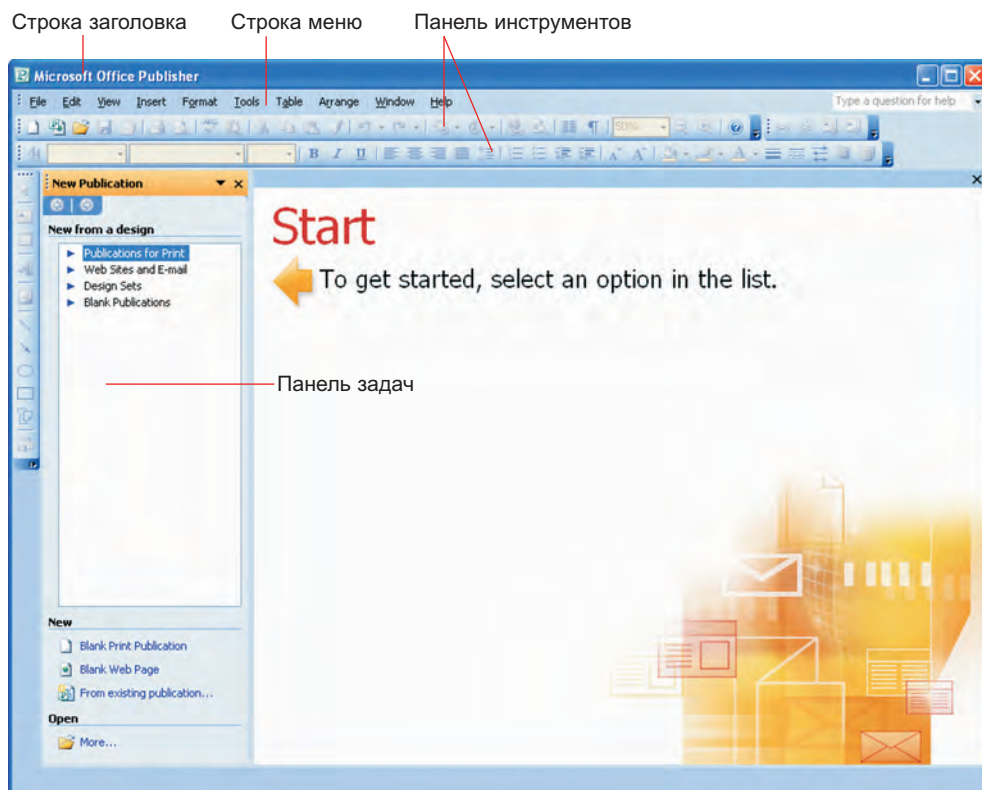
Информационный бюллетень — это краткая официальная информация об имеющих общественное значение событиях, проблеме, работе.



В этом разделе вы познакомитесь с одной из популярных издательских систем — программой **Microsoft Publisher** («Майкрософт паблишер»).

Как видно из названия, программа Publisher предусмотрена для создания публикаций. С ее помощью всего за несколько шагов можно изготовить бюллетени, брошюры, объявления, сертификаты, визитки. Конечно же, перечислен-

ную продукцию можно изготовить и в программах, обрабатывающих текст, но возможности Publisher в этой сфере намного больше. Кроме этого данная программа подходит также для создания веб-страниц.



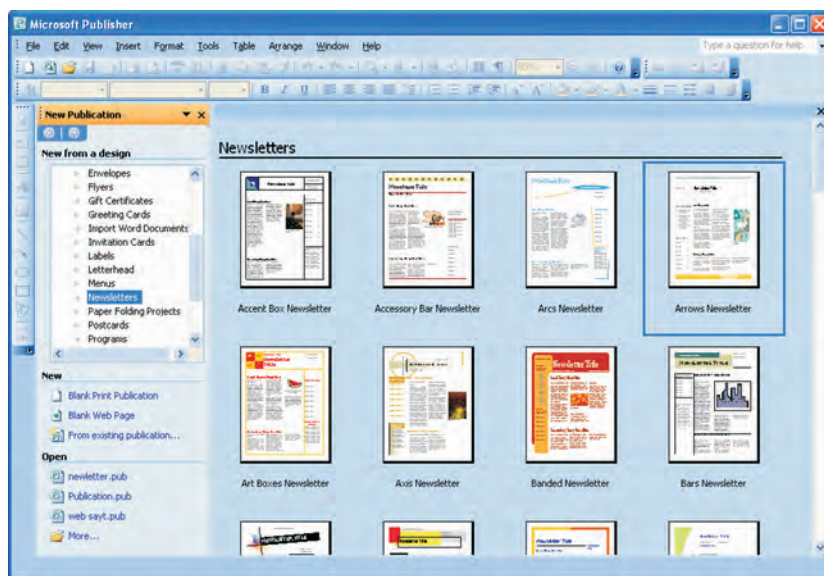
3-2. Главное окно программы Microsoft Publisher

Независимо от того, какая продукция будет создаваться, в программе Publisher нет необходимости начинать работу с нуля. Здесь предлагается много готовых профессиональных макетов для каждой публикации. Достаточно лишь открыть понравившийся макет, после чего нужно просто заполнить места, предусмотренные для текста и рисунков.

Изменив цветовую схему и шрифт, добавив или удалив определенные элементы, можно создать макет, соответствующий вашим требованиям.

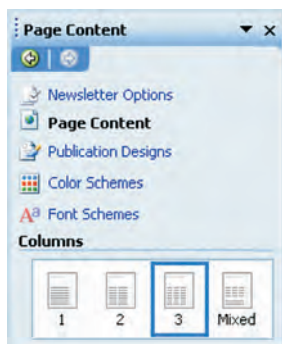
Задание 1. Создание нового бюллетеня

1. Запустите программу Microsoft Publisher (3-2).
2. Из открывшегося списка публикаций в левой части экрана на панели задач **New Publication** выберите сначала вид **Publications for Print**, а затем вид **Newsletters**. В правой части окна появятся образцы разных шаблонов информационных бюллетеней.



3-3. Образцы шаблонов информационных бюллетеней

3. Выберите необходимый шаблон. Запустится Мастер создания информационного бюллетеня.



3-4. Выбор количества столбцов бюллетеня

4. Мастер выведет на экран диалоговое окно **Personal Information**. Щелкните по кнопке **Cancel**. Выберите на находящейся в левой части экрана панели задач **Newsletter Options** пункт **Page Content** и укажите необходимое для вашего документа количество столбцов (3-4).
5. Выполните в меню **File** команду **Save**.
6. Дайте название файлу. Программа Publisher автоматически сохранит этот файл с расширением **.pub**.
7. Щелкните по кнопке **Save**.

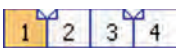
В Microsoft Publisher вы можете использовать для своего информационного бюллетеня специальные рамки — как текстовые, так и графические. Щелкнув по одной из рамок бюллетеня, можно изменить заголовок, текст или рисунок.

Для того чтобы изменить текст, находящийся в определенной текстовой рамке, необходимо щелкнуть левой кнопкой мыши по этой рамке и с клавиатуры ввести свой текст. Новый текст заменит предложенный в шаблоне бюллетеня.

Все элементы публикации, создаваемой в программе Publisher, в том числе и текстовые блоки, независимы друг от друга. Можно расположить любой элемент в необходимом месте или изменить его размеры, форму, внешний вид.



Задание 2. Заполнение текстовых рамок

1. На стандартной панели инструментов щелкните по кнопке  **Zoom In**. Бюллетень, с которым вы будете работать, увеличится в размерах.
2. Щелкните по текстовой рамке **Lead Story Headline** и с клавиатуры введите заголовок основной статьи.
3. Ознакомьтесь с рекомендациями, помещенными в поле основной статьи, и щелкните по рамке. Теперь эти рекомендации можно заменить своим текстом. Точно так же вы можете заменить тексты в других текстовых рамках.
4. Для изменения размеров текстовой рамки щелкните один раз по текстовому полю и наведите указатель мыши на один из небольших белых *маркеров*, расположенных по периметру текстовой рамки. Указатель изменит свою форму и примет вид двусторонней стрелки. **Удерживая левую кнопку мыши, протяните рамку.** В зависимости от направления движения рамка будет увеличиваться или уменьшаться в размерах.
5. Для перехода на другую страницу бюллетеня щелкните по значку , расположенному в нижней части окна программы.
6. Сохраните информационный бюллетень в памяти компьютера.

3.3. РАБОТА С ТЕКСТОВЫМИ РАМКАМИ

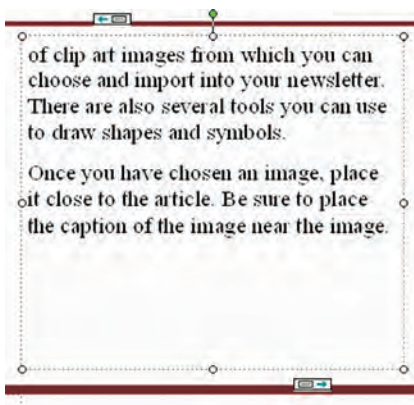
Что имеют в виду под редактированием текста?

Как вы уже знаете, в настольных издательских программах для текстов имеются готовые рамки (блоки). Эти рамки заполняются текстами. Исходные тексты, как правило, создаются при помощи текстовых редакторов. Так как каждый текстовый блок и каждое изображение не зависят друг от друга, их очень просто можно расположить в нужном месте.

Текст может не поместиться в заданной рамке. В этом случае он автоматически переходит в другую рамку, прикрепленную к первой. Взаимосвязанные текстовые рамки могут удлиняться на нужное количество страниц и состоять из нескольких колонок.

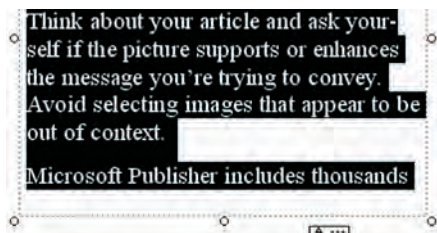
Задание 3. Соединение и разъединение текстовых рамок

1. Если несколько текстовых рамок соединены друг с другом, то есть в них расположен текст одной статьи, то в верхнем левом или нижнем правом углу поля, объединяющего эти рамки, отображаются кнопки  Go to Previous Text Box и  Go to Next Text Box. Щелкните по любой из них и узнайте, какие именно рамки соединены друг с другом.



2. Чтобы разъединить взаимосвязанные рамки, необходимо щелкнуть сначала по одной рамке, а потом по кнопке  Break Forward Link на панели инструментов. Текст отделенной рамки поместится в специальное поле Text in Overflow. Это поле не видно на экране. В нижней части текстовой рамки появится значок , который показывает, что часть текста скрыта.

3. Увеличьте текстовую рамку или переместите скрытую часть текста на другое поле.



4. Для того чтобы поместить продолжение текста в другую рамку, щелкните сначала по рамке, в которую помещен основной текст, а затем по кнопке  **Create Text Box Link** на панели инструментов. Указатель мыши примет форму кружки с «текстом внутри» ().
5. Чтобы разместить избыточный текст в пустой текстовой рамке, переместите кружкообразный указатель мыши в эту рамку и щелкните левой кнопкой. Если импортируемый файл не поместится в рамку целиком, на экране появится запрос об автоматическом использовании функции перемещения текста.

К информационному бюллетеню можно добавлять необходимое количество текстовых полей.

Задание 4. Создание нового текстового поля

1. Щелкните по кнопке  **Text Box**, расположенной вдоль левого края главного окна на панели инструментов **Objects**.
2. Переместите указатель мыши на нужное место бюллетеня и, удерживая левую кнопку мыши, нарисуйте новую рамку.



3. Для перехода в режим редактирования щелкните в произвольном месте текстовой рамки.
4. Если необходимо, измените стиль текста (шрифт, размер символов, начертание букв), используя панель инструментов форматирования, а затем начните ввод текста с клавиатуры.

Некоторые части бюллетеня могут оказаться вам не нужными.

Задание 5. Удаление текстовых рамок

1. Для удаления одной текстовой рамки установите указатель мыши на ней и щелкните правой кнопкой. Из открывшегося контекстного меню выберите команду **Delete Object**.
2. Для удаления нескольких объектов выделите их методом «лассо». Для этого установите указатель мыши в стороне от этих объектов и, удерживая левую кнопку мыши, двигайте ею так, чтобы она охватила рамкой все ненужные объекты.
3. Нажмите клавишу <Delete> или установите указатель мыши на группе выделенных объектов и щелкните правой кнопкой. Из открывшегося контекстного меню выберите команду **Delete Object**.

3.4. ИНФОРМАЦИОННЫЙ БУКЛЕТ



Слово «буклет» происходит от латинского «bouclett», которое переводится как «колечко». Обычно буклеты изготавливают на листах формата А4 или А3 и печатают на мелованной бумаге.

Ознакомьтесь с переводом слова «буклет», данным в тексте. Как по-вашему, почему подобное издание так названо?

Буклеты — одна из наиболее популярных малых полиграфических форм. Это лист с двумя параллельными сгибами. Такая форма делает буклет более лаконичным и делит информацию на блоки.

Буклеты — это идеальное средство для информирования клиентов о предлагаемых товарах и услугах на выставках, презентациях или во время рекламных кампаний. В грамотно составленном буклете можно найти ответы на основные вопросы, интересующие клиента.

Для разработки дизайна буклета в основном используют следующие элементы: логотип, контактная информация, графический материал и тексты.



Задание 6. Создание информационного буклета

1. Запустите программу **Microsoft Publisher**.
2. Из списка вида изданий, открывшегося на панели задач **New Publication** в левой части экрана, выберите сначала тип **Publications for Print** (Публикации для печати), а потом **Brochures** (Буклеты). В правой части окна появятся образцы разных шаблонов информационных буклетов.
3. Выберите нужный шаблон. Запустится Мастер создания информационных буклетов.
4. Мастер выведет на экран диалоговое окно **Personal Information**. Щелкните по кнопке **Cancel**.
5. Если вы собираетесь отправлять буклет по почте, на панели задач **Brochures Options** в разделе **Customer** щелкните по варианту **Include** (3-5).
6. Если в буклет необходимо добавить какую-то форму, в разделе **Form** выберите **Order form** (Форма запроса), **Response form** (Форма ответов) или же **Sign-up form** (Форма подписи).
7. Замените тексты в текстовых рамках на свои.
8. Если вас не устраивает цветовая схема буклета, можете изменить ее. Для этого выбрав на панели задач строку **Brochures Options**, щелкните по ссылке **Color Schemes**. Так же, как и в бюллетенях, вы можете выбрать одну из предлагаемых цветовых схем или создать свою, пройдя по ссылке **Custom color scheme**.



3-5. Панель задач
Brochures Options

Часто размер текста регулируется автоматически таким образом, чтобы было заполнено все текстовое поле. Однако размеры текста можно менять и вручную.

Задание 7. Изменение размеров текста в текстовом поле

1. Щелкните по текстовой рамке.
2. В меню **Format** выберите пункт **AutoFit Text** и дайте команду **Do Not AutoFit**.
3. Выделите текст и из списка **Font Size** на панели инструментов форматирования выберите требуемые размеры текста.

Задание 8. Замена рисунков

1. Установите указатель мыши на рисунке, который хотите заменить, и щелкните правой кнопкой мыши. В открывшемся контекстном меню выберите пункт **Change Picture**.
2. Выберите папку, в которой находится новый рисунок.
3. Выберите рисунок, который хотите разместить в буклете, и дважды щелкните по нему. Программа Publisher автоматически установит необходимые размеры рисунка.
4. При необходимости эту процедуру можно повторить и для других рисунков.

Если команда **Change Picture** не отображается в меню, значит рисунок, возможно, сгруппирован с другими объектами. В этом случае щелкните по рисунку правой кнопкой мыши и в открывшемся контекстном меню выберите команду **Ungroup**. Затем для выделения рисунка, который будет заменен, опять щелкните правой кнопкой мыши. В открывшемся меню выберите пункт **Change Picture**.

После придания буклету необходимого вида сохраните его в памяти компьютера. Для этого в меню **File** выберите команду **Save As**.

Вопросы и задания

1. Какие виды публикаций вы знаете?
2. Что из себя представляют настольные издательские системы?
3. Что такое шрифт и какими параметрами он характеризуется?
4. Какой шрифт использован в заголовках и основных текстах этого учебника?
5. Какой размер имеет лист формата А2?
6. Что такое информационный бюллетень и как его создают в программе Microsoft Publisher?
7. Подготовьте бюллетень на определенную тему по любому учебному предмету.
8. Используя макеты, предложенные в Publisher, подготовьте информационный бюллетень о вашем любимом времени года.
9. Создайте информационный бюллетень на интересующую вас тему.
10. Что такое буклет и для каких целей его используют?
11. Где вы видели буклеты?
12. Подготовьте буклет на любую тему.

ИСТОРИЧЕСКАЯ СПРАВКА

Книгопечатание

Книги в том виде, в котором мы их видим сегодня, начали печатать в средние века. Основоположителем книгопечатания считают **Иоганна Гуттенберга** (1400-1468) из немецкого города Майнца. Но возраст примененной и усовершенствованной им техники печати более древний.

Еще в 175 году до нашей эры придворные ученые китайского императора нашли способ сохранения для потомков важных указов, стихов, исторической информации. Тратя много времени и сил, они высекали необходимые тексты на камне. Затем к камню прикладывали увлажненную бумагу, которую потом покрывали тушью. Вырезанные письменные символы оставались белыми на темном фоне бумаги. Таким образом китайцы могли делать необходимое количество оттисков и распространять их по всей империи.

Впоследствии китайцы вместо камня стали использовать деревянные дощечки. Каждый знак на дереве был вырезан в перевернутом виде и был выпуклым. Потом выпуклую часть закрашивали и прижимали к бумаге. Китайцы уже в VII веке таким способом печатали не только религиозные и научные тексты, но и игральные карты, календари и даже бумажные деньги. Напечатанная подобным образом в 868 году книга «Алмаз сутра» сохранилась до наших дней.



«Алмаз сутра» (868 г.) хранится в Британском музее

ПРОВЕРЬ СЕБЯ**Какие из следующих высказываний верны?**

1. Microsoft Word является настольной издательской системой.
2. В больших текстовых блоках обычно используют шрифт с засечками.
3. Площадь листа формата A1 составляет 0,5 м².
4. Microsoft Publisher является инструментом программирования.
5. Слово «буклет» в переводе с французского означает «прямоугольник».

Выбери правильный ответ**1. ... не относится к книжному типу издания.**

- | | |
|------------|--------------|
| A. брошюра | B. журнал |
| C. газета | D. календарь |

2. ... являются настольными издательскими системами.

- A. Microsoft PowerPoint, Microsoft Publisher
- B. QuarkXPress, Adobe PageMaker
- C. Adobe PageMaker, Microsoft Word
- D. Microsoft Publisher, Microsoft Word

3. Шрифты по стилю делятся на ...

- A. нормальный и жирный
- B. нормальный и курсив
- C. Times и Arial
- D. с засечками и без засечек

4. Площадь листа формата A2 составляет

- | | |
|-----------------------|-----------------------|
| A. 1/2 м ² | B. 1/4 м ² |
| C. 1 м ² | D. 2 м ² |

5. В программе Microsoft Publisher невозможно создать

- | | |
|----------------------|-----------------|
| A. графические файлы | B. буклеты |
| C. бюллетени | D. веб-страницы |

ПРОЕКТЫ

ПРОЕКТ 3-1

Используя текстовый редактор или настольную издательскую программу, напишите короткий рассказ о себе (имя, возраст, класс, чем интересуетесь). Добавьте к тексту иллюстрации. Окончив работу, распечатайте ее на принтере. К какому виду публикации относится ваша работа?

ПРОЕКТ 3-2

Используя возможности программы Publisher, создайте календарь. Выберите для этой цели вид публикации Calendars.

СОВМЕСТНЫЙ ПРОЕКТ

Соберите информацию о визитных карточках и их видах. С помощью программы Publisher (Invitation Cards) создайте для себя личную и официальную визитки. Обсудите в группе, соответствуют ли ваши визитки требованиям.

КРИТИЧЕСКИЙ ВЗГЛЯД

Вам поручено выполнить на компьютере следующее:

- написать письмо;
- составить расписание уроков;
- произвести вычисления;
- подготовить настенный календарь;
- создать визитку.

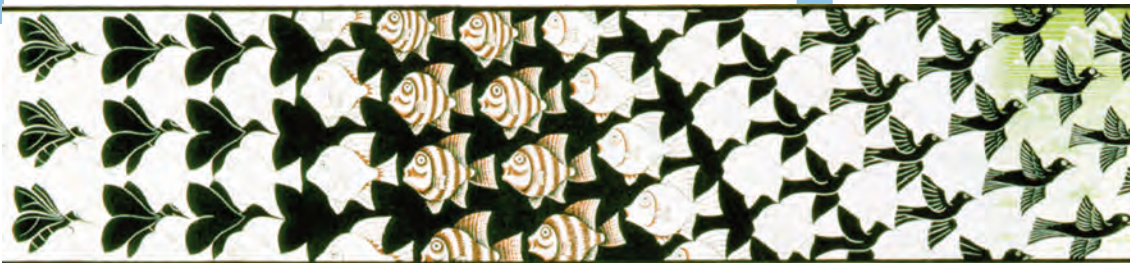
Какие из известных вам программ вы будете использовать при выполнении каждого из этих заданий? Составьте таблицу в подходящей программе, записав в первом столбце названия перечисленных заданий, в другом – необходимое для их выполнения программы.

4

АЛГОРИТМИЗАЦИЯ



- Понятия алгоритма и алгоритмизации
- Основные свойства алгоритма
- Основные виды алгоритмов с точки зрения структуры
- Примеры линейных, разветвленных и циклических алгоритмов
- Нахождение максимального и минимального элемента в последовательности чисел
- Упорядочение последовательности чисел в порядке возрастания или убывания



4.1. ПОНЯТИЕ АЛГОРИТМА

Гюнель и Фатима испекли пирог по рецепту из кулинарной книги. Но несмотря на то, что каждая из них использовала качественные продукты, пирог Фатимы получился вкуснее. Почему, по-вашему мнению, так случилось?



Человек постоянно сталкивается с задачами разной сложности, которые необходимо решить. Некоторые из них настолько трудны, что над их решением приходится очень долго думать. Порой же решение найти так и не удастся. А иногда эти задачи настолько простые, что решаются моментально. Задача «Поступить в университет» намного сложнее, чем задача «Купить

мороженое», и реализация ее требует выполнения большего количества действий. Но вместе с тем даже решение очень простой задачи состоит из нескольких последовательных этапов или требует последовательного выполнения определенных действий. Если заранее составить план действий, то достичь намеченной цели будет намного легче.

Обычно мы выполняем привычные действия не задумываясь, механически. Например, каждый из вас знает, как открывать замок ключом, но если вы захотите научить этому маленького ребенка, придется объяснить ему каждое действие и последовательность их выполнения.

1. Достань ключ из кармана.
2. Вставь ключ в замочную скважину.
3. Поверни ключ два раза по часовой стрелке.
4. Вынь ключ из замочной скважины.



Если ваша цель – открыть замок двери, то, выполнив последовательно перечисленные действия, вы достигнете ее.

Алгоритм – это последовательность действий, которые приводят к решению поставленной задачи. Процесс разработки алгоритма для решения задачи называется **алгоритмизацией**.

Для того чтобы какая-либо последовательность действий называлась алгоритмом, она должна обладать следующими свойствами:

1. *Определенность*: каждый шаг алгоритма должен быть ясным, точным, однозначным.
2. *Массовость*: алгоритм может быть применим к разным наборам исходных данных.
3. *Результативность*: алгоритм должен завершиться получением результата.

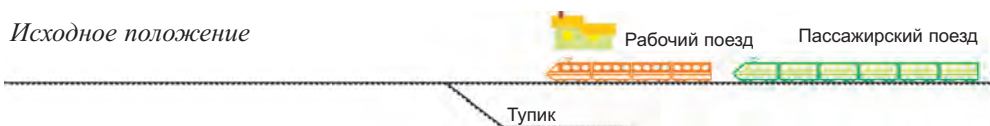


Был проведен анализ ошибок, допущенных во многих сложных с технической точки зрения проектах XX столетия (создание новых автомобилей, зданий, самолетов и т.д.). Результаты анализа показали, что в основном это были **алгоритмические ошибки**, то есть причиной их возникновения являлась неправильная последовательность действий в алгоритмах.



Задача. На небольшой станции однопутной железной дороги остановился поезд в составе тепловоза и трех вагонов, доставивший бригаду рабочих для строительства второго пути. Пока же на этой станции имеется небольшой тупик, где при необходимости может поместиться тепловоз с вагоном или два вагона. Вскоре следом за поездом со строительной бригадой к той же станции подошел пассажирский поезд. Как пропустить пассажирский поезд?

Исходное положение



Требуемое положение

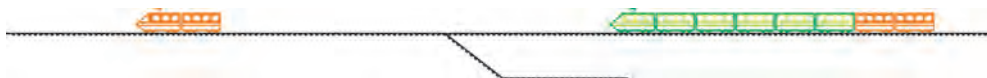


Решение.

Шаг 1. Рабочий поезд доезжает по основному пути до начала тупика. Затем, дав задний ход, заходит в тупик. Два последних вагона отсоединяются, а сам тепловоз с одним вагоном возвращается на основной путь и продолжает движение вперед.



Шаг 2. Пассажирский поезд движется до начала тупика. Два вагона рабочего поезда, стоящие на запасном пути, присоединяются к последним вагонам пассажирского поезда и поезд выводит их на основной путь. Затем пассажирский поезд, двигаясь назад, освобождает путь к тупику.



Шаг 3. Рабочий поезд (тепловоз и вагон), двигаясь назад, заходит в тупик.



Шаг 4. Пассажирский поезд отцепляет два последних вагона рабочего поезда и по свободному пути продолжает движение.



Шаг 5. Рабочий поезд (тепловоз и вагон) выезжает из тупика. Затем, двигаясь назад, подъезжает к своим вагонам и, присоединив их, возвращается на свою первоначальную стоянку.

Эта последовательность шагов удовлетворяет всем трем свойствам алгоритма:

1. *Определенность*: каждый шаг показан ясно, например, «начинает движение от начала тупика», «отсоединяются два вагона» и т.д.
2. *Массовость*: у пассажирского поезда может быть разное количество вагонов, вместо пассажирского поезда можно указать грузовой и т.д.
3. *Результативность*: после выполнения данных шагов получен ожидаемый результат, то есть пассажирский поезд продолжит движение по своему пути.

Слово «алгоритм» происходит от имени **Абу Абдуллах Мухаммеда ибн Муса ал-Хорезми**. Этот ученый в сочинении «**Китаб ал-джабр ва-л-мукабала**» в 820 году впервые дал описание позиционной десятичной системы счисления. Ал-Хорезми сформулировал правила вычислений в новой системе и впервые использовал цифру 0 для обозначения пропущенной позиции в записи числа. В первой половине XII века книга ал-Хорезми в латинском переводе попала в Европу. Переводчик дал ей название «*Algoritmi de numero Indorum*» («Алгоритмы о счете индийском»). Таким образом слово «алгоритм» попало в европейские языки именно благодаря этому сочинению.



Каждую последовательность действий нельзя назвать алгоритмом. Что произойдет, если в алгоритме «Открытие замка», поменять местами шаги 2 и 3? Будет ли достигнута цель?



Из курса математики вы знакомы со многими алгоритмами: сложение, вычитание, умножение, деление многозначных чисел, схожие действия над обычными дробями, нахождение НОД, НОК и т.д.

- ✓ Алгоритм – это последовательность действий, которые приводят к решению поставленной задачи.
- ✓ Слово «алгоритм» происходит от имени ученого Абу Абдуллах Мухаммеда ибн Муса ал-Хорезми, жившего в IX веке. Сочинение, которое он написал в 820 году, – «Китаб ал-джабр ва-л-мукабала», в первой половине XII века было переведено на латинский язык как «*Algoritmi de numero Indorum*» («Алгоритмы о счете индийском»).
- ✓ Для того чтобы последовательность действий считалась алгоритмом, она должна обладать свойствами определенности, массовости и результативности. Другими словами, 1) каждый шаг алгоритма должен быть ясным, точным, однозначным; 2) алгоритм, как правило, может быть применен не только для одних значений исходных данных, он может быть также использован и для других значений; 3) алгоритм должен приводить к получению результата задачи.



4.2. ОСНОВНЫЕ АЛГОРИТМИЧЕСКИЕ КОНСТРУКЦИИ.

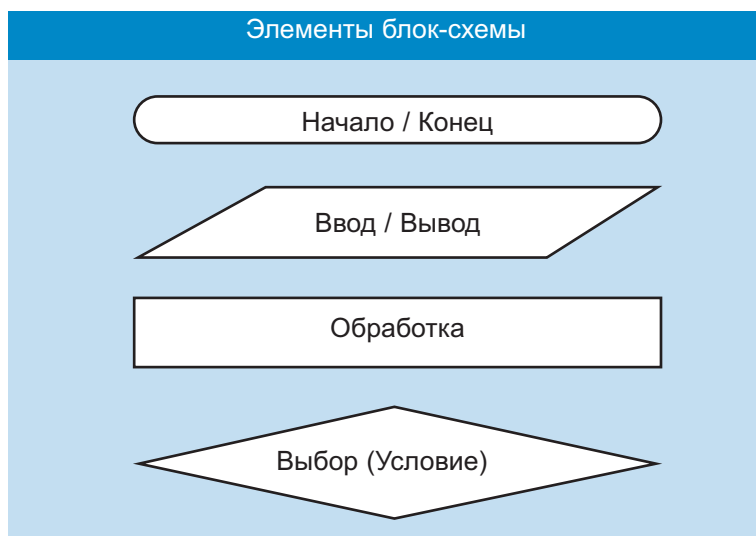
ЛИНЕЙНЫЕ АЛГОРИТМЫ

Почему для записи алгоритмов не всегда подходят естественные языки?

Алгоритмы можно представлять в различной форме: на естественных, специальных языках, посредством формул, геометрических фигур. Выбор формы представления алгоритма зависит от характера задачи. Алгоритмы, связанные с вычислениями, удобнее представлять с помощью формулы или последовательности формул. А алгоритм, например, заваривания чая – словесно при помощи пронумерованных предложений, то есть пошагово.

Самая простая форма представления алгоритма – **словесная**. Эта форма эффективна, когда необходимо передать смысл алгоритма. Однако не всегда идею можно точно сформулировать словами.

Наиболее распространенная форма наглядного представления алгоритма – это задание его графически в виде **блок-схемы**. Это наглядная, относительно компактная и более точная, чем словесная, форма представления алгоритма. Блок-схема состоит из стандартных геометрических фигур.

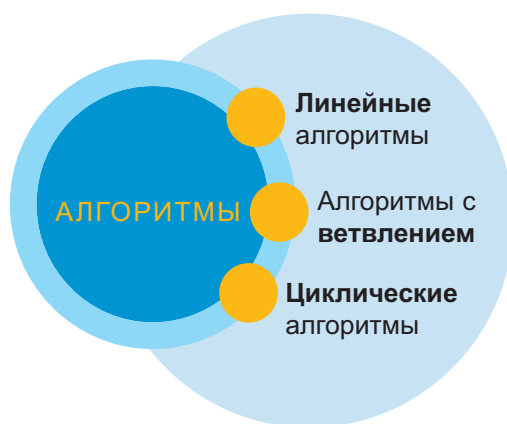


Каждая фигура (элемент) является частью алгоритма. В блок-схеме операции с данными (действия) изображают в виде прямоугольника, проверку условия представляют в виде ромба. Для соединения элементов используют вертикальные и горизонтальные прямые. Чтобы показать направление выполнения дей-

ствий, на конце прямых ставят стрелки. От прямоугольников всегда отходит только одна стрелка, но войти может несколько, от ромба же отходят две стрелки.

Одна из стрелок, отходящих от ромба, помечена словом «да», вторая – словом «нет» (они указывают выполнение – невыполнение условия). Этот способ представления удобен при анализе алгоритма, проверке логической правильности его работы. Овал или скругленный прямоугольник указывает начало и конец алгоритма.

Действия алгоритма показывают последовательно одно за другим, однако выполняются они не всегда в заданной последовательности. Причиной этого является проверка в алгоритме определенных условий. Увеличивая количество условий в алгоритме, можно получать алгоритмы различной структуры. С точки зрения построения алгоритмы можно разделить на три основные группы: *линейные алгоритмы, алгоритмы с ветвлением, циклические алгоритмы.*



Любой сложный алгоритм можно разделить на части, каждая из которых будет относиться к одной из указанных выше групп. Поэтому очень важно знать структуру и принцип построения этих трех типов алгоритмов.

Задача. Имеется два стакана. В одном стакане вода, во втором – молоко. Требуется поменять местами содержимое стаканов, то есть молоко перелить в стакан с водой, а воду – в стакан с молоком.



Понятно, что для решения этой задачи необходим еще один стакан.

Сначала воду из первого стакана выливаем в пустой стакан, потом молоко из второго стакана выливаем в первый стакан и, наконец, из вспомогательного стакана выливаем воду во второй стакан. Как видите, алгоритм решения этой задачи состоит из трех последовательно выполняемых действий.



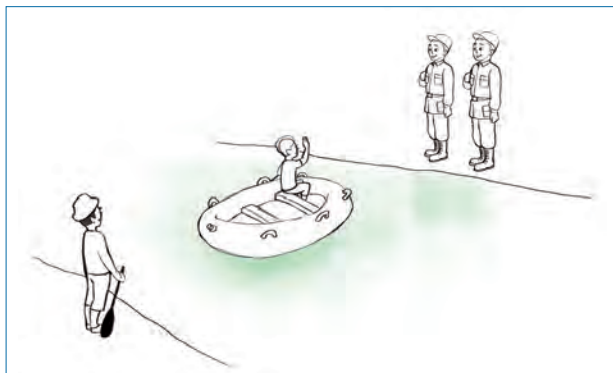
Алгоритм, при выполнении которого не пропускается и не повторяется ни один шаг и все шаги выполняются последовательно, называется **линейным алгоритмом**.

и B , то есть значение A присвоить B , а значение переменной B – переменной A можно так:

$$X := A; A := B; B := X.$$

Этот метод перестановки широко используют и в программировании. Если надо присвоить какой-то переменной значение другой переменной, и наоборот, то возникает необходимость в третьей – вспомогательной переменной. Например, поменять местами значения переменных A

Задача о переправе. Двум солдатам необходимо попасть на другой берег реки. Они видят лодку с двумя детьми в ней. Так как лодка небольшая, в нее могут поместиться или двое детей, или один солдат. То есть один ребенок и один солдат или два солдата в лодку не поместятся. Как солдатам переправиться на другой берег?



Решение.

Условно обозначим солдат буквами C1 и C2, а детей – P1 и P2. Направление движения лодки укажем стрелкой. Тогда алгоритм решения задачи можно представить в таком виде:

Шаг	На этом берегу	В лодке	Направление	На другом берегу
1	C1, C2	P1, P2	→	
2	C1, C2	P1	←	P2
3	C2, P1	C1	→	P2
4	C2, P1	P2	←	C1
5	C2	P1, P2	→	C1
6	C2	P1	←	P2, C1
7	P1	C2	→	P2, C1
8	P1	P2	←	C1, C2
9		P1, P2	→	C1, C2

Пять разведчиков должны переправиться через реку. Река очень глубокая, а моста вблизи нет. У берега в лодке сидят два мальчика. Разведчики просят детей переправить их на другой берег. В лодку помещается либо один солдат, либо двое детей. Солдат и ребенок вместе не могут сесть в лодку. Составьте алгоритм переправы солдат через реку. Если движение лодки в одном направлении примем за один рейс, то сколько рейсов как минимум придется сделать, чтобы всех разведчиков перевезти через реку?



- ✓ Алгоритмы можно представлять в разных формах: словесно, посредством формул, геометрических фигур. Выбор способа представления алгоритма зависит от характера задачи.
- ✓ Наиболее распространенная форма наглядного изображения алгоритма – графическая, в виде блок-схемы.
- ✓ Вычисление (действие) в блок-схеме изображается в виде прямоугольника, проверка условия – в виде ромба.
- ✓ С точки зрения построения алгоритмы можно разделить на три основные группы: (1) линейные, (2) алгоритмы с ветвлением, (3) циклические.
- ✓ Любой сложный алгоритм можно разделить на части, каждая из которых будет относиться к одной из основных групп.
- ✓ Все шаги линейного алгоритма выполняются последовательно. Ни один из шагов не пропускается и не повторяется.

4.3. АЛГОРИТМЫ С ВЕТВЛЕНИЕМ. НАХОЖДЕНИЕ МАКСИМАЛЬНОГО И МИНИМАЛЬНОГО ЭЛЕМЕНТА

Какие слова мы используем, когда проверяем условие? В каком случае можно убрать эти частицы, не нарушив общий смысл предложения?

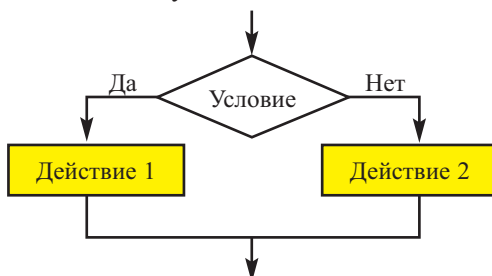


Если в зависимости от условия можно выбрать один из вариантов решения задачи, то такой алгоритм называют **алгоритмом с ветвлением**.

В повседневной жизни редко можно встретить линейные алгоритмы. Гораздо чаще приходится совершать в зависимости от какого-то условия те или иные действия по разным «ветвям» алгоритма. Ветвление в алгоритме используют в двух случаях:

1. В алгоритме имеются шаги, которые при определенных условиях необходимо *пропустить*;
2. В зависимости от определенного условия приходится делать *выбор* из необходимых действий в алгоритме.

В блок-схеме *условие ветвления* алгоритма записывается в ромбе и от него отходят две стрелки. Стрелка «да» указывает, какие действия будут совершаться при выполнении условия, стрелка «нет» – при невыполнении условия. В общем виде ветвление в алгоритме имеет следующий вид:

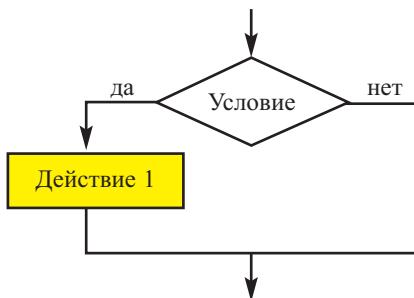
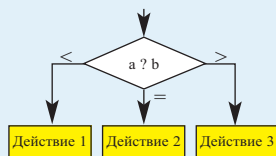


Словесно эту схему можно выразить так: «**если** Условие выполнено, **то** Действие 1, **иначе** Действие 2». Эту форму алгоритма с ветвлением называют *полной*.

Если при невыполнении условия в алгоритме с ветвлением не предусмотрено совершение какого-то действия, то такую форму алгоритма называют *неполной формой* и словесно записывают: «**если** Условие выполнено, **то** Действие 1».



При сравнении двух чисел ветвление в алгоритме удобнее представлять таким образом:



Задача о заключенном. В камере подземелья две двери. Одна из них ведет на свободу, а вторая – к месту казни. Возле каждой двери стоит стражник. Один из стражников всегда лжет, второй всегда говорит правду. Но заключенный не знает, кто из них кто. Ему дана возможность выйти из тюрьмы. Но дверь, ведущую к свободе, нужно найти, задав стражникам лишь один вопрос. Какой вопрос должен задать заключенный?



Решение.

Указав на одну из дверей, нужно спросить у стражника: «Подтвердит ли твой товарищ, что эта дверь ведет на свободу?» Если дверь, на которую показал заключенный, ведет на свободу и стражник, к которому он обратился лгун, то он ответит: «Нет», так как его честный товарищ дал бы на этот вопрос положительный ответ.

Если дверь ведет к смерти, то стражник-лжец ответит: «Да». Рассуждая подобным образом, можно быть уверенным, что стражник-правдивец, видя, что указывают на дверь, ведущую на свободу, даст отрицательный ответ, в противном же случае ответит: «Да».

	Дверь, ведущая на свободу	Дверь, ведущая к смерти
Стражник-лжец	<i>Нет</i>	<i>Да</i>
Стражник-правдивец	<i>Нет</i>	<i>Да</i>

Таким образом, если заключенный, задав указанный выше вопрос любому стражнику, услышит отрицательный ответ, то дверь, на которую он указал, ведет на свободу. Если же он услышит утвердительный ответ, то значит на свободу ведет другая дверь.

Задача о взвешивании. Было изготовлено 8 внешне одинаковых медалей, одна из которых получилась легче остальных. Как без гирь, взвесив на чашечных весах всего два раза, определить медаль, имеющую меньший вес?



Решение.

Разделим медали на три группы: 3+3+2. Сравним группы из трех медалей, положив эти группы на разные чаши весов. Если их вес окажется одинаковым, то медаль, которую мы ищем, – это одна из двух оставшихся. Иначе она в группе, которая весит меньше.

В обоих случаях, взвесив один раз, мы определим группу, в которой находится легкая медаль.

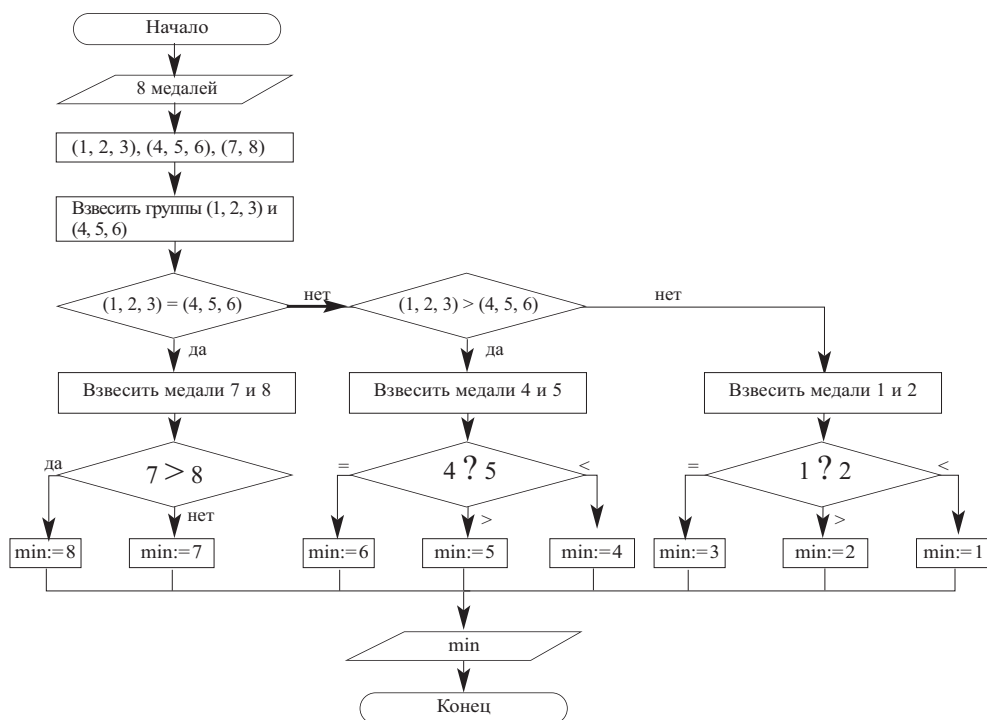
Теперь нужно, взвесив второй раз, определить легкую медаль в группе из двух или трех медалей.

Для наглядности покажем решение задачи с помощью схемы.

Дано: ①②③④⑤⑥⑦⑧

<i>1-е взвешивание</i>	<i>2-е взвешивание</i>	<i>Результат</i>
①②③ = ④⑤⑥	⑦ > ⑧	⑧
①②③ > ④⑤⑥	④ = ⑤	⑥
	④ > ⑤	⑤

Блок-схема этого алгоритма может выглядеть так:



- ✓ Если в зависимости от поставленного условия в алгоритме есть возможность выбрать один из вариантов действий, то такой алгоритм называют алгоритмом с ветвлением.
- ✓ Алгоритмы с ветвлением используют в двух случаях: (1) в алгоритме имеются такие шаги, которые при определенных условиях необходимо пропустить; (2) в зависимости от определенного условия приходится делать выбор из необходимых действий в алгоритме.
- ✓ В блок-схеме условие ветвления записывается в ромбе, и от него отходят две стрелки. Стрелка «да» указывает, какие действия будут совершаться при выполнении условия, стрелка «нет» – при невыполнении условия.
- ✓ Алгоритм с ветвлением имеет две формы: полную и неполную. Полную форму словесно можно выразить так: «если Условие выполнено, то..., иначе...».
- ✓ При невыполнении условия в алгоритме может быть не предусмотрено совершение какого-либо действия. Такую форму алгоритма называют неполной и словесно ее выражают так: «если Условие выполнено, то...».

4.4. ЦИКЛИЧЕСКИЕ АЛГОРИТМЫ. СОРТИРОВКА

Какие виды физической деятельности человека основаны на повторяющихся действиях ?

Многие процессы, происходящие в окружающем нас мире, повторяются. Например, каждый день начинается с рассвета, каждый год происходит смена времен года, зеленеют и цветут деревья.



Циклический алгоритм – это алгоритм, в котором повторяется определенная последовательность действий.

Повторяющиеся действия очень часто встречаются и во время вычислений. Из урока математики вы знаете, что умножение одного числа на другое – это многократное сложение одного и того же числа. Например, чтобы найти произведение 15 и 5, необходимо 5 раз сложить число 15.

$$15 \times 5 = 15 + 15 + 15 + 15 + 15.$$

Составляя алгоритм решения такого типа задач, нередко приходится повторять последовательность определенных действий несколько раз. Конечно, можно записать эту последовательность столько раз, сколько это необходимо. Однако этот путь не очень удобен. Если число повторяющихся действий немалое, то алгоритм получится очень большим. В таком алгоритме легко допустить

ошибку. Кроме этого во многих алгоритмах заранее неизвестно количество повторений – оно определяется по ходу алгоритма. Все эти проблемы решаются при помощи специальной алгоритмической структуры – цикла. Использование цикла существенно сокращает объем алгоритма.

Циклические алгоритмы могут быть двух видов. В первом случае заранее известно количество повторений (с начала цикла). Например, если необходимо перенести 100 кирпичей, то вы знаете, что должны повторить это действие 100 раз. В другом случае действия повторяются до выполнения определенного условия.

Представьте, что вы находитесь в темной комнате и вам необходимо дойти до стены. Заранее вы не знаете,



сколько шагов для этого потребуется сделать. Как в этом случае не столкнуться со стеной? Для того чтобы узнать, дошли вы до стены или нет, нужно вытянуть руку вперед и сделать один шаг. Повторять это действие до тех пор, пока ваша рука не коснется стены. То есть каждый раз, прежде чем делать шаг, необходимо проверять наличие стены и в зависимости от этого делать очередной шаг или остановиться.

Задача на взвешивание. *Имеются четыре арбуза разной массы. Как расположить эти арбузы в порядке увеличения их массы, взвесив их не более 5 раз на чашечных весах, не используя гири.*



Решение.

Алгоритм решения этой задачи пошагово можно представить так:

1. Сравнить первую пару арбузов по весу;
2. Сравнить вторую пару арбузов по весу;
3. Сравнить самый тяжелый арбуз первой пары с самым тяжелым арбузом второй пары. В результате выбрать самый тяжелый арбуз.
4. Сравнить самый легкий арбуз первой пары с самым легким арбузом второй пары. В результате выбрать самый легкий арбуз.
5. Сравнить два оставшихся арбуза. По весу они займут второе и третье место, соответственно.

Имеется 100 шаров разной массы. Как, взвесив их не более 150 раз на чашечных весах без гирь, можно найти среди них самый тяжелый и самый легкий шар?

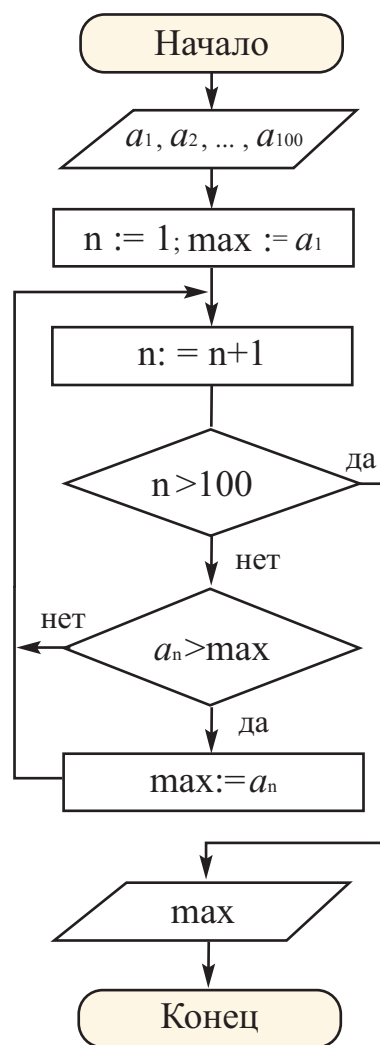




Рассмотрим следующую ситуацию. На рыбалке проводится состязание на то, кто поймает самую крупную рыбу. Сортировать рыбы можно следующим образом: первая пойманная рыба кладется в ведро. Вторая пойманная рыба сравнивается с той, что уже в ведре. Если новая рыба больше, то рыба, которая была в ведре, отпускается обратно в воду, а вновь пойманная кладется в ведро. Если же вторая рыба меньше той, которая в ведре, то она отпускается в воду, а рыба в ведре остается на месте.

Понятно, что в ведре всегда будет находиться самая крупная из пойманных рыб.

Допустим, дана таблица из 100 чисел. Используя алгоритм предыдущей задачи, алгоритм нахождения максимального числа в таблице можно представить в виде блок-схемы таким образом:



Блок-схема алгоритма
нахождения максимального
элемента



- ✓ Алгоритм, в котором повторяется определенная последовательность действий, называется циклическим.
- ✓ Использование цикла существенно сокращает объем алгоритма.
- ✓ Количество повторений в цикле может быть заранее известно или не известно.
- ✓ Циклические алгоритмы используются при нахождении максимального и минимального элементов, а также в задачах сортировки.

Вопросы и задания

1. Что такое алгоритм?
2. Что такое алгоритмизация?
3. Какими свойствами должна обладать последовательность действий, чтобы она называлась алгоритмом?
4. Составь алгоритм задачи «Волк, коза и капуста».
5. В какой форме может быть представлен алгоритм?
6. Каковы основные элементы блок-схемы и что выражает каждый из них?
7. Что такое линейный алгоритм?
8. Приведите примеры линейных алгоритмов.
9. Составьте блок-схему для задачи о переливании (стаканы с водой и молоком).
10. В каких случаях в алгоритме используется ветвление?
11. Какие формы алгоритма с ветвлением вы знаете?
12. Составьте блок-схему алгоритма нахождения наибольшего из двух чисел.
13. Составьте блок-схему алгоритма для задачи о заключенном.
14. В каком случае в алгоритме используют цикл?
15. Высота дерева 20 метров. Каждый день улитка поднимается утром по дереву вверх на 2 метра, а ночью опускается вниз на 1 м. За сколько дней улитка при таком движении доползет до вершины дерева? То есть сколько раз будут повторяться действия улитки вверх-вниз, прежде чем она достигнет вершины?

ИСТОРИЧЕСКАЯ СПРАВКА

Первые алгоритмы

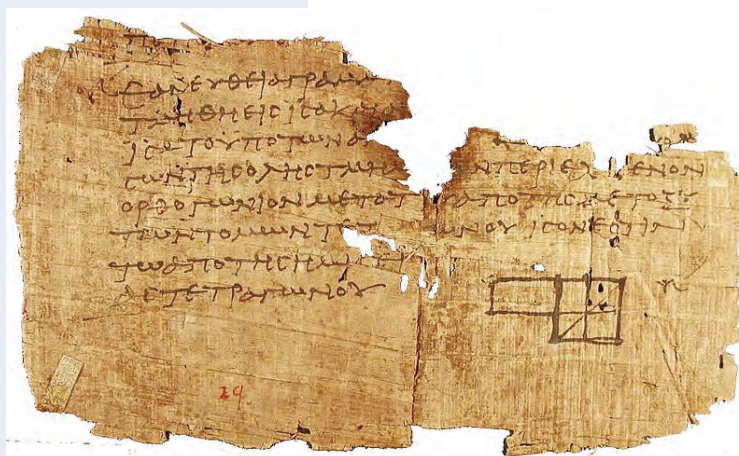
Возраст некоторых математических алгоритмов исчисляется несколькими тысячелетиями. Самым древним оригинальным алгоритмом считается **алгоритм нахождения наибольшего общего делителя (НОД) двух целых чисел**. Этому алгоритму, описанному в седьмой книге «Начала» Евклида, приблизительно 2300 лет. Он известен как **алгоритм Евклида**.

Алгоритм Евклида (нахождение наибольшего общего делителя двух положительных целых чисел). *Предположим, требуется найти наибольший общий делитель двух заданных положительных чисел. Для этого сначала наибольшее число делится на наименьшее, потом второе число делится на полученный остаток, затем первый остаток делится на второй остаток и т.д. В этом процессе последний положительный остаток, отличный от нуля, будет наибольшим общим делителем данных чисел.*

Например. Предположим, $a = 777$, $b = 629$. Тогда $777 = 629 \cdot 1 + 148$,
 $629 = 148 \cdot 4 + 37$, $148 = 37 \cdot 4$.

Последний отличный от нуля остаток 37 и является наибольшим общим делителем чисел 777 и 629.

Справедливости ради следует отметить, что в первоначальной трактовке алгоритма речь шла о последовательном вычитании из большего числа меньшего. Последнее положительное, отличное от нуля число и является наибольшим общим делителем.



Один из древних фрагментов сочинения Евклида «Начала»
 (приблизительно 100 год н.э.).

ПРОВЕРЬ СЕБЯ

Что из перечисленного можно считать алгоритмом?

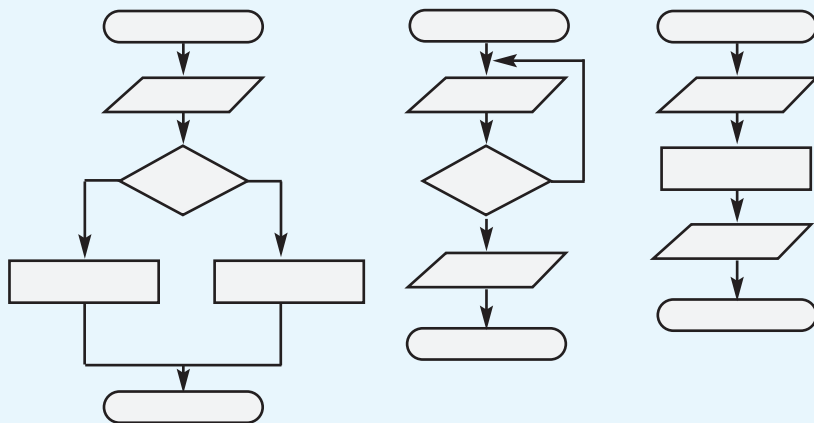
1. Правила безопасного перехода дороги при отсутствии пешеходной дорожки.
2. Набор правил дорожного движения.
3. Правило сложения двух простых дробей.
4. Правила склонения существительных.
5. Перечень продающихся в магазине товаров.

Выбери правильный ответ

1. В блок-схеме условие записывают в

- | | |
|-------------------|-------------------------------|
| A. прямоугольнике | B. скругленном прямоугольнике |
| C. ромбе | D. параллелограмме |

2. К какому виду алгоритма относится вторая блок-схема?



- | | |
|-----------------|-----------------|
| A. линейному | B. с ветвлением |
| C. циклическому | D. смешанному |

3. Алгоритм, в котором все шаги выполняют последовательно, называют

- | | |
|----------------|---------------------|
| A. линейным | B. с ветвлением |
| C. циклическим | D. последовательным |

4. Алгоритм, в котором повторяется определенная последовательность действий, называется

- | | |
|----------------|------------------|
| А. линейным | В. с ветвлением |
| С. циклическим | Д. повторяющимся |

5. Если в алгоритме в зависимости от условия можно выбрать один из вариантов решения задачи, то такой алгоритм называют

- | | |
|----------------|-----------------|
| А. линейным | В. с ветвлением |
| С. циклическим | Д. условным |

ПРОЕКТЫ

ПРОЕКТ 4-1

Два грузовых поезда движатся навстречу друг другу по одному и тому же пути железной дороги. В каждом поезде по 80 вагонов. На станции, где они встретятся, есть тупик, но туда помещается только 40 вагонов и локомотив. Как должны действовать машинисты, чтобы поезда смогли продолжить свой путь?

Составьте и запишите в текстовом редакторе пошаговый алгоритм решения этой задачи.

ПРОЕКТ 4-2

Используя предложенные формулы, составьте алгоритм перехода от календаря Хиджры к Григорианскому (европейскому) календарю, и наоборот.

- $X - X / 33 + 622 = Q$,
- $Q - 622 + (Q - 622) / 32 = X$.

Здесь X показывает год Хиджры, Q – год по Григорианскому календарю. Оформите этот алгоритм в форме отчета. Включите в отчет информацию о календаре Хиджры и григорианском календаре. Выполнив алгоритм, определите, какому году по календарю Хиджры соответствует текущий год.

ПРОЕКТ 4-3

Способ, описанный в задаче о наибольшей пойманной рыбе, используется в алгоритме нахождения наибольшего и наименьшего элементов в последовательности чисел. Используя этот метод, составьте и запишите пошагово алгоритм нахождения наименьшего элемента в последовательности чисел и представьте его в виде блок-схемы.

ИНТЕГРАТИВНЫЙ ПРОЕКТ

Выберите отрывки из произведений, изученных на уроках литературы. Используя предложенный алгоритм, исследуйте эти отрывки:

1. Прочитайте отрывок литературного произведения.
2. Найдите в нем такое событие, которое можно показать в форме алгоритмической конструкции.
3. Представьте ход события блок-схемой.
4. Оформите результат на листе или подготовьте его в текстовом редакторе, а затем распечатайте.

ПРОГРАММИРОВАНИЕ

5



- Отличие понятия объекта от понятия элемента
- Методы объекта
- Объектно-ориентированное программирование
- Основные элементы среды программирования Lazarus
- Понятие формы
- Событие и реакция на него



5.1. ОБЪЕКТЫ И ИХ СВОЙСТВА

Что такое объект?

Как вы знаете, программное обеспечение компьютеров делят на три класса: системные программы, прикладные программы и инструментарии программирования. С программами, относящимися к первым двум классам, вы уже знакомы. Это Windows, Calculator, Paint, Notepad, WordPad и др.



К какому классу программ относятся программы Windows, Paint, Notepad, WordPad, MS DOS?

В этом разделе рассматривается программное обеспечение, относящееся к третьему классу. Точнее, вы познакомитесь со средой программирования **Lazarus**, в основе которой лежит один из инструментариев программирования – язык **Pascal**. Для того чтобы лучше понять принципы работы в этой программе, рассмотрим одно понятие.

Вы часто слышите и употребляете в разговоре слово «*объект*», особенно на уроках информатики. То, на что люди обращают внимание, они именуют объектом. Объект может быть и одушевленным и неодушевленным. События или процессы, которые происходят в природе, также можно рассматривать как объекты. Все эти объекты физические, их изучают на уроках физики, химии, биологии, географии.

А можно ли назвать объектами буквы, слова, предложения, знаки препинания, которые вы изучаете на уроках родного языка? Являются ли объектами используемые на уроках математики числа, математические действия, формулы? Можно ли рассматривать как объект таблицы, графики, диаграммы, схемы, рисунки? Конечно же, они тоже объекты. Но в отличие от физических объектов это *информационные объекты*.



Объектом может быть практически все. Если мы наблюдаем за чем-то или, изучая что-то, выделяем его свойства, значит это объект. Если мы не можем выделить свойства чего-то, то оно не является объектом наблюдения и изучения. Но, если что-то рассматривая под другим углом зрения, все-таки можно найти у него отличительные свойства, то это можно считать объектом.

Наблюдая за облаками, порой можно увидеть стаю птиц. Если нас не интересуют ее особенности (или не различаем стаю в небе), то в данный момент стая птиц не является для нас объектом наблюдения. Но если наблюдать за стаей, то можно выделить ее свойства (форму стаи, высоту полета, направление и скорость, количество птиц и т.д.). В этом случае стая станет для нас объектом, а каждая птица в стае будет не объектом, а элементом стаи. Но если обратить внимание на определенную птицу и выделить ее особенности, то объектом наблюдения будет конкретная птица в стае.

Нас окружают тысячи объектов, и каждый из них обладает своими определенными свойствами. В то же время у разных объектов могут быть одинаковые свойства. Например, стол и стул могут быть изготовлены из одного и того же материала. Однако набор всех свойств объекта *неповторим (уникален)*. Если бы было не так, то невозможно было бы выделить среди похожих объектов какой-то конкретный. В таком случае он бы не был объектом, а лишь частью другого объекта.

Уникальность набора свойств объекта – это не свойство его, а требование, предъявляемое к любому объекту. Если это требование нарушается, то приходится либо выявлять дополнительные (уникальные) свойства объекта, либо воспринимать его не как объект, а как элемент (часть) объекта более высокого уровня. Например, предположим, у Аллая и Гюная есть совершенно одинаковые карандаши. По внешнему виду их невозможно отличить друг от друга. И поэтому, если мы хотим посмотреть на эти карандаши как на объекты, то должны указать дополнительные свойства. Например, указать, кому они принадлежат – «карандаши Аллая».

Свойства объекта могут быть произвольными. Однако в информатике для удобства выделяют два типа свойств: *свойства-признаки* и *свойства-действия*. Например, свойствами-признаками карандаша являются его цвет, форма, качество заточки, место хранения. Свойства-действия объекта – это действия, которые может совершить объект или которые можно произвести с данным объектом.

Предмет, явление или процесс, на который направлены наши мысли и внимание и отличающийся своими свойствами, является **объектом**. Составные части объекта, не имеющие своих отличительных свойств, называют **элементами объекта**.



Иногда очень важно знать свойства и методы объекта. Например, если объект «гвоздь» необходимо вбить в объект «стена», мы обязательно должны знать свойства этих объектов. Если стена деревянная, гвоздь должен быть железным (наоборот быть не может).



Если посмотрим на карандаш как на инструмент, с ним можно связать такие действия, как «писать», «рисовать», «заточить».

В программировании свойства-признаки объекта называют просто **свойствами объекта**, а для свойств-действий объекта используют специальный термин – **методы объекта**.



Объект: Леопард

Свойства объекта		Методы объекта
Имя свойства	Значение свойства	
Цвет:	Желтый или рыжий (с черными или коричневыми пятнами)	Сидеть Раздирать Прыгать Подстергать
Высота	45 – 80 см	
Длина туловища	90 – 190 см	
Длина хвоста	60 – 110 см	
Вес	40 – 80 кг	



- ✓ Предмет, явление или процесс, на который направлены наши мысли и внимание и отличающийся своими свойствами, является объектом.
- ✓ Составные части объекта, не обладающие своими отличительными свойствами, называют элементами объекта.
- ✓ Некоторые свойства у разных объектов могут быть одинаковыми. Однако набор всех свойств объекта неповторим (уникален).
- ✓ В информатике выделяют два типа свойств объектов: свойства-признаки и свойства-действия. Свойства-признаки называют свойствами, а свойства-действия – методами объекта.

5.2. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Вследствие какого события начинает утром звонить будильник?

Понятие «объект» играет важную роль и в программировании. Объекты программ очень похожи на объекты из реальной жизни — они тоже отличаются друг от друга своим внутренним строением, действиями, свойствами. Все, что вы видите на экране компьютера, можно назвать объектом: окно любой программы, каждая кнопка, значок, любое меню.



Чтобы понять значение объектно-ориентированного программирования, совершим краткое путешествие в историю. В 50-60 годах прошлого века компьютеры были немислимо дорогими. По сравнению с их стоимостью зарплата программистов была мизерной. Это оказывало влияние и на программирование. В то время было не особенно важно, сколько времени тратит программист на написание и налаживание программы. Основной целью было загрузить компьютер так, чтобы он и минуты не простаивал, так как каждая минута работы компьютера исчислялась большой суммой.

Все изменилось с середины 80-х годов, когда появились персональные компьютеры. Массовое производство компьютеров потребовало наличия большого количества программ для них. И стало ясно, что стоимость месячной работы профессионального программиста ничуть не меньше стоимости его компьютера. И тогда люди, для того чтобы напрасно не тратить труд программистов стали усовершенствовать саму среду программирования.

В первую очередь появился принцип повторного использования кода (программы). Суть этого принципа в том, что однажды написанные кем-то программные модули должны быть собраны и сохранены, а затем могут быть использованы другими программистами в виде готовых блоков. Такие программные блоки называли *объектами*. При необходимости создания новых программ берутся нужные объекты из предыдущих программ и просто изменяются в соответствии с новыми требованиями.

Вероятно, вы встречали в разных программах окна одинаковой формы, похожие меню, одинаковые кнопки. В большинстве программ используется одинаковый шрифт. Все это объекты, соответствующие новым требованиям.

Объектно-ориентированное программирование можно сравнить со строительством бетонных жилых домов. Дом возводится из готовых блоков, а их свойства изменяются на месте. Блоки ставятся на определенное место, подсоединяются вода, электричество, канализационная линия — готова ванная или кухня. Не подсоединяя водопровод и канализационную линию, изменив свойства пола и стен, можно получить спальню или гостиную.



Первым *объектно-ориентированным языком программирования* стал C++ (читается как си-плюс-плюс). Потом появилась объектно-ориентированная версия программы Pascal – Object Pascal. В настоящее время в мире таких языков программирования очень много.

С появлением системы графического управления компьютерами изменился и метод разработки программ. Он тоже стал графическим, то есть визуальным. В среде *визуального программирования* готовые блоки будущей программы выделяются при помощи мыши и изменяются в соответствии с новыми требованиями.



Программа **Lazarus**, с которой мы познакомимся на следующем уроке, – именно из таких объектно-ориентированных, визуальных программ. Как было отмечено на прошлом уроке, каждый объект имеет свойства и действия (методы). Объекты программ тоже имеют *свойства*. Кажущиеся очень простыми на первый взгляд программные объекты могут иметь очень много свойств. Например, рассмотрим командную кнопку в окне. Эта кнопка имеет размер, место, цвет, надпись и другие свойства. Во время программирования нужно работать только с необходимыми свойствами, а остальные свойства, используя один раз (при создании программы), можно «забыть». Свойства объектов программ можно менять. Для этого применяют специальные процедуры – *методы*.

Возможно, многие из вас для того, чтобы вовремя проснуться утром, пользуются будильником. Как объект будильник обладает определенными свойствами и действиями. Но здесь следует обратить внимание на следующее: после завода будильника он без вашего контроля совершает действие (звонит). Это происходит потому, что наступило время, на которое его заводили. Для будильника на этот факт можно посмотреть как на *событие*. Когда происходит событие, то начинается его обработка. Для будильника это приведение в действие звуковых сигналов. Значит любое событие, связанное с объектом, является причиной его определенного действия. То есть объект использует определенные *методы* для обработки события.

Этот принцип работает и в операционной системе Windows. Если сказать более точно, Windows – это среда, *управляемая событиями*. В *системе, управляемой событиями*, компьютер постоянно ожидает, что пользователь при помощи мыши, клавиатуры или других устройств станет производить какие-то действия. Такие действия – это событие для программного обеспечения, которое должно на них реагировать (действовать).

Таким образом, для каждого объекта существует определенный набор событий. Например, командная кнопка поддерживает события Click (щелчок), MouseDown (нажата кнопка мыши), MouseUp (отпущена кнопка мыши).

Возможно, эти понятия покажутся немного сложными для вас, однако на следующем уроке, когда вы будете в объектно-ориентированной среде создавать свою программу, они станут для вас понятными.

OK

Объект: Кнопка

Свойства объекта		События
Имя свойства	Значение свойства	
Цвет	Синий	Click
Ширина	30 пикселей	MouseDown
Длина	100 пикселей	MouseUp
Заголовок	OK	



- ✓ Все что вы видите на экране компьютера – окно любой программы, каждая кнопка, значок, любое меню – можно назвать объектом.
- ✓ Объекты программы обладают свойствами и методами.
- ✓ Первым широко распространенным объектно-ориентированным языком программирования является C++.
- ✓ Lazarus – инструмент объектно-ориентированного визуального программирования.
- ✓ Windows – система, управляемая событиями. В этой системе компьютер постоянно ожидает, что пользователь с помощью мыши, клавиатуры или других устройств произведет какие-то действия.

5.3. СРЕДА ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ LAZARUS

Из каких элементов состоит окно программы?

Как отмечалось на прошлом уроке, существуют различные среды визуального программирования: Visual Basic, Delphi, Borland C++ Builder и др. Каждый из них имеет свой язык программирования. Например, в основе среды Visual Basic лежит язык программирования **Basic**, Delphi – **Pascal**, Borland C++ Builder основывается на языке C++. Каждая из этих сред программирования является очень мощной системой, но все они в то же время представляют собой коммерческие продукты. То есть для законного использования их необходимо заплатить определенную сумму. Поэтому в образовательных системах большинства стран пред-

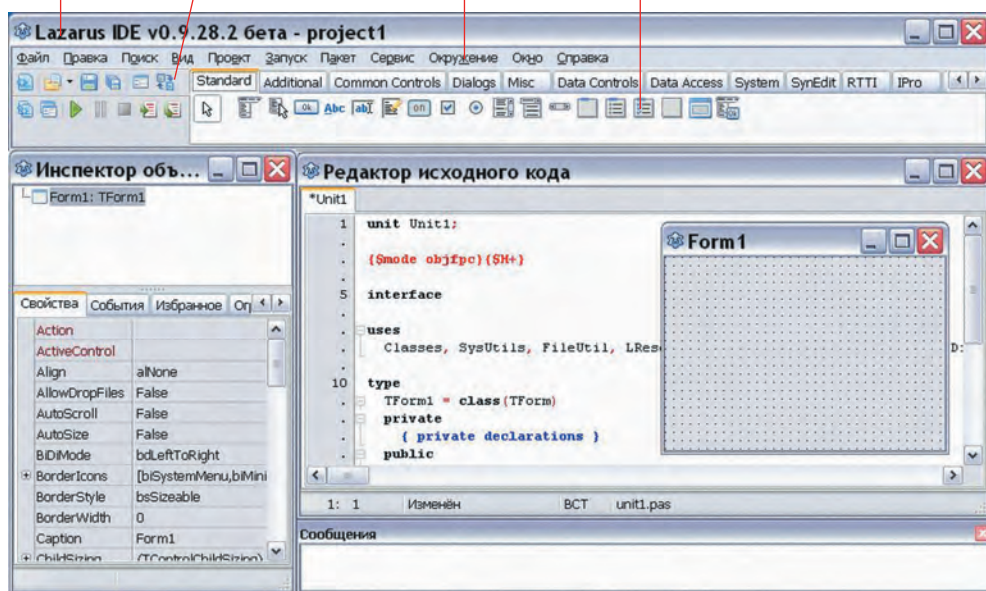


Если на вашем компьютере не установлена программа Lazarus, ее можно скачать с веб-страницы <http://www.lazarus.freepascal.org>

почтение начинают отдавать свободно распространяемым программам (free soft). На этом уроке познакомимся с одним из таких продуктов – **Lazarus**. В основе среды программирования Lazarus лежит язык **Pascal**, точнее, **Free Pascal**.

Если на рабочем столе вашего компьютера есть значок , дважды щелкните по нему или в меню Start выберите команду All Programs ⇨ Lazarus ⇨ Lazarus. Откроется не одно окно, как в знакомых вам до этого программах, а несколько: главное окно программы, Object Inspector (Инспектор объектов), Source Editor (Редактор исходного кода), Form1 (Окно формы) и Messages (Сообщения) (5-1). Месторасположение любого окна можно изменить на экране независимо от других.

Строка заголовка Панель инструментов Строка меню Палитра компонентов



5-1. Окна программы Lazarus

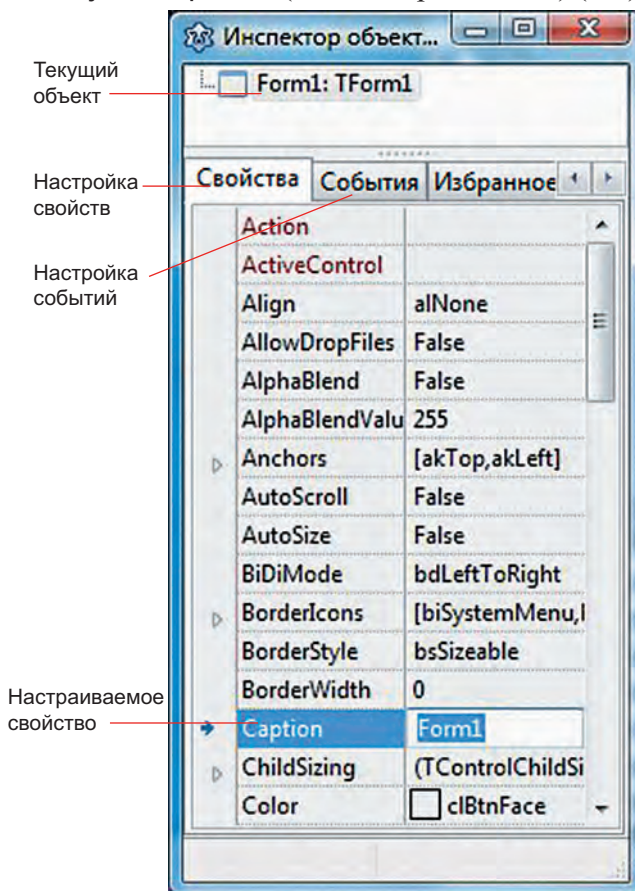
Главное окно. Главное окно системы Lazarus обычно расположено в верхней части экрана. Как и в других программах, главное окно состоит из нескольких частей: строки заголовка, строки меню и панели инструментов.

В *строке заголовка* главного окна указано название открытого в данный момент проекта: project1. Под строкой заголовка находится *строка меню*.

Строка меню – один из наиболее важных элементов каждой программы или системы. Здесь можно найти основные команды системы. Ниже строки меню расположена *панель инструментов*. Здесь находятся наиболее часто используемые кнопки команд. Для того чтобы узнать название кнопки, достаточно навести на нее указатель мыши. В главном окне есть панель инструментов, которая отличается от тех, что были известны вам ранее. Эта панель называется *палитрой компонентов*. Здесь находятся компоненты, которые будут нам необходимы во время программирования. Если навести указатель мыши на кнопку компонента, отобразится его название.

Палитра компонентов состоит из многочисленных вкладок. На каждой вкладке имеются свои компоненты. Хотя число компонентов более 100, мы будем использовать только некоторые из них.

Инспектор объектов. В левой части экрана находится очень важное окно Object Inspector (Инспектор объектов) (5-2).

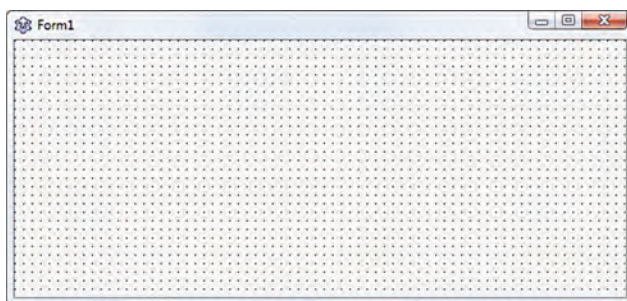


5-2. Инспектор объектов предназначен для настройки объектов программы

По ходу программирования это окно используется особенно часто. Объекты программы настраиваются именно здесь. Выбрав любой объект, здесь можно просмотреть и изменить его свойства. События, на которые реагируют объекты, также выбираются и настраиваются здесь.

Окно формы. *Форма* – это бланк будущей программы. Каждая программа имеет по крайней мере одно окно и, следовательно, не менее одной формы. Поэтому во время работы с программой вы всегда видите на экране окно формы.

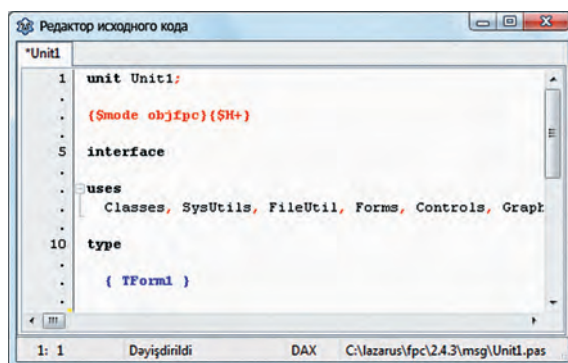
Макет первого окна называется Form1 (5-3). Если в программе будет два окна, то второму макету будет дано название Form2 и т.д.



5-3. В окне формы размещаются элементы управления будущей программы

Вы, разумеется, можете заменить стандартные названия более подходящими. Как видно из рисунка, изначально окно формы пустое. На следующем уроке, создав программу, вы добавите в нее необходимые элементы управления. Сетка из точек в окне формы позволяет размещать объекты правильно и аккуратно. Во время работы программы сетки не видно.

Окно исходного кода. В этом окне находится код программы. Может возникнуть вопрос: ведь мы ничего не писали, откуда взялись записи в окне кода (5-4)?



5-4. В окне кода находится текст программы

Дело в том, что большую часть кода вашей программы система Lazarus автоматически формирует сама. И это не преувеличение. В операционной системе Windows даже самые простые программы состоят из нескольких тысяч операторов языка программирования. Если надо добавить какой-то оператор в свою программу, это можно сделать именно в этом окне. Некоторые операторы система Lazarus добавляет сама, остальные добавляются вручную.

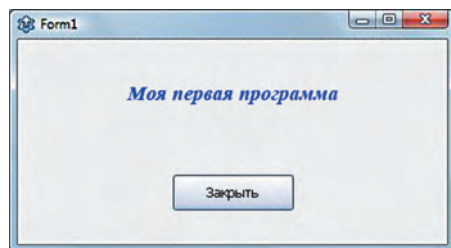
- ✓ Существуют различные среды визуального программирования: *Visual Basic, Delphi, Borland C++ Builder* и др. Одной из них является свободно распространяемая (бесплатная) среда программирования *Lazarus*.
- ✓ В основе среды программирования *Lazarus* лежит язык *Pascal*.
- ✓ При запуске программы *Lazarus* на экране появляется не одно, а несколько окон: главное окно программы, *Object Inspector* (Инспектор объектов), *Source Editor* (Редактор исходного кода), *Form1* и *Messages* (Сообщения).
- ✓ Как и в других программах, главное окно *Lazarus* состоит из нескольких частей: строки заголовка, строки меню и панели инструментов. В главном окне есть и другая панель инструментов, которую называют палитрой компонентов.
- ✓ Объекты программы настраиваются в окне *Object Inspector* (Инспектор объектов). Выбрав любой объект, можно просмотреть и изменить здесь его свойства. События, на которые реагируют объекты, также выбираются и настраиваются здесь.
- ✓ Форма – это окно будущей программы. Каждая программа имеет по крайней мере одно окно и, следовательно, не менее одной формы.
- ✓ В окне кода располагается код программы.

5.4. СОЗДАНИЕ ПЕРВОЙ ПРОГРАММЫ

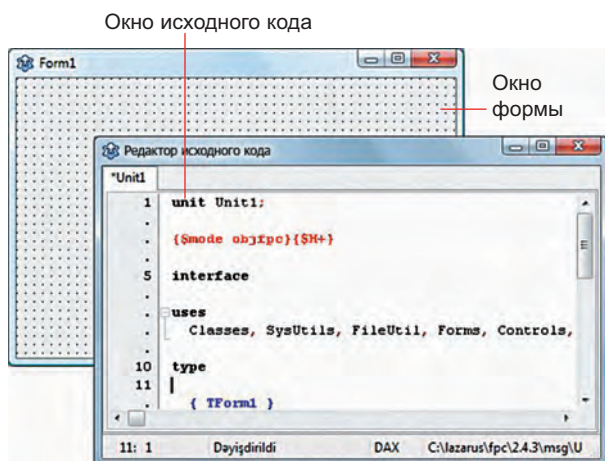
Какие действия подразумевают понятия «выделить мышью» и «перетаскивать мышью»?

На этом уроке вы научитесь создавать в системе *Lazarus* свою первую программу. Она будет очень простой – ваша программа ничего не будет делать, – ее можно будет только открыть и закрыть. Для того чтобы что-то открыть, необходимо окно программы, а чтобы закрыть – нужна соответствующая кнопка. Кроме этого в окно программы будет добавлена некоторая запись. В результате получится то, что вы видите на рисунке (5-5).

В системе *Lazarus* любая незаконченная программа является *проектом*. Система распознает все файлы проекта и записывает их в заданную папку. В проекте всегда много файлов, большинство из которых система создает сама. Для вас



5-5. В итоге получится такая программа



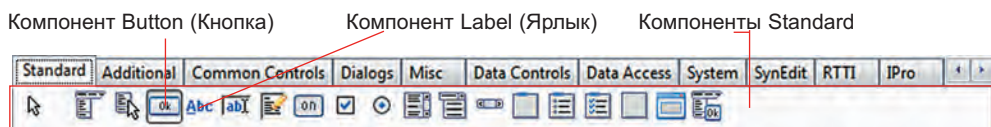
5-6. Окно формы и ее окно исходного кода

одну форму, потому что объекты сохраняются именно в форме. Например, программа не может состоять только из одной командной кнопки. Командная кнопка, как и другие элементы управления, с которыми вы познакомитесь на следующем уроке, может размещаться только на форме.

Запустите систему Lazarus. Система создаст форму для окна новой программы. Одновременно система подготовит общий для всех вновь созданных программ фрагмент исходного кода (5-6). В форме нет ни одного элемента управления, а во фрагменте кода – ни одного действия, которое может выполнить ваша программа.

Естественно, пустая форма и пустое окно программы никому не нужны. Обычно окно программы не пустует – в нем размещаются элементы управления: разнообразные кнопки, ярлыки и т.д. В системе Lazarus имеются готовые макеты таких элементов. Их называют *компонентами*.

В Lazarus для компонентов есть специальная панель инструментов – **Component Palette** (Палитра Компонентов) (5-7). Она состоит из многочисленных вкладок, на каждой из которых собраны компоненты определенной категории. Самые простые компоненты расположены на вкладке **Standard**. Каждый компонент имеет свой значок. Чтобы выбрать какой-либо компонент, необходимо щелкнуть по его значку. При двойном щелчке компонент добавляется в форму.



5.7. Палитра компонентов системы Lazarus

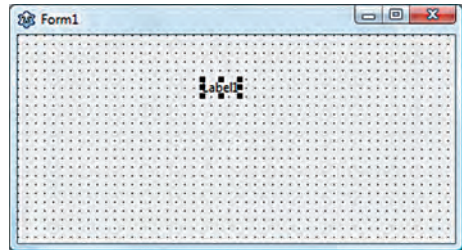
В вашей первой программе будет использовано только два компонента – **Label** (Ярлык) и **Button** (Кнопка).

важны из них только три файла: *файл формы*, *файл кода* и *файл проекта*. На остальные файлы можно не обращать внимания – система сама контролирует их.

Как уже отмечалось, *форма* – это окно будущей программы. Во время программирования вы работаете с формой. После запуска готовой программы форма превращается в полноценное окно. Каждая программа должна иметь хотя бы одно окно и, следовательно,

Задание 1. Добавление нового компонента в форму

1. Запустите программу Lazarus.
2. Установите указатель мыши на компоненте `TLabel` и щелкните по нему. Компонент выделится.
3. Переместите указатель мыши в любое место формы и нажмите ее левую кнопку. В этом месте появится новый объект – `Label1`. По краям и углам этого объекта будут размещены небольшие квадратики – маркеры (5-8).
4. Используя метод *перетаскивания* мышью, разместите объект в любом месте формы.



5-8. Вновь созданный объект, выделенный с помощью маркеров

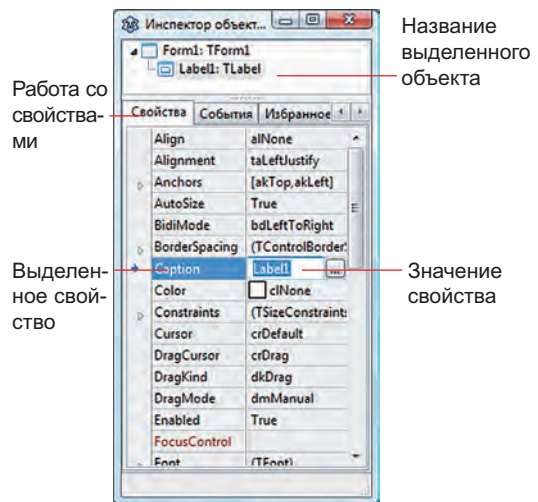
Вы создали свой первый объект `Label1`, но в нем пока нет нужного вам текста. Его тоже необходимо создать. Для этого сначала познакомимся со свойствами объектов.

Как вы знаете, все объекты имеют свойства, и у разных объектов они разные. Для объекта `Label1` одним из значимых свойств является *содержание записи*.

До тех пор, пока вы его не указали, название объекта будет восприниматься как содержание записи. Поэтому на экране вы увидите запись `Label1`.

Для настройки свойств объекта в системе Lazarus предусмотрена специальная панель **Object Inspector** (Инспектор объекта) (5-9). Правила работы с этой панелью следующие:

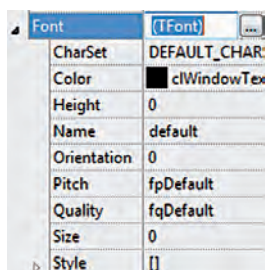
1. Выделяется объект, который будет настраиваться (меняться свойства).
2. Выбирается требуемое свойство.
3. Выбранному свойству задается значение.



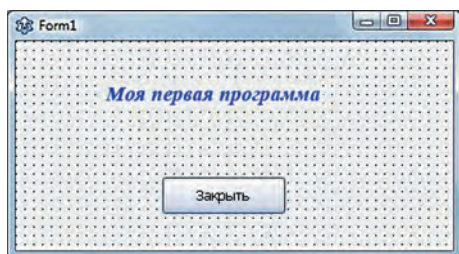
5-9. Настройка свойств объекта при помощи Инспектора объекта.

Задание 2. Настройка свойств объекта

1. Выделите объект **Label1**.
2. Для изменения текста ярлыка на панели **Object Inspector** щелкните по свойству **Caption**. Это свойство выделится (5-9).
3. Введите текст, например, **Моя первая программа**. Как видите, размеры объекта меняются в соответствии с объемом текста. Такое поведение объекта определяется особым свойством, которое называется **AutoSize**.
4. Найдите свойство **AutoSize** в списке и убедитесь, что его значение **True (Истина)**. Это указывает на то, что размер текстового поля меняется в зависимости от содержания текста.



5. Для изменения шрифта текста предусмотрено свойство **Font (Шрифт)**. Это составное свойство, о чем говорит небольшой треугольник (или знак +) перед ним. Если щелкнуть по значку, откроются подсвойства свойства **Font** (5-10).
6. Щелкните по подсвойству **Color (Цвет)**. Справа откроется список набора цветов.
7. Выберите из списка необходимый цвет, например, пункт **clPurple**. Текст станет фиолетового цвета.
8. Щелкните по подсвойству **Style (Стиль)**. Это подсвойство в свою очередь также составное. Щелкните по значку треугольника перед этим свойством. Откроются свойства подсвойства **Style**.
9. В открывшемся списке свойствам **fsBold** и **fsItalic** задайте значение **True**. Текст станет жирным и наклонным.



5-11. Окно формы после настройки параметров кнопки

Методом, изложенным в задании 1, создайте командную кнопку в форме. Для этого на вкладке **Standard** палитры компонентов выберите компонент **TButton**. Затем, используя метод из задания 2, замените надпись на кнопке (**Button1**) любой требуемой записью (например, **Закреть**) (5-11).

Таким образом на форму были внесены необходимые компоненты. На следующем уроке будет продолжена работа над этой формой.

- 
- ✓ В системе Lazarus мы разрабатываем проекты.
 - ✓ В проекте всегда много файлов. Большинство из них система создает сама. Для программистов важны три из них: файл формы, файл кода и файл проекта.
 - ✓ Каждая программа должна иметь хотя бы одно окно (форму), потому что другие объекты хранятся в форме. Например, программа не может состоять только из одной командной кнопки. Командная кнопка должна размещаться в форме.
 - ✓ Окно программы обычно не бывает пустым – в нем имеются элементы управления: разнообразные кнопки, ярлыки и другие компоненты.
 - ✓ Компоненты в системе Lazarus расположены на специальной панели инструментов *Component Palette* (Палитре компонентов).
 - ✓ Для добавления нового компонента в форму необходимо выделить нужный компонент на палитре компонентов, установить указатель мыши в любом месте формы и нажать левую кнопку мыши.
 - ✓ Для изменения свойств объекта предусмотрена специальная панель. Правила работы с этой панелью, которая называется *Object Inspector* (Инспектор объектов), следующие: (1) выделяется объект, который будет настраиваться (изменяться свойства); (2) выделяется необходимое свойство; (3) задается значение выделенному свойству.

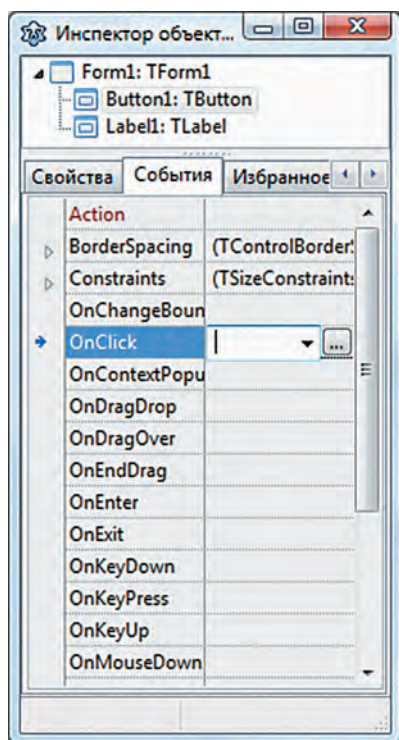
5.5. СОБЫТИЯ И ИХ ОБРАБОТКА

Какие события, по-вашему, могут произойти для объекта «кнопка»?

На прошлом уроке была создана форма для вашей первой программы. Туда были добавлены два объекта – ярлык и командная кнопка. Потом были настроены свойства этих двух объектов. Продолжим работу в этой программе.

Созданная кнопка пока бездействует. Если щелкнуть по ней, ничего не произойдет, так как программе никакого указания не было дано, то есть реакция кнопки на щелчок по ней пока не определена. Теперь необходимо написать программный код – *процедуру*, который обработает щелчок.

Программисты не любят писать очень длинные программы в виде одного модуля. Чем больше операторов в программе, тем сложнее их контролировать. А если их больше ста, то не избежать ошибок.



5-12. Вкладка События

Современные игровые программы, например, состоят из миллиона строк, и поэтому с такими программами сложно работать не разделив их на подпрограммы.

Одна из подпрограмм отвечает за походку героя, вторая – за прыжки, третья – за то, как он стреляет. В современных программах таких небольших подпрограмм десятки, иногда сотни. Все они хранятся в памяти компьютера и ждут, когда их вызовут. Например, как только игрок нажимает кнопку «Огонь», запускается подпрограмма, которая обрабатывает режим стрельбы. Одна из подпрограмм проверяет, куда попадают снаряды, другая контролирует количество боеприпасов, третья ведет учет набранных баллов и т.д.

Различают два вида подпрограмм: *процедуры* и *функции*. Такое деление условно. В некоторых языках программирования все подпрограммы рассматривают как функции. Но в любом случае нужно знать разницу между ними.

Функция – это подпрограмма, которую используют для определенных вычислений или проверок. Как только она заканчивает работу, результат вычисления возвращает обратно в программу, вызвавшую ее.

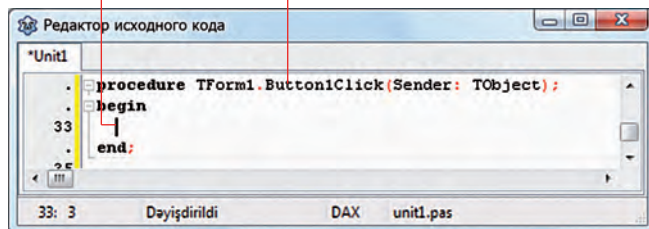
Процедура также является подпрограммой, которую используют для проведения определенных подсчетов или проверок. Но процедура ничего не возвращает в программу, вызвавшую ее.

Задание 3. Определение реакции объекта на событие

1. Щелкнув указателем мыши по кнопке в форме, выделите ее. Кнопка выделится маркерами.
2. Перейдите в окно Object Inspector (Инспектор объектов) и щелкните по вкладке Events (События) (5-12). В левом столбце Инспектора объектов отобразятся все события, на которые может отреагировать объект (кнопка). Возле каждого события указана процедура, обрабатывающая данное событие. Так как мы пока не показали ни одной процедуры, правый столбец пустой.

3. Рассмотрим событие «щелчок по кнопке». Это событие называют **OnClick**. Для создания обрабатывающей процедуры для него, дважды щелкните в правом от события столбце. Lazarus автоматически активизирует окно кода и добавит в него обработчик-процедуру данного события (5-13).

Сюда вводится оператор процедуры Название процедуры



5-13. Бланк процедуры обработки в окне кода

4. Название этой процедуры – `TForm1.Button1Click` – говорит само за себя: в форме **Form1** для объекта кнопки **Button1** обрабатывается событие **Click** (щелчок). Все команды, записанные между словами **begin** и **end**, исполняются при указанном событии. Эти операторы программа Lazarus устанавливает сама, для того чтобы программист видел начало и конец процедуры. Нам лишь остается записать операторы, которые опишут, что произойдет при щелчке по кнопке. Чтобы еще более облегчить работу, текстовый курсор уже установлен в нужном месте. Так как на кнопке написано «Закрыть» (мы ее так наименовали), щелчок по ней должен привести к завершению работы программы и закрытию окна. Для того чтобы сделать это в программе Pascal, предусмотрен оператор `Close ()`. Введите этот оператор, там где находится текстовый курсор. Теперь текст в редакторе кода будет иметь следующий вид:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    Close ();
end;
```

Программа готова! Как видите, текст в окне кода достаточно длинный, но мы «в ответе» только за эти четыре оператора. Все остальные действия система визуального программирования берет на себя.

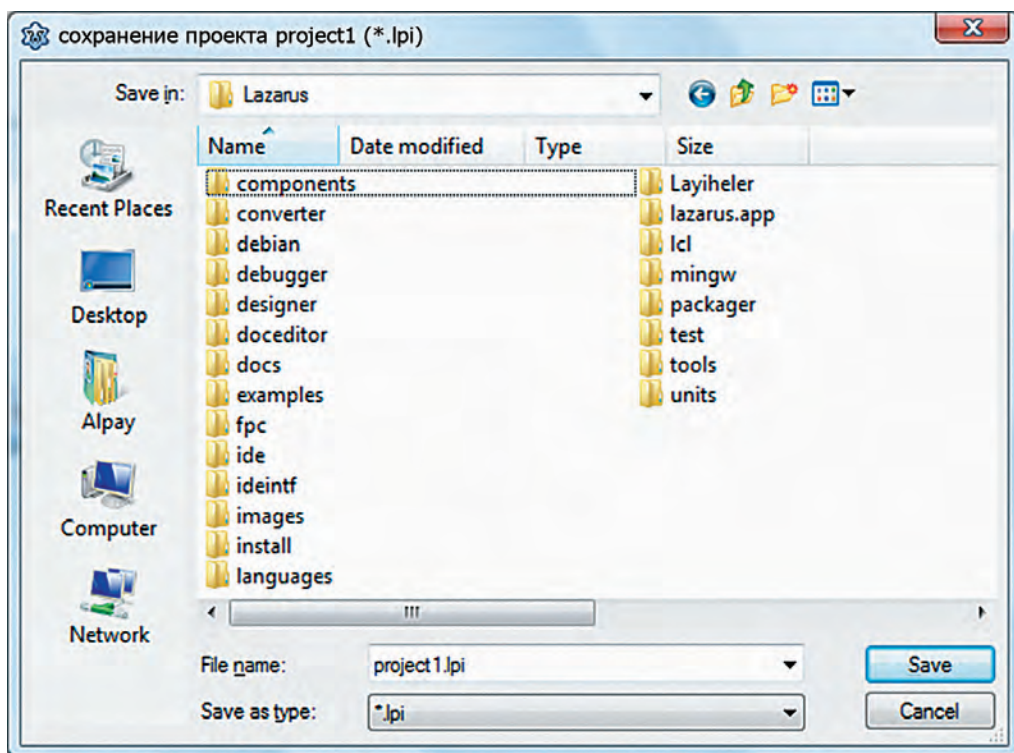
Теперь проверим, как работает программа. Для этого необходимо запустить ее. Это можно сделать при помощи команды меню **Run ⇨ Run (Запуск ⇨ Запуск)**, клавиши **<F9>** или кнопки  **Run (Запуск)** на панели инструментов.

Система Lazarus компилирует программу, создает исполняемый файл и запускает его. Обратите внимание, как выглядит окно программы после запуска (5-5).

Таким образом, мы создали полностью работающую (правда, не приносящую пользы) программу. Но осталось одно дело: мы не сохранили созданный проект. Любая создаваемая в системе Lazarus программа, даже очень простая, является проектом. Поэтому мы должны сначала дать своему проекту имя, а потом определить, где мы его сохраним. Назовем свою первую программу **«Первая»** и, создав в папке C:\Lazarus новую папку, сохраним ее там. Новую папку назовем соответствующим именем, например, «Проекты». Для каждого проекта надо создавать новую папку. Таким образом, сохранение проекта состоит из трех этапов:

1. Создание в папке C:\Lazarus новой папки (только вначале), а в ней – папки для сохранения конкретного проекта;
2. Сохранение файла проекта в соответствующей папке;
3. Сохранение программных файлов в папке проекта.

Это действие можно начать с команды меню File ⇨ Save As (Файл ⇨ Сохранить как). Откроется диалоговое окно Save project project1 (*.lpi) (сохранение проекта project1) (5-14).



5-14. Диалоговое окно сохранения файла

Если в папке C:\Lazarus нет папки **Проекты**, создайте ее, а потом внутри нее создайте папку **Первая**. В поле **File Name** название project.lpi замените на pervaya.lpi. При щелчке по кнопке **Save** файл проекта запишется на выделенном месте. После этого откроется диалоговое окно **Save Unit1 (*.pas)** (**Сохранение Unit1**) для сохранения текста программы. Не меняя предложенное название **Unit1**, щелкните по кнопке **Save**. В результате все файлы, связанные с программой, сохранятся в указанном месте.

- 
- ✓ *Современные программы состоят из десятков, а иногда и сотен подпрограмм.*
 - ✓ *Выделяют два типа подпрограмм: процедуры и функции.*
 - ✓ *Функция – это подпрограмма, которая вызывается для выполнения определенных вычислений или проверок. Как только она заканчивает работу, результат вычисления возвращается обратно в программу, вызвавшую ее.*
 - ✓ *Процедура тоже является подпрограммой. При вызове ее также производятся определенные вычисления, проверки. Но процедура ничего не возвращает в программу, вызвавшую ее.*

5.6. ЯЗЫК ПРОГРАММИРОВАНИЯ PASCAL

Как по-вашему, почему язык программирования назвали в честь Блеза Паскаля?

На прошлом уроке вы впервые создали программу в системе Lazarus. Хотя она и не делала ничего полезного, но была интересна как ваша первая программа. На следующем уроке предстоит знакомство с еще одной программой, которая станет образцом ваших будущих программ, а пока рассмотрим некоторые важные понятия, которые используют в программировании.

Компьютер «понимает» только один язык – *машинный*. Писать программы на машинном языке и проверять их правильность очень сложно и трудоемко. Поэтому программисты стали искать пути для облегчения своей работы и в результате создали *языки программирования*. Процесс перевода программы, написанной на языке программирования, на машинный язык называется *компиляцией*.



Блез Паскаль



Никлаус Вирт

В основе системы Lazarus лежит язык программирования Free Pascal. Free Pascal берет свое начало от классического языка Pascal. В средних школах многих стран основам программирования обучают именно на базе языка Pascal. Одна из используемых сегодня систем визуального программирования Delphi тоже основывается на языке Pascal.

Язык программирования Pascal (Паскаль) был разработан в 1971 году шведским физиком **Никлаусом Виртом** (1934) и назван в честь французского математика и философа **Блеза Паскаля** (1623-1662). Н.Вирт создал этот язык для обучения своих студентов. Отметим, что в 1984 году Н.Вирт за разработку серии инновационных компьютерных языков ЭЙЛЕР, ALGOL-W, МОДУЛА и Pascal получил премию Тьюринга – самую престижную премию в информатике. Pascal – один из наиболее распространенных языков программирования. Он отличается от других программ своим ясным и логичным написанием, и это делает его приемлемым как для начинающих, так и для профессиональных программистов.

Как известно, каждый шаг алгоритма – это команда или указание для его исполнителя. Отдельная команда на языке программирования называется *оператором* (statement). На языке программирования каждая **программа** состоит из последовательности операторов.

Отделение операторов друг от друга в программе определяется правилами соответствующего языка программирования. В большинстве первых языков программирования правила требовали, чтобы *каждый отдельно взятый оператор был записан с новой строки*. На языке BASIC это правило остается в силе до сих пор.

Современные языки программирования обычно дают возможность записывать на одной строчке несколько операторов, или, наоборот, один оператор записывать на нескольких строчках. В этом случае, чтобы отделить операторы друг от друга, между ними ставится *разделительный символ*. В большинстве языков программирования, в том числе и в языке Pascal, в качестве разделителя используют точку с запятой (;).

Когда говорят об операторах, то обычно отмечают их назначение, например, *оператор присваивания*, *условный оператор*, *оператор цикла* и т.д.

При решении любой задачи в программе происходит обработка *данных*. В программе данные называют *величинами*. Величины, значения которых меняются по ходу программы, называют *переменными*, а не изменяющие свое значение – *постоянными*, или *константами*.

Каждая переменная имеет свое имя, тип и значение. *Имя переменной* служит для ее обозначения и в программировании его называют *идентификатором*. Вообще идентификаторы используют для того, чтобы на языках программирования именовать разные объекты – переменные, константы, функции и т.д.

Идентификатор состоит из букв, цифр и некоторых других символов. И хотя правила написания идентификаторов на разных языках программирования могут быть различными, есть и общие принципы:

1. Идентификатор может состоять из букв и цифр, но должен начинаться с буквы.
2. В идентификаторе не используют знак пробела и знаки препинания. Некоторые знаки могут быть исключением, например, «`_`» или «`$`».
3. Во всех языках программирования для написания операторов существуют ключевые слова. Идентификатор не должен совпадать ни с одним из ключевых слов.
4. Идентификатор может быть записан как строчными, так и заглавными буквами. К идентификаторам, написанным одними и теми же, но в разных регистрах буквами, различный подход. Например, идентификаторы `One` и `one` на языках BASIC и Pascal считаются одинаковыми, а на языке C – разными.

В качестве примеров возможных идентификаторов можно показать `i`, `a`, `t0123456789`, `NoClass`. А вот такие идентификаторы использовать нельзя: `1stPlace` (начинается с цифры), `one and one` (есть пробелы), `yes (no)` (использованы скобки).

Может возникнуть вопрос: в чем смысл давать названия постоянным величинам? Действительно, это не оказывает влияния на работу программы, но делает программу более понятными. Кроме этого, обычно, в программах, к одной и той же величине обращаются по несколько раз. В этом случае, если для постоянной величины не используется идентификатор, то при работе программы с ее другим значением, придется внести изменения в соответствующие места программы.



В математике, физике и других областях знаний также дают названия постоянным величинам. Какие величины тебе известны?



Для того чтобы задать или изменить значение переменной, предусмотрен *оператор присваивания*. В целом этот оператор состоит из трех частей:

<идентификатор> <символ присваивания> <выражение>

Например, для того чтобы присвоить значение 2 переменной `x`, оператор присваивания нужно записать так: `x := 2;`. В левой части оператора присваивания показан идентификатор переменной. На языке Pascal знак присваивания показывается как `:=`. Во время выполнения оператора вычисляется значение выражения в правой части и полученный результат присваивается переменной в левой части. Для выполнения действий с числами пользуются *арифметическими выражениями*.

В арифметических выражениях используют четыре действия (сложение, вычитание, умножение, деление) и скобки. На языке программирования знаки сложения и вычитания представляют теми же символами (+ и –), что и в математике, а для умножения и деления используют символы * и / соответственно. Последовательность выполнения действий такая же, как в математике. Ниже приведены примеры арифметических выражений.

<code>a := b + c;</code>	сумма значений переменных <code>b</code> и <code>c</code> присваивается переменной <code>a</code>
<code>i := i + 1;</code>	значение переменной <code>i</code> увеличивается на 1
<code>z := x * x + y * y;</code>	переменной <code>z</code> присваивается сумма квадратов <code>x</code> и <code>y</code>

Эти примеры показывают, что знак присваивания нельзя путать со знаком равенства. Так, во втором примере левая и правая части никогда не равны друг другу.

Тип переменной определяет множество его возможных значений. В стандартном языке Pascal существует четыре типа переменных: Integer (целый), Real (вещественный), Boolean (логический) и Char (символьный).

Любая переменная, используемая в программе, написанной на языке Pascal, должна быть описана с самого начала, то есть должен быть указан ее тип. Для этого в начале программы записывается ключевое слово **var**. После этого слова записываются идентификатор каждой переменной и ее тип (между ними ставится двоеточие). Например, внизу указаны две переменные – одна целого, а вторая вещественного типа.

```
var i: Integer;
    r: Real;
```



- ✓ Программа, написанная на языке программирования, переводится на машинный язык с помощью компилятора. Этот процесс называется компиляцией.
- ✓ Язык программирования Pascal был создан в 1971 году шведским физиком Никлаусом Виртом.
- ✓ Отдельная команда на языке программирования называется оператором (statement). На языке программирования каждая программа состоит из последовательности операторов.
- ✓ Величина, значение которой меняется по ходу программы, называется переменной, а величины, не меняющие свое значение, – постоянными (константами).
- ✓ Каждая переменная имеет свое имя, тип и значение. Имя является обозначением (символом) переменной, и в программировании его называют идентификатором.
- ✓ В стандартном языке Pascal имеется четыре типа переменных: Integer (целый), Real (вещественный), Boolean (логический) и Char (символьный).
- ✓ Для задания или изменения значения переменной используют оператор присваивания.
- ✓ Все переменные, которые имеются в программе, должны быть объявлены заранее.

5.7. ОЧЕНЬ ПОЛЕЗНАЯ ПРОГРАММА

Что вам известно об истории календаря Хиджры?

Первая ваша программа в системе Lazarus была очень простой и, можно сказать, что ничего не делала. Целью ее было показать, что такое визуальное программирование. Следующая программа тоже будет несложной, но полезной. Вы создадите программу, которая будет автоматически переводить год календаря Хиджры в год по европейскому (Григорианскому) календарю, и наоборот. Для этого воспользуемся формулами, которые известны вам из материалов предшествующего раздела:

$$Q = X - X/33 + 622 \qquad X = Q - 622 + (Q - 622)/32$$

Таким образом ваша новая программа должна будет переводить год календаря Хиджры (X) в год по Григорианскому календарю (Q), и наоборот, а также отображать результаты вычислений на экране.

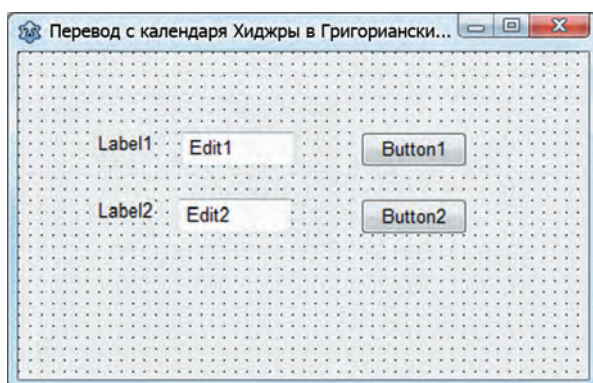
Задание 4. Календари

1. Запустите программу Lazarus.
2. Для создания нового проекта выберите команду меню **File⇒New...** (**Файл⇒Создать...**). Откроется соответствующее диалоговое окно.
3. Выбрав из списка пункт **Project⇒Application (Проект⇒Приложение)**, щелкните по кнопке **ОК**. Будет создан новый проект.
4. Сохраните новый проект в папке **Calendar**. Для этого используйте команду **File⇒Save As (Файл⇒Сохранить как ...)**.
5. Используя предложенную таблицу, измените свойства формы.

Свойство	Значение	Описание свойства
Caption	Перевод с календаря Хиджры в Григорианский, и обратно	Название формы
Height	230	Высота формы
Width	400	Ширина формы
Font.Name	Arial	Название шрифта
Font.Size	10	Размер шрифта

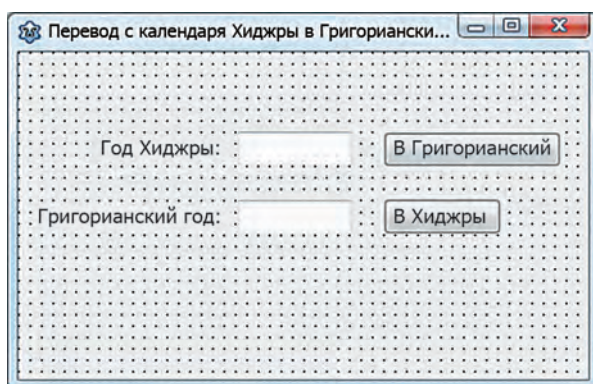
6. Используя вкладку **Standard** палитры компонентов, создайте два объекта – ярлыки **Label1** и **Label2**.
7. Используя во вкладке **Standard** компонент **TEdit**, создайте после объекта **Label1** объект **Edit1**. Точно так же создайте после объекта **Label2** объект **Edit2**.
8. Используя компонент **TButton**, создайте перед объектом **Edit1** объект **Button1**. Аналогично создайте перед объектом **Edit2** объект **Button2**. Выровните объекты относительно друг друга.

В результате получится приблизительно такой вид, как на рисунке 5-15.



5-15. Создание формы

9. Выделите объект **Label1** и измените его текст на **Год Хиджры**. Аналогично измените текст объекта **Label2** на **Григорианский год**.
10. Выделите объект **Button1** и измените его текст на **В Григорианский**. Аналогично текст объекта **Button2** замените на **В Хиджры**.
11. Выделите объект **Edit1** и в значении его свойства **Text** поставьте пробел. Повторите эти действия с объектом **Edit2**. В итоге интерфейс программы готов (5-16).



5-16. Форма после изменения свойств объекта

Программа должна работать так: в текстовом поле **Год Хиджры** записывается любая дата, например, **1419**. Для того чтобы узнать, какому году будет соответствовать этот год по Григорианскому календарю, щелкните по кнопке **В Григорианский**. В текстовом поле **Григорианский год** появится год по европейскому календарю (например, **1998**).

12. Для создания обработчика события `OnClick` для кнопки `В Григорианский (Button1)` дважды щелкните по объекту `В Григорианский (Button1)`. Управление перейдет к редактору исходного кода.
13. Процедуру `TForm1.Button1Click` составьте так, как указано ниже:

```
procedure TForm1.Button1Click(Sender: TObject);
var
    X, Q: integer;
begin
    X := StrToInt(Edit1.Text);
    Q := X-(X div 33) + 622;
    Edit2.Text := IntToStr(Q);
end;
```

Рассмотрим каждую строчку программного кода.

Строка 1. Заголовок подпрограммы. Он состоит из ключевого слова **procedure** и имени подпрограммы `TForm1.Button1Click`. В скобках обычно указывается список параметров подпрограммы.

Строка 2. Открывает блок описания переменных (слово **var** – сокращение английского «variable» – переменная).

Строка 3. Описывает переменные `X` и `Q`.

Строка 4. Начинается программный блок (**begin** – начало).

Строка 5. Выполняется следующее действие: считывается информация, введенная со свойства `Text` объекта `Edit1`. Эта информация воспринимается компилятором как строка текста (например, '360' - это строка из трех символов).

А поскольку нам для вычислений нужны числовые значения, функция `StrToInt` преобразовывает ее в целое число. Результат присваивается переменной `X`.

Строка 6. Вычисляется значение выражения и результат присваивается переменной `Q`.

Строка 7. У объекта `Edit2` изменяется свойство `Text`. Поскольку значение переменной `Q` – целое число, его необходимо перевести в строку. Для этого используется функция `IntToStr`.

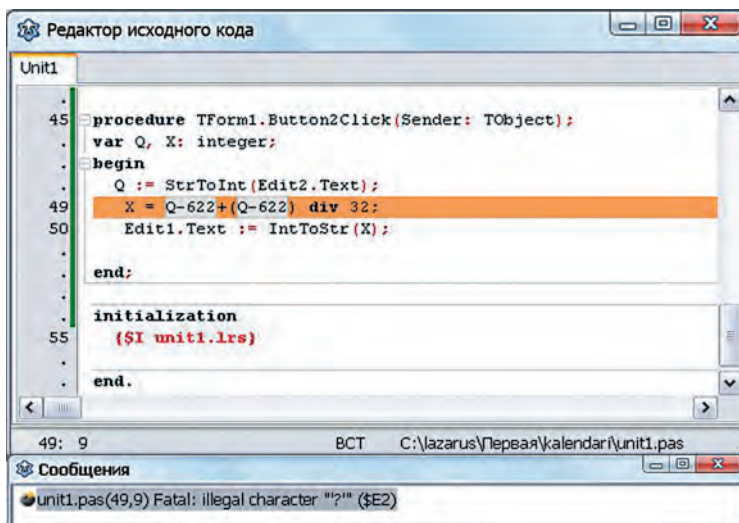
Строка 8. Конец программного кода (**end** – конец).

14. Для создания обработчика события **OnClick** для кнопки **В Хиджры (Button2)** дважды щелкните по объекту **В Хиджры (Button2)**. Управление перейдет к редактору исходного кода.
15. Процедуру **TForm1.Button2Click** составьте так, как указано ниже:

```
procedure TForm1.Button2Click(Sender: TObject);  
var  
    X, Q: integer;  
begin  
    Q := StrToInt(Edit2.Text);  
    X = Q-622+(Q-622) div 32;  
    Edit1.Text := IntToStr(X);  
end;
```

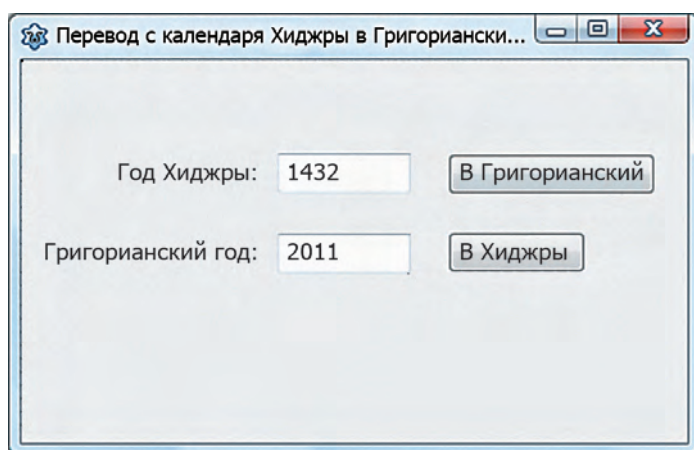
Наконец программа готова. Теперь ее можно откомпилировать и запустить. Для компиляции необходимо выполнить команду меню **Run⇒Run (Запуск⇒Запуск)**. Во время компиляции проверяется наличие синтаксических ошибок. Если выявится ошибка, процесс компиляции остановится. На 6-й строке процедуры **TForm1.Button2Click** была специально допущена ошибка (в операторе присваивания пропущено двоеточие).

Результат компиляции показан на рисунке 5-17. Строка с ошибкой выделена. В окне **Сообщения** указан номер строки, где была обнаружена ошибка, и вид ошибки. Исправив ошибку, нужно снова произвести компиляцию.



5-17. Строка с ошибкой выделена

После удачного завершения компиляции запускается программа (5-18). Теперь вы можете реализовать цель, поставленную в начале урока. То есть ввести в текстовое поле Год Хиджры определенную дату (например **1432**) и щелкнуть по кнопке **В Григорианский**. В текстовом поле Григорианский год появится соответствующая дата (**2011**). Можно сделать и обратное: ввести любую дату в текстовом поле Григорианский год и щелкнуть по кнопке **В Хиджры**. Программа, вычислив согласно данной формуле соответствующий год, покажет его в текстовом поле Год Хиджры.



5-18. Результат работы программы

Вопросы и задания

1. Объясните отличие понятий «объект» и «элемент».
2. Может ли в информатике быть два объекта с абсолютно одинаковыми свойствами?
3. Что такое метод объекта?
4. Перечислите несколько свойств и методов объекта «доска».
5. Какие свойства имеет значок «Мой компьютер» на рабочем столе?
6. Если происходит событие для какого-то объекта, как он обрабатывает это событие?
7. Что означает выражение «управляется событиями»?
8. Какие языки программирования вы знаете?
9. Какие окна открываются при запуске Lazarus? Объясните функции каждого окна.
10. Где находится Палитра компонентов и что имеется на ней?
11. Что такое проект в системе Lazarus?
12. Каковы основные файлы проекта?
13. Как добавить новый объект на форму?
14. Что необходимо для изменения свойств объектов программы?
15. Что такое составное свойство?
16. Что такое подпрограмма и для чего ее используют?
17. В чем различие процедуры и функции?
18. Просмотрите список событий объекта «кнопка». Попробуйте понять, когда происходят некоторые из этих событий.
19. Что необходимо сделать для перевода программы с языка программирования на машинный язык?
20. Какой сейчас год по календарю Хиджры?
21. Какой процедурой обрабатывается событие, которое произойдет, если установить указатель мыши на любом объекте и щелкнуть левой кнопкой?
22. Как описываются переменные на языке Pascal?
23. Используя программу Календарь, определите, в каком году по календарю Хиджры вы родились.

ИСТОРИЧЕСКАЯ СПРАВКА

Неоднозначное толкование

Было бы намного проще отдавать компьютеру команды на естественном языке. Но, к сожалению, машины не могут понимать все нюансы человеческой речи. Люди во время разговора используют жесты, мимику, намеки, разные речевые обороты. Это позволяет им, говоря одно, дать понять совершенно другое. Люди нередко используют слова и выражения, имеющие несколько значений.

Человеческий мозг может разгадать «загадки» естественного языка, а компьютер понимает только конкретную, математически точную систему. В такой системе каждый символ или группа символов всегда должны обозначать одно и то же, а каждое предложение должно быть понятно до буквы. Не должны допускаться неясные высказывания или запутанные намеки.

В этом смысле очень поучительна такая история:

*Поэт **Хабиби**, живший в XV веке во дворце **Шаха Исмоила Хатаи** и получивший псевдоним **Маликиуара**, в детстве был пастухом в гейчайском селе **Баргюшад**. Как-то он встретил отправившегося со своей свитой на охоту правителя **Аггоюнлу Султана Ягуба**, который спросил у него:*

– Чьи это ягнята?

– Овцы!

– Кто главный в селе?

– Бык.

– Ты не понял! Кто выходит навстречу, если к вам в деревню приходит важный человек?

– Собаки.

Султана Ягуба очень разозлили слова Хабиби и он с трудом удержался от того, чтобы не ударить его саблей.

ПРОВЕРЬ СЕБЯ

Какое из приведенных высказываний верно?

1. У разных объектов не могут быть одинаковые свойства.
2. Lazarus – визуальная среда программирования.
3. Основу среды программирования Lazarus составляет язык C++.
4. В каждой программе есть хотя бы одна форма.
5. Функция является одним из видов подпрограммы.

Выбери правильный ответ

1. ... – это действия объекта.

- | | |
|----------|--------------------|
| A. форма | B. составные части |
| C. цвет | D. методы |

2. Для обработки происшедшего события объект использует определенные

- | | |
|-------------|--------------|
| A. свойства | B. процедуры |
| C. методы | D. функции |

3. При нажатии на кнопку мыши происходит событие

- | | |
|--------------|------------|
| A. Click | B. Press |
| C. MouseDown | D. MouseUp |

4. Среда программирования Lazarus основана на языке

- | | |
|----------------|-----------------|
| A. Basic | B. Pascal |
| C. Free Pascal | D. Turbo Pascal |

5. Каждая отдельная команда на языке программирования называется

- | | |
|---------------|-----------------|
| A. оператором | B. указателем |
| C. функцией | D. предложением |

ПРОЕКТЫ

ПРОЕКТ 5-1

Почему на логотипе программы Lazarus изображен леопард? Соберите в Интернете информацию об истории этой программной системы. На основе собранных материалов подготовьте отчет.

ПРОЕКТ 5-2

На первой странице каждого раздела помещен по одному фрагменту картины «Метаморфозы III», созданной известным художником-графиком Маурицем Корнелисом Эшером (Maurits Cornelis Escher) в 1967-1968 годах. Найдите в Интернете информацию о художнике и картине. Создайте буклет, посвященный творчеству этого художника. Включите в буклет предложения о впечатлениях от картины «Метаморфозы III».

СОВМЕСТНЫЙ ПРОЕКТ

Сыграйте в игру, связанную с нахождением объекта по его свойствам и действиям. Один игрок мысленно задумывает любой объект. Остальные, для того чтобы определить задуманное, задают вопросы, связанные со свойствами и действиями объекта. Вопросы должны быть такими, чтобы на них можно было отвечать только «да» или «нет». Например, если вы задумали объект «верблюд», вопросы и ответы могут быть такими:

- Это объект живой природы? – Да.
- Растет? – Нет.
- Может летать? – Нет.
- Живет на севере? – Нет.

Если кто-то из игроков решит, что знает ответ, то может сказать его. Если он ошибется, то выходит из игры.

ФАЙЛ И ПАПКА

Для дополнительного чтения



Одним из основных понятий системы Windows является файл. **Файл** – это информация, которая хранится на определенном носителе (например, диске, магнитной ленте). Информация хранится в файле в закодированном виде. Это может быть текст, рисунок, программа или видеоизображение. Файлу обязательно нужно дать имя, с помощью которого его можно будет найти.

Полное имя файла состоит из двух частей: собственно имени и расширения, указывающего тип файла. Имя файлу задается во время его создания. В операционной системе DOS имя файла может состоять максимум из 8 символов. А в операционной системе Windows таких ограничений нет. Название файла может быть как длинным, так и коротким; со смыслом и бессмысленным; состоять как из букв, так и из цифр. Например, «Информатика-7», «Lesson1», «sgfg123», «Раздел 1». Некоторые символы, имеющие специальное назначение, использовать в имени файла нельзя. Тип файла указывает тип и назначение сохраненных данных. Как по фамилии узнают человека, так и по типу файла можно понять, какую информацию он хранит: текст, графику, звуковые данные или программу.

Нередко вместо понятия «тип файла» используют понятие «расширение файла». Между названием файла и расширением ставят точку. Например, в записи BAY.DOC слово BAY – это название файла, DOC – это его расширение. Если бы расширение файла каждый задавал по своему желанию, то оно не имело бы смысла. Поэтому для расширения файла приняты некоторые договоренности:



DOC, TXT	Расширения файлов, содержащих текст
BMP, JPG	Расширения файлов, содержащих графику
AVI, MPG	Расширения файлов, содержащих видеоизображение
WAV, MP3	Расширения файлов, содержащих звуковую информацию
BAK	Расширение файла, который является прежней версией данного файла
EXE, COM	Расширения программных файлов

Каждый файл в операционной системе Windows, кроме имени и расширения, имеет свой значок. Значок тоже соответствует типу файла, который он представляет.



Заглавные и строчные буквы в имени файла одинаково воспринимаются компьютером. Имя файла может содержать не более 255 символов. Нельзя использовать символы * / : < > ? \ | “, так как для Windows они имеют особое значение.

Параметрами файла являются также его размер, дата и время создания. *Размер файла*, как и объем информации, измеряют в байтах, килобайтах и мегабайтах. *Время и дата создания файла* регистрируются в системе как параметр файла в момент его создания. Они записываются в определенном порядке. Например, запись 20.05.11 22:16 указывает на то, что дан-

ный файл был создан 20 мая 2011 года в 22 часа 16 минут.

ПАРАМЕТРЫ ФАЙЛА			
Имя параметра	Значение параметра		
Имя файла	1_Algoritmika	Лес	Anketa
Тип файла	DOC	BMP	EXE
Значок файла			
Размер файла	126 Kb	15 Kb	136 Kb
Дата и время создания	09.10.10 00:41	07.03.07 19:28	29.09.98 16:08

Файлы можно создавать, хранить, открывать, переименовывать, копировать, перемещать, удалять, закрывать.

Как известно, в компьютерах информация хранится на дисках. Чтобы в дальнейшем ее легко можно было найти, нужно размещать ее правильно и аккуратно. Для этого пользуются **папками**. Файлы помещают в папки, как и листы бумаги.

Как и любой объект, папка имеет свое *имя*. Правила присвоения имен папкам такие же, как и файлам, единственное отличие в том, что папка не имеет расширения. Папку можно назвать по-любому, но для удобства желательно выбрать понятное и соответствующее ее содержанию имя, хотя имя папки A.B.C.D – корректное имя.

Одним из параметров папки является ее *размер*. Размер папки определяется суммой размеров всех файлов и папок, которые содержатся в ней. При соз-

дании папки система регистрирует дату и время ее создания. Как и с файлами, стандартные операции «создать», «открыть», «переименовать», «копировать», «переместить», «удалить», «закрыть» применимы и к папкам.



В некоторых операционных системах (например, DOS и UNIX) вместо термина «папка» используется термин «каталог».

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Алгоритм	72	Компилятор	111	Программное обеспечение	17
Алгоритмизация	72	Компиляция	111	Проект	103
Алгоритм с ветвлением	80	Компьютерная модель	28	Процедура	108
Арифметическое выражение	114	Линейный алгоритм	78	Пункт	55
Блок-схема	76	Математическая модель	27	Рабочий стол	114
Буклет	64	Материальная модель	27	Разделитель	112
Ветвление	80	Машинный язык	111	Размер файла	125
Грань	33	Мейнфрейм	11	Расширение имени файла	125
Графическая модель	27	Метод объекта	96	Ребро	33
Графический интерфейс пользователя	19	Микрокомпьютер	12	Свойство объекта	95
Графический примитив	31	Миникомпьютер	11	Сетевой этикет	15
Графический редактор	29	Модель	26	Системные программы	18
Двухмерная графика	29	Моделирование	17	Смайлик	16
Значок	19, 125	Настольные издательские системы	18	Событие	98
Идентификатор	113	Окно формы	101	Среда Lazarus	94
Имя файла	125	Окно кода	102	Стилус	12
Инспектор объектов	101	Окно моделирования	29	Стиль	41
Инструментарий программирования	18	Объект	94	Строка заголовка	103
Интернет	11	Оператор	112	Строка меню	29, 100
Интерфейс	57	Операционная система	18	Суперкомпьютер	11
Интерфейс пользователя	19	Палитра компонентов	101	Тень	43
Информационный бюллетень	60	Панель задач	59	Тип файла	125
Информационная культура	14	Панель инструментов	29, 101	Трехмерная графика	29
Информационная модель	27	Папка	126	Файл	125
Информационное общество	14	Переменная	113	Форма	101
Карманный компьютер	12	Перетаскивание	28	Формат бумаги	57
Каталог	126	Периферийные устройства	29	Функция	108
Кегль	55	Планшетные компьютеры	52	Шрифт	54
		Постоянная	113	Шрифт без засечек	55
		Прикладные программы	29	Шрифт с засечками	55

ПРОГРАММА

по информатике для 7-го класса

общеобразовательной школы

Один час в неделю, всего 32 часа

1. Информационное общество (2 ч.).

Информационная культура. Сферы применения компьютерной техники. Сетевой этикет.

2. Информационные технологии (2 ч.).

Классификация компьютеров. Виды программного обеспечения. Графический интерфейс пользователя и его объекты.

3. Прикладные программы (9 ч.).

Программа Microsoft Publisher. Информационный бюллетень. Информационный буклет. Программа Google SketchUp. Трехмерное моделирование. Моделирование на основе фотографий. Стили и тени.

4. Алгоритм (4 ч.).

Стандартные алгоритмы (нахождение максимальных и минимальных элементов, сортировка). Создание алгоритмов решения разных задач.

5. Программирование (14 ч.).

Объектно-ориентированная среда программирования на базе Pascal (Lazarus или Delphi). Формы, элементы управления. События. Математические операции. Типы данных и переменные. Строки. Условие. Цикл.

6. Заключение (1 ч.).