

Р.А. Кадиркулов, Г.К. Нурмуханбетова

ИНФОРМАТИКА

Учебник для учащихся 8 класса
общеобразовательной школы

*Рекомендовано
Министерством образования и науки
Республики Казахстан*

Алматыкітап баспасы
2018

УДК 373.167.1
ББК 32.973 я 72
К 79

Условные обозначения

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

Кадиркулов Р.А.

К 79 **ИНФОРМАТИКА:** Учебник для учащихся 8 класса общеобразовательной школы с русским языком обучения./Р.А. Кадиркулов, Г.К. Нурмуханбетова – Алматы: Алматыкітап баспасы, 2018. – 184 с.

ISBN 978-601-01-3518-5

УДК 373.167.1
ББК 32.973 я 72

ISBN 978-601-01-3518-5

© Кадиркулов Р.А., Нурмуханбетова Г.К., текст, 2018
© ТОО «Алматыкітап баспасы», 2018

ДОРОГИЕ ДРУЗЬЯ!

Вы продолжаете изучение информатики. В течение учебного года вы познакомитесь со структурой компьютера, компьютерными сетями, с представлением и измерением информации, электронными таблицами и моделированием, основами составления алгоритмов и программирования.

Для развития критического мышления весь учебный материал построен на основе познавательных уровней таксономии Блума. При работе с учебником вы научитесь понимать и использовать теоретический и практический материал. На уровне анализа научитесь его глубоко осмысливать, на уровне синтеза – формулировать выводы, предлагать свои идеи, разрабатывать проекты. На уровне оценивания научитесь давать оценку важности изученного материала в области информатики и его использования в повседневной жизни. Для более легкого усвоения материала каждая тема по спиральной системе обучения переходит от простого к сложному. Практические задания составлены таким образом, чтобы их удобно было выполнять с учебником и самостоятельно. Изученные темы найдут свое логическое продолжение в 9 классе. Например, в этом году мы будем рассматривать простые способы работы с электронными таблицами, и эта тема будет усложнена и продолжена в 9 классе.

При работе с учебником вы приобретете умения и навыки:

- разработки проектов заданий на основе анализа, синтеза и оценки как теоретического материала, так и практических работ, – в группе или в паре;
- понимания и конструктивного оценивания роли информационных процессов в обществе;
- понимания и определения проблем использования информационных технологий в различных отраслях человеческой деятельности;
- эффективного использования информационных технологий в повседневной жизни, учебе и в будущей трудовой деятельности;
- анализа, абстрагирования, моделирования и решения задач по программированию;
- освоения научно-технического языка.

Желаем вам удачи!

1.1

ИЗМЕРЕНИЕ ИНФОРМАЦИИ. ВЕРОЯТНОСТНЫЙ ПОДХОД



Как можно применить вероятностный подход к измерению объема информации?



Подумай

- Для чего нужно определять объем информации?
- Какие виды измерения объема информации вы знаете?



Новые знания

Вероятностный подход

Объем информации в сообщении о каких-либо событиях зависит от вероятности его получения. Имеются различные устройства и системы, которые дают возможность собрать и обработать информацию.

Сообщение – это информация, представленная в определенной форме, которая в процессе передачи данных поступает к получателю.

Сообщение – это *слова*, которые мы слышим (радиопередачи, объяснение учителя), *видеоизображение*, которое мы визуальное воспринимаем (фильм, сигнал светофора), *текст* книги, который мы читаем и т.д.

Информативность сообщения характеризуется содержащейся в нем полезной информацией, для определения объема которой необходима единица измерения. Для человека получение новой информации приводит к расширению знаний или к уменьшению неопределенности. Есть два вида измерения информации.

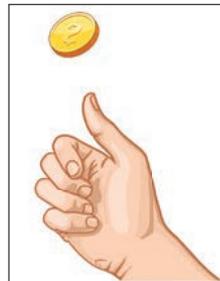
**Вероятностный подход определения количества информации***

Сообщение, уменьшающее неопределенность знания в два раза, несет 1 бит информации.

Сообщение об одном из двух равновероятных результатов некоторого события несет 1 бит (условно 0 или 1) информации.

Примечание: Определение «бит» в данной формулировке может быть сложным для восприятия, потому что содержит непонятный термин «**неопределенность знаний**». Сначала нужно раскрыть смысл этого термина. Рассмотрим примеры, чтобы понять его.

Пример 1. Подбрасывая монету, вы загадываете, что выпадет: орел или решка. Есть всего два возможных результата подбрасывания монеты. Причем ни один из этих результатов не имеет преимуществ перед другим. В таком случае говорят, что они равновероятны. Узнав результат подбрасывания монеты, вы получили один бит информации.



Так как у монеты есть две стороны, и мы не знаем, какой стороной она упадет, то перед ее подбрасыванием **неопределенность знаний** равна двум. Следовательно, результатом будет решение уравнения $2^x = 2$ и получаем, что $x = 1$ бит.

Пример 2. В игре в кости используют игральный кубик с шестью гранями. Сколько бит информации каждый бросок кубика несет игроку?

Примечание: Каждая из шести сторон кубика имеет равную вероятность выпадения. Тогда неопределенность знаний о результате броска кубика уменьшится в 6 раз. Количество информации определится при решении уравнения $2^x = 6$.



Вероятностное событие с ограниченной возможностью числа N исходов считается **случайным событием**.

В 1928 г. американский инженер **Ральф Хартли** рассматривал процесс получения информации как выбор одного сообщения из конечного, наперед заданного множества N равновероятных сообщений.

Пусть в некотором сообщении содержатся сведения о том, что произошло одно из N возможных равновероятных событий. Тогда количество информации, заключенное в этом сообщении, – x бит и число N – количество событий, связаны формулой Хартли. С основными особенностями формулы Ральфа Хартли вы ознакомитесь в старших классах.

$$N = 2^x$$

N – количество равновероятных событий;
 x – количество бит в сообщении.

Таблица 1

Количество бит	0	1	2	3	4	5	6	7	8	9	10
Количество событий	1	2	4	8	16	32	64	128	256	512	1024

Примечание: По таблице можно определить количество бит, соответствующее числу событий. Например, на уроке по пятибалльной системе можно получить одну из четырех оценок «5», «4», «3», «2» с равной вероятностью, то есть имеется 4 события (оценку «1» мы не будем рассматривать). В соответствии с таблицей 4 события несут 2 бита информации. Разумеется, удобно, когда количество событий соответствует целой степени числа 2. Если же количество событий будет 14 или 23, то появляется необходимость использования математического понятия – логарифм. С понятием «логарифм» вы познакомитесь в старших классах.



Анализ задачи

Пример №1

Ученики пошли в бассейн с 8 дорожками. Инструктор сообщил им, что они будут плавать по 3 дорожкам. Какое количество информации получили ученики из данного сообщения?

Запишите краткую информацию к примеру по следующему образцу:

Дано:

$$N = 8$$

Найти: x -?

Решение:

$$N = 2^x$$

$$8 = 2^x \Rightarrow 2^3 = 2^x, x = 3$$

Ответ: 3 бита

Примечание: Мы должны ответить на вопрос: число 2 в какой степени равно 8.

Пример №2

Тимур живет на третьем этаже многоэтажного дома. Это сообщение несет 4 бита информации. Сколько этажей в доме Тимура?

Дано:

$$x = 4$$

Найти: N -?

Решение:

$$N = 2^x \Rightarrow N = 2^4 \Rightarrow N = 16$$

Ответ: 16 этажей

Примечание: в данном случае на ответ не влияет на каком этаже живет Тимур. Даже если Тимур жил бы на 7 этаже, ответ был бы таким же.

Пример №3

В школьной библиотеке книги расположены на 16 стеллажах с 8 полками каждая. Библиотекарь сообщила Диане, что книга, которая ей нужна, находится на пятой полке седьмого стеллажа. В каком количестве ученица получила информацию?

Дано:

$$N_1 = 16$$

$$N_2 = 8$$

Найти: x -?

$$N_1 = 2^{x_1}; N_2 = 2^{x_2};$$

$$\text{Если } 16 = 2^{x_1}, x_1 = 4;$$

$$\text{Если } 8 = 2^{x_2}, x_2 = 3$$

Общая информация, воспринятая ученицей:

$$x = x_1 + x_2 \qquad x = 4 + 3 = 7$$

Ответ: 7 бит**Примечание:** Для получения результата достаточно знать количество стеллажей и полок.

Задания

1. Поиск числа из промежутка $1..N$ дает 9 бит информации. Чему равно число N ?
2. Известно, что сообщение «Анна потеряла букварь» несет 4 бита информации. Сколько учебников у Анны?
3. Какое количество информации необходимо для кодирования координат поля шахматной доски размером 8×8 ?
4. Было принято сообщение размером 10 бит. Какое количество информации можно получить из данного сообщения?
5. В коробке имеется 16 разноцветных шаров. Какое количество информации несет сообщение о том, что из коробки взят белый шар?
6. Чему равен объем сообщения о событии с вероятностью, равной 64 ?
7. Сообщение, что книга, интересующая ученика, расположена на 5 полке, несет 3 бита информации. Сколько полок на стеллаже?
8. В коробке имеется 40 шаров, из которых 8 белых и 32 синих. Какое количество информации несет сообщение о том, что взят белый шар? Синий шар?
9. В коробке имеется 64 цветных карандаша. Сообщение о том, что взят красный карандаш, несет 4 бита информации. Сколько красных карандашей в коробке?
10. Поиск числа из промежутка $1..N$ несет 6 бит информации. Чему равно число N ?
11. Получена информация объемом 8 бит. Какого размера информацию можно составить из этих данных?
12. В школьной библиотеке число стеллажей несет 5 бит информации, а число полок на стеллаже несет 4 бита информации. Какое количество стеллажей и полок на каждом стеллаже в школьной библиотеке?



Вопросы

1. Чем определяется объем знаний человека, получающего сообщение?
2. Что такое вероятностный подход измерения объема информации?
3. Чему равно сообщение, уменьшающее неопределенность знаний в два раза?
4. Что характеризует формула Ральфа Хартли?
5. Что такое случайное событие?

1.2 ИЗМЕРЕНИЕ ИНФОРМАЦИИ. АЛФАВИТНЫЙ ПОДХОД.



Как применить алфавитный подход для измерения объема информации?



Подумай

- Что такое алфавитный подход измерения объема информации?



Новые знания

Алфавитный подход

Алфавитный подход к измерению информации эквивалентен ранее рассмотренному вероятностному подходу. Здесь рассматривается измерение количества информации текстового (символьного) сообщения, составленного из символов какого-нибудь алфавита. При алфавитном подходе к измерению информации количество информации не зависит от содержания. Алфавитный подход – **единственный способ**, дающий возможность измерения информации, обрабатываемой на компьютерах, вычислительной технике, мобильных устройствах.

А Э Б В Г Г
 Д Е Ё Ж З И
 Й К Қ Л М Н
 Њ О Ө П Р С
 Т У Ұ Ү Ф Х
 Һ Ц Ч Ш Щ Ъ
 Ы І Ь Э Ю Я

Основным понятием данного подхода является – алфавит. **Алфавит** – ограниченное множество символов, предназначенное для представления информации. Число символов алфавита называют **мощностью алфавита**.

Если считать, что алфавит – это событие, то каждый из символов в сообщении можно рассматривать как одно из состояний события. Если появление символов равновероятно, то можно рассчитать, сколько бит информации несет каждый символ (X) с помощью уравнения Хартли.

$$N = 2^X$$

N – мощность используемого алфавита.

X – **информационный вес** символа.

Отсюда можно сделать вывод: Для вычисления информационного объема сообщения I , состоящего из K символов, необходимо информационный вес одного символа X умножить на K .

$$I = K * X$$

I – информационный объем сообщения;
 K – мощность;
 X – **информационный вес** символа.

Величину количества информации можно назвать информационным объемом текста. Наименьшая мощность алфавита для передачи информации равна 2. Такой алфавит называется **двоичным**.

В двоичном алфавите информационный вес символа равен одному биту, так как $2^x = 2$. Таким образом, один символ двоичного алфавита несет 1 бит информации. В дальнейшем более подробно рассмотрим алфавит внутреннего (машинного) языка компьютера – язык двоичного кодирования.

Бит – основная единица измерения информации. **Байт** – информационный вес символа алфавита. В современных вычислительных системах байт состоит из восьми битов и, соответственно, может принимать одно из 256 (2^8) различных значений (состояний, кодов).

Количество информации зависит от объема текста (от числа знаков в тексте) и от мощности алфавита.

Пример. Сообщение, состоящее из 20 символов, записано буквами алфавита, представленного 64 символами. Какое количество информации оно несет? Ответ: 120 бит информации.

Примечание: Здесь задан алфавит мощностью $N=64$, количество символов $K=20$. Тогда по формуле Хартли информационный вес (X) составляет 6 бит, количество информации (I) составляет: $20 * 6 = 120$ бит.



Анализ задач



Пример №1

Сообщение, содержащее 40 символов, составлено с помощью 32-символьного алфавита. Каков информационный объем сообщения?

Дано:

$$N=32$$

$$K=40$$

Найти: I -?

Решение:

$$N = 2^x \Rightarrow 32 = 2^x \Rightarrow x = 5$$

$$I = K * x \Rightarrow I = 40 * 5 \Rightarrow I = 200 \text{ бит.}$$

Ответ: 200 бит.

Пример №2

Электронная книга содержит 150 страниц, на каждой странице – 40 строк, в каждой строке – 60 символов. Определите информационный объем книги.

Дано:

$$K_1=150$$

$$K_2=40$$

$$K_3=60$$

$$N=256$$

Найти: I –?

Решение:

Формула для вычисления информационного

объема: $I = K * X$

$$N=2^x \Rightarrow 256=2^x \Rightarrow X=8 \text{ бит};$$

$$K=K_1 * K_2 * K_3; K=150 * 40 * 60; K=360000;$$

$$I=8 \text{ бит} * 360000=2880000 \text{ бит}=360000 \text{ байт}=351,56 \text{ Кб.}$$

Ответ: 351,56 Кб

Примечание: Мощность алфавита, используемого в компьютере, равна 256.

Пример №3

Будем считать, что алфавит казахского языка состоит из 42 букв, 12 знаков препинания и 10 цифр, то есть 64-символьный алфавит. Алфавит английского языка пусть состоит из 32 символов, включающих 26 букв и 6 знаков препинания. Сравните объем информации в сообщениях, составленных из 70 символов алфавита казахского языка и из 70 символов алфавита английского языка.

Дано:

$$N_1=64$$

$$N_2=32$$

$$K_1=70$$

$$K_2=70$$

Найти: $\frac{I_1}{I_2}$ –?

Решение:

Формула для вычисления информационного

объема: $I = K * X$

$$N_1=2^{X_1} \Rightarrow 64=2^{X_1} \Rightarrow X_1=6 \text{ бит};$$

$$N_2=2^{X_2} \Rightarrow 32=2^{X_2} \Rightarrow X_2=5 \text{ бит};$$

$$I_1=70 * 6 \text{ бит}=420 \text{ бит}; \quad I_2=70 * 5 \text{ бит}=350 \text{ бит}$$

$$\frac{I_1}{I_2} = \frac{420}{350} = 1.2$$

Ответ: больше в 1,2 раза



Применение

1. В Южной Америке на берегах реки Амазонки найдено неизвестное до сих пор племя индейцев. Племенной алфавит состоит только из 8 букв. Какой объем информации несет одна буква алфавита?
2. История развития планеты Земля показывает, что на ней существовали и исчезли несколько цивилизаций. Согласно дошедшей до нас информации, в одной из этих цивилизаций

информационный объем одного символа составлял 6 бит. Определите количество символов алфавита потерянной цивилизации.

3. Индейское племя, имеющее алфавит из 128 символов отправило письмо, состоящее из 30 букв, с курьером в соседнее племя. Какой объем информации доставил курьер?
4. Определите количество килобайт в сообщении из 512 символов, составленном с помощью 16-символьного алфавита?



Анализ



Сравните вероятностный и алфавитный подходы измерения информации. Проведите анализ сходства, особенностей, эффективности, удобства и легкости применения.



Синтез



Составьте примеры на измерение количества информации алфавитным подходом. При составлении примеров сделайте акцент на актуальность их содержания.



Оценка



Какой подход, вероятностный или алфавитный, вы выбрали бы при измерении количества информации? Почему?



Вопросы

1. Что такое алфавит?
2. Что определяет мощность алфавита?
3. Что такое информационный вес символа алфавита?
4. Как можно измерить информационный объем текста с помощью алфавитного подхода?
5. Какие формулы и величины используются при алфавитном подходе измерения информации?
6. Что такое байт, килобайт, мегабайт, гигабайт?

Исследовательская работа	При исследовании обратите внимание на следующие темы
Подходы к измерению информации	<ul style="list-style-type: none"> • История возникновения измерения информации с древности до наших дней. • Информация, которая нас окружает. • Единицы измерения информации. • Методы измерения информации.

1.3

**ВНУТРЕННЕЕ УСТРОЙСТВО КОМПЬЮТЕРА
И ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА**

Как устроена архитектура персонального компьютера?

**Подумай**

- Что такое бионика?
- Какое отношение имеет бионика к компьютеру?
- Что такое архитектура компьютера?
- Как вы думаете, какие идеи взяты за основу при разработке архитектуры компьютера?

Бионика (от др.-греч. βίον – живущее) – прикладная наука о применении в технических устройствах и системах принципов организации, свойств, функций и структур живой природы, то есть живые формы в природе и их промышленные аналоги.

**Новые знания****Внутреннее устройство компьютера**

Персональный компьютер – универсальная техническая система. Его конфигурацию (основная конфигурация: системный блок, монитор, клавиатура) можно изменять по своему усмотрению.

Системный блок считается основным устройством. Устройства, расположенные внутри системного блока, называются внутренними, а устройства, присоединяемые к нему снаружи, называются внешними или периферийными.

Внутренние устройства системного блока:

- Материнская плата;
- Процессор или микропроцессор;
- Оперативная память (ОЗУ);
- Жесткий диск;
- Привод оптических дисков (CD и DVD);
- Видеографический адаптер;
- Звуковая карта;
- Сетевая плата;
- Порты ввода-вывода;
- Блок питания.



Рис. 1. Материнская плата

Материнская плата

Материнская плата (англ. motherboard или mainboard) – это основа компьютера, к которой подключаются все его остальные элементы. Она представляет собой печатную плату, на которой установлены различные микросхемы, элементы и разъемы, и служит посредником при взаимодействии различных узлов компьютера (рис. 1).

На материнской плате располагаются основные элементы: процессор, оперативная память, набор управляющих микросхем (чипсет), BIOS (Basic Input Output System – базовая система ввода-вывода), кеш-память, шины, слоты расширения и другие устройства. На материнской плате имеются разъемы для параллельных и последовательных портов (для подключения клавиатуры и мыши), источника питания, встроенного динамика, индикаторов и кнопок, находящихся на передней панели системного блока.



Процессор – это электронная микросхема, состоящая из множества полупроводниковых элементов (кристаллов), управляющая работой каждого устройства компьютера, а также выполняющая все вычисления и обработку информации на нем. Полупроводниковый кристалл выполняется из особо чистого кремния. При его изготовлении используют высокоточные технологии.

На одном кристалле располагаются сотни тысяч взаимосвязанных электронных молекул, выполняющих сложные преобразования информации. Разрядность процессора определяет, какое количество бит информации он примет и обработает через свои регистры за один такт. Количество элементарных операций (тактов), выполняемых процессором за единицу времени, определяется тактовой частотой. Единицы измерения тактовой частоты – герц (Гц), мегагерц (МГц), гигагерц (ГГц).

Компьютерный блок питания (или сокращенно – блок питания, БП) – вторичный источник электропитания, предназначенный для снабжения узлов компьютера электроэнергией постоянного тока путем преобразования сетевого напряжения до требуемых значений. Как правило, компьютер использует 5 или 12 вольт низкого напряжения. Различие блоков питания зависит от мощности и стабилизации напряжения. Разноцветные провода определяют величину напряжения: желтый +12 В, красный +5 В, черный 0. Мощность БП от 150 ватт до 320 ватт. В блоке питания расположен вентилятор, который охлаждает его и другие устройства, расположенные в системном блоке. Он охлаждает не только блок питания, но и все устройства, расположенные в системном блоке.



Шина (англ. computer bus) в архитектуре компьютера — подсистема, служащая для передачи данных между его функциональными блоками. Имеется три основных вида шин: шина данных, адресная шина и командная шина. Шина данных — часть общей шины, предназначенная для параллельной передачи данных двоичного разряда между устройствами компьютера. Шина адреса представляет собой набор проводников, по которым передается адрес ячейки памяти, в которую или из которой пересылаются данные. Типы шин: ISA, MCA, EISA, SCSI, VLB, PCI, AGP, USB и т.д.



Чипсет (англ. chipset) — набор микросхем, спроектированных для совместной работы с целью выполнения набора заданных функций. Обеспечивает взаимодействие центрального процессора с различными типами памяти, устройствами ввода-вывода, контроллерами и адаптерами периферийных устройств.



BIOS — считается одним из основных устройств компьютера и располагается на материнской плате. Микросхема BIOS предназначена для установления параметров системы (рис. 2).



BIOS выполняет следующие функции:



Рис. 2. UEFI BIOS

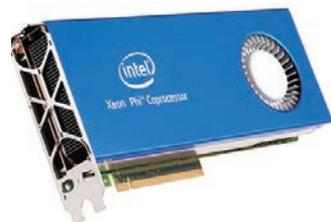
- Тестовая система POST проверяет после включения компьютера его основные компоненты;
- Настраивает параметры аппаратных устройств;
- Обеспечивает загрузку операционной системы с диска;
- Обеспечивает взаимодействие системных и аппаратных компонентов.

В настоящее время существует множество видов BIOS, самыми распространенными из них являются Award BIOS, AMIBIOS, Phoenix BIOS и UEFI BIOS.

До 2000 года в большинстве случаев применяли Award BIOS. Это довольно старая система, которая использовалась на компьютерах с процессорами Pentium II, Pentium III. BIOS была очень надежно защищена, так как в случае ее «поломки» будет невозможно запустить компьютер. На современных компьютерах возможность перезаписи BIOS существует, поэтому перешли на систему Phoenix BIOS.

UEFI (Unified Extensible Firmware Interface, унифицированный расширенный интерфейс) – самый последний вариант BIOS. Базовая система ввода-вывода определяет ход процесса загрузки компьютера, когда происходит загрузка операционной системы. А дальнейшая работа происходит под управлением ОС, обеспечивается взаимодействие различных устройств и всех комплектующих компьютера между собой.

Сопроцессор – специализированный процессор, расширяющий возможности центрального процессора компьютерной системы. У сопроцессора нет своей программы, он расширяет систему команд и регистров основного процессора, вместе с ним выполняет работу. Он не управляет системой, только ждет команду от центрального процессора. Сопроцессор дает возможность сократить выполнение арифметических операций на 80% и более, ускорить обработку фрагментов видеоизображений.



Слот расширения – разъем на материнской плате, соединенной с системной шиной, который позволяет подключать к нему дополнительные платы, расширяющие конфигурацию устройства. Через слоты обычно подключаются видеокарты, звуковые карты, сетевые карты.



Слоты расширения компьютера:

- AGP – видеокарта;
- AMR – модем, звуковая карта;
- CNR – модем, сетевая карта, звуковая карта;
- EISA – SCSI, сетевая карта, видеокарта;
- ISA – сетевая карта, звуковая карта, видеокарта;
- PCI – сетевая карта, звуковая карта, SCSI;
- PCIe – видеокарта, сетевая карта, звуковая и др.

Дополнительные карты, подключаемые к материнской плате

Видеокарта – это устройство, предназначенное для вывода на экран монитора графической информации. При помощи видеокарты данные, передаваемые компьютером, преобразовываются в видеосигнал и передаются на монитор. Видеопамять (Video Memory), или VRAM (Video RAM) – это внутренняя оперативная память для хранения данных, которые используются для формирования изображения на экране монитора. Чем больше объем памяти видеокарты, тем больше данных в ней может храниться.



Звуковая карта дает возможность создавать на компьютере высококачественный звук и записывать его. Карта предоставляет расширенные возможности работы со звуком, например, в компьютерных играх и других программах.



Сетевая карта дает возможность соединения компьютера с компьютерной сетью. Имеется несколько видов сетевых карт: **Ethernet, Token Ring** и платы, предназначенные для подключения к беспроводной сети (WI-FI).



Порты ввода-вывода – разъемы на передней или задней панели корпуса, к ним с помощью кабеля подсоединяются различные устройства. Виды и количество устройств зависят от вида и количества портов.



Периферийные устройства персонального компьютера



Рис. 3. Периферийные устройства компьютера

Анализ

Выполните анализ рисунка 3:

- определите названия периферийных устройств компьютера;
- выясните функции этих устройств и опишите, к каким дополнительным платам они подсоединяются;
- определите, какие еще устройства, кроме периферийных, имеются у компьютера.

Синтез

Составьте общие правила использования периферийных устройств компьютера. Поясните каждый пункт составленных правил: почему именно так составлено данное правило.

Оценка

Оцените роль материнской платы в устройстве компьютера.

**Вопросы**

1. Назовите внутренние устройства компьютера.
2. Что такое материнская плата?
3. Какие устройства подсоединяются к материнской плате?
4. Какие функции выполняют периферийные устройства?
5. Что такое процессор?
6. Каково назначение графического адаптера памяти?
7. Какие дополнительные платы есть в компьютере?
8. Какие устройства подсоединяются к портам ввода-вывода системного блока?

**Задания****Темы для исследовательской работы:**

№	Темы исследования	Вопросы и направления, раскрывающие тему исследования
1	История развития принтеров.	<ul style="list-style-type: none"> • Какие устройства в истории человечества, предназначавшиеся для записи и печати, вам известны? • Анализ современных печатающих устройств • Какие виды принтеров вам известны, и как они работают? • Что вы знаете о 3D-принтерах?
2	Я и интернет.	<ul style="list-style-type: none"> • Исследуйте возникновение интернета и процесс его развития. • Исследуйте устройства, необходимые для подключения компьютера к интернету и опишите их виды. • Исследуйте пути развития и варианты использования интернета в Казахстане.
3	Развитие USB-устройств.	<ul style="list-style-type: none"> • Как появились USB-порты? • В каком направлении USB-порты будут развиваться в дальнейшем?
4	Что такое архитектура компьютера?	<ul style="list-style-type: none"> • Исследуйте тенденции развития архитектуры современных компьютеров.

Раздел I. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ КОМПЬЮТЕРА И СЕТЕЙ

1.4 Процессор и его характеристики



Что вам известно о процессоре компьютера и его основных характеристиках?



Подумай

- Какие технические устройства можно представить моделью человека? Почему? Приведите примеры.
- Как ты думаешь, какие сходства есть между деятельностью человека и работой компьютера?
- Может ли компьютер копировать действия человека?
- Может ли компьютер полностью заменить человека?

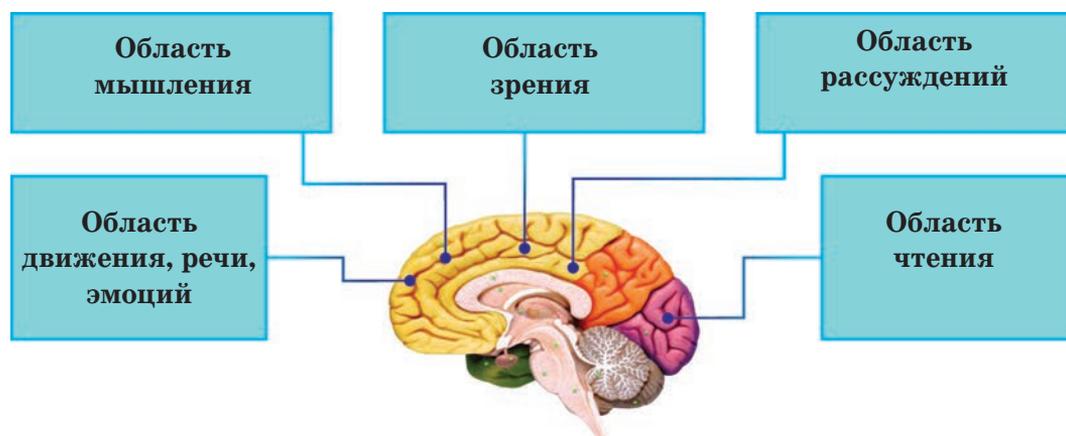


Новые знания

Мозг человека регулирует работу всех органов, ответственных за жизнь.

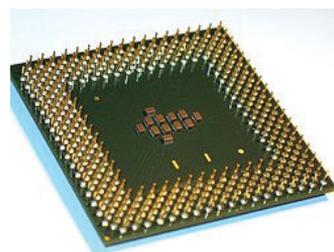
Компьютер (англ. computer – «вычислитель») – электронное устройство, предназначенное для автоматизации получения, обработки, хранения, передачи и использования информации.

Функция	Человек	Компьютер	Устройства
Хранение информации	Память	Устройства памяти	Постоянная, оперативная, кэш-память и внешняя память
Обработка информации	Система мышления	Процессор	Процессор
Восприятие информации	Органы чувств	Устройства ввода	Клавиатура, микрофон, сканер, камера, мышь
Передача информации	Речь, органы движения	Устройства вывода	Звуковые колонки, принтер, плоттер



Области человеческого мозга, отвечающие за восприятие, обработку, хранение информации

Центральный процессор (англ. central processing unit, CPU – центральное обрабатывающее устройство) – это «мозг» компьютера, его основная микросхема. Процессор управляет работой каждого устройства компьютера и разрешает выполнение программного кода. Быстродействие компьютера напрямую зависит от скорости процессора. Работа процессора заключается в выполнении и выборе из памяти команд и данных в определенной последовательности. Это и является основой выполнения программ.



Центральный процессор вычисляет и обрабатывает данные, а также является самой дорогой микросхемой компьютера.

Основные производители процессоров – компании Intel и AMD.

Двухъядерный процессор производителя Intel называется Core 2 Duo, а производителя AMD – Dual Core.

Чем выше частота процессора, тем больше тепла он выделяет. Эта проблема была решена размещением в одном корпусе двух процессоров. Такие процессоры называют многоядерными. В настоящее время лидерами рынка являются процессоры фирмы Intel.

Процессоры различаются разрядностью, тактовой частотой и объемом кеш-памяти.

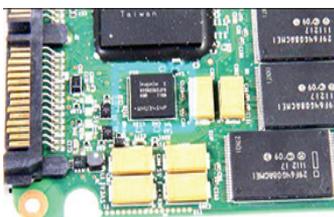
Основные параметры процессора



Разрядность процессора – величина, определяющая количество информации, которое центральный процессор способен обработать за один такт. Наименьший разряд единицы информации – двоичный. В мире ЭВМ один разряд называется битом (bit – часть). Если ЭВМ за один раз может обработать восемь разрядов, тогда процессор считается 8-разрядным, если 32 разряда, процессор называется 32-разрядным. Чем выше разрядность процессора, тем быстрее идет обработка данных.



Частота процессора (frequency). Важной характеристикой, влияющей на функциональность процессора, является тактовая частота. Она измеряется в мегагерцах. Один мегагерц – это миллион тактов в секунду. 100 МГц – сто миллионов тактов в секунду. За один такт процессор вычисляет какой-нибудь фрагмент. Поэтому, чем выше тактовая частота, тем быстрее процессор обработает поступившие данные.



Кеш-память процессора. Данные, обрабатываемые процессором, поступают из оперативной памяти. Как правило, процессор у себя ничего не хранит. В нем очень мало ячеек, в которых обрабатываются данные. Эти ячейки называются регистрами. Для облегчения работы процессора долгое время используется технология кеширования. Кеш – это промежуточный буфер с быстрым доступом. Когда какие-то данные обрабатываются в основной памяти, их копия попадает в кеш-память. В случае повторного использования этих данных, их сразу можно извлечь из буфера. Если какие-либо данные нужны процессору, он, в первую очередь, проверяет регистры. Если нужных данных в реги-

страх нет, то процессор проверяет кеш-память. Если этих данных нет в кеш-памяти, процессор проверяет оперативную память. Если же и в оперативной памяти нет нужных данных, процессор делает запрос HDD-диску.

Устройство процессора

В настоящее время процессор является техническим устройством, играющим важную роль в нашей жизни. Люди используют его в электронных устройствах в медицине, образовании, военном деле и других отраслях. Процессоры используются не только в компьютерах, но и в телевизорах, телефонах, холодильниках. Развитие микропроцессоров расширило функциональные и вычислительные возможности электронных устройств. Современные процессоры могут содержать два (двухъядерные) и более (многоядерные) вычислительных ядер. Первоначально двухъядерные процессоры имели два независимых ядра. Каждое процессорное ядро представляет собой отдельный процессор с кеш-памятью первого и второго уровней. Самым производительным считается компьютер для игр, оснащенный процессором с четырьмя ядрами (рис. 1). В настоящее время центр ядра составляет кеш-память второго уровня, она еще больше увеличивает производительность. Использование



Рис 2. 8-ядерный процессор



Рис 1. 4-х ядерный процессор

Американская компания Intel впервые выпустила 8-ядерный процессор Intel Core i7-5960X Extreme Edition (рис. 2). В процессоре размещен 1 млрд транзисторов.

многоядерного процессора увеличивает производительность процессора. У него появляется возможность одновременно выполнять несколько текущих команд. По-другому можно сказать, что многоядерный процессор – это множество одноядерных процессоров. На материнской плате они объединяются и взаимосвязываются. Производительность многоядерных процессоров увеличивается за счет того, что данные разделяются между несколькими ядрами для обработки. На рис. 3 представлена простейшая схема ядра процессора.

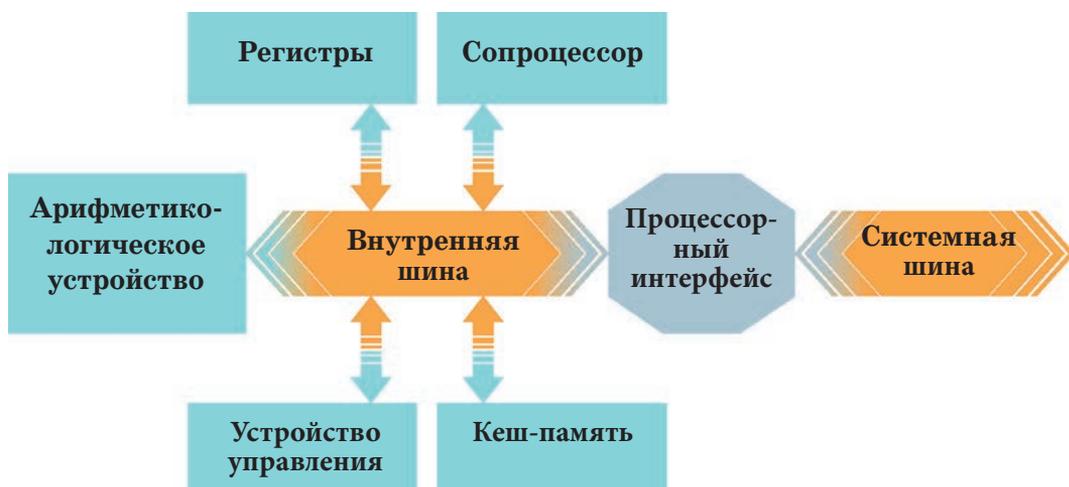


Рис. 3. Простейшее устройство ядра процессора

Арифметико-логическое устройство (АЛУ) – выполняет все математические и логические операции.

Устройство управления (УУ) – обеспечивает последовательность выполнения команд процессора.

КЕШ-память – быстросействующая память малого объема, используемая процессором для временного хранения информации. Кеш используется в качестве буфера для ускорения обмена данными между процессором и оперативной памятью. Значения из кеш-памяти извлекаются напрямую, без обращения к основной памяти.

Регистры – ячейки памяти, расположенные внутри процессора. Регистр является устройством временного хранения данных, числа или команды и используется с целью облегчения арифметических, логических и пересылочных операций.

Сопроцессор – устройство, расширяющее вычислительные возможности основного процессора и предназначенное для обработки чисел с плавающей запятой.

Процессорный интерфейс – разъем на материнской плате, который предназначен для размещения процессора. Каждый разъем материнской платы предназначен для определенной группы процессоров.

Системная шина – информационная магистраль, обеспечивающая связь всех устройств компьютера.



Анализ



Проведите сопоставительный анализ работы процессора с мозгом человека.



Синтез



«Компьютер внутри нас». Задумайтесь над информационным процессом в организме человека. К примеру, человек от полученной информации расстраивается или радуется и т.д.



Оценка



Оцените значимость процессора в организации работы компьютера.



Вопросы

1. Что такое процессор?
2. Каковы его функции?
3. Какими параметрами различаются процессоры? Опишите их.
4. Опишите устройство процессора.
5. Что такое многоядерный процессор?



Задание

Темы исследования	Вопросы и направления, раскрывающие тему исследования
«Central Processor Unit»	<ul style="list-style-type: none"> • Изучите историю появления первых процессоров и историю их развития. • Какие фирмы, выпускающие процессоры, являются ведущими игроками рынка? • Каково внутреннее устройство процессора? • На каких принципах основывается работа процессора?
Бионика	<ul style="list-style-type: none"> • Что такое Бионика? • Становление Бионики как науки. • Взаимосвязь природы и техники. • Какую пользу человечеству приносит Бионика?
Схожесть жизнедеятельности живых организмов с техникой	<ul style="list-style-type: none"> • Копирование элементов живой природы. Моделирование. • Заимствование технологий и идей из живой природы и применение их в технике. • Жизнедеятельность птиц и животных, строение их организма. Структура живой природы. Применение в быту и технике принципов, свойств, функций и структур живой природы.

1.5 КОМПЬЮТЕРНЫЕ СЕТИ



Какие вычисления производят для определения пропускной способности сети?



Подумай

- По каким основным параметрам оцениваются компьютерные сети?
- Какие устройства необходимы для организации компьютерной сети?



Новые знания

Устройства, используемые для организации компьютерной сети

Компьютерная сеть (англ. *computer network*) – совокупность компьютеров, соединенных с помощью каналов связи и средств коммутации в единую систему для обмена сообщениями и доступа пользователей к ресурсам сети. Сети соединяются кабелями, телефонной сетью или бывают беспроводными. Самая большая сеть в мире – Интернет.

Ребята, в 7 классе вы познакомились с понятием «компьютерная сеть», с особенностями организации сетей, с топологией сетей и видами кабелей, используемых для их создания.

- Какие устройства необходимы для организации компьютерной сети?
- В чем заключаются функции этих устройств?



Сетевой коммутатор или **хаб**, свитч (англ. *Switch* – переключатель) – устройство, соединяющее несколько компьютеров в одну сеть.

Сетевая карта – дает возможность для присоединения компьютера к компьютерной сети. Имеются типы карт, обеспечивающие скорость работы 10, 100, 1000 Мбит/сек.



Роутер предназначен для подключения нескольких устройств (компьютер, ноутбук, смартфон, планшет) к локальной сети с доступом в интернет. А также

устройство для подключения к беспроводной сети, с его помощью без кабеля можно подключиться к интернету.

Патч-корд (коммутационный шнур) представляет из себя электрический или оптоволоконный кабель для подключения или соединения между собой электрических устройств. Обе стороны кабеля имеют соединительные разъемы, называемые коннекторами.



Облачная технология. Облачные вычисления

В настоящее время все большую популярность набирают так называемые **облачные технологии** (англ. cloud technology или cloud computing).

Говоря простым языком, облачные вычисления – это предоставление вычислительных служб (серверов, хранилищ, баз данных, сетевого оборудования, программного обеспечения, аналитики и т.д.) через интернет.

Это и есть так называемое «облако». Компании, которые предоставляют такие вычислительные службы, называются поставщиками облачных служб. Обычно они взимают плату за облачные вычислительные службы на основе использования так же, как с вас взимают плату за воду или электричество дома.

Для того чтобы лучше представить, что такое cloud computing, можно привести простой пример. Раньше пользователь для доступа к электронной почте прибегал к определенному ПО (мессенджеру и программы), установленному на его ПК. Теперь же он просто заходит на сайт той компании, услуги электронной почты которой ему нравятся, непосредственно через браузер. Без использования посредников абоненту предоставляется инфраструктура различных веб-сервисов, предоставляющих клиенту удаленные вычислительные мощности.

Существуют следующие модели обслуживания облачных вычислений, предоставляемых потребителю:

- программные средства – приложения провайдера, выполняемые в облачной инфраструктуре;
- средства для развертывания в облачной инфраструктуре создаваемых потребителем приложений, разрабатываемых с использованием поддерживаемых провайдером инструментов и языков программирования;
- средства обработки данных, хранения, сетей и других базовых вычислительных ресурсов, на которых потребитель может разверты-

вать и выполнять произвольное программное обеспечение, включая операционные системы и приложения.

Преимущества облачных сервисов

Доступность. Доступ к информации, хранящейся в облаке, может получить каждый, кто имеет компьютер, планшет или любое мобильное устройство, подключенное к сети Интернет.

Мобильность. У пользователя нет постоянной привязанности к одному рабочему месту. Из любой точки мира менеджеры могут получать отчетность, а руководители – следить за производством.

Экономичность. Одним из важных преимуществ облачных сервисов называют снижение затрат. Пользователю не надо покупать дорогостоящие компьютеры большой мощности и ПО, а также он освобождается от необходимости нанимать специалиста по обслуживанию локальных IT-технологий.

Арендность. Пользователь получает необходимый пакет услуг только в тот момент, когда он ему нужен, и платит только за количество приобретенных функций.

Гибкость. Все необходимые ресурсы предоставляются провайдером автоматически.

Высокая технологичность. Большие вычислительные мощности, которые предоставляются в распоряжение пользователя, которые можно использовать для хранения, анализа и обработки данных.

Надежность. Надежность современных облачных сервисов намного выше, чем у локальных ресурсов.

Пропускная способность сети

Эффективность работы компьютерной сети определяется следующими параметрами:

- **Пропускная способность компьютера** определяет скорость передачи данных в сети: какое количество битов передано за одну секунду. Единица измерения – бит в секунду (бит/с).
- **Надежность** – способность передачи точной информации без потери данных.
- **Ценность сети** – возможность подсоединения к сети компьютеров и других устройств.

Пропускная способность сети зависит от технических возможностей и используемых кабелей.

Тип соединения	Пропускная способность, Мбит/с	Надежность	Возможность расширения
Электрические кабели: • витая пара • коаксиальный кабель	10-500 до 10	низкая высокая	простая сложная
Телефонная линия	1-2	низкая	без проблем
Оптоволоконный кабель	100-200	абсолютная	без проблем

Пропускная способность любого канала связи ограничена. Эта величина зависит от свойств устройства и кабелей, используемых для организации сети. **Объем передаваемой информации Q** определяется по следующей формуле.

$$Q = q * t$$

q – пропускная способность передачи данных
(единица измерения бит/с)
 t – время передачи

Рассмотрим примеры вычисления пропускной способности сети.

Пример №1

Какой объем информации можно передать за 5 секунд с пропускной способностью 20 Кбит в секунду?

Дано:
 $q = 20 \text{ Кбит/с}$
 $t = 5 \text{ с}$

Найти: Q – ?

Решение:

$$Q = q * t$$

$$\begin{aligned} Q &= q * t = 20 * 1\,024 = 20\,480 \text{ бит} / \text{с} * 5 \text{ с} = \\ &= 102\,400 \text{ бит} / 8 = 12\,800 \text{ байт} / 1\,024 = \\ &= 12,5 \text{ Кбайт} \end{aligned}$$

Ответ: 12,5 Кбайт

Пример №2

За какое время можно передать файл объемом 525 Кбайт с пропускной способностью сети 33 600 бит/с?

Дано:

$$q = 33\,600 \text{ бит/с}$$

$$Q = 525 \text{ Кбайт}$$

Найти: t – ?**Решение:**

$$Q = q * t \Rightarrow t = \frac{Q}{q}$$

$$Q = 525 \text{ Кбайт} * 1024 = 537\,600 \text{ байт} * 8 = 4\,300\,800 \text{ бит}$$

$$t = 4\,300\,800 \text{ бит} / 33\,600 \text{ бит/с} = 128 \text{ с}$$

Ответ: 128 с**Пример №3**

Файл объемом 1687,5 Кбайт был передан за 4 минуты. Определите пропускную способность сети.

Дано:

$$t = 4 \text{ минут} = 240 \text{ с}$$

$$Q = 1\,687,5 \text{ Кбайт}$$

Найти: q – ?**Решение:**

$$Q = q * t \Rightarrow q = \frac{Q}{t}$$

$$Q = 1\,687,5 \text{ Кбайт} * 1\,024 = 1\,728\,000 \text{ байт} * 8 = 13\,824\,000 \text{ бит}$$

$$q = 13\,824\,000 \text{ бит} / 240 \text{ с} = 57\,600 \text{ бит/с}$$

Ответ: 57 600 бит/с

ADSL-модем – устройство, предназначенное для подключения к сети Интернет через телефонную линию.

Аббревиатура **ADSL** – расшифровывается как «*Asymmetric Digital Subscriber Line*» — «*Ассиметричная цифровая абонентская линия*».



Задания

1. Данные переданы через ADSL-соединение с пропускной способностью 256 000 бит/с. С помощью этого соединения для передачи файла необходимо 16 секунд времени. Определите объем файла в килобайтах.
2. Скорость передачи данных устройства 56 000 бит/с. За сколько времени будет передан файл объемом 210 000 байт?
3. Данные переданы через ADSL-соединение с пропускной способностью 2 621 440 бит/с. Определите, сколько секунд понадобится для передачи файла объемом 10 Мбайт с помощью этого соединения.
4. Данные переданы через ADSL-соединение с пропускной способностью 5 242 880 бит/с. Определите сколько секунд понадобится для передачи файла объемом 1 Гбайт с помощью этого соединения.
5. Данные переданы через ADSL-соединение с пропускной способностью 128 000 бит/с. С помощью этого соединения для передачи файла необходимо 7 минут времени. Определите объем файла в Кбайтах.
6. За какое время будет передана информация объемом 225 Кбайт модемом с пропускной способностью 14 400 бит/с?
7. Данные переданы через ADSL-соединение с пропускной способностью 16 800 бит/с. Сообщение, записанное с помощью 64-символьного алфавита, состоит из 20 480 символов. Сколько времени необходимо для передачи данной информации по сети?
8. Книга, набранная на компьютере, состоит из 128 страниц, на каждой странице 32 строки, в каждой строке 64 символа. На компьютере используется 256-символьный алфавит. Определите пропускную способность устройства, передающего информационный объем данной книги за 2 минуты 8 секунд.
9. За сколько времени модем с пропускной способностью 28 800 бит/с передаст разноцветный растровый рисунок размером 640x480 пикселей (каждый пиксель цвета кодируется 3 байтами)?



Вопросы

1. Что такое компьютерная сеть?
2. Какие устройства необходимы для настройки сети?
3. Какова функция сетевого коммутатора?
4. Что такое патч-корд?
5. По какой формуле вычисляется объем передаваемой информации?
6. По каким параметрам определяется эффективность работы компьютерной сети?
7. Какова зависимость пропускной способности сети от используемых кабелей?

Словарь – сөздік – dictionary

Таблица

На русском языке	На казахском языке	На английском языке
Измерение информации	Ақпаратты өлшеу	Information measurement
Сообщение	Хабар	Message
Информационное сообщение	Хабардың ақпараттылығы	Information message
Вероятностный метод	Ықтималдық әдісі	Probabilistic method
Бит	Бит	Bit
Неопределенность знаний	Білімнің белгісіздігі	Uncertainty of education
Алфавитный метод	Алфавиттік әдіс	Alphabetic method
Мощность алфавита	Алфавиттің қуаттылығы	Power of the alphabet
Количество информации	Ақпарат саны	Amount of information
Вес информации	Символдың ақпараттық салмағы	Weight of information
Объем информации в тексте	Мәтіннің ақпараттық көлемі	The amount of information in the text
Двоичный алфавит	Екілік алфавит	Binary alphabet
Язык двоичного кодирования	Екілік кодтау тілі	Binary coding language
Байт	Байт	Byte
Внутреннее устройство компьютера	Компьютердің ішкі құрылысы	Internal computer structure
Персональный компьютер	Дербес компьютер	Personal computer
Материнская плата	Аналық тақша	Motherboard
Процессор, микропроцессор	Процессор, микропроцессор	Processor, microprocessor
Оперативная память	Жедел жад	RAM – Random Access Memory
Жесткий диск	Қатты диск	HDD – Hard Disk Drive
Привод оптических дисков	Оптикалық диск жетектері	ODD – Optical Disk Drive
Видеоадаптер, видеокарта	Бейнесызбалық бейімдеуіш	Video adapter
Звуковая карта	Дыбыстық тақша	Sound card
Сетевая плата	Желілік тақша	Network card

Продолжение таблицы

Порты ввода-вывода	Енгізу-шығару порттары	Ports input-output
Блок питания	Қоректендіру блогы	Power supply unit
Шина	Шина	Computer bus
Микросхема	Чип	Chip
Сопроцессор	Сопроцессор	Coprocessor
Слот	Слот	Slot
Слот расширения	Слот кеңейтімдері	Expansion slot
Периферийное устройство	Қосымша құрылғылар	Peripheral
Разряд процессора	Процессордың разрядтылығы	Processor bit
Частота процессора	Процессордың жиілігі	CPU frequency
Кеш-память	Процессордың кеш жады	Cache
Устройство процессора	Процессордың құрылысы	Processor construction
Число ядер	Ядролардың саны	Number of cores
Арифметико-логическое устройство	Арифметикалық -логикалық құрылғы	Arithmetic and logic unit
Устройство управления	Басқару құрылғысы	Control device
Регистр	Регистр	Register
Интерфейс	Интерфейс	Interface
Системная шина	Жүйелік шина	System bus
Компьютерная сеть	Компьютерлік желі	Computer network
Интернет	Ғаламтор	Internet
Коммутационный шнур	Коммутатор бауы	Switch cord
Облачная технология	Бұлттық технология	Cloud technology
Облачные вычисления	Бұлттық есептеулер	Cloud computing
Пропускная способность сети	Желінің өткізу қабілеті	Network bandwidth
Надежность	Сенімділік	Reliability
Пропускная способность передачи канала	Арнаның өткізу жылдамдығы қабілеті	Channelspeed
Время доставки	Тасымалдау уақыты	Delivery time
Ассиметричная цифровая абонентская линия	Ассиметриялық цифрлық абоненттік линия	Asymmetric Digital Subscriber Line
Аспекты использования компьютеров	Компьютерді пайдаланудың аспектілері	Aspects of computer use

Раздел II. ЗДОРОВЬЕ И БЕЗОПАСНОСТЬ

2.1 НЕГАТИВНЫЕ АСПЕКТЫ ИСПОЛЬЗОВАНИЯ КОМПЬЮТЕРА



Каково влияние электронных устройств на организм человека и какие эффективные способы защиты от них применяются?



Подумай

- Какие факторы окружающей среды и быта отрицательно влияют на организм человека?
- Что такое электромагнитное излучение, и каково его влияние на организм человека?
- Что вы можете сказать о чрезмерном использовании компьютера, а также о полученных при этом вредных привычках?



Новые знания

Влияние электромагнитного излучения на организм человека

Электромагнитное поле – особый вид материи, с помощью которого электрические заряженные частицы начинают взаимодействовать между собой. Работающее электрическое оборудование, приводит в движение молекулы, электроны, ионы и атомы. Учитывая, что клетки человеческого организма состоят из молекул, сильное электромагнитное поле вызывает их колебания. Это приводит к негативным изменениям в организме.

В повседневной жизни мы не замечаем, какой вред нашему организму наносят электрические устройства, используемые в быту. Электромагнитные волны присутствуют в каждом доме, офисе, то есть, везде, где живет и работает человек. Электромагнитные волны излучают телевизор, утюг, холодильник, микроволновая печь, пылесос, компьютер, телефон и другая бытовая техника.

Волны наносят очень большой вред организму. Результаты биологических и медицинских исследований показывают, что электромагнитные волны негативно влияют на нервную систему, внутренние органы и физиологическое развитие человека. Под воздействием электромагнитных волн возникает множество заболеваний. Они

оказывают воздействие на мыслительную деятельность человека, нарушают память. Существуют нормы электромагнитного излучения электрических устройств, некоторые из них представлены в таблице 1.

Таблица 1. Нормы электромагнитного излучения

Источники электромагнитных волн	Показатели излучений (мкТл)	Превышение нормы
Компьютер	1-100	5-500
Мобильный телефон	40	200
Холодильник	1	5
Электрический чайник	10	50
Микроволновая печь	8-100	40-500
Фен	15-17	75-85
Трамвай и троллейбус	150	750
Метро	300	1500

Негативное влияние электромагнитных волн, излучаемых компьютером, на организм человека довольно велико.

Длительная и непрерывная работа за компьютером может привести к ухудшению зрения и головным болям.

Электромагнитные волны, исходящие от компьютера, имеют следующие негативные последствия:

- снижение остроты зрения;
- искривление позвоночника;
- психологические нагрузки;
- усталость и слабость;
- снижение иммунитета;
- замедление кровообращения.

Если вы не хотите столкнуться с перечисленными негативными последствиями, следует соблюдать определенные правила работы с компьютером. Помните, очень важно уделять внимание своему здоровью.

Способы уменьшения воздействия электромагнитных излучений от экрана компьютера

В процессе работы необходимо каждый час устраивать перерыв и выполнять разминочные упражнения. Нельзя использовать ноутбук, поставив его на колени. Есть вероятность попасть под воздей-

ствии сильной радиации. При работе на компьютере или ноутбуке рекомендуется одевать защитные очки для глаз.

Выбирая монитор, отдайте предпочтение жидкокристаллическому варианту с надписью «Low Radiation», свидетельствующей о низком уровне излучаемой радиации.

Влажная уборка помещения и наличие ионизатора послужат дополнительной защитой от электромагнитного излучения компьютера.

Не рекомендуется ставить системный блок компьютера на стол, а также сидеть около задней стороны монитора – в этих местах радиация имеет наибольшую силу излучения. При установке монитора, его заднюю часть следует развернуть к окну или стене.

От компьютера и ноутбука исходит тепло, по этой причине в организме пользователя уменьшается количество кальция. По этой причине у человека ухудшается состояние ногтей и волос.

В комнатах, где расположены компьютер, ноутбук, телевизор и другая бытовая техника, по возможности, вместо пластиковых окон, лучше установить деревянные. Дело в том, что пластиковые окна не пропускают наружу радиацию, излучаемую этими приборами, она остается в помещении.

И, наконец, не забывайте выключать компьютер по окончании работы.



10 правил правильного расположения компьютера

1. Экран монитора должен располагаться на расстоянии 50–70 сантиметров от глаз.
2. Верхняя часть экрана должна находиться на уровне глаз.
3. Если в классе или офисе установлено несколько компьютеров, расстояние между ними должно быть не менее 1,2 метра.
4. По возможности, компьютер и монитор задней стороной должны быть направлены к стене.
5. В помещении, где установлен компьютер, не должно быть пыли. Регулярно проводите влажную уборку.

6. Не рекомендуется находиться за компьютером более 1–2 часов. В процессе работы, по возможности, надо делать перерывы продолжительностью 5–10 минут.
7. Во время перерыва выполните несколько упражнений для глаз: перемещайте взгляд вверх-вниз, влево-вправо; сделайте круговые движения глазами; меняйте направление взгляда – посмотрите вдаль (например, в окно), потом сфокусируйте взгляд на предмете на столе.
8. Не допускайте развития «синдрома компьютерного зрения», который развивается у людей, проводящих весь рабочий день за компьютером. Источник освещения не должен быть направлен вам прямо в глаза, а также создавать блики на экране. Работайте на устройствах с матовыми мониторами или используйте специальные антибликовые пленки. Шторы или жалюзи также дают возможность регулировать уровень освещения.
9. Если за компьютером сидит школьник, рекомендуется после 20 минут работы устраивать перерыв не менее 15 минут. Для взрослых пользователей рекомендуются ежечасные 10-минутные перерывы в работе за компьютером.

АСПЕКТ (латин. aspectus – взгляд) – точка зрения, с которой рассматривается какое-либо явление, понятие, перспектива.



Анализ



Выберите 3 электрических устройства, которыми вы пользуетесь ежедневно. Сделайте полный анализ влияния этих устройств на организм человека. Результаты анализа оформите в письменном виде.



Синтез



Составьте 10 анкетных вопросов о негативном влиянии компьютера на организм человека.

**Оценка**

Назовите пять способов защиты организма человека от негативного воздействия компьютера, приведите доказательства. Объясните причину. Напишите итоговый отчет.

**Вопросы**

1. Каково воздействие электромагнитных излучений на организм человека?
2. Какие заболевания может вызвать воздействие электромагнитных волн?
3. Каково воздействие электромагнитного излучения, исходящего от экрана компьютера?
4. Каковы способы борьбы с негативными последствиями для человеческого организма от использования компьютера?

**Домашнее задание**

№	Темы исследования	Вопросы и направления, раскрывающие тему исследования
1	Воздействие компьютера на здоровье человека	<ul style="list-style-type: none"> • Оказывает ли компьютер негативное воздействие на организм человека? • Сколько времени можно проводить перед компьютером? • Как уменьшить негативное воздействие компьютера на организм человека?
2	Полезны или вредны компьютерные игры?	<ul style="list-style-type: none"> • Какое влияние на организм подростка оказывает длительное времяпрепровождение за компьютером? • Как уменьшить вред от компьютерных игр?
3	Влияние ежедневного использования электрических устройств на организм человека	<ul style="list-style-type: none"> • Какое воздействие оказывает электромагнитное излучение на организм человека? • Насколько опасны бытовые электроприборы для человека? • Как уменьшить воздействие электрических устройств на организм человека?

2.2 БЕЗОПАСНОСТЬ В СЕТИ



Какие правила безопасности надо соблюдать при работе в сети Интернет?



Подумай

- Что такое информационная безопасность?
- Какие опасности подстерегают пользователей интернета?
- Что вы знаете об обмане и мошенничестве в Сети?



Новые знания

Информационная безопасность

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

Защита информации – единый комплекс взаимосвязанных мер, направленный на защиту информации от несанкционированного доступа, воровства, потери, удаления, изменения или копирования без разрешения правообладателя.

Существуют три направления информационной безопасности.

Доступность (оптимальность) – возможность получения информации в течение ограниченного периода времени. Доступность информации – свойство технических средств и информационных технологий обработки, обеспечивающее своевременный беспрепятственный доступ к интересующей информации.

Целостность – существование информации в неискаженном виде, гарантия того, что при хранении или передаче информации не было произведено ее несанкционированного изменения или удаления.

Конфиденциальность – недопустимость несанкционированного доступа к информации, гарантия того, что конкретная информация доступна только тем, для кого она предназначена.

Система информационной безопасности имеет два направления, (схема 1).

Схема 1. Система информационной безопасности



Мошенничество в сети Интернет

Сегодня почти не осталось людей, которые не пользовались бы интернетом. У многих есть личные страницы в популярных социальных сетях и даже персональные интернет-сайты.

Получение практически любой информации и неограниченные возможности для общения сделали этот сегмент интернета самым востребованным. Сеть знакомит людей друг с другом, налаживает деловые связи, помогает обсудить широкий круг вопросов, влияет на формирование общественного мнения особыми явлениями.

Однако не только положительные моменты связаны с интернетом. Наряду с благородными целями, сеть таит в себе ряд угроз. Встречаются пользователи с дурными намерениями, или проще говоря мошенники.

В виртуальной реальности, как и в обычной жизни: всегда найдутся желающие получить легкие деньги.



С какими видами мошенничества вы можете столкнуться в сети Интернет, и как от них защититься? Деятельность мошенников направлена в основном на тех, кто не имеет достаточного опыта работы в сети и хочет заработать деньги.

Сегодня существует огромное количество различных способов обмана людей в интернете. Очень важно, изучая информацию на различных сайтах, не поддаваться ни на какие сомнительные и «очень выгодные» предложения быстрого заработка.

Давайте поговорим о наиболее распространенных видах обмана пользователей.

Выигрыш в лотерею или наследство от дальнего родственника. На адрес вашей электронной почты приходит сообщение о выигрыше крупной суммы денег или письмо от юриста, который сообщает, что вы являетесь наследником недавно умершего богатого родственника.

Для получения выигрыша или наследства вам предлагают выслать деньги для оформления документов.

Волшебные кошельки. Это самая распространенная схема обмана в сети. Вам предлагают пополнить электронный счет на определенную сумму, обещая, что вы получите 2-кратную прибыль. Маленькая сумма, вложенная в проект, может вернуться к вам в удвоенном размере. Однако, чем большую сумму вы переведете на указанный мошенниками счет, тем больше вероятность ее полной потери денег.

Предложения работы или заработка. В настоящее время это – самый распространенный вид мошенничества в сети. Например, размещается объявление о вакансии наборщика текста. За работу предлагают хорошие деньги. Но для регистрации кандидата в базе данных или для открытия вашего электронного счета требуется сделать денежный перевод. Отправив указанную сумму, вы потеряете деньги и не получите работу.

Обучение заработку в интернете. Это старый и широко распространенный способ мошенничества. Вам предложат список сайтов, на которых за денежное вознаграждение вас готовы научить зарабатывать в сети. Но, переведя деньги, вы потеряете их, так как никто и не собирался заниматься вашим образованием.

Благотворительные акции. Этот вид мошенничества популярен в социальных сетях. Автор страницы сообщает, что кто-то в семье пострадал от тяжелой болезни или несчастного случая, стал жертвой мошенников или остался без средств к существованию. Вас просят помочь – обычно деньгами.

Чтобы не стать жертвой мошенников каждый пользователь должен соблюдать предельную осторожность при работе в интернете. Не

делиться информацией, фотографиями, видеороликами о себе и своих родных с новыми малознакомыми людьми.

Защищайте или удаляйте переписку с родственниками и близкими друзьями о своих секретах.

Что такое кибербуллинг?

Кибербуллинг (*cyberbullying*)

– подростковый виртуальный террор, получил свое название от английского слова *bull* – бык, с родственными значениями: агрессивно нападать, бередить, задирать, придирааться, провоцировать, донимать, терроризировать, травить. Опасность кибербуллинга заключается в том, что в интернете (в социальных сетях, на видеопорталах) размещается ложная информация о подростке и начинается его травля. Наносится вред здоровью ребенка. В ряде случаев даже требуется помощь психолога. Как мошенничество, так и кибербуллинг в сети не остаются безнаказанными. Те, кто идет на такие преступления, не смогут избежать наказания. И хотя пользователей интернета миллиарды, злоумышленника можно вычислить. Дело в том, что каждому компьютеру при подключении к сети присваивается IP-адрес.

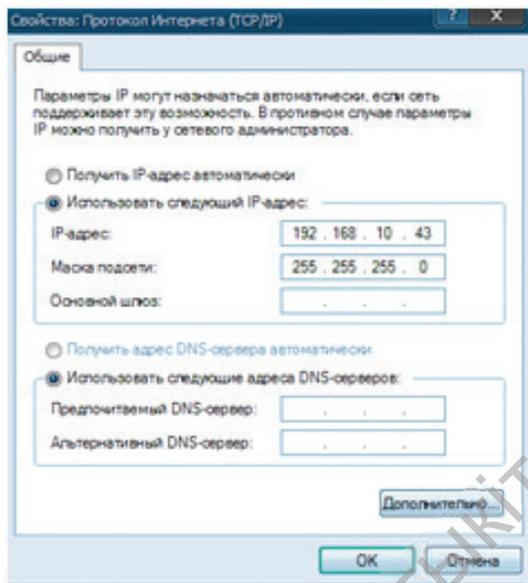


Что такое IP-адрес?

IP-адрес (*Internet Protocol Address*) – сетевой адрес узлов

компьютерной сети. Провайдером (организация, которая предоставляет доступ в интернет) каждому пользователю сети выдается уникальный IP-адрес. IP-адрес состоит из четырех десятичных чисел (от 0 до 255), разделенных точками. В сети компьютер определяется по этим числам. Например, 192.168.10.43 или 128.29.15.124.

По такой системе адресации миллиардам компьютеров можно задать уникальный адрес.





Анализ



Какие виды мошенничества вы встречали в сети Интернет? Проанализируйте информацию о них в парах, в группах. Отсортируйте по видам.



Синтез



Какую опасность подросткам несет кибербуллинг? Как можно противодействовать кибербуллингу? Разработайте правила борьбы с кибербуллингом. По каждому пункту правил приведите примеры.



Оценка



Насколько важна безопасность при работе в сети Интернет?



Вопросы

1. Что такое информационная безопасность?
2. Какова значимость информационной безопасности?
3. Назовите направления системы информационной безопасности?
4. Что такое интернет-мошенничество?
5. Какую опасность несет кибербуллинг?
6. Можно ли поймать мошенников в сети Интернет?
7. Что такое IP-адрес?



Задания

№	Темы исследования	Вопросы и направления, раскрывающие тему исследования
1	Меры безопасности в сети Интернет.	<ul style="list-style-type: none"> • Основные правила, которые должен знать пользователь сети Интернет. • Пути защиты информации в интернете.
2	Безопасное использование социальных сетей.	<ul style="list-style-type: none"> • Социальные сети и их особенности. • Риски, которым пользователь подвергается в социальных сетях.
3	Интернет – это хорошо или плохо?	<ul style="list-style-type: none"> • История возникновения сети Интернет. • Полезные и вредные аспекты использования интернета. • Советы по безопасному использованию сети Интернет.

Словарь – сөздік – dictionary

На русском языке	На казахском языке	Ағылшын тілінде
Электромагнитное излучение	Электромагниттік сәуле	Electromagnetic radiation
Сетевая безопасность	Желідегі қауіпсіздік	Online security
Информационная безопасность	Ақпараттық қауіпсіздік	Information security
Выигрыш в лотерею или наследство от дальнего родственника	Лотерея жеңімпазы немесе алыс туысқанының мұрагері	Winning the lottery or inheritance from a distant relative
Волшебные кошельки	Сиқырлы әмияндар	Magic wallets
Предложения работы или заработка	Жұмыс немесе ақша табуды ұсыну	Work or earnings offer
Заработок в интернете	Интернетте ақша табу	Making money online
Кибербуллинг	Кибербуллинг	Cyberbullying
IP-адрес	IP-адрес	Internet Protocol Address
Провайдер	Провайдер	Provider
Доступность	Қол жетімділік (оңтайлық)	Availability
Целостность	Тұтастық	Integrity
Конфиденциальность	Жасырындылық	Confidentiality
Статистика	Статистика	Statistics
Ссылка	Сілтеме	Reference
Относительная ссылка	Салыстырмалы сілтеме	Relative reference
Абсолютная ссылка	Абсолют сілтеме	Absolute reference
Встроенные функции	Кірістірілген функциялар	Built-in functions
Функции	Функциялар	Functions
Финансовые	Финанстық	Financial
Логические	Логикалық	Logic
Текстовые	Мәтіндік	Text
Дата и время	Мерзім мен уақыт	Date and time
Ссылки и массивы	Сілтемелер және жиымдар	References and arrays
Математическое выражение	Математикалық өрнек	Mathematically expression

3.1 СТАТИСТИЧЕСКИЕ ДАННЫЕ. ИСПОЛЬЗОВАНИЕ АБСОЛЮТНЫХ И ОТНОСИТЕЛЬНЫХ ССЫЛОК



Как применять абсолютные и относительные ссылки при обработке статистических данных?



Подумай

- Что такое статистические данные?
- Какие специалисты работают со статистическими данными?



Новые знания

Что такое статистические данные?

Из истории древнего мира нам известно, что уже тогда для определения численности населения государства, количества людей, годных к воинской службе, численности скотоводческих хозяйств, площади земельных наделов (участков) выполнялись постоянные расчеты. Эти данные использовались для начисления налогов, для призыва на воинскую службу и т.д. В результате изменения и развития общества возникли новые методы проведения вычислений. В середине XVII века в Англии возникло научное направление «Политическая арифметика», направленное на ведение учета различных изменений и анализа развития общества. Одновременно с этим в Германии появилась школа «Обязательная статистика». Благодаря таким шагам возникло новое направление науки – **статистика**. Слово «статистика» происходит от латинского status – «состояние дел». В настоящее время статистика является самостоятельной общественной наукой.

В процессе исследования развития общества получен значительный объем информации. Эту информацию необходимо было собрать, обработать, упорядочить и оформить. На примере выполнения этих работ рассмотрим применение табличного процессора MS Excel.

Использование абсолютных и относительных ссылок

Ссылка – адрес ячейки, используемый в записи формулы.

В табличном процессоре самое важное действие – копирование формулы из одной ячейки в другую. В 7 классе упоминалось, что ссылка (адрес ячейки) может быть *относительной, абсолютной и смешанной*. Теперь мы подробно остановимся на способах использования этих ссылок. В прикладной программе MS Excel по умолчанию используется относительная ссылка, в которой при копировании формулы из одной ячейки в другую адреса в ссылках изменяются автоматически.

При абсолютной адресации адрес ячейки не изменяется во время копирования формулы, так как ячейка рассматривается как постоянная.

Для обозначения абсолютной адресации используется символ \$.

Относительная ссылка. При копировании формулы из одной ячейки в другую результат напрямую зависит от адреса ссылки. В простейшем случае используются относительные ссылки. Это означает, что при копировании формулы из одной ячейки в другую, адрес ссылки автоматически изменится. Например: если в ячейке B2 формула будет содержать ссылку на ячейку B3, расположенную на одну строку ниже. Тогда при копировании данной формулы достаточно использовать относительную ссылку. Например, A9, D25, F5 являются относительными адресами (ссылками).

Абсолютная ссылка. При абсолютной адресации в копируемой формуле адреса используемых ячеек не изменятся, они останутся постоянными. При преобразовании формулы из относительной адресации в абсолютную, и наоборот, необходимо выделить адрес ссылки и нажать клавишу F4. При абсолютной адресации знак \$ ставится как перед именем столбца, так и перед номером строки. Например: \$A\$16. Адреса \$A9, A\$7 содержат знак \$ или перед именем столбца, или перед номером строки. Такие адреса называются **смешанными**. Теперь рассмотрим примеры использования ссылок для закрепления полученных знаний.

Ссылка на ячейку	Ссылка на следующую ячейку	
	Заполнение вправо	Заполнение вниз
A1 (относительная)	B1	A2
\$A1 (по столбцу абсолютная) смешанный адрес	\$A1	\$A2
A\$1 (по строке абсолютная) смешанный адрес	B\$1	A\$1
\$A\$1 (абсолютная)	\$A\$1	\$A\$1

Рассмотрим фрагмент электронной таблицы, заполненной числами и формулами. Обратим внимание как изменится формула при использовании ссылок.

	Абсолютная ссылка		Относительная ссылка	
	A	B	C	D
1	5	=A1*\$B\$2	=A1*B1	
2		8		=B2*C2
3				

В ячейке B1 размещена формула =A1*\$B\$2. После набора формулы, нажав клавишу Enter, получим результат 40. Здесь число, расположенное в ячейке A1, будет умножено на число из ячейки B2, ответ разместится в ячейке B1. Теперь скопируем эту формулу в ячейку C2. Какие изменения произошли? Для получения ответа на данный вопрос проанализируем ссылки, используемые в формуле.

На ячейку A1 ссылка относительная, то есть при изменении местоположения формулы изменится и данная ссылка.

На ячейку B2 ссылка абсолютная, поэтому при копировании формулы всегда будут использоваться данные из этой ячейки.

При копировании в ячейку C2 формула изменит свой вид на =B2*\$B\$2. Обратите внимание, что ссылка на ячейку A1 изменилась на B2: при переходе на следующую строку она изменилась на A2, а сместившись на 1 столбец вправо, приняла вид B2. При нажатии на клавишу Enter, получим ответ 64.

	A	B	C	D
1	5	=A1*\$B\$2		
2		8	=B2*\$B\$2	
3				
4				
5				

В чем причина? При перемещении формулы вправо на одну ячейку адрес A1 изменится на B1. При нажатии на клавишу Enter получим результат. При копировании этой формулы в ячейку D3 формула примет вид =C3*\$B\$2, и результат будет равен 0, потому что в ячейке C3 пусто и ее значение равно нулю.

Какой результат мы получим, если формулу из ячейки B1 скопируем в ячейку A4?

	A	B	C	D
1	5	=A1*\$B\$2		
2		8	= B2*\$B\$2	
3				=C3*\$B\$2
4	=A1*\$B2			
5				

Теперь рассмотрим примеры, встречающиеся на практике.



Практическая работа

Пример. В таблице представлены данные о заработной плате сотрудников. Каким будет размер зарплаты при ее увеличении на 10%?

При повышении заработной платы на 10% , надо зарплату умножить на коэффициент 1,1 ($100\% + 10\% = 110\% / 100\% = 1,1$). На рисунке 1 формула расчета зарплаты задана в ячейке D4. Если эту формулу скопировать в ячейки D5:D10, расчет будет неверным. Потому что при копировании вместо адреса D2, будет указан адрес D3 и т.д., что приведет к ошибке, как показано на рисунке 2.

СУММ			
A	B	C	D
Зарботная плата сотрудников			
Процент повышения зарплаты 10%			1,1
№	Сотрудники	Зарботная плата	Повышенная зарплата
1	Сотрудник №1	100000	=C4*D2
2	Сотрудник №2	120000	
3	Сотрудник №3	110000	
4	Сотрудник №4	150000	
5	Сотрудник №5	130000	
6	Сотрудник №6	80000	
7	Сотрудник №7	105000	

Рис. 1

A	B	C	D
Зарботная плата сотрудников			
Процент повышения зарплаты 10%			1,1
№	Сотрудники	Зарботная плата	Повышенная зарплата
1	Сотрудник №1	100000	110000
2	Сотрудник №2	120000	#ЗНАЧИ
3	Сотрудник №3	110000	12100000000
4	Сотрудник №4	150000	#ЗНАЧИ
5	Сотрудник №5	130000	1,573E+15
6	Сотрудник №6	80000	#ЗНАЧИ
7	Сотрудник №7	105000	1,65165E+20

Рис. 2

Для правильного автоматического вычисления заработной платы сотрудников в формуле в ссылке D2 зададим абсолютную адресацию (рис. 3). На рисунке 4 показано, как автоматически вычислить зарплату при ее повышении на 10%.

СУММ			
A	B	C	D
1	Зарботная плата сотрудников		
2	Процент повышения зарплаты 10%		
3	№	Сотрудники	Зарботная плата
4	1	Сотрудник №1	100000
5	2	Сотрудник №2	120000
6	3	Сотрудник №3	110000
7	4	Сотрудник №4	150000
8	5	Сотрудник №5	130000
9	6	Сотрудник №6	80000
10	7	Сотрудник №7	105000

Рис. 3

СУММ			
A	B	C	D
1	Зарботная плата сотрудников		
2	Процент повышения зарплаты 10%		
3	№	Сотрудники	Зарботная плата
4	1	Сотрудник №1	100000
5	2	Сотрудник №2	120000
6	3	Сотрудник №3	110000
7	4	Сотрудник №4	150000
8	5	Сотрудник №5	130000
9	6	Сотрудник №6	80000
10	7	Сотрудник №7	105000

Рис. 4



Задания

Задание №1. По заданному образцу составьте таблицу, выполните вычисления.

1. Составьте таблицу.

	A	B	C	D	E
1	№	Наименование товара	Кол-во	Цена	Общая цена
2	1	Тетрадь	50	15	=C2*D2
3	2	Ручка	80	50	
4	3	Карандаш	25	6	
5	4	Ластик	10	0,45	
6	5	Линейка	20	8,15	

2. В ячейку E2 запишите формулу расчета общей суммы за тетради: для этого умножим количество тетрадей на цену за 1 штуку и получим формулу =C2*D2.

3. Протяните данную формулу для копирования по столбцу E. Отформатируйте таблицу и сохраните.

Задание №2. При расчете основной заработной платы служащего, нужно коэффициент его должностного оклада умножить на величину базовой зарплаты, закрепленной в Казахстане. Закончите заполнение таблицы по приведенным данным на рисунке 5.

	A	B	C	D	E
2	Какую зарплату получают служащие?				
4	Размер базовой зарплаты =				17697 ₸
6	№	Список служащих	Коэффициент	Зарплата	
7	1	№1 служащий	4,8	84945,6	
8	2	№2 служащий	3,9		
9	3	№3 служащий			
10	4	№4 служащий			
11	5	№5 служащий			

Таблица. Таблица Пифагора

	1	2	3	4	5	.	.
1	1	2	3	.	.		
2	2	4	6	.	.		
3	3	6	9	.	.		
4		
5		
.							
.							

Рис. 5

Задание №3: Создайте таблицу умножения Пифагора для чисел от 1 до 10 (Таблица).



Анализ



Проанализируйте и сравните пути выполнения заданий 1 и 2. В процессе анализа найдите ответы на следующие вопросы:

- 1) В чем различие этих двух заданий?
- 2) Какие ссылки были использованы в процессе выполнения заданий?



Вопросы

1. Что такое статистические данные?
2. Для чего их собирают?
3. Что такое ссылка?
4. Что такое относительная ссылка?
5. В чем отличие абсолютной ссылки от относительной?
6. Что такое смешанная ссылка?
7. В чем польза использования ссылок при решении практических задач?

3.2

ИСПОЛЬЗОВАНИЕ АБСОЛЮТНЫХ И ОТНОСИТЕЛЬНЫХ ССЫЛОК
В РЕШЕНИИ ПРАКТИЧЕСКИХ ЗАДАЧ

Как применять абсолютные или относительные ссылки при обработке данных разных типов?



Подумай

- Какие преимущества дает использование абсолютных и относительных ссылок при решении задач в MS Excel?



Новые знания

Абсолютные и относительные ссылки можно использовать не только внутри одного файла, но и при одновременной работе с несколькими файлами. Пусть имеется два файла, в одном из которых делается ссылка на другой файл. Выясним, отразятся ли изменения, внесенные в первый файл, на втором файле?

Рассмотрим пример. Пусть имеется прайс-лист с указанием стоимости товара в долларах США. С изменением курса доллара соответственно цена товара в тенге тоже будет меняться. В 1-м файле указаны цены на товар в долларах и соответствующая цена в тенге (рис. 1). Здесь курс доллара по отношению к тенге записан в ячейке C2, и будет использоваться абсолютная ссылка на эту ячейку. При изменении курса доллара цена товара в тенге автоматически изменится. Во 2-м файле дано количество товаров, купленных за один квартал (рис. 2). Для изменения цены товара в 1 файле в соответствии с курсом доллара выполним следующую последовательность действий.

1. Чтобы во 2-м файле цену товара №1 записать в ячейку C4, поставим в ней знак =, в 1-м файле выделим ячейку D4 с указанной ценой товара №1, при нажатии клавиши Enter между файлом 2 и файлом 1 установится связь. Ссылка в ячейке C2 из-

	A	B	C	D
1		Перевод долларов в тенге		
2			1\$= 320	〒
3				
4		Наименование товара	Цена, \$	Цена, 〒
5	№			
6	1	№1 товар	20	6400
7	2	№2 товар	40	12800
8	3	№3 товар	100	32000
9	4	№4 товар	250	80000
10	5	№5 товар	300	96000
11	6	№6 товар	400	128000

Рис. 1

меняется: ‘=[1.xlsx] Лист1!\$D\$4’. Теперь цена товара в 1-м файле будет меняться в соответствии с курсом доллара, во 2-м файле тоже будет меняться стоимость товара.

	A	B	C	D	E	F	G	H	I	J
1	Общее количество товаров, купленных за квартал									
2		Наименование товара	Цена, ₸	Количество купленных товаров			Сумма проданных товаров			Общая сумма
3	№			январь	февраль	март	январь	февраль	март	
4	1	№1 товар		20	30	80	0	0	0	0
5	2	№2 товар		30	20	70	0	0	0	0
6	3	№3 товар		40	50	90	0	0	0	0
7	4	№4 товар		60	40	30	0	0	0	0
8	5	№5 товар		80	50	20	0	0	0	0
9	6	№6 товар		40	50	70	0	0	0	0

Рис. 2

2. При копировании формулы из ячейки C2 2-го файла в другие ячейки скопируется именно стоимость товара №1 из указанной ячейки. Чтобы этого не случилось, изменим ссылку (рис. 3).

	A	B	C	D	E	F	G	H	I	J
1	Общее количество товаров, купленных за квартал									
2		Наименование	Цена, ₸	Количество купленных			Сумма проданных товаров			Общая сумма
3	№			январь	февраль	март	январь	февраль	март	
4	1	№1 товар	6400	20	30	80	128000	192000	512000	832000
5	2	№2 товар	6400	30	20	70	192000	128000	448000	768000
6	3	№3 товар	6400	40	50	90	256000	320000	576000	1152000
7	4	№4 товар	6400	60	40	30	384000	256000	192000	832000
8	5	№5 товар	6400	80	50	20	512000	320000	128000	960000
9	6	№6 товар	6400	40	50	70	256000	320000	448000	1024000

Рис. 3

3. Чтобы у каждого товара записывалась его стоимость, в формуле из ячейки C2 ‘=[1.xlsx]Лист1!\$D\$4’ перед номером строки уберем знак \$, получим ‘=[1.xlsx]Лист1!\$D4’ (рис. 4).

	A	B	C	D	E	F	G	H	I	J
1	Общее количество товаров, купленных за квартал									
2		Наименование	Цена, ₸	Количество купленных			Сумма проданных товаров			Общая сумма
3	№			январь	февраль	март	январь	февраль	март	
4	1	№1 товар	6400	20	30	80	128000	192000	512000	832000
5	2	№2 товар	12800	30	20	70	384000	256000	896000	1536000
6	3	№3 товар	32000	40	50	90	1280000	1600000	2880000	5760000
7	4	№4 товар	80000	60	40	30	4800000	3200000	2400000	10400000
8	5	№5 товар	96000	80	50	20	7680000	4800000	1920000	14400000
9	6	№6 товар	128000	40	50	70	5120000	6400000	8960000	20480000

Рис. 4

Вот мы с вами и познакомились с принципами использования абсолютных и относительных ссылок в электронной таблице. Теперь для решения следующих примеров, попробуйте применить полученные знания на практике.



Практическая работа

Движение легкового автомобиля.

Составьте таблицу движения легкового автомобиля (рис. 5), равномерно движущегося со скоростью 72 км/ч, за каждые 2 секунды.

Задание:

- 1) Скорость автомобиля переведите в м/с.
- 2) Определите длину пути за 20 секунд.
- 3) За какое время автомобиль со скоростью 72 км/ч проедет 1 км?

Составьте электронную таблицу по следующему образцу. Для вычисления результата в первый столбец запишите время движения, во второй – пройденный путь за указанное время.

	А	В
1	Скорость автомобиля	
2	Скорость (км/ч)	72
3	Скорость (м/с)	20
4	Промежуток времени (с)	2

В 7 классе на уроках физики вы научились переводить км/ч в м/с. Запишите в ячейку В3 формулу $=B2*1000/3600$. Используя абсолютную ссылку, в ячейку В6 запишите формулу $=B3*A6$. В таблице задан общий путь легковой машины за 20 секунд и изменение длины пути за каждые 2 секунды.

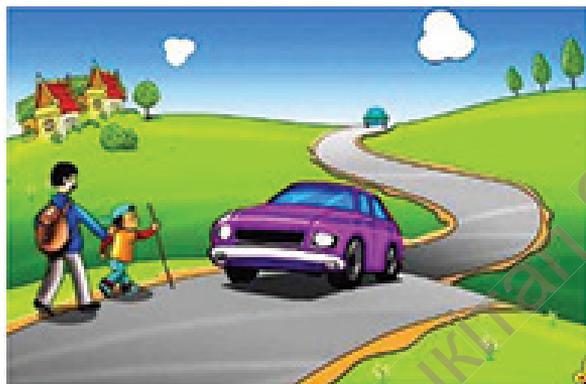


Рис. 5

	А	В
1	Скорость автомобиля	
2	скорость (км/час)	72
3	скорость (м/с)	20
4	промежуток времени(с)	2
5	Время(с)	Путь(м)
6	0	0
7	2	40
8	4	80
9	6	120
10	8	160
11	10	200
12	12	240
13	14	280
14	16	320
15	18	360
16	20	400

Рис. 6

Результаты задания:

- 1) Скорость легкового автомобиля равна 20 м/с.
- 2) Из рисунка 5 видно, что через 20 секунд путь равен 400 м.
- 3) 1 км пути со скоростью 72 км/ч автомобиль проезжает за 50 секунд. Чтобы получить ответ, необходимо записанные в столбцах **время** и **путь** методом автозаполнения протянуть до получения значения пути 1000 м, тогда мы и узнаем время, потраченное на его прохождение.

**Вопросы**

1. Какие еще примеры на использование абсолютных и относительных ссылок вы можете привести?
2. Как можно связать два файла с помощью абсолютных и относительных ссылок?
3. Из каких шагов состоит алгоритм, связывающий два файла с помощью ссылок?

**Задание**

Задание №1. Ниже представлен фрагмент электронной таблицы в режиме формул. Если скопировать содержимое ячейки C1 в ячейку C3, какие значения мы получим в ячейках C1 и C3?

	A	B	C
1	10	15	= B1 + \$A\$2
2	20	20	
3	30	25	
4	40	30	

Задание №2. Скопируйте формулу из ячейки C2, вставьте в ячейку C3. Какую формулу мы получим в результате?

	A	B	C
1	30		
2	12	4	=\$A\$1*A2 + B2
3	23	5	
4	43	2	

Задание №3: У Рамазана большой дом, в котором 5 комнат. Измеряя длину и ширину каждой комнаты, он получил таблицу и набрал ее в табличном процессоре. Помогите Рамазану вычислить площадь и периметр каждой комнаты.

	A	B	C	D	E	F
1	№	Название комнаты	длина (м)	ширина (м)	площадь (м ²)	Периметр (м)
2	1	Гостиная	5	6		
3	2	Детская	4	3		
4	3	Столовая	5	3		
5	4	Спальня	4	4		
6	5	Игровая	3	3		

Задание №4. Самат учится в 8 классе. Он на летних каникулах помогал отцу откармливать скот. Отец был доволен трудолюбием Самата. Скот ежедневно прибавлял в весе 1% от предыдущего веса. Заданы первоначальный вес и 10 дней, в течение которых Самат откармливал скот.

Задание: Составьте формулу для вычисления и вычислите (рис. 7):

1. Какой вес наберет откормленный Саматом скот через 10 дней?
2. Какое количество килограмм набирает скот, откармливаемый Саматом ежедневно?

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	Процент ежедневного прибавления веса от первоначального 1% =										0,01	
3												
4	день		1	2	3	4	5	6	7	8	9	10
	первоначальный	280	282,6									
5	вес (кг)											

Рис. 7

Задание №5. В 1-м файле задана стоимость товара с учетом изменений за каждый месяц одного квартала. Во 2-м файле дано количество проданных товаров за каждый месяц. Во 2-м файле вычислите сумму, вырученную от продажи товаров за каждый месяц.

Данные 1-го файла

Данные 2-го файла

	A	B
1	Название месяцев	Стоимость товара (т)
2	Апрель	100000
3	Май	105000
4	Июнь	103000

	A	B	C
1	Название месяцев	Количество проданных товаров	Стоимость проданных товаров
2	Апрель	80	=
3	Май	57	
4	Июнь	68	

3.3 ВСТРОЕННЫЕ ФУНКЦИИ



Как применять встроенные функции при решении задач в электронных таблицах?



Подумай

- Что такое функция?
- Какую роль она играет в жизни человека?
- Как вы думаете, что такое встроенные функции в программе?



Новые знания

Функции. Определение понятия «функция»

В приложении Microsoft Excel в качестве функции рассматриваются вычислительные операции, проводимые по определенному алгоритму или формулам. У каждой функции имеется соответствующее имя. В приложении Excel имена функций можно набрать на клавиатуре или выбрать из библиотеки, выполнив команду **Формулы** -> **Библиотека функций**, также на панели инструментов за это отвечает кнопка f_x (**Вставить функцию**). «Мастер функций» предлагает вам готовые шаблоны, пользователю достаточно записать аргументы.

В приложении MS Excel имеется более 200 встроенных функций, предназначенных для решения простейших и сложных задач. А также пользователи приложения имеют возможность создавать необходимые формулы с помощью встроенного приложения – Visual Basic for Applications (VBA).

Для ускорения поиска функции поделены на основные категории: финансовые, логические, текстовые, дата и время, ссылки и массивы, математические, статистические, инженерные, аналитические, проверка свойств и значений (рис. 1).

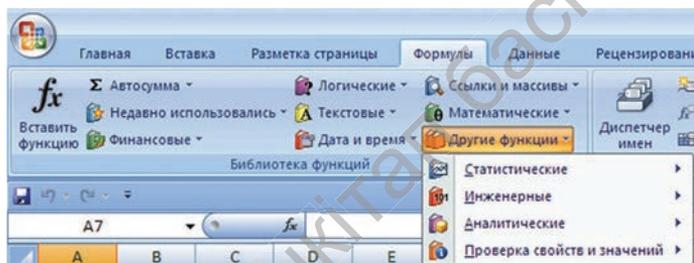


Рис. 1. Виды функций

В качестве аргументов функции могут использоваться числа, числовые или буквенно-числовые выражения, адрес ячейки, адрес диапазона.

Например:

= СУММ(A5:A9) – здесь СУММ – функция, вычисляющая сумму значений ячеек A5, A6, A7, A8, A9;

= СРЗНАЧ(G4:G6) – здесь СРЗНАЧ – функция, вычисляющая среднее арифметическое содержимого ячеек G4, G5, G6.

При составлении сложных функций (одна функция в другой) используются круглые скобки: = ОКРУГЛ(СРЗНАЧ(H4:H8); 2).

Если функция, записываемая с помощью мастера функций, имеет аргументы, то открывается диалоговое окно, как показано на рисунке 2. Это окно предназначено для ввода аргументов.

Способы ввода аргументов функции:

- используемые адреса (ячейки или диапазон) можно набрать на клавиатуре;
- используемые ячейки и диапазоны можно указать, выделив их на рабочем листе.

Теперь рассмотрим функции, которые используются при выполнении вычислений чаще остальных.

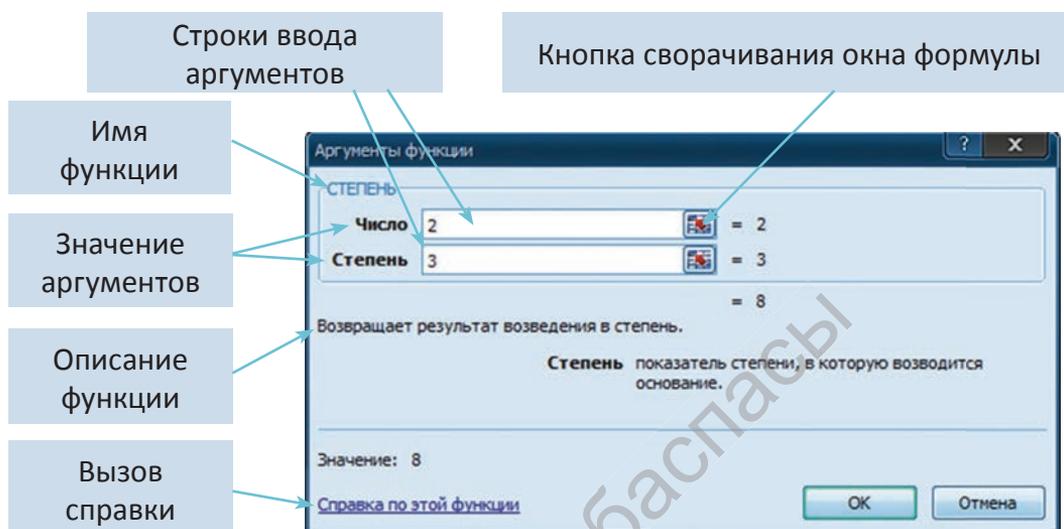


Рис. 2. Окно мастера функций

Математические функции

ABS (число) – модуль числа;

SIN (число), COS (число), TAN (число) – вычисляет синус, косинус, тангенс заданного угла;

ГРАДУСЫ (угол) – переводит радианы в градусы;

ЗНАК (число) – определяет знак числа; возвращает значение 1 – если число положительное, 0 (ноль) – если число равно 0, и -1 – если число отрицательное;

КОРЕНЬ (число) – функция, вычисляющая квадратный корень заданного числа;

СУММ (число 1; число 2; ...) – вычисляет сумму всех чисел, указанных в качестве аргументов;

СТЕПЕНЬ (число; степень) – заданное число возводится в указанную степень.

Статистические функции

МАКС (аргумент 1; аргумент 2;...) – определяет максимум среди заданных аргументов;

МИН (аргумент 1; аргумент 2;...) – определяет минимум среди заданных аргументов;

СРЗНАЧ (аргумент 1; аргумент 2;...) – определяет среднее значение заданных аргументов;

СЧЕТЕСЛИ (диапазон; условие) – подсчитывает количество пустых ячеек в диапазоне, удовлетворяющих заданному условию.

Логические функции

Логические функции служат для проверки результата какого-либо условия. При составлении условий используются операторы сравнения: =, >, <, <> (не равно), >= (больше или равно), <= (меньше или равно).

ЕСЛИ (логическое_выражение; значение 1; значение 2) – значение 1 будет результатом, если логическое выражение истинно, в противном случае результатом будет значение 2. Например: в ячейке A1 записано число 30000, а в ячейке B1 – формула =ЕСЛИ(A1<20000; 12; 15). Так как условие не выполнено, то результатом будет 15.

Логические функции И, ИЛИ предназначены для составления сложных условий;

И (логическое_выражение 1; логическое_выражение 2; ...) – принимает значение ИСТИНА, если все аргументы принимают значение ИСТИНА, в противном случае принимает значение ЛОЖЬ;

ИЛИ (логическое_выражение 1; логическое_выражение 2; ...) – принимает значение ИСТИНА, если хотя бы один аргумент принимает значение ИСТИНА, в противном случае принимает значение ЛОЖЬ.

Например: =ЕСЛИ(И(A1>=20000; A1<40000); 15; 18) – если A1 принимает значение из промежутка от 20000 до 40000, тогда значение функции равно 15, в противном случае – 18.

С помощью встроенной функции ЕСЛИ можно составлять сложные условия. Например:

=ЕСЛИ(A1<20000; 12; ЕСЛИ(A1<40000; 15; 18)) – если в ячейке A1 находится число меньше 20000, то функция примет значение 12, если ячейка A1 содержит число меньше 40000, тогда значение функции равно 15, в противном случае функция принимает значение 18. Если приложение MS Excel не может **правильно определить формулу**, тогда в ячейке возможно наличие одной из приведенных ошибок:

Ошибка	Пояснение
# Дел/0!	В формуле содержится деление на ноль
# Знач!	Недопустимый тип аргумента
# Имя?	Ошибка в написании имени
# Н/Д	Неопределенные данные
# Пусто!	Пересечение двух областей, которые не имеют общих ячеек.
#Ссылка!	Задана некорректная ссылка
# Число!	Используется аргумент, не входящий в область определения функции



Практическая работа

Исследование чисел

Задано пятизначное число, например: 54289. При исследовании выполните следующие задания:

- 1) найдите квадрат числа;
- 2) найдите квадратный корень числа;
- 3) найдите сумму цифр в составе числа;
- 4) найдите среднее арифметическое цифр, составляющих число;
- 5) найдите наименьшее значение среди цифр, составляющих число.

Практическое задание выполним в приложении MS Excel в соответствии с рисунком 3.

МНН			
	A	B	D
1			
2	№	Результаты исследования числа 54289	
3	1	Вычисление квадрата числа	2947295521
4	2	Извлечение квадратного корня из числа	233
5	4	Вычисление суммы цифр числа	28
6	5	Нахождение наименьшего из цифр, составляющих число	2
7	3	Вычисление среднее арифметическое цифр числа	=СРЗНАЧ(5;4;2;8;9)

Рис. 3. Таблица выполнения задания

При выполнении практической работы будем использовать мастер функций, а также функции, приведенные ниже. В ячейки можно записать формулы, составленные с их помощью:

Ячейка C3: =СТЕПЕНЬ (54289; 2)

Ячейка C4: =КОРЕНЬ (54289) (рис. 4)

Ячейка C5: =СУММ (5; 4; 2; 8; 9)

Ячейка C6: =МИН (5; 4; 2; 8; 9)

Ячейка C7: =СРЗНАЧ (5; 4; 2; 8; 9)

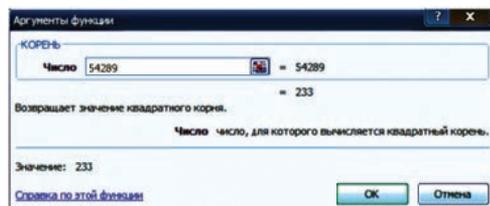


Рис. 4. Вычисление корня



Вопросы

1. Дайте математическое определение понятию «функция».
2. Назовите основные функции приложения MS Excel?
3. Какие категории функций имеются в приложении?
4. Какие существуют способы ввода аргументов функции?
5. Какую структуру имеет рабочее окно MS Excel?
6. Назовите математические функции программы MS Excel?
7. Какую роль выполняют логические функции?



Домашнее задание

Успеваемость

Ренат учится в 8 классе. В его школе 8 параллельных классов, в которых по результатам I четверти получены значения качества знаний. Ренат, изучив полученные данные, решил занести их в таблицу. Помогите ему выполнить задание.

№	Содержание задания	Ячейка для записи результатов		A	B	C	D
1	Определение класса с самым высоким уровнем качества знаний	C1	1	8 «А»	74%		
2	Определение класса с самым низким уровнем качества знаний	C2	2	8 «Б»	72%		
3	Определение среднего качества знаний среди классов	B9	3	8 «В»	51%		
4	На сколько процентов нужно поднять уровень знаний класса с самым низким качеством знаний чтобы достичь среднего уровня	C3	4	8 «Г»	73%		
5	Разница между высоким и низким уровнем качества знаний	C4	5	8 «Д»	81%		
			6	8 «Е»	37%		
			7	8 «Ж»	64%		
			8				

Раздел III. ОБРАБОТКА ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ

3.4

ИСПОЛЬЗОВАНИЕ ВСТРОЕННЫХ ФУНКЦИЙ ПРИ РЕШЕНИИ ПРАКТИЧЕСКИХ ЗАДАЧ



Какие возможности открывает использование встроенных функций при решении практических задач?



Подумай

- Как применять встроенные функции для обработки вычислительных задач, встречающихся в повседневной жизни?
- Как вы думаете, в каких изученных вами разделах алгебры можно эффективно применять встроенные функции при решении задач?

Следующие задания ориентированы на практическое использование встроенных функций табличного процессора. Вы научитесь строить графики функций для задач из повседневной жизни, например, сможете считать расходы на электроэнергию. В задачах также описана последовательность их решения.



Практическая работа

Практическая работа №1

	A	B	C	D
1	Итоговые оценки ученика			
2				
3		Предмет	Оценка	Анализ
4	1	Казахский язык	5	1
5	2	Казахская литература	5	1
6	3	Русский язык	4	0
7	4	Русская литература	5	1
8	5	Английский язык	4	0
9	6	История	5	1
10	7	Алгебра	4	0
11	8	Геометрия	5	1
12	9	Информатика	5	1
13	10	Физика	5	1
14	11	Химия	3	0
15	12	Биология	5	1
16	13	География	4	0
17	14	Физическая культура	5	1
18		Средняя оценка	4,6	9
19		Освобожден ли от экзамена?		нет

Рис. 1

Экзамен

Известно, что в конце учебного года отличники 5-8 и 10 классов освобождаются от экзаменов. Рамазан учится в 8 классе. Его оценки представлены в виде таблицы (рис. 1).

Задание: Определите будет ли Рамазан в конце учебного года в числе отличников, освобожденных от экзаменов?

Идея выполнения задания: В столбец «Анализ» ставим 1 в том случае, если оценка «5», в ином случае ставим 0. Если

у ученика по всем предметам оценка «5», тогда сумма чисел столбца «Анализ» будет равняться количеству предметов. В нашем примере это – 14.

Последовательность выполнения задания:

1. В столбце «Анализ» в ячейку D4 запишем формулу с логической функцией: =ЕСЛИ (C4=5; 1; 0)
2. Формулу из ячейки D4 скопируем до ячейки D17 включительно.
3. В ячейку D18 запишем сумму ячеек с D4 до D17: =СУММ (D4:D17)
4. В ячейку D19 запишем логическую функцию: =ЕСЛИ (D18=14; «Да»; «Нет»)
5. В ячейку C18 запишем формулу вычисления среднего арифметического числа с ячейки C4 до ячейки C17: =СРЗНАЧ (C4:C17)

Практическая работа №2

График функции

Задание: Постройте график функции $y=x^2-4$. Для этого в приложении MS Excel составим таблицу, в которой каждому аргументу x соответствует ордината y (табл. 1).

Таблица 1. Таблица зависимости Y от X

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	x	-3,0	-2,5	-2,0	-1,5	-1,0	-0,5	0,0	0,5	1,0	1,5	2,0	2,5	3,0
2	y	5,0	2,3	0,0	-1,8	-3,0	-3,8	-4,0	-3,8	-3,0	-1,8	0,0	2,3	5,0

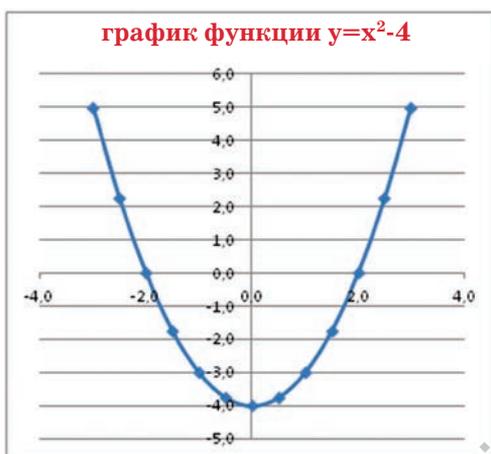


Рис. 2

Последовательность выполнения задания:

1. При построении таблицы начальное значение $x = -3$ с шагом 0,5.
2. В ячейку B2 запишем формулу =B1*B1-4 и скопируем ее до ячейки N2 включительно.
3. Выделив таблицу зависимости Y от X и выполнив последовательность команд **Вставка=>Диаграмма Точечная => Точечная гладкими кривыми**, получим диаграмму, изображенную на рисунке 2. При изменении данных в таблице, изменится и диаграмма.

Практическая работа №3

Вычисление расхода электроэнергии

Даны стоимость электроэнергии за 1кВт/ч и показания электросчетчика. Также известен расход электроэнергии за прошлый и текущий месяцы.

Задание:

1. Вычислить общее количество потраченной электроэнергии за прошлый и текущий месяцы.
2. Вычислить средний показатель потраченной энергии потребителями за текущий месяц.
3. Определить потребителя с наибольшим показателем использования электроэнергии.
4. Определить потребителя с наименьшим показателем использования электроэнергии.
5. Построить диаграмму потраченной электроэнергии потребителями.

В процессе выполнения задания в приложении MS Excel вы научитесь строить таблицы и форматировать их, вставлять в ячейки формулы, а полученные знания по работе с абсолютными ссылками, функциями и диаграммами использовать на практике.

Последовательность выполнения задания:

1. Создадим таблицу по образцу, представленному на рисунке 3, отформатируем данные в ячейках.
2. Внесем в ячейки соответствующие данные.
3. По заданию 1 в ячейках B11, C11, D11, E11, используя функцию суммирования, разместим соответствующие данные. Например, в ячейку B11 вставим формулу $=СУММ(B4:B10)$.
4. Для определения среднего показателя электроэнергии, потраченной в текущем месяце, в ячейку D12 запишем формулу $=СРЗНАЧ(D4:D10)$.
5. Для определения потребителя с наибольшим показателем использования электроэнергии в ячейку D13 запишем формулу $=МАКС(D4:D10)$. Для определения потребителя с наименьшим показателем использования электроэнергии в ячейку D14 запишем формулу $=МИН(D4:D10)$.
6. Чтобы построить диаграмму потраченной потребителями электроэнергии, выделим данные диапазона D4:D10 и с помощью команды **Вставка** \Rightarrow **Диаграмма** получим новую диаграмму (рис. 4).

	A	B	C	D	E
1	Вычисление электроэнергии				
2	стоимость 1кВт/час 15 тенге				
3	Дом №	Показатель счетчика за прошлый месяц(кВт/ча	Показатель счетчика за текущий месяц (кВт/час)	Расход (кВт/час)	цена за электроэнергию (тенге)
4	1-дом	2243	2347	104	1560
5	2-дом	4256	4401	145	2175
6	3-дом	356	499	143	2145
7	4-дом	7411	7789	378	5670
8	5-дом	8685	8888	203	3045
9	6-дом	25	231	206	3090
10	7-дом	485	567	82	1230
11	Общий расход			1261	18915
12	Средний расход			180,1	2702,1
13	наибольшее потребление			378	
14	Наименьшее потребление			82	

Рис. 3



Рис. 4



Домашнее задание

1. Заданы фамилия, имя, возраст и рост учащихся. Если в секцию волейбола набирают ребят не старше 14 лет и ростом выше 165 см, то сколько учеников могут в нее записаться?
2. В зависимости от длительности хранения продуктов на складах цена на них уменьшается. Если продукт хранится более 10 месяцев, то его цена снизится на 80 процентов, если срок хранения более 6 и менее 10 месяцев, цена упадет на 50 процентов. Составьте сводную таблицу товаров, хранящихся на торговом складе. В таблицу должны быть внесены наименования товаров, сроки хранения, первоначальная цена, цена после уценки.
3. Постройте график функции $y = -x^2 + 3$.

Примечание. Данные для таблицы возьмите из повседневной жизни. Например, если вы напишете, что рост ученика составляет 300 см, то это не будет соответствовать действительности.

3.5 АНАЛИЗ ДАННЫХ НА ОСНОВЕ ИМЕЮЩЕЙСЯ ИНФОРМАЦИИ



Каким образом в приложении MS Excel можно сделать прогноз на основе имеющейся информации?



Подумай

- Что такое прогноз?
- Какие прогнозы мы делаем в повседневной жизни?
- Делают ли прогнозы в науке и технике?



Новые знания

В статистике **динамическими рядами** называют ряды, отражающие развитие событий, которые происходят систематизированно и в хронологическом порядке. В динамическом ряду для каждого временного отрезка приводятся два показателя: показатель времени t и уровень ряда y . Динамические ряды образуются в результате измерения некоторых показателей: технических и социальных систем, природы, экономики. С помощью динамических систем можно исследовать изменение температуры объекта, пропускную способность сети, провести анализ продажи товара за определенный промежуток времени. Динамические ряды дают возможность определить направление или тренд основного развития определенных показателей.

Анализ динамических рядов – совокупность математико-статистических методов, предназначенных для определения структуры динамических рядов и их прогнозирования.

Слово «**прогноз**» (греч. πρόβλεψις – предвидение, предсказание) – означает предсказание будущего с помощью научных методов, а также сам результат предсказания. Прогноз – научная модель явлений или предстоящих событий.

Прогнозирование – процесс разработки прогнозов.

К методам прогнозирования относятся:

- статистический метод;
- экспертная оценка (метод Дельфи);
- моделирование.

Линия тренда в Excel – это график функции, который используется для составления прогнозов на основе имеющихся табличных данных.

Для наглядной иллюстрации тенденций изменения значений исходных табличных данных применяется линия тренда. Главной задачей линии тренда является составление прогноза дальнейшего развития событий.

Виды линий тренда в программе Excel

В программе Excel можно выбрать один из шести видов линий тренда: линейный, логарифмический, полиномиальный, степенной, скользящий средний, экспоненциальный. Рассмотрим подробнее линейный тренд.

Линейный тренд хорош при моделировании характеристик, значения которых увеличиваются или убывают с постоянной скоростью. Это наиболее простая в построении модель исследуемого процесса.

Используйте этот вид тренда для создания прямой линии, которая очень хорошо описывает простой линейный набор данных, когда точки данных расположены близко к прямой. Линейный тренд используется для анализа числовых данных, например, для оценки зависимости между двумя видами значений. При построении линейного тренда связи между двумя параметрами определяются коэффициентом линейной функции по методу наименьших квадратов. Для нахождения значений коэффициента линейной функции используется коэффициент R-квадрат.

Коэффициент R-квадрат

Коэффициент R-квадрат называют величиной достоверности, с его помощью можно определить уровень точности прогноза. При значении коэффициента, равном единице (или близком к единице), можно говорить о возможности использования полученных данных для прогноза. При сопоставлении данных линии тренда в программе Excel автоматически вычисляется значение коэффициента R-квадрат. При необходимости можно эти значения отразить на диаграмме.

Руслан любит читать страницы блогеров в социальных сетях. Он обратил внимание, что число посетителей страниц зависит от количества постов и репостов. Если на странице опубликовано 100 статей, то количество ее посетителей составляет 228 тысяч (табл. 1).

Теперь Руслана интересуют ответы на следующие вопросы: сколько людей просмотрят страницу, если количество опубликованных постов будет равным 120? а если 150?

Для определения отношения между двумя величинами Руслан составил формулу, по которой, зная количество постов, можно вычислить количество посетителей страницы. Что представляет собой данная формула?

Таблица 1

№	Количество статей на странице	Количество посетителей страницы (тысяч человек)
1	30	67
2	42	88
3	57	92
4	69	120
5	75	145
6	87	184
7	95	199
8	100	228
9	120	?
10	150	??

Для составления равенства, определяющего отношение между двумя величинами, введем обозначение. Количество статей обозначим x , количество посетителей обозначим y . По таблице 1 можем определить, что отношение между этими величинами линейное. Тогда запишем формулу линейного уравнения, известную из курса алгебры:

$$y=kx+b.$$

Если в этом уравнении будут известны значения коэффициента k и свободного члена b , то, взяв любое значение x , вычислим соответствующее ему значение y .

На основе табличных данных построим линейный график.

Для этого, выделив табличные данные, необходимо выполнить следующую последовательность команд: **Вставка => Диаграммы => Точечная => Точечная с маркерами**. В результате получим диаграмму, как на рисунке 1.

На диаграмме по оси x расположено количество постов, по оси y – количество пользователей страницы (тысяч человек). Точки не расположены на одной линии (прямой), а разбросаны. Можем ли мы провести одну линию, максимально близко расположенную ко всем

точкам? Для ответа на данный вопрос нужно проверить корреляцию этих точек. Если при изменении одной величины, изменяется среднее значение другой, то такая статистическая зависимость называется **корреляционной зависимостью**. В программе Excel имеется функция **КОРРЕЛ** (корреляция) (рис. 1). Значение корреляции функции обозначается буквой **R**. **Корреляция** в переводе с латинского *correlatio* – отношение, взаимосвязь. Если значение корреляции

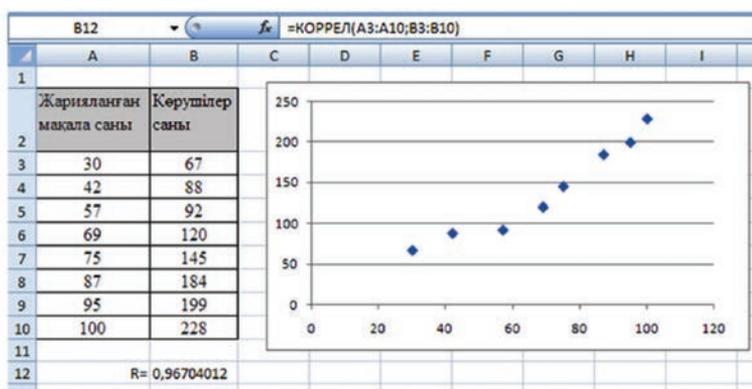


Рис. 1

функции $R > 0$, то связь положительно возрастающая, если $R < 0$, то связь отрицательно убывающая. При приближении значения корреляции функции к 1, линии связи приближаются к прямой. В нашем случае $R=0,96704012$, то есть отчетливо видна линейная зависимость.

Для изображения на диаграмме линий линейного тренда выберем одну точку, например, пятую по счету. Выделим ее левой кнопкой мыши, в контекстном меню выполним команду «Добавить линию тренда...» (рис. 2). Из шести видов, представленных на рисунке 2, выберем тренд «Линейная». Для отражения на диаграмме уравнения зависимости линейного тренда установим флажок в строке «показывать уравнение на диаграмме». Для размещения на диаграмме достоверной величины аппроксимации установим

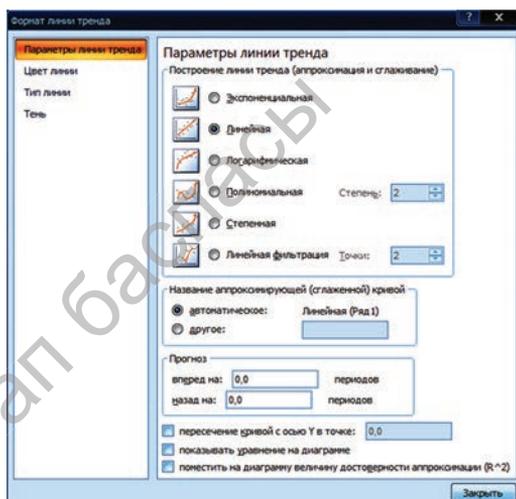


Рис. 2

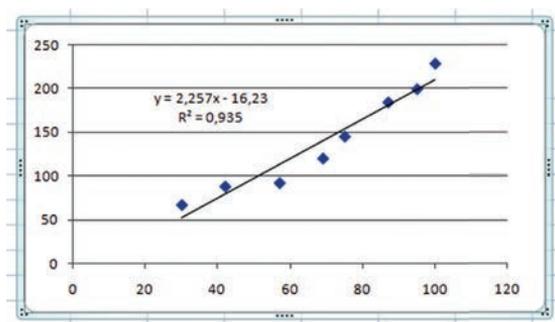


Рис. 3

флажок в строке «поместить на диаграмму величину достоверности аппроксимации R^2 ». При нажатии на кнопку «Закреть» получим диаграмму, на которой будет отражена линия тренда, записаны уравнение зависимости и значение R^2 (рис. 3).

Теперь найдем ответы на вопросы Руслана. Для этого мож-

но использовать следующие методы.

1-й метод. На диаграмме для $x=120$ по линейной зависимости $y=2,257x-16,23$ нужно выполнить прогнозирование (таблица 1). Если в результате вычислений количество посетителей страницы составит 254 тысячи, а количество статей – 150, то можно предположить, что посетителей будет 322 тысячи.

$$y = 2,257 * 120 - 16,23$$

$$y = 254,31$$

2-й метод. Для вычисления прогноза используем функцию **ПРЕДСКАЗ**. Она относится к категории статистических. Результаты действия функции можно увидеть на рисунке 4:

= ПРЕДСКАЗ (известные_значения_y; известные_значения_x)

Вместо x из таблицы 1 возьмем для прогноза значение 120, вместо y укажем количество пользователей. Для построения прогноза вместо x возьмем промежуток значений, расположенных выше $x=120$. Результат прогнозирования получился равным 254,65, как и в первом методе.

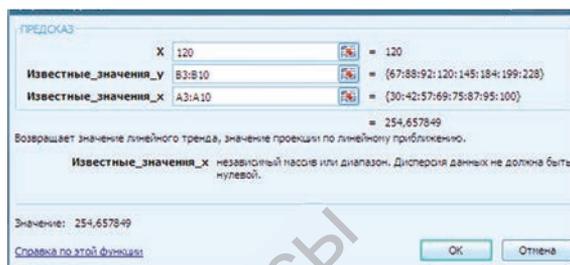


Рис. 4



Анализ



Выполните анализ задач Руслана. При выполнении анализа покажите отдельно теоретическую и практическую части прогнозирования.

**Синтез**

Составьте 2 примера на прогнозирование из повседневной жизни. Докажите зависимость между двумя величинами в составленных вами примерах. Например, зависимость рождаемости в Казахстане от экономического роста в ближайшие 10 лет.

**Оценка**

Какова значимость прогнозирования на основе имеющихся данных? Что дает такое прогнозирование людям?

**Вопросы**

1. Что такое динамические ряды? Какие функции они выполняют?
2. Какие функции используются в программе Excel для прогнозирования?
3. Объясните назначение функции КОРРЕЛ.
4. Что такое линейный тренд?
5. Какие виды линейных трендов вы знаете?
6. Каково назначение функции ПРЕДСКАЗ?

**Задания**

1. Один из составленных вами примеров используйте для прогнозирования на компьютере.
2. Имеются данные об изменении погоды за 10 дней (табл. 2). Составьте прогноз погоды на 11 день.

Таблица 2. Погода за 10 дней

Дни	Температура за этот день
1	+2
2	+3
3	+5
4	+4
5	+6
6	+10
7	+13
8	+9
9	+15
10	+18
11	?

3.6-3.7 РЕШЕНИЕ ПРИКЛАДНЫХ ЗАДАЧ



Как применять встроенные функции, различные типы и форматы данных, а также на их основе строить графики функций при решении прикладных задач в приложении MS Excel?



Подумай

- Что такое прикладная задача?
- Какое преимущество дает решение прикладных задач на компьютере перед выполнением расчетов вручную?
- Какие прикладные задачи можно решить с помощью функций?



Практическая работа

Прикладные задачи – задачи профессиональной направленности, возникающие в какой-либо сфере человеческой деятельности: в математике, физике, химии, экономике, бухгалтерском учете и др. Применение теоретических материалов на практике называют прикладными задачами. Многие из них можно легко и быстро решить с помощью приложения MS Excel.

Применение электронных таблиц для решения математических задач

Задание. Даны функции $y = 2x + 2$ и $y = -2x + 6$. Постройте графики этих функций и найдите точку их пересечения.

Алгоритм выполнения задания

1. При построении таблицы примем значение x в интервале от -5 до $+5$ с шагом 1 .

2. По значениям x построим график функции $y = 2x + 2$, для вычисления значения y в ячейку **A3** запишем формулу $y = 2 * B1 + 2$, скопируем ее в ячейки с **A4** по **A13**.

3. По значениям x построим график функции $y = -2x + 6$, для вычисления значения y в ячейку **C3** запишем формулу $y = -2 * B1 + 6$, скопируем ее в ячейки с **C3** по **C13**.

4. Для построения графика функции $y=2x+2$ выделим данные из диапазонов **В3:В13** и **А3:А13**, выполним команду **Вставка=>Точечная=>Точечная гладкими кривыми** и получим красную линию на диаграмме, изображенной на рисунке 1.

5. Для построения графика функции $y=-2x+6$ на этой же диаграмме, выделим данные из диапазона **С3:С13**, нажмем кнопку **«Вставить»**. На диаграмме появится синяя линия. Координаты точки пересечения графиков можно увидеть в 9-й строке таблицы, изображенной на рисунке 1, других таких совпадений нет.

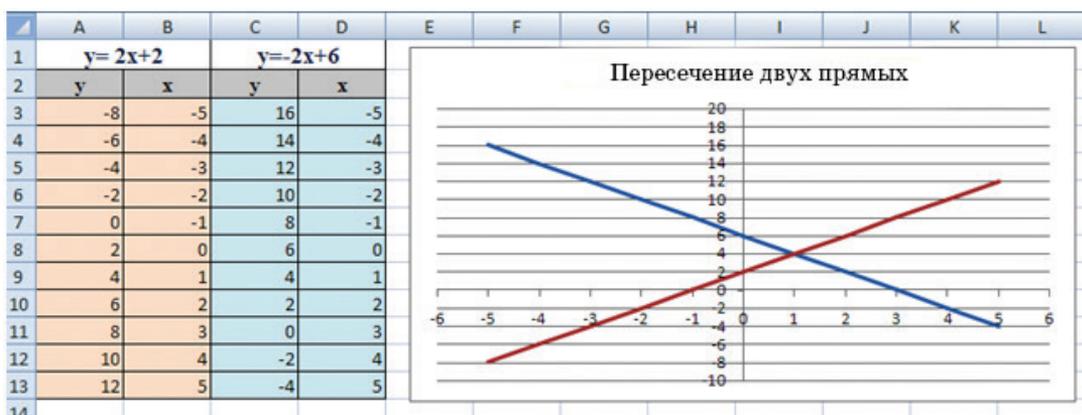


Рис. 1

Применение электронных таблиц для решения физических задач

Задание. Постройте график зависимости силы тока от сопротивления $I(R)$. Сопротивление (R) изменяется в пределах от 1 до 10 Ом с шагом 1 Ом.

Алгоритм выполнения задания

Для выполнения задания запишем закон Ома для части цепи из курса физики. Из формулы видно, что сила тока и сопротивление обратно пропорциональны.

$$I = \frac{U}{R}$$

В ячейку D1 запишем постоянное значение напряжения. Например, $U=40$ В. Ячейку D1 будем использовать с абсолютной адресацией. В столбец B запишем значения сопротивления, а в ячейку A2 запишем формулу $=D\$1/B2$ и скопируем ее до ячейки A11 (рис. 2)

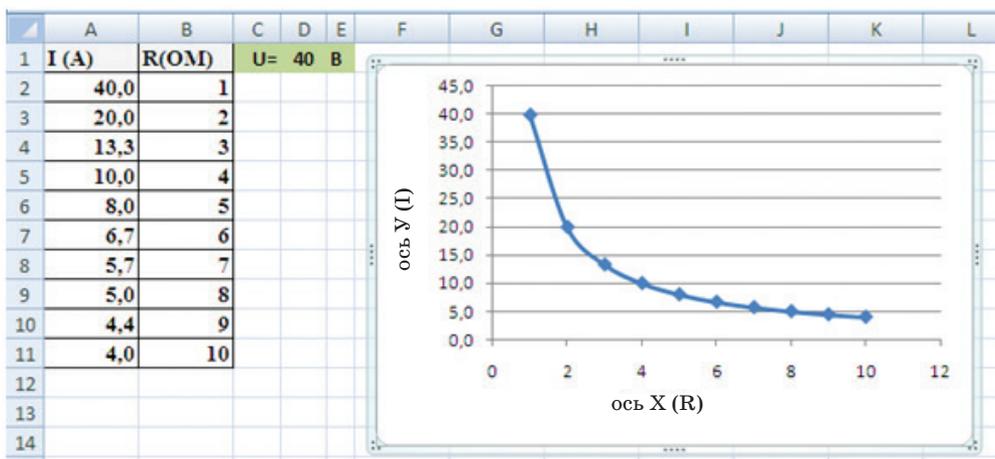


Рис. 2

Использование электронных таблиц для решения геометрических задач

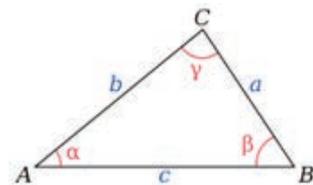
Задание. Заданы все стороны нескольких треугольников. Используя формулу Герона, вычислите площади треугольников. Определите треугольник с наибольшей и наименьшей площадью.

Алгоритм выполнения задания

Используя заданные величины трех сторон (a , b и c) треугольника, вычислим его площадь (S) по формуле Герона из курса геометрии:

$$S = \sqrt{p(p-a)(p-b)(p-c)},$$

где p – полупериметр треугольника: $p = \frac{a+b+c}{2}$



В приложении Excel на примере рисунка 3 заполним таблицу. Задайте свои значения сторон. По условию существования треугольника, сумма любых двух его сторон должна быть больше третьей.

В ячейку E3 запишем формулу $=\text{КОРЕНЬ}(\text{E3}*\text{E3}*\text{E3}*\text{E3})/2$ и скопируем ее в ячейки с E4 по E12.

В ячейку F3 запишем формулу Герона:
 $=\text{КОРЕНЬ}(\text{E3}*(\text{E3}-\text{B3})*(\text{E3}-\text{C3})*(\text{E3}-\text{D3}))$

и скопируем ее в ячейки с F4 по F12.

Чтобы вычислить наибольшую площадь треугольника, используем формулу $=\text{МАКС}(\text{F3}:\text{F12})$.

Чтобы вычислить наименьшую площадь треугольника, используем формулу =МИН(F3:F12).

Чтобы вычислить среднее значение площади, используем формулу =СРЗНАЧ(F3:F12).

A15 Средняя площадь треугольника						
	A	B	C	D	E	F
1	Треугольник №	Стороны треугольника			Полупериметр	Вычисление площади треугольника
2		a	b	c		
3	1	10	12	13	17,5	57,00
4	2	8	7	9	12	26,83
5	3	5	4	3	6	6,00
6	4	8	5	11	12	18,33
7	5	4	6	8	9	11,62
8	6	7	11	13	15,5	38,50
9	7	14	15	16	22,5	96,56
10	8	2	4	3	4,5	2,90
11	9	4	4	7	7,5	6,78
12	10	3	4	5	6	6,00
13	Наибольшая площадь треугольника					96,56
14	Наименьшая площадь треугольника					2,90
15	Средняя площадь треугольника					27,05

Рис. 3

Применение электронных таблиц для решения экономических задач

Задание. Телефонная компания начала использовать новую систему оплаты за услугу пользования телефоном. Если потребитель в течение 1 месяца разговаривает до А минут, то он за каждую минуту платит 10 тенге, а если больше А минут, тогда за каждую минуту, превышающую норматив, оплачивает по В тенге. Как с помощью электронной таблицы вычислить сумму оплаты первых десяти потребителей?

Алгоритм выполнения задания

1. Для выполнения задания подготовим таблицу по образцу, представленному на рисунке 4.
2. Используя логическую функцию, вычислим стоимость разговоров, превышающих лимит А минут, а также стоимость разговоров, которые длились меньше А минут. Для этого в ячейку С4 запишем формулу =ЕСЛИ(B4>\$C\$1;\$C\$1*\$F\$1;B4*\$F\$1) и скопируем ее в ячейки с С5 по С13. В формуле граница разговоров – А минут, ячейки со стоимостью разговоров имеют абсолютную адресацию.

- Если время разговоров потребителя превышает А минут, то лишние минуты занесем в столбец D. Для этого в ячейку D4 запишем формулу $=ЕСЛИ(B4>C4;B4-C4;0)$ и скопируем ее в ячейки с D5 по D13.
- В ячейку E4 запишем формулу $=D4*F4$, вычисляющую стоимость разговоров, превышающих норматив, и скопируем ее в ячейки с E5 по E13.
- В столбце F будем вычислять стоимость разговоров за один месяц. Для этого в ячейку F4 запишем формулу $=C4+E4$ и скопируем ее в ячейки с F5 по F13.

	A	B	C	D	E	F
1	A=		300 минут меньше		Цена= 10	
2	B=		300 минут больше		Цена = 25	
3	Дома	Время потраченное на разговоры по телефону за месяц (минуты)	Цена за разговоры (тенге)	Превышение допустимого времени (минут)	Цена за разговоры (тенге)	Общая сумма за телефонную связь(тенге)
4	1 -дом	500	3000	200	5000	8000
5	2 -дом	173	1730	0	0	1730
6	3 -дом	468	3000	168	4200	7200
7	4 -дом	392	3000	92	2300	5300
8	5 -дом	100	1000	0	0	1000
9	6 -дом	500	3000	200	5000	8000
10	7 -дом	267	2670	0	0	2670
11	8 -дом	308	3000	8	200	3200
12	9 -дом	150	1500	0	0	1500
13	10 -дом	87	870	0	0	870

Выполните задания, приведенные ниже, используя образцы решения прикладных задач.



Практическая работа

- Даны функции $y=3x-2$ и $y=-x+4$. Постройте графики этих функций и найдите точку их пересечения.
- Постройте график функции $y=-x^2+2$.
- Вычислите НОД чисел 48, 84, 36, 60, 24.
- Вычислите НОК чисел 21, 32, 7, 6, 12.
- График зависимости между силой тока и напряжением $I(U)$ называют вольт-амперной характеристикой. Напряжение U изменяется от 1 до 10 В с шагом 1 В. Постройте график зависимости между силой тока и напряжением.
- Тело движется с постоянной скоростью 5 м/с. Постройте график зависимости пройденного пути от времени $S(t)$. Время изменяется в интервале $t=0...20$ секунд с шагом 2 секунды.

7. Постройте график зависимости между кинетической энергией тела и его скоростью $E_k(U)$. Масса тела постоянна и равна $m=100$ кг. Скорость тела изменяется в интервале $U=0...20$ м/с с шагом 2 м/с. Определите, к какому виду относится график зависимости. Обоснуйте ответ.
8. Постройте график зависимости теплоемкости от температуры $Q(\Delta t)$. Сколько потребуется количества тепла Q , чтобы изменить температуру воды массой $m=10$ кг. $C=4200$ Дж/кг $^{\circ}$ С.
9. Используя данные, приведенные в таблице, вычислите неизвестные величины.

	А	В	С
1	Радиус окружности	Диаметр окружности	Длина окружности l (дм)
2	20	?	?
3	38	?	
4	15		
5	45		

10. В нижеприведенной таблице заданы координаты вершин треугольника. Например, одна вершина $A(x_1, y_1)$. Используя эти координаты, вычислите длины сторон треугольника. Можно ли из этих сторон построить треугольник? Определите. Запишите «Да» или «Нет» рядом с величинами сторон в соответствующем столбце.

Примечание. Для нахождения длины сторон и определения условия существования треугольника воспользуемся формулой вычисления расстояния между двумя точками и условием существования треугольника.

	А	В	С	Д	Е	Ф	Г	Н	І	Ј
1	А		В		С		Длина стороны			Существует ли треугольник? «Да» или «Нет»
2	x_1	y_1	x_2	y_2	x_3	y_3	а	б	с	
3	2	3	6	8	17	8				
4	-4	-3	8	9	-7	1				
5	4	6	9	9	5	7				
6	9	8	90	99	3	3				
7	7	7	11	20	17	26				

11. Составьте таблицу, содержащую баллы 10 учеников по профильным предметам математике и физике, сдававших ЕНТ. По профильным предметам максимально возможно набрать 40 баллов. Для перевода баллов в оценки имеется соответствующая шкала. Оценка «5» за 81-100% правильных ответов, «4» – за 61-80%, «3» – за 41-60%.

Если процент верных ответов ниже, то ставится оценка «2». Определите сколько учеников по результатам тестирования получили оценки «5», «4», «3» и «2».

	A	B	C	D	E
1	Ученики	Математика	Оценка	Физика	Оценка
2	Ученик №1	25		18	
3	Ученик №2	40		29	
4	Ученик №3	12		14	
5	Ученик №4	32		36	
6	Ученик №5	40		40	
7	Ученик №6	20		5	
8	Ученик №7	13		6	
9	Ученик №8	7		21	
10	Ученик № 9	4		18	
11	Ученик № 10	35		25	
12	Количество оценок «2»				
13	Количество оценок «3»				
14	Количество оценок «4»				
15	Количество оценок «5»				

12. Даны демографические изменения численности населения Казахстана за последние 10 лет. Какова будет численность населения страны к 2027 году? Составьте прогноз.

	A	B
1	Годы	Численность населения
2	2008	15 571 506
3	2009	15 776 492
4	2010	16 204 617
5	2011	16 441 959
6	2012	16 675 392
7	2013	16 911 911
8	2014	17 165 239
9	2015	17 417 447
10	2016	17 670 957
11	2017	17 926 500

Творческие задания

Задание 1. Составьте таблицу, отражающую изменение атмосферного давления в течение недели в Астане. Проведите анализ изменения атмосферного давления. При выполнении задания используйте абсолютную адресацию и логические функции.

Нормальное атмосферное давление составляет 760 мм ртутного столба.

Если:

- значение в интервале 755 -765 мм ртутного столба, оно «соответствует нормативу»;
- значение в интервале 720-754 мм ртутного столба, оно «ниже норматива»;

- значение в интервале 766-780 мм ртутного столба, оно «выше норматива».

	A	B	C	D	E
1	№	Дни недели	Атмосферное давление (мм ртутного столба)	Отклонение от нормативных требований	Оценка
2	1	Понедельник	746	16	Ниже норматива
3	2	Вторник	764	4	Соответствует нормативу
4	3	Среда	755	5	Соответствует нормативу
5	4	Четверг	778	32	Выше норматива
6	5	Пятница	719	41	Ниже норматива
7	6	Суббота	735	25	Ниже норматива
8	7	Воскресенье	758	2	Соответствует нормативу

Дескрипторы для выполнения:

- полностью форматирует таблицу;
- умеет использовать абсолютную адресацию ячеек;
- умеет правильно записывать в ячейки выражение с логическими функциями.

Задание 2. Определить физическое развитие учеников 8 класса.
 Норма веса = (Рост – Вес) * 1,1

	№	A	B	C	D	E
1	1	ФИО учеников	Рост (см)	Вес (кг)	Норма веса (кг)	Отклонение от нормы (%)
2	2	1-й ученик	161	49		
3	3	2-й ученик	157	50		
4	4	3-й ученик	149	55		
5	5	4-й ученик	172	60		
6	6	5-й ученик	165	50		
7		Среднее значение				

Дескрипторы для выполнения:

- полностью форматирует таблицу;
- использует формулы в ячейках;
- использует типы данных в ячейках;
- использует функции в ячейках.

3.8 ВЫПОЛНЕНИЕ ПРОЕКТНЫХ РАБОТ В ПРИЛОЖЕНИИ MS EXCEL



Как в повседневной жизни применять знания, полученные при изучении процессов обработки информации в электронных таблицах?



Подумай

- Как систематизировать теоретические знания и практические навыки, полученные при изучении возможностей электронных таблиц?

Творческие темы

Выберите одну из тем, приведенных ниже. Выполните проект самостоятельно или в паре и презентуйте его. При выполнении задания, кроме рассмотрения теоретического материала, желательно создать программный продукт, который высоко оценивается.

Тема проекта №1: Эффективное применение программы MS Excel в повседневной жизни.

Краткое содержание

Раскрыть возможности программы MS Excel, показать в каких сферах ее используют, а также каково ее применение в повседневной жизни.

Тема проекта №2: Диаграммы вокруг нас.

Краткое содержание

Рассказать о видах диаграмм, которые можно построить на основе имеющихся данных в программе MS Excel. Показать историю использования диаграмм и способы их построения. Исследовать и доказать необходимость построения диаграмм на примерах из повседневной жизни.

Тема проекта №3: Исследование графиков линейной функции, параболы и гиперболы в программе MS Excel.

Краткое содержание

Рассказать о видах математических функций, провести исследование графиков линейной функции, параболы и гиперболы на основе данных в программе MS Excel.

Тема проекта №4: Пути решения систем линейных уравнений в программе MS Excel.

Краткое содержание

Провести исследование и составить алгоритм путей решения системы линейных уравнений в программе MS Excel. Исследовать пути представления, составления автоматизированных образцов, облегчающих вычисления в программе MS Excel. Представить программный продукт. Сделать презентацию.

Тема проекта №5: Создание теста в программе MS Excel.

Краткое содержание

Исследовать возможности создания тестов в приложении MS Excel. Объяснить особенности составленного теста по сравнению с аналогами. Выявить достоинства и недостатки, представить свой программный продукт. Создать презентацию.

Тема проекта №6: Эффективное использование программы MS Excel для определения качества знаний.

Краткое содержание

Объяснить, как с помощью программы MS Excel можно оценить качество знаний учащегося в каждой четверти. Составить автоматизированные образцы для расчета качества знаний и представить программный продукт. Создать презентацию.

Тема проекта №7: Увеличение точности вычислений в лабораторных работах по физике с помощью программы MS Excel.

Краткое содержание

В курсе школьной физики проводить вычисления в лабораторных работах вручную нелегко. Представить образцы программ для точного вычисления абсолютной и относительной погрешностей при проведении расчетов в программе MS Excel. Представить программу. Создать презентацию.

Тема проекта №8: Использование программы MS Excel для решения прикладных математических задач.

Краткое содержание

Представить исследование использования программы MS Excel для решения прикладных математических задач, а также о применении математики в повседневной жизни. Разработать программу. Сделать презентацию.

Рекомендации по эффективной защите проекта

- Представьте выбранную тему.
- Объясните причину выбора темы.
- Сформулируйте цели и задачи.
- Опишите использованные средства.
- Сделайте краткий обзор темы.
- Поделитесь своими мыслями о реализации проекта.
- Расскажите о новых знаниях и навыках, полученных при работе над проектом.
- Познакомьте слушателей с заключением проекта.
- Выслушайте мнения и оценки аудитории, умейте принять их.

Критерии успешного проекта

- Самостоятельность выполнения проекта.
- Актуальность темы.
- Раскрытие темы.
- Свой способ представления решения проблемы.
- Артистические данные при защите проекта и умение держать себя на публике;
- Умение использовать технические средства, а также возможность использовать визуальные средства.
- Умение правильно и точно ответить на поставленные вопросы.

Каким должен быть отчет о выполненном проекте

- Почему я выбрал эту тему.
- Что я узнал нового, чему научился.
- Как я осуществлял поиск необходимой информации.
- Как я редактировал информацию.
- Как я совмещал теорию и практику.
- Какие части проекта наиболее эффективны.
- Что я не смог выполнить.
- Что было самым легким в проекте.
- Какие трудности возникали во время работы над проектом.
- Если бы я начал проект заново, что бы изменил в нем.
- Мое впечатление от проекта.

Словарь – сөздік – dictionary

На русском языке	На казахском языке	На английском языке
Статистический	Статистикалық	Statistical
Инженерный	Инженерлік	Engineering
Аналитический	Аналитикалық	Analytical
Если	Егер	If
И	Және	And
Или	Немесе	Or
Истина	Ақиқат	True
Ложь	Жалған	False
Динамический ряд	Динамикалық қатар	Dynamic series
Прогноз, прогнозирование	Болжам	Forecast, prediction
Линейный	Сызықтық	Linear
Логарифмический	Логарифмдік	Logarithmic
Многочлен	Көпмүшелі	Polynomial
Степень	Дәреже	Degree
Визуальный	Көрнекі	Visual
Квадратное значение	Квадрат мәні	Square value
Корреляция	Корреляция	Correlation
Электронная таблица	Электрондық кесте	Spreadsheet
Вставить	Кірістіру	Insert
Диаграмма	Диаграмма	Chart

РАЗДЕЛ IV. ИНТЕГРИРОВАННЫЕ СРЕДЫ РАЗРАБОТКИ ПРОГРАММ

4.1

КЛАССИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



Каковы отличия между системным программным обеспечением, прикладными программами и системами программирования?



Подумай

- Какие программы вы используете при работе на компьютере?
- Знаете ли вы, на какие типы подразделяются компьютерные программы?
- Какова роль программ при работе на компьютере?



Новые знания

Программное обеспечение компьютера

Весь набор программ для компьютера называется программным обеспечением. Для успешной работы на компьютере необходимо знать, какой программой надо пользоваться в каждом конкретном случае. По своему назначению программное обеспечение делится на следующие типы (*Схема 1*):



Схема 1. Типы программного обеспечения



Системное программное обеспечение

Этот тип программного обеспечения считается самым важным для компьютера. С его помощью обеспечивается управление компонентами компьютера, осуществляется связь между компьютером и пользователем. Системное программное обеспечение – это комплекс

программ. Центральное место среди системных программ занимают операционные системы. Они управляют всей работой компьютера от загрузки до выключения.

Операционные системы (ОС) – комплекс программ, обеспечивающих управление процессором, организующих работу с файлами и выполнение прикладных программ, а также осуществляющих ввод и вывод данных. Составной частью любой операционной системы является файловая система. По сути, все программы и операционные системы хранятся в виде файлов. Ранее вы изучали действия с файлами – удаление, копирование, переименование и т.д. Все эти действия обеспечиваются работой файловой системы.

Операционная система:

- выполняет прикладные программы;
- управляет внешней и внутренней памятью компьютера, процессором и внешними устройствами;
- осуществляет связь между компьютером и пользователем.

Наиболее распространенные операционные системы персональных компьютеров – MS DOS; WINDOWS 95/98; WINDOWS 2000/XP; WINDOWS 7, 8, 10; OS/2; Linux.

Прикладные программы

Прикладные программы (или приложения) – программы, предназначенные для выполнения определенных задач и рассчитанные на непосредственное взаимодействие с пользователем. С помощью прикладных программ можно слушать музыку, смотреть кино, пользоваться интернетом, читать, решать задачи в электронных таблицах, играть и т.д. Можно использовать прикладную программу для оценки финансовой деятельности фирмы или выполнения дру-



гой практической работы – разработки дизайна, печати документов и т.д. Прикладные программы могут работать только в операционной системе. Каждая прикладная программа предназначена для выполнения конкретной работы. Например, для рисования используется графический редактор, а для работы с текстом – текстовый редактор.

Группу прикладных программ называют **приложением операционной системы Windows**, или пакетом прикладных программ. Наиболее популярные приложения:

- текстовые редакторы;
- графические редакторы;
- электронные таблицы (MS Excel);
- системы управления базами данных;
- обучающие программы;
- исследовательские и дизайнерские программы;
- компьютерные игры.

Системы программирования

Системы программирования – это совокупность программ для разработки, отладки и внедрения новых программных продуктов для обеспечения работы персональных компьютеров и техники, автоматизации вычислительных процессов и эффективного выполнения прикладных программ. С помощью систем программирования можно создать алгоритмы новых операционных систем и прикладных программ.

В системах программирования имеются операторы с такими же служебными функциями, но имеющие другие формы записи. Например, если в языке программирования Бейсик вывод результата на экран осуществляется оператором PRINT, то в языке Паскаль – оператором WRITE. Но принцип работы у них одинаковый. Наиболее распространенные языки программирования – Basic, Pascal, C++, Delphi, ЛОГО и др. Количество программ на компьютере влияет на скорость работы и на использование его ресурсов. Поэтому пользователь должен руководствоваться необходимостью применения той или иной программы.



Анализ



Исследуйте программное обеспечение, имеющееся на компьютере, и сделайте анализ. Выберите три программы, относящиеся к трем типам программного обеспечения. Покажите их сходства и различия в виде таблицы.

**Синтез**

Выберите профессию. Например, дизайнер или врач. Какие прикладные программы используют эти специалисты. Ответ обоснуйте.

**Оценка**

Оцените важность программного обеспечения компьютерного оборудования. Вывод обоснуйте примерами.

**Вопросы**

1. Что такое программное обеспечение?
2. Какие бывают типы программного обеспечения?
3. Что такое прикладные программы?
4. Какие прикладные программы вы знаете?
5. Какие системы программирования вам известны?

**Задание**

Выберите три прикладные программы на компьютере и заполните таблицу. Например, MS Word, MS Excel, MS PowerPoint. Для анализа можете выбрать другие программы.

Параметры анализа	MS Word	MS Excel	MS PowerPoint
Назначение			
Сходство интерфейсов			
Различие интерфейсов			
Сходство инструментов			
Различие инструментов			

4.2 КОМПОНЕНТЫ ИНТЕГРИРОВАННОЙ СРЕДЫ РАЗРАБОТКИ ПРОГРАММ



Как применять компоненты интегрированной среды разработки программ?



Подумай

- С какими компонентами среды Lazarus вы знакомы?
- Какие преимущества имеют данные компоненты?



Новые знания

Компоненты среды Lazarus

Визуальная среда Lazarus имеет большое количество готовых компонентов для написания программ. В 7 классе при первом знакомстве с этой программной средой вы изучили некоторые компоненты из вкладки Standard. Сейчас мы продолжим изучение и знакомство с компонентами и их применением.

В окне среды Lazarus имеется 15 иконок-приложений, содержащих около 250 компонентов, используемых при создании программ. Изучить их все в рамках одной главы учебника невозможно. Однако мы рассмотрим основные и научимся их применять.

Если в будущем вы планируете стать программистом, то Lazarus поможет вам в освоении данной профессии.

В главном окне приложения Lazarus (рис. 1) отражены компоненты одной из вкладок.

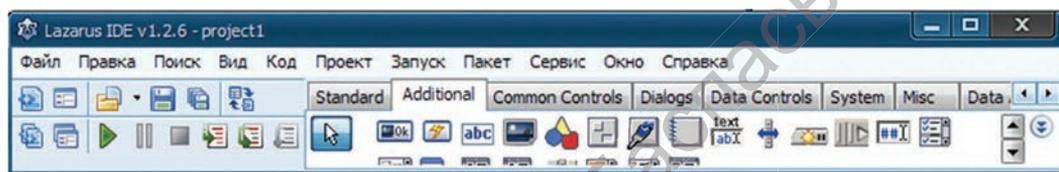


Рис. 1

Полный список компонентов среды программирования показан на *рисунке 2*. Изучим возможности и работу некоторых из них, а также свойства компонентов, необходимые для создания приложения.

Компонент BitBtn

Командная кнопка. В отличие от стандартной кнопки Button, здесь текст можно вставить в виде отдельного рисунка. Компонент обладает несколькими свойствами.

Например, свойство **Glyph**, с помощью которого можно загрузить рисунки. В стандартном случае они располагаются по адресу **C:\lazarus\images** или загружаются из сети Интернет (рис. 3).



Рис. 3

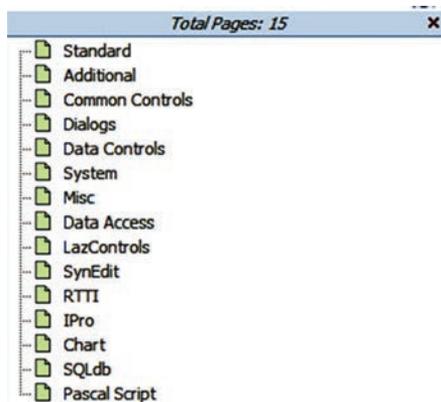


Рис. 2

Нажимая кнопку справа от свойства **Glyph** «...», выбираем рисунок из большого количества изображений, затем нажимаем последовательно на кнопки **Загрузить** и **ОК**. Свойство **Layout** определяет где будет располагаться пиктограмма:

- **blGlyphBottom** – ниже текста;
- **blGlyphLeft** – слева от текста;
- **blGlyphRight** – справа от текста;
- **blGlyphTop** – выше текста.

Компонент RadioGroup

Если в программе нужно использовать несколько взаимоисключающих альтернатив, то применяется компонент **RadioGroup** (табл. 1, рис. 4). Надписи кнопок и их количество определяются свойством **Items**, имеющим тип **TStrings**.

Таблица 1. Свойства компонента RadioGroup

Свойство	Содержание
property Columns: Integer;	Определяет количество взаимоисключающих альтернатив
property ItemIndex: Integer;	Показывает индексы взаимоисключающих альтернатив
property Items: TStrings;	Определяет заголовок кнопок и переключателей
property Caption: TCaption;	Текстовый заголовок переключателя (по умолчанию пишется заголовок RadioGroup1)



Рис. 4

Компонент CheckBox

 Используется в качестве индикаторов. По щелчку выбирает значение true и false. Обладает свойством Checked.

Компонент RadioButton

 Зависимый переключатель. Из нескольких кнопок может быть выбрана только одна. Также обладает свойством Checked.

Компонент Memo

 Компонент для работы с большим количеством строк – ввод, отображение и редактирование текстового материала (табл. 2, рис. 5). Вмещает объем 32 Кб или 20 рядов текстового ряда.

Таблица 2. Текстовой редактор

Свойство	Содержание
Name	Имя компонента
Text	Текст на поверхности Мемо
Lines	Группа текстов. Переход на соответствующий номеру текст. Нумерация начинается с нуля
Lines. Count	Количество тестовых рядов в Мемо
ScrollBar	Образует прокручивание, если объем текста не вмещается в окно
ReadOnly	Разрешает или запрещает редактирование текста
Add (s)	Добавляет текст в рабочее поле
Delete (n)	Удаляет N-й ряд.
Insert (n; s)	Ввод текста с S-го по N-й ряд

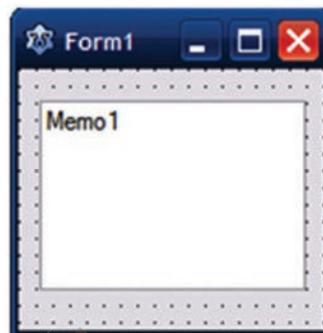


Рис. 5

Компонент Image

Компонент Image предназначен для вывода на форму графического рисунка (табл. 3, рис. 6). По умолчанию рисунки представлены в формате **bmp**. Для показа рисунков в формате **jpg**, нужно добавить модуль JPEG.

Таблица 3. Компонент Image

Свойство	Содержание
Picture	Рисунок на поверхности формы
Proportional	Пропорциональное увеличение или уменьшение рисунка
Stretch	Пропорциональное растяжение и сжатие рисунка
AutoSize	Автоматическое определение размеров
Center	Уточняет расположение рисунка
Visible	Если true , то рисунок отображается на форме, если false , то не отображается

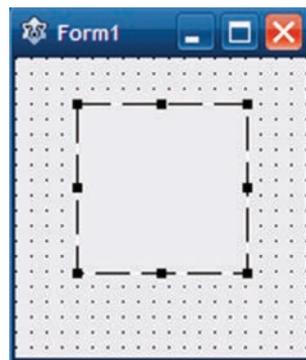


Рис. 6



Практическая работа

Рассмотрим практическое применение изученных компонентов. Для этого разработаем проект «Случайные числа».

С помощью оператора `Random[n]` на определенном интервале выбрать случайные числа и расставить по порядку.

Использование оператора `Random[n]`

<code>random (i:integer):integer</code>	Использование оператора <code>Random[n]</code>
<code>randomise</code>	Выбираются случайные цифры. Если в программу не введен <code>randomise</code> , значение случайных цифр не меняется.

Для построения проекта форматируем окно будущего приложения, как показано на *рисунке 7*. Конечно, можно этот проект выполнить по-другому. Это всего лишь образец. Вместе с кнопкой **Button**, которая выбирает случайное число, используем компонент **Memo**, который выстраивает случайные числа. Очень важно правильно задать свойства компонента Мемо. Если количество случайных чисел превысит 15, они могут не вместиться в окно экрана. В этом случае используем свойство «**ScrollBars ssAutoVertical**» в компоненте Мемо. Исполнение проекта показано на *рисунке 8*. Программный код для кнопки **Button** написан внутри рамки.

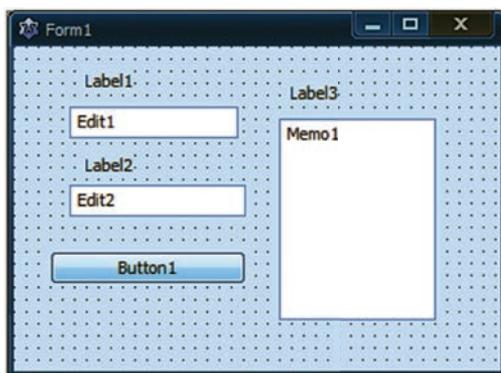


Рис. 7

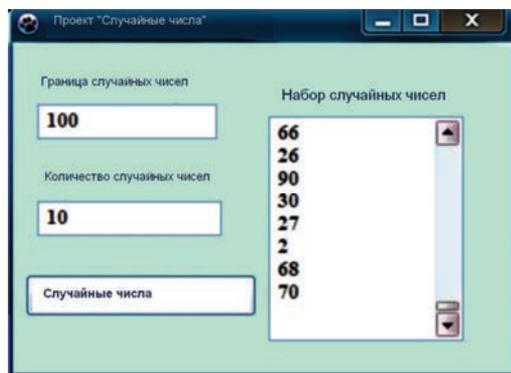


Рис. 8

```

Программный код проекта
«Случайные числа»
procedure TForm1;
Button1Click(Sender: TObject);
var
tir,n,i:integer;
begin
randomize;
tir:=StrToInt(Edit1.Text);
n:=StrToInt(Edit2.Text);
for i:=1 to n do
memo1.Lines.
Add(inttostr(random(tir)));
end;

```

Теперь давайте рассмотрим, как работает программа.

Преобразуйте числовой текст из полей ввода в числа и вставьте его в оператор цикла, работу которого мы будем рассматривать в этом параграфе. Достаточно скопировать циклический оператор в программе проекта. Работу с циклом продолжим в следующем параграфе.

Правильное создание формы и написание текста программы при выполнении практической работы требует определенного умения и навыков, а также большой внимательности.



Вопросы

1. Какую функцию в среде программирования выполняют компоненты визуальной среды?
2. Что вы можете сказать о компоненте RadioGroup и его свойствах?
3. Как вы опишете работу компонента CheckBox?
4. Какие свойства компонента Memo вы можете перечислить?
5. Какова функция компонента Image?
6. Как выбираются случайные числа?



Задание

1. Исследуйте компонент ListBox и его свойства.
2. Исследуйте компонент ComboBox и его свойства.
3. Исследуйте компонент Shape и его свойства.
4. Выберите компоненты и создайте свой проект.

РАЗДЕЛ IV. ИНТЕГРИРОВАННЫЕ СРЕДЫ РАЗРАБОТКИ ПРОГРАММ

4.3 ОПЕРАТОР ВЫБОРА



Как осуществляется работа алгоритма выбора в визуальной среде?



Подумай

- Как мы выбираем нужную вещь среди многих в повседневной жизни?



Новые знания

Оператор выбора

Условный оператор позволяет выбрать одно из нескольких возможных продолжений программы, образующих альтернативные варианты. Параметром, по которому осуществляется выбор, служит так называемый ключ выбора (или селектор) – выражение любого типа, кроме REAL и STRING.

Построение оператора выбора

Построение оператора выбора:
Case ключ выбора **of**
 ряд выбора
else оператор
end;
 case-случай, of – рабочие термины.

или



case выражение **of**
 1- константа: 1- оператор;
 2- константа: 2- оператор;
 ...
 n- константа: n- оператор
else оператор
end;

Работа оператора. Сначала вычисляется значение выражения, затем находится его константа. Соответственно, после константы выполняется работа назначенного оператора и заканчивается работа оператора выбора. Может случиться, что в списке выбора не окажется константы, равной вычисленному значению ключа. В этом случае управление передается оператору, стоящему за словом ELSE.

Пример. Ниже показан код модульной программы вывода названия месяца по порядковому номеру.

<pre>var n: byte; begin write ('введите число от 1 до 12'); read (n); case n of 1,2,12: Writeln ('Зима'); 3..5: Writeln ('Весна'); 6..8: Writeln ('Лето'); 9..11: Writeln ('Осень'); else writeln ('Не верно'); end; end.</pre>	или →	<pre>procedure TForm1; Button1Click(Sender: TObject); var Month: Integer; begin Month:=StrToInt(Edit1.Text); case Month of 1,2,12: ShowMessage('Зима'); 3..5: ShowMessage('Весна'); 6..8: ShowMessage('Лето'); 9..11: ShowMessage('Осень') else ShowMessage('выполняется выбор числа между 1; 12'); end; end.</pre>
---	----------	---

Попробуем выполнить это задание в визуальной среде Lazarus.

На рисунке 1 показан интерфейс проекта «**Времена года**», на следующем экране – результат проекта (рис. 2).

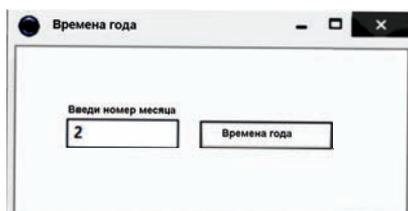


Рис. 1. Построение проекта



Рис. 2. Выполнение проекта

Применяется переключатель, выполняющий выбор из нескольких пунктов либо выбирающий опцию в среде Lazarus.

Переключатели типа **RadioButton** позволяют пользователю выбрать один параметр из двух или более предлагаемых вариантов. Каждый параметр представлен одним переключателем, пользователь может выбрать только один переключатель из группы.

Таблица 1. Свойства компонентов

Свойство компонента	Работа, значение
Caption	Выводит заголовок компонента.
Alignment	Определяет способ выравнивания текста. Может принимать 3 состояния taLeftJustify – по левому краю компонента, taCenter – выравнивание по центру и taRightJustify – выравнивание по правому краю.
Checked	Показывает использование кнопок: выбрано или нет. Если значение true , кнопка выбрана, false – не выбрана.

Компонента **CheckBox** – индикатор, с помощью которого пользователь либо подключает какую-либо опцию, либо отключает. Свойства этой компоненты похожи на свойства компонент **RadioButton**. Вышеописанные компоненты часто используются в программах тестирования.



Практическая работа

Используя компонент выбора, создайте небольшую программу по тестированию.

Создание формы проекта показано на рисунке 3, выполнение проекта – на рисунке 4. На схеме 2 даны свойства компонентов, использованных в интерфейсе проекта.

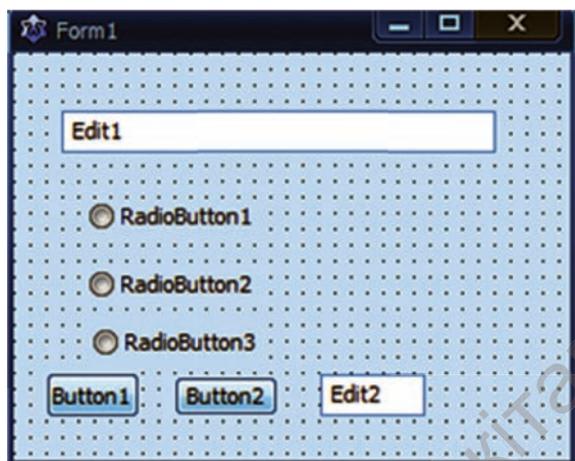


Рис. 3. Создание проекта



Рис. 4. Выполнение проекта

Схема 2. Свойства компонентов проекта

Выбранный компонент	Свойство и значение
Edit1	Свойство TEdit1 «Найти оператор ввода»
Edit2	Очищаем свойство TEdit2
Label1	Caption => Ответы
Button1	Caption => Ок
Button	Caption => Следующий
RadioButton	Caption => «Write»
RadioButton	Caption => «Read»
RadioButton	Caption => «Begin»

Код программы кнопки «Тест»

Procedure TForm1.Button1Click(Sender:TObject); {код кнопки «>»}.
begin

 if (n=1) and (RadioButton2.Checked) and (z=0) then z:=z+1;
 if (n=2) and (RadioButton1.Checked) and (z<=1) then z:=z+1;
 if (n=3) and (RadioButton2.Checked) and (z<=2) then z:=z+1;

end;

end.

procedure TForm1.Button2Click(Sender: TObject);

begin

Button1.Click;

inc(n);

if n=2 then begin

 edit1.text:= 'Найти оператор цикла с параметром';
 radiobutton1.Caption:= 'While';
 radiobutton2.Caption:= 'Case';
 radiobutton3.Caption:= 'Copy';

end;

if n=3 then begin

 edit1.text:= 'Назови функцию оператора DIV';
 radiobutton1.Caption:= 'Умножает';
 radiobutton2.Caption:= 'Деление нацело';
 radiobutton3.Caption:= 'Слагает';

end;

if n>3 then edit2.text:=inttostr(z);

end;

procedure TForm1.Button3 Click(Sender: TObject);

begin

```

Button1.Visible:=True;
Edit1.Visible:=True;
Edit2.Visible:=True;
GroupBox1.Visible:=True;
Button2.Visible:=True;
n:=1;
edit1.text:='Найти оператор ввода';
radiobutton1.Caption:='Write';
radiobutton2.Caption:='Read';
radiobutton3.Caption:='Begin';

```

```
end;
```

**Анализ**

Проведите анализ проекта «Тестовая программа». Обратите внимание на код и формат построения. Как бы вы изменили формат проекта?

**Синтез**

Создайте проект из повседневной жизни. В качестве примера возьмите стартап, придуманный школьниками, а также полученные за его реализацию денежные премии.

**Оценка**

Сравните и оцените алгоритмы разветвления и выбора.

**Вопросы**

1. Какова функция оператора Case of?
2. Что такое Выбор параметра?
3. Какое действие осуществляет компонента RadioButton?
4. Какое действие выполняет компонента CheckBox?
5. Какова функция свойств компонента Caption?
6. Назовите свойства компонентов выбора, какие типы компонентов вы знаете?
7. Каковы свойства и функции компонента Alignment, и когда он применяется?

8. Какие сходства и различия существуют между компонентами выбора в среде Lazarus и операторами выбора в языке Паскаль?
9. Что вы знаете о «синдроме компьютерного зрения»? Выполнение каких правил позволит защитить глаза во время работы на компьютере?



Задание

Создайте проект в программируемой среде Lazarus.

1. Напишите название месяца по порядковому номеру.

Ввод	Ответ
3	Наурыз

2. Напишите программу, которая по индексу автомобильных номеров выдает соответствующее название региона Казахстана.

Ввод	Ответ
01	Астана

3. В странах Азии календарь имеет 12-летний цикл, а годам соответствуют названия различных животных. Традиционный казахский календарь выглядит так: 1) крыса (тышқан); 2) бык (сиыр); 3) тигр (барыс); 4) заяц (қоян); 5) улитка (ұлу); 6) змея (жылан); 7) лошадь (жылқы); 8) овца (қой); 9) обезьяна (мешін); 10) петух (тауық); 11) собака (ит); 12) свинья (доныз). Напишите программу, которая по номеру года выдает название соответствующего животного на казахском и русском языках.

Ввод	Ответ
4	заяц (қоян)
27	тигр (барыс)

4. Вывод информации о количестве дней в месяце по его порядковому номеру. Напишите программу, учитывающую количество дней в феврале.

Ввод	Ответ
7	31
2	28

Создайте проект в визуальной среде.

Создайте проект – тест из 5 вопросов по информатике. В проекте должны быть отражены оценки.

РАЗДЕЛ IV. ИНТЕГРИРОВАННЫЕ СРЕДЫ РАЗРАБОТКИ ПРОГРАММ

4.4

ЦИКЛЫ С ПАРАМЕТРАМИ. ОПЕРАТОР FOR



Как применять циклы с параметрами в интегрированной среде обработки программ?



Подумай

- Что вы знаете о циклах?
- Какое отношение это понятие имеет к нашей жизни?
- Какое значение имеют циклы в природе? Приведите примеры.
- Как вы думаете, какую роль играют циклы в жизни человека?
- Где встречаются циклы в математике?



Новые знания

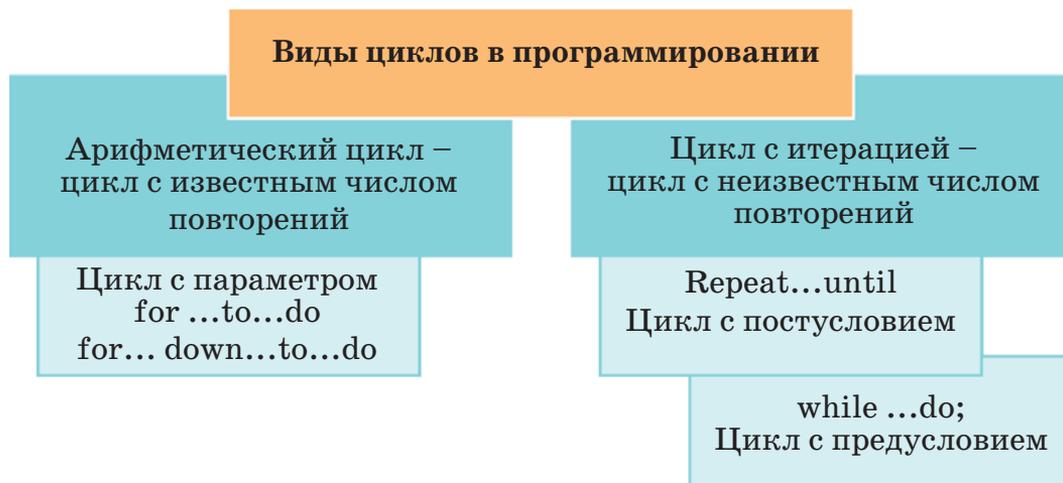
Циклы в программировании

Цикл – это повторение одних и тех же действий. Значение циклов в природе и жизни человека довольно велико. Известно множество циклических процессов. Например, времена года, дни, недели, месяцы – все это циклически повторяющиеся события.

Циклические процессы встречаются во всех сферах человеческой деятельности: в экономике (зарплата, пенсионные отчисления, пособия, стипендии); в истории (выборы депутатов в Мажилис, Сенат); в математике (сложение, умножение нескольких чисел); в медицине (ежегодный медосмотр); в военном деле (ежегодный набор на службу солдат) и т.д. В программировании, используя циклический алгоритм, можно разработать компьютерные модели циклических процессов в различных отраслях.

Последовательность действий, которые повторяются в цикле, называют телом цикла. Один проход цикла называется шагом или итерацией. Переменные, которые изменяются внутри цикла и влияют на его окончание, называются параметрами цикла. Существуют операторы, которые конвертируют алгоритм в язык программирования. Операторы циклов в программировании делятся на два вида (*схема 1*).

Схема 1



Оператор цикла с параметром FOR

Оператор **For** работает до достижения определенного значения, указанного пользователем. Если в алгоритме организовывается процесс с заранее известным числом повторений, то применяется оператор цикла с параметром, имеющий вид:

For i:=p to n do

<оператор или выражение>;

Например: **For i:=1 to 100 do**

Читается: значение *i* меняется с шагом 1 от 1 до 100.

Здесь: **FOR** (для), **TO** (к), **DO** (выполнение) – ключевые слова.

i – цикл параметра типа integer, longint, boolean

Существует два варианта использования оператора For:

1) цикл в котором значение параметра цикла увеличивается (*схема 2*);

2) цикл, в котором значение параметра цикла (*схема 3*) уменьшается.

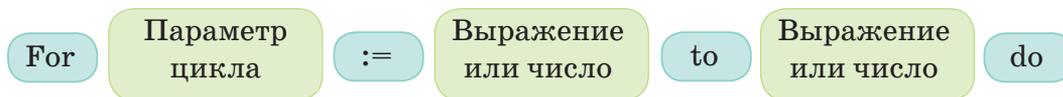
Для того чтобы понять работу цикла с параметром рассмотрим примеры.

Пример 1. Напишите программу вычисления суммы чисел от 1 до 100.

Пример 2. Выведите на экран числа от 1 до 10, записанные в обратном порядке.

При написании программы для первого примера используем первый вид цикла с параметром (схема 2).

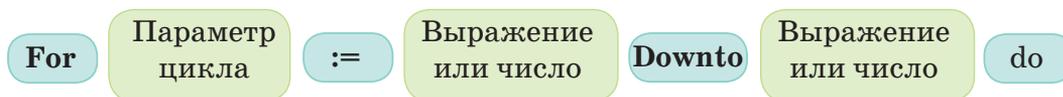
Схема 2



<p>Код примера 1: var s,i : integer; begin s:=0; for i:=1 to 100 do s:=s+i; writeln(s); end. Результат: 5050</p>	<p>Код примера 2: var i : integer; begin for i = 10 downto 1 do write (i, ' '); end. Результат: 10 9 8 7 6 5 4 3 2 1</p>
--	---

При написании программы для второго примера применим второй вид цикла с параметром (схема 3).

Схема 3



Если в процессе применения оператора For тело цикла содержит более одного оператора или выражения, то тело цикла заключается между операторами **begin...end**. Рассмотрим пример.

Пример 3. В 8 «А» классе учатся 20 учеников. Рост каждого ученика нам известен. Если он меньше 140 см, то ребенок считается невысоким. Определите сколько в классе высокорослых учеников.

Применение цикла с параметром в визуальной среде.

В решении задачи будем применять одновременно циклический и разветвленный операторы. Для того чтобы записать рост всех 20 учеников, размещаем внутри цикла оператор ввода, присваивающий на каждом шаге цикла переменной x рост ученика. В последующем шаге сравним рост ученика со значением 140 см. Как видим, в теле цикла работает не один, а два оператора. Для ввода двух операторов в тело цикла используем структуру **begin...end**. Здесь оператор значения z определяется выражением z:=z+1.

Код примера 3:
var
x, i, z: integer;
begin
for i:=1 to 20 do
begin
readln(x);
if x>=140 then z:=z+1;
end;
writeln(z);
end.

В среде программирования Lazarus можно создавать проекты с использованием циклов с параметром. В применении циклов в языке Паскаль и среде программирования Lazarus нет отличий. Свои особенности есть только в выводе на экран результата. На примере рассмотрим применение цикла с параметром в среде Lazarus.



Практическая работа

Пример 4. Создайте проект «Цикл», в котором значение параметра будет увеличиваться и уменьшаться.

В 7 классе мы уже познакомились с этапами создания такого проекта. В процессе выполнения задания подготовим форму, как показано на рисунке 1. На рисунке 2 продемонстрирован процесс выполнения проекта «Цикл». Выберем свойства (качества) и записи, соответствующие компонентам выполненного проекта, который состоит из кода программы, описывающей три кнопки **Button**. Тогда, используя код на языке Паскаль и переводя значение параметра цикла в текстовый формат, выводим результат на экран.

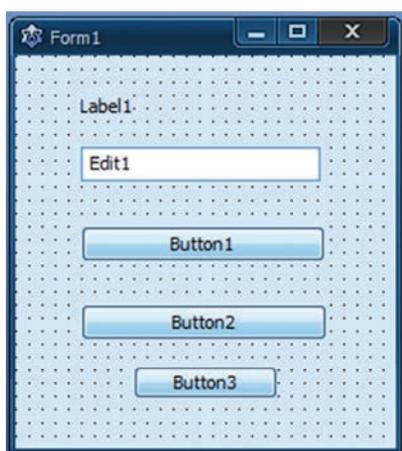


Рис. 1

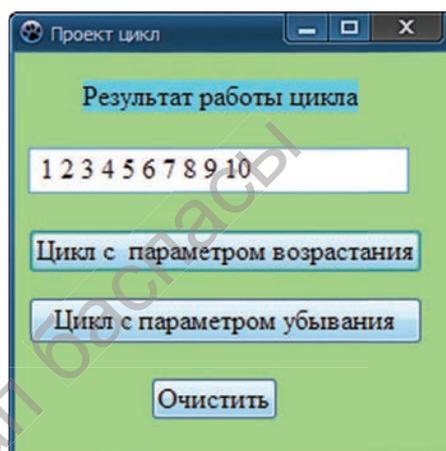


Рис. 2

```

procedure TForm1.
Button1 Click (Sender:
 TObject);
var i:integer;
s:string;
begin
for i:=1 to 10 do
s:=s+inttostr(i)+' ';
edit1.text:=s;
end;

```

```

procedure TForm1.Button2
Click (Sender: TObject);
var i:integer;
s:string;
begin
for i:=10 downto 1 do
s:=s+inttostr(i)+' ';
edit1.text:=s;
end;

```

```

procedure TForm1.
Button3
Click (Sender:
 TObject);
begin
edit1.Text:='';
end;

```

Рассмотрим еще один способ вывода результата на экран в визуальной среде с использованием цикла с параметром.

Пример 5. Разработайте проект «Сложение», выполняющий сложение чисел от 1 до 10.

Выполнение задания показано на рисунке 3, а вывод на экран — на рисунке 4. Для вывода результата на экран используется процедура **ShowMessage()**. Процедура **ShowMessage** выводит на экран текст и командную кнопку ОК. В коде программы используется прием перевода результата **S** в текст. Команда “**#13**” выполняет работу ключа.

```

ProcedureTForm1.Button1Click(Sender: TObject);

```

```

var
s: string; i,t:ShortInt;
begin
t:=0;
for i:= 1 to 10 do begin
s:=s+IntToStr
(t)+''+IntToStr(i)+'
='+intToStr(t+i)+ #13;
t:=t+i;
end;
ShowMessage(s);
end;

```

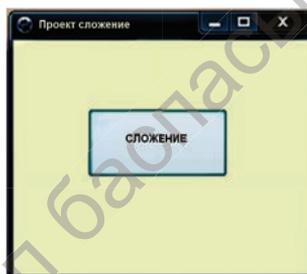


Рис. 3

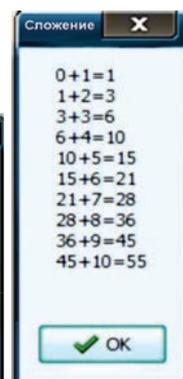


Рис. 4

Для остановки цикла и прекращения повторений внутри цикла с параметром используем команду **break**.

Таким образом, вы познакомились с работой цикла с параметром в модульной и визуальной средах, а также убедились, что написание программы и получение результата – работа сложная, но интересная.



Анализ



Каковы особенности использования цикла с параметром в модульной и визуальной средах? Назовите анализ. Назовите сходства и различия вывода результатов на экран.



Синтез



Составьте один пример с использованием цикла с параметром. Предложите код программы в модульной среде, а также в визуальной среде программирования Lazarus.



Оценка



Оцените важность операторов цикла для программирования.



Вопросы

1. Что такое цикл?
2. Какую роль играют циклы в жизни человека?
3. Какие бывают типы циклов в программировании?
4. При решении какого типа задач необходимо использовать циклические операторы?
4. Какие виды циклов с параметром вам известны?
5. Каковы различия в использовании циклов в модульной и визуальной средах?



Задание

1. Удвойте числа от 1 до 10 и выведите на экран.
2. Вычислите значение функции $y = 3x^2 - 4x + 2$ в интервале $[0...10]$ с шагом 1.
3. Постройте таблицу умножения на 4.
4. Из числового промежутка выведите те числа, в записи которых есть хотя бы одно нечетное число. Например, 12, 23, 98, 99.
5. Выведите на экран трехзначные числа, делящиеся на 3 и 9.
6. Уточните сколько нулей есть в числах от 1 до 100.

Подготовка проекта «Числовая последовательность» в визуальной среде

Подготовьте проект последовательности чисел по данному образцу (рис. 5):

1. Вставка кнопки, показывающей последовательность нечетных цифр в составе записи чисел от 10 до 20.
2. Вставка кнопки, показывающей последовательность четных цифр в составе записи чисел от 80 до 100.
3. Вставка кнопки, показывающей последовательность чисел Фибоначчи.

Числа Фибоначчи – элементы числовой последовательности, каждый последующий член которой равен сумме двух предыдущих. Элементы ряда Фибоначчи: 1, 1, 2, 3, 5, 8, 13, Рекуррентные соотношения чисел Фибоначчи:

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

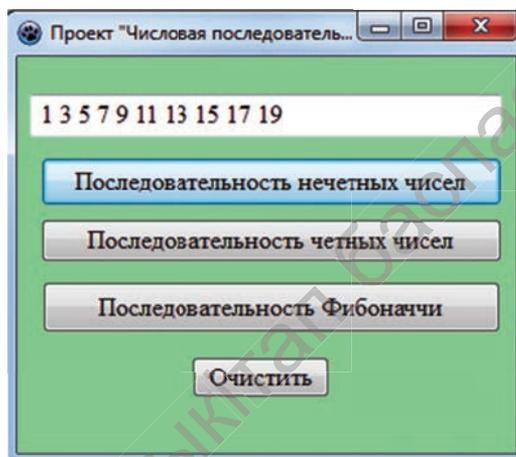


Рис. 5

РАЗДЕЛ IV. ИНТЕГРИРОВАННЫЕ СРЕДЫ РАЗРАБОТКИ ПРОГРАММ

4.5 ВЛОЖЕННЫЕ ЦИКЛЫ



Как применяются вложенные циклы при написании программ в интегрированной среде?



Подумай

- Что такое вложенные циклы?
- Какие преимущества дает использование вложенных циклов в процессе решения задач?

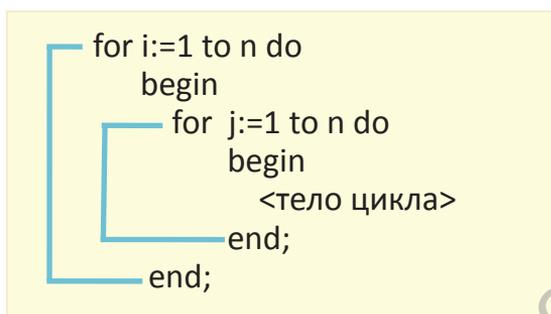


Новые знания

Вложенные циклы

Циклы называются вложенными, если они последовательно размещаются один внутри другого. Во вложенных циклах внутренний цикл должен быть полностью размещен во внешнем (*схема 1*).

Схема 1



где n – целое число.

Если в программе один цикл, то он выполняется n раз. Если 2 вложенных цикла, то эта группа выполняется $n*n$ раз. Например, показанный ниже цикл выполняется 100 раз.

```

for i:=1 to 10 do
for j:=1 to 10 do

```

Тело цикла по внешнему циклу выполняется один раз, затем по внутреннему циклу выполняется указанное количество раз. Увеличение количества вложенных циклов увеличивает и время работы программы. Например, если мы используем 3 вложенных цикла, то время работы будет равно n^3 . Увидеть это можно на следующем примере. Если $n=100$, то 3 вложенных цикла будут исполняться 1 000 000 раз.

```
for i:=1 to 100 do
  for j:=1 to 100 do
    for k:=1 to 100 do
```

Если на 1 000 000 итераций цикла требуется время 1 с, то при $n=1000$ будет уже миллиард итераций. На это потребуется время, равное 1 000 секунд. Следовательно, для выполнения одной задачи необходимо приблизительно 16 минут. Поэтому задачи должны быть корректно составлены.

С помощью вложенных циклов можно вывести цифры или символы на экран в табличном (матричном) виде. Вывод цифровой информации в **матричном виде** очень удобен.

Программа вывода на экран информации в матричном виде показана ниже. Например, матрица чисел размером 4x4 может быть выведена кодом следующей программы.

```
var
k,i,j: integer;
begin k:=0;
for i:=1 to 4 do
begin
for j:=1 to 4 do
begin inc(k);
write(k, ' ');
end;
writeln;
end;
end.
```

Как видно из кода программы, при выполнении внешнего цикла 1 раз, внутренний цикл выполняется 4 раза. Во внутреннем цикле значение **K** **возрастает** каждый раз на **один**. По окончании внутреннего цикла курсор беспараметрного оператора **writeln** спускается на один ряд и готовит вывод на экран следующего шага. Результат программы выводится в следующем виде:

```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

Стандартные ошибки при написании вложенных циклов:

1. Запись одинаковых параметров как во внутреннем, так и во внешнем циклах;
2. При построении конструкции **begin...end** ошибочный либо лишний оператор может не войти во вложенный цикл;
3. Ввод реальных чисел в параметры цикла.

Рассмотрим работу вложенных циклов на практике.

**Практическая работа****Задание. Простые числа**

Дана последовательность чисел длиной N ($N < 100$). Требуется вывести на экран все простые числа последовательности. Числа разделить между собой одним пробелом.

Пример	Результат
10 5 45 98 91 300 197 77 87 53 47	5 197 53 47

Алгоритм задачи

Для решения задачи необходимо использовать вложенный цикл. **Простое число** – натуральное число, которое делится только на единицу и на самого себя. Для его определения извлечем квадратный корень из этого числа и округлим до целого. Первый цикл используем для ввода числа, второй – для проверки условия простого числа. Параметр внутреннего цикла изменяется от 2 до округленного целого числа с шагом 1. Если среди этих чисел нет делителей заданного числа, то оно является простым.

Примечание. Ввести все заданные числа в строку через пробел.

```
var
n,i,j,k,y:longint; x:real;
begin
readln(n);
for i:=1 to n do
begin
read(y); k:=0; x:=sqrt(y);
for j:=2 to round(x) do
if y mod j= 0 then k:=1;
if k=0 then write(y, ' ');
end;
end.
```

round(x) – округляет число до целого значения. Здесь **x** – вещественное число.

Примеры:

round(23.5)=24

round(23.14)=23

sqrt(y) – извлекает квадратный корень из **y**.

round(sqrt(18))=4

round(sqrt(23))=5

Создание проекта «Простые числа» в визуальной среде

Выполни задание в визуальной среде Lazarus. Для ввода чисел воспользуемся функцией **InputBox**, с которой вы познакомились в 7 классе.

Как показано на рисунке 1, проект начинаем с создания формы. На рисунке 2 показано выполнение проекта «Простые числа», в котором будут указаны свойства компонентов и код программы. Код программы включает кнопку **Button**, функцию **InputBox**, чем и отличается от вывода на экран в языке программирования Паскаль (рис. 3).

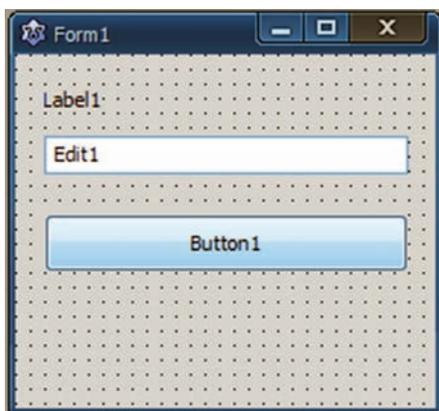


Рис. 1

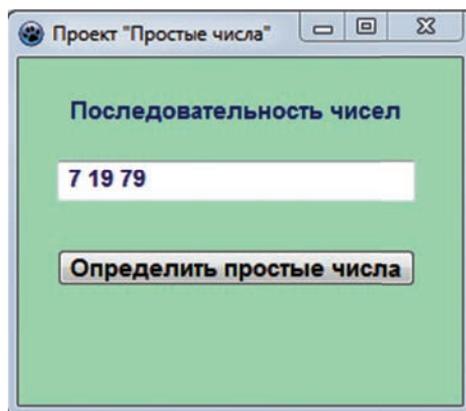


Рис. 2

```

procedure TForm1.Button1Click(Sender:
TObject);
var
l,s:string;
i,j,x,k,n,g:integer;
begin s:=inputbox ('Длина
последовательности', 'Ввод длины
последовательности', "");
val(s,n,g );
for i:=1 to n do
begin
s:=inputbox ('Члены
последовательности', 'Ввод членов
последовательности', "")
val(s,x, g ); k:=0;
for j:=2 to round(sqrt(x)) do
if x mod j=0 then k:=1;
if k=0 then l:=l+inttostr(x)+' ';
end;
edit1.text:=l;
end;

```

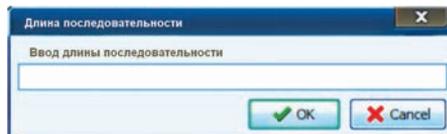


Рис. 3

Оператор Val – преобразовывает значение строки в числовое представление.
Val (s, x, p) где
s: String; X: real или integer;
p: integer;
 Переменная **S** строкового типа, **X** – переменная типа **real** или **integer**, **p** – числовой тип **integer**

**Анализ – синтез**

Проанализируйте роль вложенных циклов при создании проекта «Простые числа» в модульной и визуальной средах. Покажите сходство и различие двух программ. Как еще можно создать проект «Простые числа»?

**Оценка**

Оцените полезность вложенных циклов для решения задач.

**Вопросы**

1. Что такое вложенные циклы?
2. Как можно подсчитать время работы вложенных циклов?
3. Какие трудности связаны с увеличением количества вложенных циклов?
4. Как вывести результат на экран в табличном (матричном) виде?
5. Какие распространенные ошибки встречаются при использовании вложенных циклов?

**Задание**

1. С помощью вложенных циклов выведите на экран матрицу с $n=5$:

1)	2)	3)	4)	5)
1 1 1 1 1	1 1 1 1 1	1 1 1 1 1	1 0 0 0 0	0 0 0 0 1
1 1 1 1 1	0 0 0 0 0	2 2 2 2 2	0 1 0 0 0	0 0 0 1 0
1 1 0 1 1	1 1 1 1 1	3 3 3 3 3	0 0 1 0 0	0 0 1 0 0
1 1 1 1 1	0 0 0 0 0	4 4 4 4 4	0 0 0 1 0	0 1 0 0 0
1 1 1 1 1	1 1 1 1 1	5 5 5 5 5	0 0 0 0 1	1 0 0 0 0

2. Приведены фрагменты программы. Какое значение получит x после завершения программы?

```
1)
x:=0;
for i:=1 to 5 do
for k:=i+1 to 5 do
x:=x+1;
```

```
2)
x:=0;
for i:=1 to 5 do
for k:=i+1 to 5-i do
x:=x+1;
```

```
3)
x:=0;
for i:=1 to 5 do
for k:=i-1 to i+1 do
x:=x+k;
```

```
4)
x:=10;
for i:=1 to 3 do
for k:=0 to i do
x:=x+(k-i);
```

3. Допишите программу и выведите результат.

```
1)
for i:=1 to 4 do
begin
if i mod 2 = 0 then
n:=i+1
else
n:=i;
for k:=1 to n do
write(n-i, ' ');
writeln;
end;
```

```
2)
for i:=1 to 3 do
for k:=3 downto 1 do
for n:=i-k to (i+k) div 2 do
write(n, ' ');
```

4.6

ЦИКЛ С ПОСТУСЛОВИЕМ. ОПЕРАТОР REPEAT-UNTIL



Как применяется цикл с постусловием в интегрированной среде разработки программ?



Подумай

- Приведите примеры из повседневной жизни, когда количество повторяющихся событий неизвестно?
- Что такое цикл с постусловием?



Новые знания

Цикл с постусловием

Цикл с постусловием – цикл, в котором условие проверяется после выполнения тела цикла. Как вы уже знаете, если известно количество повторений, то применяется оператор цикла **for**. В противном случае, применяются операторы **repeat** или **while**. Оператор **Repeat** называется циклом с постусловием, оператор **While** – циклом с предусловием.

Оператор Repeat-until

Общий вид цикла с условием **Repeat-until** показан на схеме 1. Результатом выражения будет тип **boolean** или логический тип данных. Операторы, заключенные между словами **repeat** и **until**, являются *телом цикла*. Сначала выполняется тело цикла, затем проверяется условие выхода из цикла. Поэтому цикл, организованный с помощью оператора **repeat**, в любом случае выполнится хотя бы один раз. Если результат логического выражения равен **False**, то тело цикла выполнится еще раз; если результат равен **True**, то цикл остановится и произойдет выход из цикла.

Repeat

```
<оператор 1>;
< оператор 2>;
<оператор 3 >;
...
< оператор n >;
until (выражение);
```

Схема 1

Закрепим на примерах работу оператора Repeat-until. Для этого рассмотрим пример 1 из главы Циклы с параметрами. Можно показать работу операторов for и repeat при решении одной и той же задачи.

Пример 1. Напишите программу вычисления суммы чисел в интервале от 1 до 100.

Код примера 1:

```
var
  i,s: integer;
begin
  s:=0;i:=0;
  repeat
  inc(i);
  s:=s+i;
  until(i>100);
  write(s);
end.
```

1. Обратите внимание на различие в работе оператора **Repeat-until** от оператора **for**.
2. В цикле шаг **i** имеет важное значение. Например, если в примере 1 мы каждый раз будем увеличивать значение **i** не на единицу, а на 2, то легко получим сумму всех четных чисел до 100.
3. Работа цикла не всегда зависит от одного условия. Как показано в примере 2, в тело цикла можно вставить несколько условий одновременно, а также с помощью других дополнительных операторов прервать работу цикла.

Пример 2. Два брата Руслан и Тимур с сестрой Дианой пошли в сад собирать яблоки. Через некоторое время они посчитали урожай и получили соответственно N, M и K яблок. Ребята решили устроить соревнование. Тот, кто соберет P яблок, прекращает работу и идет домой. Скорости сбора яблок соответственно X, Y и Z яблок в минуту. Кто из ребят первым закончит сбор и уйдет домой? Возможно, они закончат работу одновременно.

№	Пример	Результат
1	5 3 6 40 4 2 8	3
2	5 5 5 30 4 4 4	1 2 3
3	4 7 9 20 16 3 11	1 3

В задаче скорость выполнения работы можно рассматривать как циклический процесс. Как мы видим, и в операторе `until` записано составное условие.

Цикл остановится, когда количество яблок у одного из ребят достигнет значения P . Достаточно после окончания цикла уточнить, у кого из ребят количество яблок первым достигло условного предела.

Код примера 2:

```
var
n,m,k,x,y,z,i,p: integer;
begin
readln(n,m,k,p);
readln (x,y,z);
repeat
inc(n,x);
inc(m,y);
inc(k,z);
until(n>=p)or(m>=p)or (k>=p);
if n>=p then write(1, ' ');
if m>=p then write(2, ' ');
if k>=p then write(3);
end;
end.
```



Практическая работа

Применение цикла с постусловием в визуальной среде

Давайте познакомимся с применением цикла с постусловием в программируемой среде Lazarus.

Для этого научимся выполнять проект, показанный ниже.

Проект «Исследование числа»

Дано натуральное число N ($0 \leq N \leq 10^9$). Создайте проект нахождения суммы и произведения цифр этого числа.

На рисунке 1 приведена форма проекта «Исследование числа». На рисунке 2 показано выполнение проекта. Описываются соответствующие свойства используемых компонентов проекта. В коде все действия реализуются через кнопку **Button**.



Рис. 1

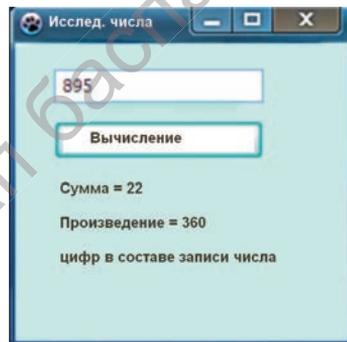


Рис. 2

Код проекта «Исследование числа»

```

procedure TForm1.Button1Click(Sender: TObject);
var t,n,s,l:longint;
begin
  n:=strtoint(edit1.text);
  l:=1;
  s:=0;
repeat
  t:=n mod 10; {выделяет последнюю цифру числа}
  n:=n div 10; {убирает выделенную цифру числа}
  s:=s+t; {отделенную цифру добавляет в сумму}
  l:=l*t; {отделенную цифру домножает в произведение}
until n=0; {проверяет логическое выражение}
label1.Caption: = 'Сумма='+inttostr(s);
label2.Caption: = 'Произведение='+inttostr(l);
end;

```



Анализ



Сравните работу цикла с параметром и цикла с постусловием в модульной и визуальной средах программирования.



Вопросы

1. Что такое цикл с постусловием?
2. В чем суть работы оператора Repeat–until?
3. Когда прекращает работу цикл с постусловием?
4. Каковы различия использования цикла с постусловием в модульной и визуальной средах?



Задания

Задание № 1

Какое значение примет x после выполнения фрагмента программы?

1)

```

x:=0;
Repeat
inc(x);
write(sqr(x), ' ');
until (x=5);

```

2)

```

x:=0; t:=10;
Repeat
inc(x);dec(t);
p:=p+x+t;
until (x=5);

```

3)

```
x:=0;
Repeat
inc(x);
s:=s+x;
until (s>20);
write(x, '');
```

4)

```
x:=1; z:=5;
Repeat
inc(x,4);inc(z);
write(x+z);
until (x=5)or (z=8)
```

Задание № 2

Задача № 1. Квадрат числа

Требуется вывести на экран квадраты натуральных чисел, не превышающие заданное число N , в порядке возрастания.

Ввод	Результат
50	1 4 9 16 25 36 49

Задача № 2. Последовательность степени числа 2

Требуется вывести на экран степени числа 2, не превышающие заданное число N , в порядке возрастания.

Ввод	Результат
50	1 2 4 8 16 32

Задача № 3. Спортивная семья

У Тимура все члены семьи любят спортивный бег. Отец, мать и Тимур каждое утро делают пробежки. В первый день все дружно пробежали X километров. В последующие дни они увеличивали длину пробега на 10% каждый день. Через сколько дней утренний пробег семьи превысит Y км?

Ввод	Результат
10 20	9

Задание № 4. Количество нулей

Дано натуральное число N . Определите количество нулей в этом числе.

Ввод	Результат
7085600	3

Задание № 5. Среднее арифметическое значение чисел

Дано натуральное число N . Определить количество цифр в числе и среднее арифметическое значение всех цифр в его записи.

Ввод	Результат
4585625	7 5

Задание № 3

Создание проекта в среде Lazarus.

Проект «Возведение в степень числа 2»

Подготовьте проект возведения числа 2 в степень N . На рисунке 3 представлен пример проекта.

Проект «Наибольшая цифра числа»

Подготовьте проект нахождения наибольшей цифры в составе записи числа N . Подготовьте проект вычисления n -й степени числа. На рисунке 4 приведен пример проекта.

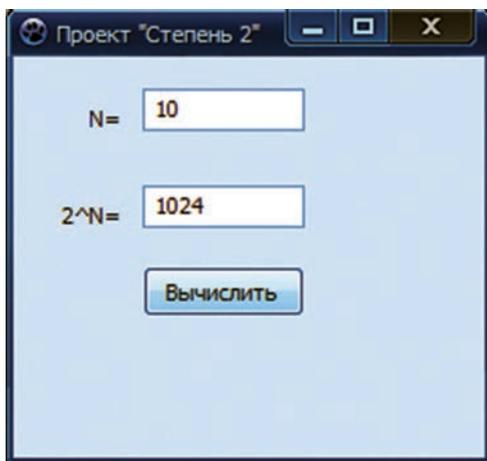


Рис. 3

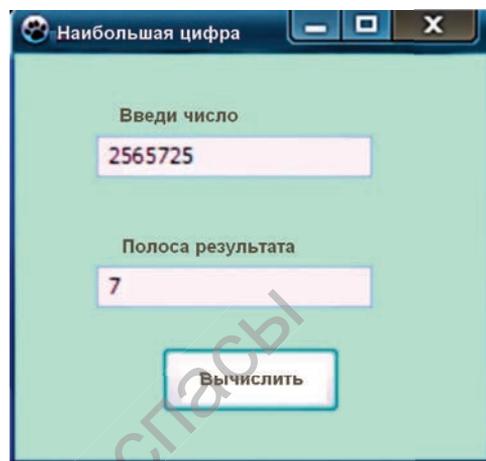


Рис. 4

4.7

ЦИКЛ С ПРЕДУСЛОВИЕМ. ОПЕРАТОР WHILE



Как применяется цикл с предусловием в интегрированной среде разработки программ?



Подумай

- Что такое цикл с предусловием?
- Каковы различия между циклами с постусловием и предусловием?



Новые знания

Оператор цикла с предусловием **While–do**

Цикл с предусловием определяется оператором **While–do** (Схема 1). При выполнении этого оператора проверяется значение условия. Если условие истинно, то тело цикла оператора **while** будет выполняться до тех пор, пока это условие не примет значение **False**. Если значение условия будет **False** изначально, то тело цикла **while** не выполнится ни разу.

Значение условия проверяется до выполнения тела цикла. Если логическое выражение имеет значение **true**, то цикл продолжит работу, если **false**, то цикл прекращает работу.

While (пока) – зарезервированное слово; <условие> – логическое выражение. Если тело цикла состоит из более чем одного оператора, то они применяется в структуре **begin–end**.

Рассмотрим примеры для понимания работы оператора **While–do**. Для этого еще раз рассмотрим пример 1 главы, посвященной работе цикла с параметром и цикла с постусловием. Вы можете увидеть различия в работе трех операторов при решении одной и той же задачи.

```
While <шапт> do
```

```
begin
```

```
<оператор 1>;
```

```
< оператор 2>;
```

```
...
```

```
< оператор n >;
```

```
end;
```

Пример 1. Напишите программу вычисления суммы чисел в интервале от 1 до 100.

Код 1-го примера:

```
var
i,s: integer;
begin
s:=0;i:=0;
while i<100 do
begin
inc(i);
s:=s+i;
end;
write(s);
end.
```

Различия между операторами **While–do** и **Repeat–until**:

1. Проверяется условие до выполнения тела цикла.
2. Если в теле цикла имеется более одного оператора, то они записываются в структуре **begin–end**.

Сходство в работе операторов:

Оба оператора применяются при решении задач с неизвестным количеством циклов.

Пример 2. Требуется определить количество четных цифр в записи натурального числа N.

Пример	Результат
26587458	5

```
Var
z,k,n: Longint; {число целого типа}
Begin
Readln (n);
While n<> 0 Do{пока n<>0, выполнение}
Begin
k:=n mod 10; {отделяет последнюю цифру числа}
if k mod 2 = 0 then Inc (z); {z := z + 1}
n:= n div 10; {удаляется последняя цифра}
end;
Writeln(z); {вывод количества четных цифр}
End.
```



Практическая работа

Применение цикла с предусловием в визуальной среде

Ознакомимся с применением цикла с предусловием в программируемой среде Lazarus. Для этого научимся строить проект, приведенный ниже. Расчет факториала выполняем созданием проекта с использованием операторов For-to-do, Repeat-until и While-do. При решении одной задачи при помощи одного из операторов цикла, можно легко понять, в чем их различия.

Проект «Факториал»

Дано натуральное число N ($1 \leq N \leq 12$). Выполните проект расчета факториала данного числа с применением операторов for, repeat-until и while-do. $N! = 1 * 2 * \dots * (N-1) * N$
 Например, $4! = 1 * 2 * 3 * 4 = 24$

На рисунке 1 показана форма проекта «Факториал». На рисунке 2 – выполнение проекта. Описываются соответствующие свойства используемых компонентов проекта. В коде проекта реализуется решение задачи с помощью кнопок **Button 1** и **Button 2**.

Компонент	Свойство	Суть свойства
Form1	Caption	Факториал
Label1	Caption	Значение числа N
Label2	Caption	N факториал
Button1	Caption	Расчет
Button2	Caption	Очищение
Edit1	Text	Пусто
Edit2	Text	Пусто
RadioGroup1 –	Caption	Виды циклов
	Items	For-to-do Repeat-until While-do

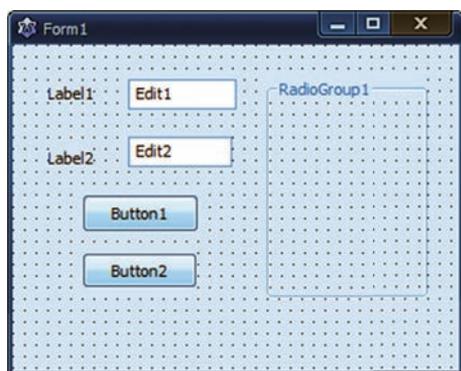


Рис. 1

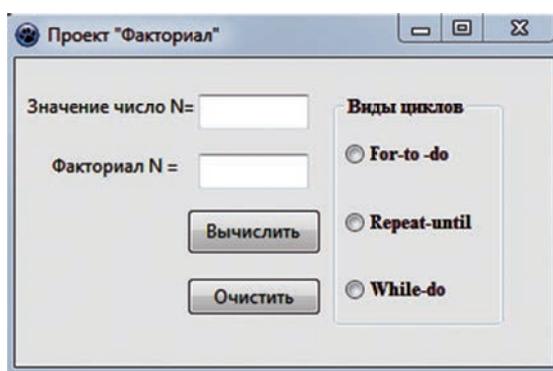


Рис. 2

Код проекта «Факториал»

```

procedure
 TForm1.Button1Click(Sender:
 TObject);
var
 p,k,j,i, c,n,t: longint;
begin
 c:=1;
 j:=0;
 n:=strtoint(edit1.text);
 case RadioGroup1.ItemIndex of
 0:for i:=1 to n do c:=c*i;
 1: repeat
 inc(j);
 c:=c*j;

```

```

until (j=n) ;
 2: while(k<n) do
 begin
 inc(k);
 c:=c*k;
 edit2.text:=inttostr(c);
 end;
 end;
 procedure TForm1.Button2
 Click (Sender: TObject);
 begin
 edit1.text:='';
 end;

```



Анализ



Сравните работу операторов for, repeat-until и while-do в модульной и визуальной средах программирования. Проанализируйте различия и сходство в выполнении работы.



Синтез



Придумайте по 2 задачи для каждого из операторов for, repeat-until и while-do. Сделайте заключение о работе операторов цикла по этой задаче.

**Оценка**

Анализ мнения. Какой из операторов for, repeat-until и while-do наиболее эффективен? Есть ли возможность заменить все три оператора одним? Если бы нужно было сократить количество операторов цикла, то какой из операторов вы удалили бы?

**Вопросы**

1. Что такое цикл с предусловием?
2. В чем суть работы оператора while-do?
3. При каких условиях останавливается цикл с предусловием?
4. Каковы различия использования цикла с постусловием в модульной и визуальной средах?

**Задания****Задача № 1. Наибольшее и наименьшее число**

Дано натуральное число N. Требуется найти и вывести на экран наибольшую и наименьшую цифру в записи этого числа.

Ввод	Результат
70856954	9 4

Задача № 2. Запись числа в обратном порядке

Дано натуральное число N. Требуется вывести на экран в обратном порядке все цифры в записи этого числа.

Ввод	Результат
152895	5 9 8 2 5 1

Задача № 3. Является ли заданное число палиндромом?

Если в данном трехзначном числе N первая и последняя цифры одинаковы, то, несмотря на значение средней цифры, оно будет палиндромом. Палиндром – число, одинаковое читающееся в обоих направлениях. Например, 858, 191 т.б.

Ввод	Результат
585	Да
956	Нет

Задача № 4. Числа больше K

Дана последовательность из N целых чисел ($N < 100$) и число K ($-100000 < K < 10000$). Требуется найти среди членов последовательности N числа больше K .

Ввод	Результат
10 236 1250 9683 321 -25 658 12 0 245 489 -125	6

Проект «Представление чисел в различных системах счисления»

Подготовьте проект перехода из десятичной записи натурального числа N в двоичную систему счисления. Пример проекта приведен на рисунке 3.

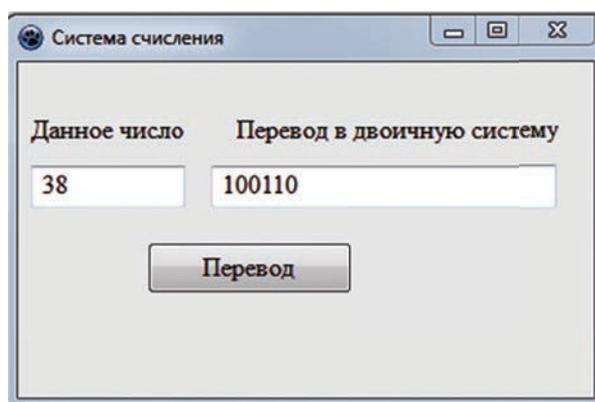


Рис. 3

4.8 ТРАССИРОВКА АЛГОРИТМА



Как осуществляется трассировка алгоритма программы?



Подумай

- Что такое алгоритм?
- Каково значение алгоритма в программировании?
- В чем заключается необходимость пошагового выполнения программы?



Новые знания

Трассировка программы

В 7 классе вы познакомились с ошибками, встречающимися в процессе программирования. Вспомним их.

Ошибки, допускаемые программистами в процессе составления программ:

1. Синтаксические ошибки;
2. Связи со временем выполнения программы;
3. Алгоритмические ошибки.

- Как выявить эти ошибки в процессе программирования?
- Как пошагово исправлять подобные ошибки в процессе написания программы?

При запуске программ, команды выполняются одна за другой со скоростью работы процессора компьютера. Программист не может уследить за тем, какое выполняется действие в данный момент, а также проконтролировать обработку алгоритма. Для того чтобы найти ошибку в программе, нужно знать точный порядок выполнения действий. Это называется трассировкой программы, или ее пошаговым (step-by-step) выполнением. При трассировке программист может выполнять каждую команду в программе по отдельности и найти ошибку. В среде Lazarus есть два способа трассировки: один из них – исправление ошибки без захода в процедуру **Шаг в обход** (Step

over), второй – с заходом в процедуру **Шаг со входом** (Trace into) (рис. 1). Трассировка без входа в процедуру выполняет только основную часть программы, трассировка сопровождающих (дополнительных) программ не выполняется, все они выполняются одним шагом. Во втором случае выполняется трассировка всех программ, а не только пошаговая проверка основной программы.

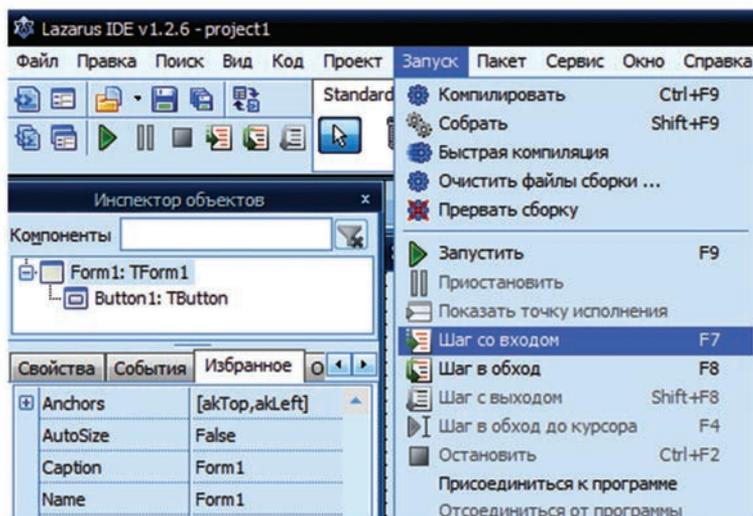


Рис. 1

Для выполнения трассировки необходимо выбрать из меню **Запуск** команду **Step over** или **Trace into**. В результате в окне редактора выделяется первое указание сопровождающей программе. Чтобы выполнить выделенную команду, из меню **Запуск** выберите команду **Step over** или **Trace into**. После выполнения команды выделяется следующее указание.

Трассировку можно активировать с помощью функциональных клавиш. Команде **Step over** соответствует клавиша <F8>, а команде **Trace into** – клавиша <F7>.

В любой момент времени можно остановить трассировку и продолжить выполнение программы. Для этого из меню **Запуск** нужно выбрать команду **Запустить**.

Если необходимо выполнить трассировку части программы, нужно установить курсор в той строке, с которой нужно начать проверку, и из меню **Запуск** выбрать команду **Run to cursor** или нажать клавишу <F4>. После этого, при помощи клавиши <F7> или <F8> можно выполнить трассировку необходимого фрагмента программы.

Теперь рассмотрим конкретный пример. В процессе программирования зададим форму с кнопкой (рис. 2).

Нажимая на кнопку, откроем поле для кода программы и впишем значение $k = 10$ (рис. 3).

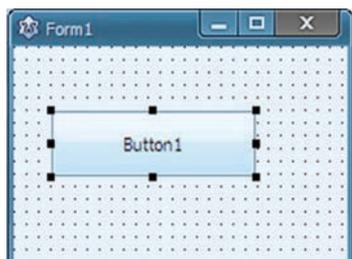


Рис. 2

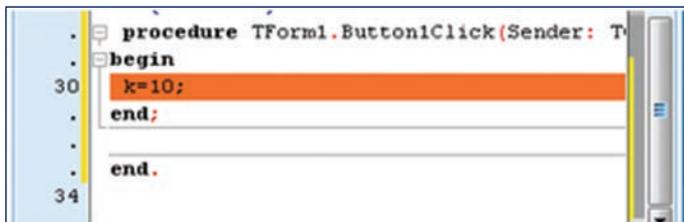


Рис. 3

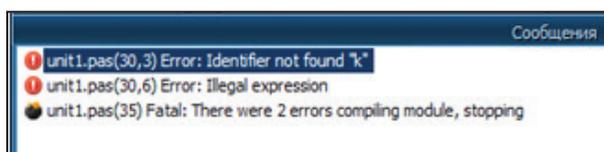


Рис. 4

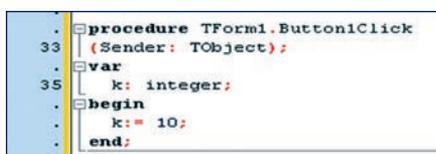


Рис. 5

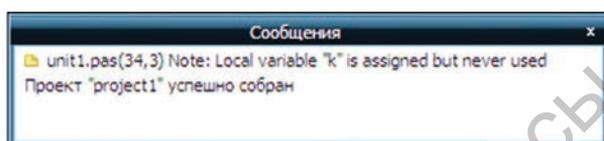


Рис. 6

Как видно из примера, при выполнении программы была допущена ошибка на 3 позиции 30-го ряда: **Identifier not found «k»** (переменная k не описана), а на 6-й позиции 30-го ряда по правилам программирования должен быть оператор присваивания (рис. 4). Для исправления ошибок опишем переменную k в разделе **var** и запишем знак присваивания "==" (рис. 5). После исправления ошибок появится сообщение «Проект «...» успешно собран» (рис. 6).

В основном, описанные здесь ошибки возникают в процессе написания программы из-за незнания операторами принципов работы и синтаксиса. К трудно обнаруживаемым ошибкам, допускаемым при написании программ, относятся алгоритмические, то есть ошибки допускаемые при построении логики программы. Для обнаружения ошибок в программе, на каждом шаге ее выполнения важно знать значения используемых переменных. Для этого в среде Lazarus предусмотрены различные инструменты. В меню программы выполните команды **Вид** ⇒ **Окна отладки**, откройте дополнительное окно (рис. 7): рекомендуем использовать окна **Окно наблюдения** и **Точки останова**.

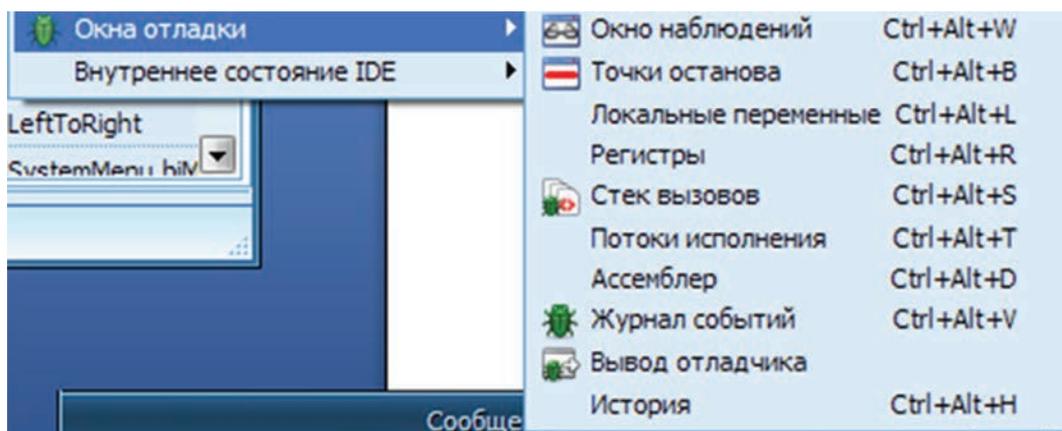


Рис. 7

Рассмотрим на примерах, как работают эти два окна. Для обнаружения допущенных в программе логических ошибок ведется наблюдение за переменными, используемыми в цикле. Из-за того, что сразу несколько переменных в теле цикла одновременно изменяют значение, эти изменения увидеть сложно. Исправить пошагово вручную еще сложнее. Одним словом, вычислить большое количество циклов с большим количеством переменных очень непросто.

- Как контролировать значения переменных и результат обработки программы в визуальной среде?



Практическая работа

Рассмотрим пример.

Нахождение суммы цифр натурального числа N.

Программа примера представлена на рисунке 8. Для трассировки мы выбрали 40-ю, 41-ю и 42-ю строки программы, они выделены красным цветом.

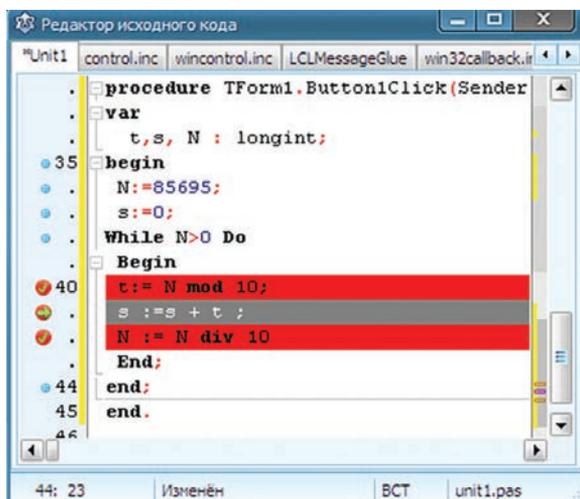


Рис. 8

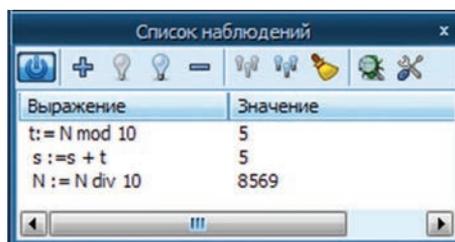


Рисунок 9

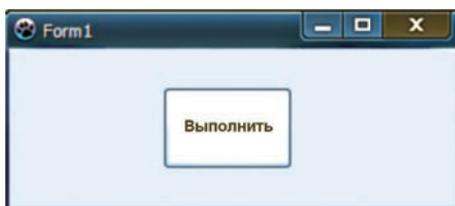


Рис. 10

В окне **Списка наблюдений** вводим подлежащие проверке формулы 40-й, 41-й и 42-й строк (рис. 9). Для наблюдения за выполнением этих формул нажимаем на кнопку «Запустить». В окне выполнения программы нажимаем на кнопку «Выполнить» (рис. 10). Для пошагового наблюдения за выполнением вычислений формул, расположенных в теле цикла, нажимаем на кнопку «Trace into» или <F7>, и следим за изменением значения формул. На рисунке 11 видно, что каждая проверяемая формула работает правильно. Поэтому в строках 40, 41 и 42 указано состояние **Enabled**.

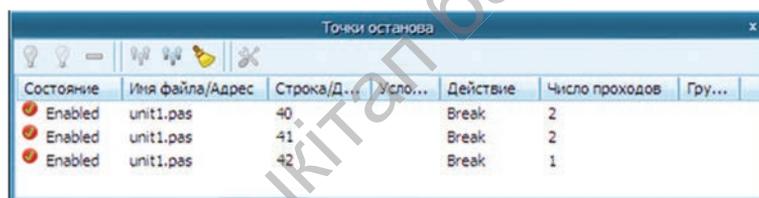


Рис. 11

Мы с вами познакомились с проверкой и отладкой программы в визуальной среде. Подобным образом осуществляется проверка и отладка программы в любой среде программирования.



Вопросы

1. Какие виды ошибок встречаются в процессе программирования?
2. Что такое трассировка?
3. Какие виды трассировки вы знаете?
4. Какова роль Окна наблюдений и Точки останова?
5. В каком пункте меню они расположены?
6. Объясните принцип выполнения алгоритма трассировки в визуальной среде программирования.



Задания

Создайте проект в визуальной среде

1. Найдите сумму цифр числа, расположенного между К и Р. В процессе выполнения задания выполните трассировку формул, расположенных в теле цикла.

Ввод	Результат
97 103	61

2. Составьте программу перевода из десятичной системы счисления в восьмеричную. Исправьте ошибки самостоятельно. Понаблюдайте за работой тела цикла.

Ввод	Результат
87	127

3. Выявите ошибки, допущенные в кодах программ, приведенных ниже.

```
var s:longint;
begin
x:=0;
Repeat
inc(x);
s=s+x;
until(x=5);
```

```
var s:longint;
begin
readln(n);
while n>0
begin
s:=s+ n mod 10;
n:=n div 10
write (s);
end.
```

```
var i,s:longint;
begin
readln(n);
for i:=10 to 1 do
if i mod 2 =0 then s:=s+i;
write (s);
end;
end.
```

4.9

ТВОРЧЕСКО-ПРАКТИЧЕСКАЯ РАБОТА ПО ТЕМЕ ЦИКЛА



Как можно использовать циклы в решении задач?



Подумай

- Каковы оптимальные методы решения задач с циклами?



Практическая работа

Задания по теме «Циклы с параметрами»

Задание № 1. Одинаковые цифры*

Задан интервал чисел от 10 до N ($10 \leq N \leq 99$). Требуется вывести на экран числа, у которых все цифры в записи одинаковые.

№	Пример	Результат
1	34	11 22 33

Задание № 2. Различные цифры**

Задан промежуток чисел от 100 до N ($100 \leq N \leq 999$). Требуется вывести на экран все трехзначные числа, у которых все цифры в записи разные.

№	Пример	Результат
1	107	102 103 104 105 106 107

Задание № 3. Выбор***

Среди восьмиклассников идет отбор для участия в соревнованиях по баскетболу. Количество восьмиклассников N ($130 \leq N \leq 200$), а критерий отбора – рост выше K сантиметров. Необходимо определить количество восьмиклассников, выбранных для участия в соревнованиях по баскетболу и порядковый номер самого высокого ученика.

№	Пример	Результат	Пояснение
1	7 170	3	Количество учащихся с ростом более 170 см – 3. Самый высокий из них – 2-й по порядку.
	165 180 171 135 143 175 169	2	

Задание № 4. Площади прямоугольников**

Даны N прямоугольников с длиной сторон a ($1 \leq a \leq 20000$) и шириной сторон b ($1 \leq b \leq 20000$), где N лежит в интервале $1 \leq N \leq 20$. Необходимо вывести на экран значение площади наибольшего прямоугольника и его порядковый номер.

№	Пример	Результат
1	3 4 3 6 5 9 2	30 2

Задание № 5. Все представления четырехзначного числа***

Найдите общее количество четырехзначных чисел, у которых в записи все цифры разные (с использованием вложенного цикла).

Задание № 6. Матричная таблица**

Составьте таблицу размером $N \times N$ из числа N ($1 < N < 20$).

№	Пример	Результат
1	4	0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0

Задание № 7. Таблица**

Выведите на экран числа в указанном порядке по образцу, показанному ниже, где N – натуральное число ($1 < N < 20$).

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

Дескрипторы выполнения заданий:

- читает задание, понимает смысл;
- строит математическую модель задачи;
- записывает программы модульного программирования;
- создает проекты в среде Lazarus (для учащихся, занимающихся по углубленному курсу).

Задания с применением циклов с постусловием и предусловием**Задание № 8. 4 цифры***

Найти количество цифр 4 в заданном числе N ($4 \leq N \leq 10^9$).

№	Пример	Результат
1	85472224	2

Задание № 9. Цифры 2 и 5 *

Из заданного числа N ($5 \leq N \leq 10^9$) запишите числа, в записи которых присутствуют только цифры 2 и 5, в обратном порядке.

№	Пример	Результат
1	54522158	52255

Задание № 10. Одинаковые цифры

Есть ли одинаковые цифры в записи чисел N ($1 \leq N \leq 10^9$) и M ($1 \leq M \leq 10^9$)? Если да, то выведите их на экран. Если нет, то выведите на экран значение «-1».

№	Пример	Результат
1	125360 267894	62
2	123 456	-1

Задание №11. Палиндром***

Является ли заданное число N ($1 \leq N \leq 10^9$) палиндромом? Уточните. Выведите на экран значение «Да» или «Нет».

№	Пример	Результат
1	2332	Да
2	12324	Нет

Задание № 12. Счастливое число***

Числа, состоящие из цифр 7 и 4 называют «счастливыми». Напишите код программы по нахождению «счастливых чисел» из условия N ($1 \leq N \leq 10^3$).

№	Пример	Результат	Пояснение
1	744747	Да	
2	12324	Нет	

Задание № 13. Ряд цифр***

Задано число N ($1 < N < 1000$). Требуется вывести на экран сумму всех цифр в записи числа, которая будет четной, при условии a_i ($1 \leq a_i \leq 10^9$)

№	Пример	Результат	Пояснение
1	2332	Да, сумма четная	
2	12324	Нет, сумма нечетная	

Примечание. Уровни: * – легкий, ** – средний, *** – сложный.

Дескрипторы выполнения заданий:

- читает задание, понимает смысл;
- строит математическую модель задачи;
- разрабатывает блок-схему алгоритма решения поставленной задачи;
- записывает программы модульного программирования;
- создает свои проекты в среде Lazarus (для учащихся, занимающихся по углубленному курсу).

4.10 СОЗДАНИЕ ПРОЕКТНЫХ РАБОТ В СРЕДЕ ПРОГРАММИРОВАНИЯ



Как можно применять теоретические и практические знания для выполнения проектных работ в средах программирования?



Подумай

- Какие полезные программы для повседневной жизни можно написать в среде программирования?

Творческие темы

Выберите одну из предложенных тем, и выполните проект самостоятельно либо в группе. Необходимо не только поработать с теоретическим материалом. Особенно ценным считается создание программного продукта.

Тема № 1. Роль объектно-ориентированного программирования в моделировании объектов

Краткое содержание

Исследование роли объектно-ориентированного программирования в моделировании объектов, развитии компьютерных технологий, особенности создания моделей объектов.

Тема № 2. Создание игр в среде программирования Lazarus

Краткое содержание

Разработка программного продукта – мини-игры с учетом технологий среды программирования Lazarus. Сделать презентацию.

Тема № 3. Создание тестовой программы в среде Lazarus**Краткое содержание**

Исследовать программы тестирования, разработанные в среде программирования Lazarus. Рассмотреть особенности создания модели теста. Представить программу тестирования. Объяснить преимущества и недостатки. Сделать презентацию.

Тема № 4. Решение прикладных задач с помощью систем программирования**Краткое содержание**

Исследовать расчетные задачи из повседневной жизни и прикладной математики с помощью систем программирования. Самостоятельно создать программный продукт. Сделать презентацию.

Тема № 5. Создание электронных учебников с помощью языков программирования**Краткое содержание**

В настоящее время существует много программ для разработки электронных учебников. Изучите теорию создания электронных учебников с помощью языков программирования.

Тема № 6. Создание приложений, предназначенных для решения физических задач в среде программирования Lazarus**Краткое содержание**

Создать приложение для автоматизации расчетов по формулам из курса физики 7 и 8 классов. Обосновать задачи программы. Сделать презентацию.

Рекомендации по эффективной защите проекта

- Представьте выбранную тему.
- Объясните причину выбора темы.
- Сформулируйте цели и задачи.
- Назовите назначение использованных компонентов.
- Сделайте краткий обзор темы.
- Поделитесь идеями, возникшими при разработке проекта.
- Обсудите новые знания и навыки, полученные при разработке проекта.
- Сделайте выводы.
- Выслушайте мнение аудитории, умейте принять его.

Критерии оценки успешного проекта

- Самостоятельное выполнение проекта.
- Актуальность темы.
- Умение раскрыть тему.
- Предложения для решения проблемы.
- Артистизм при защите проекта.
- Умение использовать технические средства и наглядные пособия.
- Умение правильно и точно отвечать на заданные вопросы.

Каким должен быть отчет о проекте

- Почему я выбрал эту тему.
- Что я узнал нового, чему научился.
- Как я нашел нужную информацию.
- Как я обрабатывал информацию.
- Как я совместил теорию и практику.
- Сильные стороны проекта.
- Что я не смог выполнить.
- Что было самым легким в проекте.
- Какие трудности возникли во время работы над проектом.
- Если бы была возможность повторного выполнения проекта, что бы я изменил в нем.
- Впечатления, полученные от выполнения проекта.

Словарь – сөздік – dictionary

На русском языке	На казахском языке	На английском языке
Программное обеспечение	Программалық қамсыздандыру	Software
Системный (файл)	Жүйелік	System (file)
Прикладной	Қолданбалы	Applied
Операционная система	Операциялық жүйе	Operating system
Приложения	Қосымшалар	Applications
Оператор выбора	Таңдау операторы	Selection operator
Сортировщик	Сұрыптаушы	Sorter
Для	Үшін	For
Цикл	Цикл, айналым	Cycle
Повтор – до тех пор пока...	Дейінгі қайталау	Repeat – until
Пока – повтор	Өзірше қайталау	While – do
Трассировка	Трассировка	Tracing
Свойства	Қасиет	Properties
Конечное условие	Соңғы шарт	The last condition
Шаг в обход	Айналып өту	Step over
Шаг со входом	Бастапқы қадам	Trace into

5.1 ПОСТАНОВКА ЗАДАЧИ



Как можно решать задачи в интегрированной среде разработки программ?



Подумай

- Что такое модель задачи?
- Как вы думаете, какие знания необходимы для построения модели задачи?
- Из каких этапов состоит решение задач в интегрированной среде разработки программ?



Новые знания

Задача и этапы ее решения

Человек с самого рождения сталкивается с необходимостью решать задачи, возникающие на его пути (делать первые шаги, учиться говорить и т.д.). Решение задач дает человеку возможность достойно жить, развиваться и быть полезным членом общества.

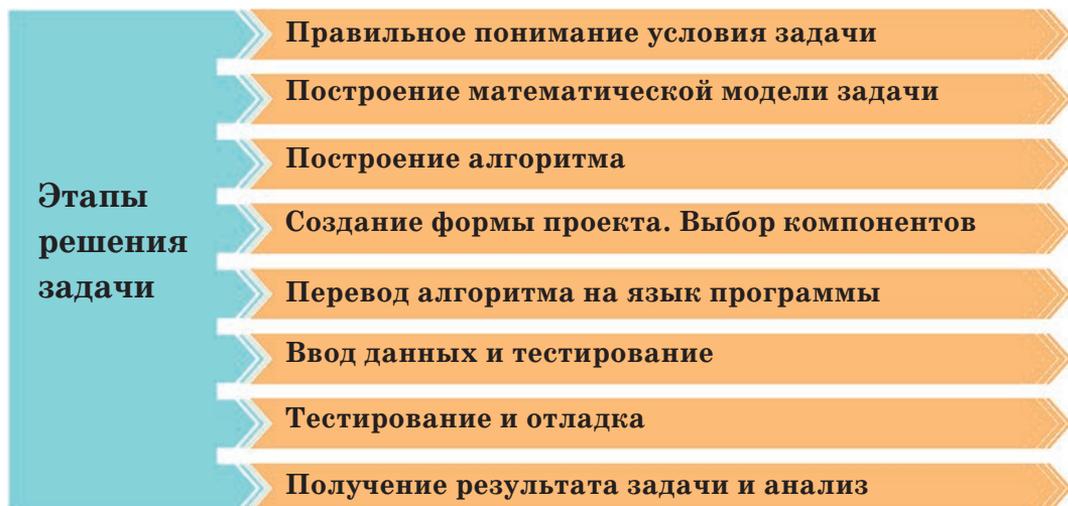
Также перед каждым из нас ежедневно возникают задачи, связанные с образом жизни и деятельностью. Своевременное и обдуманное решение таких жизненных задач считается важным показателем человеческих качеств.

Школьные предметы играют важную роль в обучении и воспитании ребенка с малых лет. Потому что в каждом школьном предмете (математика, физика, информатика, языковые предметы, история и т.д.) есть свои специфические задачи, охватывающие различные стороны и сферы жизни. В процессе решения этих задач, человек развивает умственную деятельность и логику, становясь полноценным членом общества, адаптированным к современным условиям и со сформировавшейся точкой зрения.

Из каких этапов состоит решение задачи в среде программирования?

Задачи, которые рассматриваются в среде программирования, состоят, в основном, из компьютерного моделирования определенных событий и явлений (*схема 1*).

Схема 1



Мы познакомились с этапами решения задачи в среде программирования. Теперь перейдем от теории к практике и попробуем выполнить задание. Рассмотрим проект «**Распиливание бруска на равные части**» в среде Lazarus. Этапы выполнения проекта полностью даны в виде практической работы. Внимательно проанализируйте этот проект, выполните его на компьютере.



Практическая работа

Проект «Распиливание бруска заданной величины на равные части»

Можно ли распилить брусок длиной N ($2 \leq N \leq 100$) метров на M ($2 \leq M < 20$) равных частей, где число частей на единицу больше, чем число разрезов. Если невозможно распилить брусок на равные части, разделите его на куски с разницей длин не больше одного метра.

№	Пример	Результат
1	24 6	4 4 4 4 4 4
2	13 4	3 3 3 4
3	37 8	4 4 4 5 5 5 5 5

1. Правильное понимание условия задачи. Из задачи возникает 2 условия.

Условие 1. Распилить брусок длиной N метров на M равных частей.

Условие 2. Если невозможно распилить брусок на равные части, делим его на куски с разницей длин не больше одного метра. В задаче не возникает других альтернатив, потому что обязательно выполнится одно из двух условий.

2. Построение математической модели задачи. Как видно из примера, если число N делится на M без остатка, то только тогда можно распилить брусок на равные части. То есть, для выполнения первого условия должно выполняться условие $N \bmod M = 0$. Если это условие невыполнимо, то решение задачи следует по второму условию. При рассмотрении данного примера обратите внимание на некоторые закономерности.

Рассмотрим еще один пример. Деление числа 13 на 4 дает число 3 с остатком 1. В качестве ответа задачи выводим $(M-P)$ на экран K раз и $(K+1)P$ раз. Данную закономерность можно проверить на других примерах.

Рассмотрим вариант с числами 37 и 8. При делении 37 на 8 получим целое число 4 с остатком, равным 5. Это значение выводим на экран $(8-5)$ раз в ответе. Получится последовательность: 4 4 4. После этого прибавляем единицу к целому числу $(4+1)$, и именно столько раз в ответе выводим остаток. Получится последовательность: 5 5 5 5 5.

В этом примере по первому условию брусок не делится на равные части, а по второму условию его можно поделить на несколько частей, длины которых будут отличаться друг от друга на единицу.

$$\begin{array}{r}
 \boxed{N} \rightarrow 13 \\
 \underline{12} \\
 1 \leftarrow \boxed{P}
 \end{array}
 \quad
 \begin{array}{r}
 \boxed{M} \leftarrow 4 \\
 \hline
 3 \leftarrow \boxed{K}
 \end{array}$$

3. Построение алгоритма. Необходимо выявить общую закономерность и построить математическую модель. Для написания программы надо систематизировать задачу и разделить ее на этапы. Опишем список команд на каждом этапе.

Алгоритм задачи

- Выбор компонента ввода и вывода результата. В нашем случае используем компонент **Edit**.
- Задействуем разветвленный оператор для проверки начального условия задачи $N \bmod M = 0$.
- Для вывода последовательности чисел применим оператор цикла с параметром.

4. Создание формы проекта. При создании формы проекта выбираем компоненты **Edit, Image, Label, Button**. Цель выбора компонента **Image** состоит в том, чтобы повысить наглядность задачи с помощью иллюстраций (рис. 1).

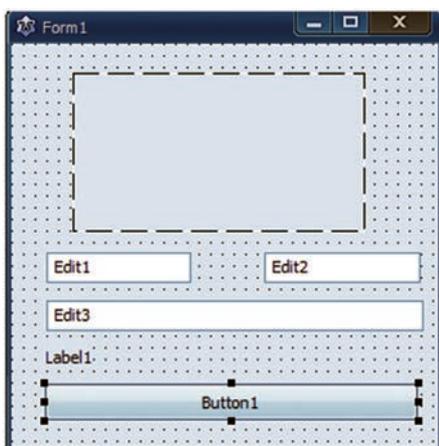


Рис. 1

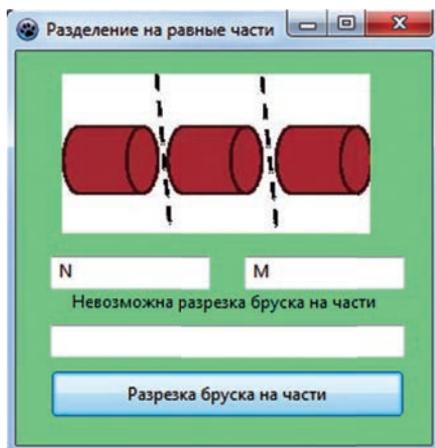


Рис. 2

5. Перевод алгоритма на язык программы. Создадим программу задачи по составленному алгоритму. При построении программы необходимо, чтобы все используемые компоненты были совместимыми друг с другом.

```

procedure TForm1.Button1Click (Sender: TObject);
var t,i,j,n,z,k,m:longint;
s:string;
begin
n:=strtoint (edit1.text);
m:=strtoint (edit2.text);
t:=n div m;
if n mod m=0 then label1.caption:='Брусок делится на равные части'
else label1.caption:='Брусок не делится на равные части';
for i:= to m-(n mod m)do
s:=s+inttostr(t)+' ';
for i = 1 to n mod m do
s:=s+inttostr (t+1)+' ';
edit3.text:=s;
end;

```

6. Ввод данных и тестирование. Запускаем проект. На экране появится форма, представленная на рисунке 2. В процессе ввода данных следует обратить внимание на то, чтобы вводились корректные данные, отвечающие условию задачи. Например, если в программу для тестирования введем 0 и 2 или 10 и 16, то в ответе будет ошибка. Потому что в первом случае длина бруска не может быть равна нулю,

а во втором случае не выполнится условие задачи – получение целых кусков.

7. Получение результата задачи и его анализ. Только в том случае, когда в результате тестирования, программа даст правильные ответы на все подготовленные заранее тесты, можно сказать, что она выполнена на все 100 процентов. Если в результате тестирования полученные ответы не будут соответствовать условиям задачи, нам придется применить метод трассировки для исправления логических и алгоритмических ошибок, допущенных в структуре программы. Как видно из рисунка 2, при вводе предложенных в 3-м примере данных, программа выдает правильный результат. Можно проверить работу программы не только на основе этих данных, а задав другие их значения.



Анализ



Сравните этапы решения задач в математике, физике и информатике. Выявите сходство и различия.



Синтез



Выберите известную вам задачу из курса математики. Разбейте ее решение на этапы, постройте проект решения выбранной задачи. Дайте название проекту.



Оценка



Исследуйте. Бывают ли другие варианты описания этапов решения задач на компьютере?

Выясните. Есть ли различие между вариантом, который мы предлагали, и другими вариантами решения задач.



Вопросы

1. Из каких этапов состоит решение задачи в программируемой среде?
2. Как вы думаете, на что нужно обратить внимание при выборе компонентов?
3. На что нужно обратить внимание при построении теста задачи и тестировании?

4. Какие особенности вы можете отметить при решении задач на компьютере? Приведите примеры.



Задания

Выполните предлагаемые проекты, используя этапы решения задачи в программируемой среде.

Подготовка проекта «Игра в баскетбол»

Баскетбольный матч состоит из 4 четвертей. Известны результаты каждой из них. Нужно определить победителя матча. Входные данные содержат 4 строки, в каждой строке находится два целых числа a и b – итоговый счет в соответствующей четверти. a – количество набранных очков за четверть первой командой, b – количество очков, набранных за четверть второй командой ($0 \leq a, b \leq 100$). Постройте проект по определению победителя по результату игры между двумя командами. По результату игры дайте ответ в виде «1 команда», «2 команда» или «Ничья».

№	Пример	Результат
1	26 17 13 15 19 11 14 16	1 команда

Проект «Выполнение действий»

Даны числа A и B , а также знаки действий, которые нужно выполнить с этими числами. Например, $+$, $-$, $/$, $*$. Подготовьте проект, который выполнит заданное действие с числами.

№	Пример	Результат
1	$7 + 9$	16
2	$4 * 10$	40

5.2

РАЗРАБОТКА АЛГОРИТМА



Как эффективно применять простые алгоритмы, для решения задач в интегрированной среде разработки программ?



Подумай

- С какими свойствами алгоритмов вы знакомы?
- Какие готовые алгоритмы вы знаете?
- Какие эффективные методы использования алгоритмов вы предложили бы для решения задач?



Новые знания

Алгоритм и его свойства

Алгоритм – это строго определенная последовательность действий для решения задачи. Остановимся на основных свойствах алгоритма (Схема 1).

Схема 1

Дискретность алгоритма означает, что алгоритм должен быть разбит на последовательность выполняемых шагов.

Определенность означает, что алгоритм должен содержать только те команды, которые входят в его набор, и которые может выполнить конкретный исполнитель.

Результативность означает, что алгоритм должен приводить к решению задачи за конечное число шагов.

Массовость алгоритма – свойство пригодности алгоритма для решения всех задач заданного класса.

Есть несколько способов описания алгоритмов

1. Описание алгоритма через слово-формулу, то есть описание алгоритма с помощью слов и формул.
2. Графическое описание алгоритма, то есть описание с помощью графических схем, блок-схем.
3. Описание алгоритмов через псевдокоды. Псевдокод – описание операторов программ на простом языке, без научных терминов и специальных знаков.

Как можно использовать алгоритм при решении задач?

Есть алгоритмы наиболее часто повторяющиеся при решении задач. Например, алгоритм определения наибольшего или наименьшего числа, нахождения среднего арифметического, выделения цифр числа и т.д.

В процессе программирования необходимо понять и запомнить часто встречающиеся алгоритмы и создать их своеобразный фонд (запас). При решении сложных задач, разбивая их на подзадачи, можно использовать эти алгоритмы.

В качестве примера рассмотрим простую задачу.

Задан размер месячной зарплаты N -го количества работников компании «Пассажирские перевозки». Определите размер наименьшей и наибольшей зарплаты работников. Найдите разницу между наибольшей и наименьшей заработной платой. Вычислите среднюю заработную плату служащих компании.

Полезные алгоритмы при решении задачи:

1. Алгоритм определения наибольшего из N чисел;
2. Алгоритм определения наименьшего из N чисел;
3. Алгоритм определения средней зарплаты.

1) Программа нахождения наибольшего числа:

```
var l,n,x,max:integer;
begin
  readln (n); max:=0
  for i:=1 to n do begin
    readln (x);
    if max<x then max:=x;
  end;
  writeln (max);
end.
```

2) Программа нахождения наименьшего числа:

```
var l,n,x,min:integer;
begin
  readln(n); min:=maxint;
  for i:=1 to n do begin
    read (x);
    if min>x then min:=x;
  end;
  writeln (min);
end.
```

3) Программа нахождения среднего арифметического значения:

```
var l,n:integer;
s:real;
begin
  readln (n); s:=0;
  for i:=1 to n do begin
    read (x);
    s:=s+x;
  end;
  write(s/n);end.
```

Создадим полный код для заданной выше задачи:

```
var
  i,n,x,max,min: integer;
  s:real;
begin
  readln(n);
  max:=0; min:=maxint;
  for i:=1 to n do
    begin
      read(x);
      if max<x then max:=x;
      if min>x then min:=x;
      s:=s+x;
    end;
  writeln(max);
  writeln(min);
  writeln(max-min);
  write(s/n:0:2);
end.
```

При написании кода программы задачи мы использовали три алгоритма.

В задаче для всех заданий можно использовать только одно тело цикла.

Такой способ помогает экономить время при выполнении программы.

В записи оператора вывода **write (s/n:0:2)** запись **0:2** означает, что дробное число округляется до 0,01. Если s/n значение равно 4.45252256855, тогда на экран выводится число 4.45.

Для чего разрабатываются специальные алгоритмы?

Сложные задачи в процессе написания программы очень удобно разбивать на несколько подзадач. Большая часть заданий для программирования дается в виде задач, поддающихся оптимизации. В них обязательно нужно определить наибольшее или наименьшее значение, самый тяжелый или самый легкий вес и т.д. В этих случаях удобно использовать заранее составленные оптимальные алгоритмы. Это, во-первых, дает возможность экономить время, во-вторых, акцентировать все внимание на главной проблеме задачи.

Остановимся на некоторых стандартных алгоритмах, известных большинству программистов. Некоторые из этих алгоритмов мы использовали в прошлых темах. В чем же необходимость их использования?

Одним из важных аспектов алгоритма является скорость его выполнения. В процессе программирования перед разработчиком постоянно стоит проблема экономии времени на выполнение программы. Например, в теме «Вложенные циклы» вы выполнили задание по определению простых чисел из последовательности и выводу результатов на экран. При определении простых чисел вы выполнили цикл `for` для вычисления квадратного корня проверяемого числа. Почему? Остановимся на причинах. Например, возьмем в качестве проверяемого число 50. Если нам нужно выяснить простое или составное это число, мы должны разделить его на все числа расположенные между 1 и 50. В таком случае для проверки числа 50 цикл должен работать 50 раз. Если мы составим алгоритм задачи таким образом, тогда для проверки числа 1 000 000, тело цикла должно работать миллион раз. На работу этого цикла уходит 1 секунда времени. Это очень много.

```
for j:=1 to n do
```

Значит, созданный нами алгоритм не является оптимальным. Мы должны придумать другой алгоритм. Если учитывать то, что число 50 делится на числа между 1 и 50, а делители любого числа, не считая само число, встречаются до середины интервала, (например, делители 50, не считая само число, встречаются до 25), тогда нет необходимости делить наше число до 50. В таком случае работа цикла сокращается в 2 раза и для проверки он будет работать 25 раз.

```
for j:=2 to n div 2 + 1 do
```

Если воспользуемся теорией чисел, работу цикла можно сократить еще больше.

Теорема: Если простое число p является наименьшим делителем числа a , тогда $P \leq \sqrt{a}$

Например, число 59 – простое. Потому что оно не делится на простые числа 2, 3, 5, 7, которые меньше 7,6 ($\sqrt{59} \approx 7,6$). Тогда для проверки числа 50 на простое число нам нужно всего 6 циклов от 2 до 7 ($\sqrt{50} \approx 7$). Этот алгоритм в 8 раз меньше первоначального. Если число, которое мы будем проверять равно 1 000 000, тогда работа цикла уменьшится в 1 000 раз.

```
for j:=2 to round(sqrt(n)) do
```

Анализируя данный пример, вы увидели, насколько важно знать в процессе написания программы оптимальные алгоритмы. Для того чтобы научиться применять алгоритмы на практике, выполните проект «Покраска деревянного забора».



Практическая работа

Диана собирается покрасить забор вокруг своего дома. У нее есть пять видов красок. Для того чтобы выяснить, в какой цвет покрасить забор, девочка нанесла краску разных цветов на доски забора. В результате весь забор оказался покрашенным в разные цвета. Диана собирается заново закрасить забор в один цвет. На эту работу она хочет потратить как можно меньше сил. Помогите Диане. Известны число досок на заборе $N(N < 100)$ и их цвет, один из пяти в интервале между 1 и 5. Ответ задачи – минимальное количество окрашиваемых досок забора.

№	Пример	Результат
1	10 1 2 2 1 5 3 1 1 4 4	6
2	7 5 5 5 5 5 5	0

На рисунке 1 дана форма начала выполнения проекта. На рисунке 2 изображен конец проекта и выданный результат. Результат проекта подходит под первый пример задачи.

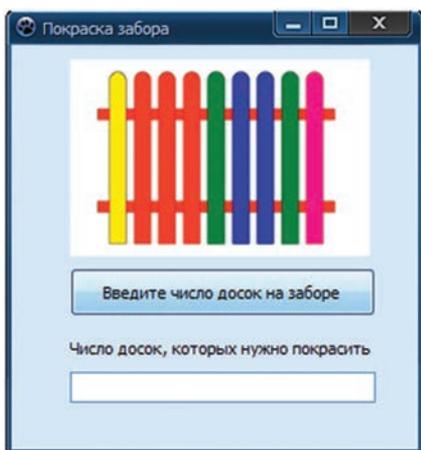


Рис. 1

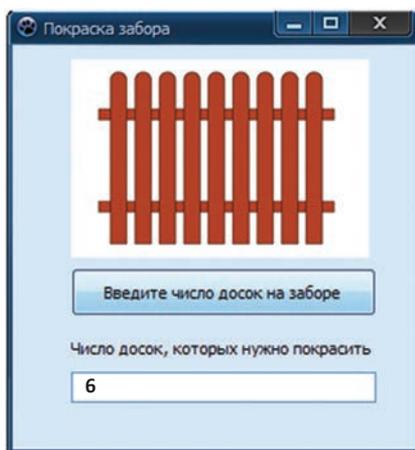


Рис. 2

Код проекта «Покраска деревянного забора»

```

procedure TForm1.Button1Click(Sender:
TObject);
var n,i,max,k,z,z1,z2,z3,z4:longint;
s,l:string;
begin
z:=0;z1:=0;z2:=0;z3:=0;z4:=0;
s:=inputbox ('Число досок на заборе',
'Введите число досок на заборе','');
val(s,n);
for i:=1 to n do
begin
s:=inputbox ('Цвета досок на заборе',
'Введите' +inttostr(i)+'цвета досок','');
val(s,k);
case k of
1: inc(z);
2: inc(z1);
3: inc(z2);

```

```

4: inc(z3);
5: inc(z4);
end;
end;
max:=z;
if z1>max then max:=z1;
if z2>max then max:=z2;
if z3>max then max:=z3;
if z4>max then max:=z4;
edit1.text:=inttostr(n-max);
end;
procedure TForm1.
Edit1Change(Sender: TObject);
begin
Image2.Visible:=True;
Image1.Visible:=False;
end;

```



Вопросы

1. Какими свойствами обладает алгоритм? Какие виды описания алгоритма существуют?
2. Почему важно применять стандартные алгоритмы в процессе программирования?
3. Какие компоненты использованы в проекте «Покраска деревянного забора»?
4. Что является самым важным в процессе программирования? Приведите примеры.



Анализ



Сделайте анализ проекта «Покраска деревянного забора». В процессе анализа:

1. Подготовьте тесты, охватывающие различные ситуации и всесторонне исследуйте работу проекта.
2. Определите компоненты, используемые в проекте, и их свойства.



Синтез



Проанализируйте алгоритм создания проекта «Покраска деревянного забора». Создайте и предложите альтернативный алгоритм.



Оценка



Проведите исследование важности алгоритмов для программиста и процесса программирования. Результаты исследования представьте в виде научной статьи.

5.3 ПРОГРАММИРОВАНИЕ АЛГОРИТМА



Какие виды алгоритмов бывают, и как их применять в процессе написания программы?



Подумай

- Какие виды алгоритмов вы знаете?
- Насколько важны в процессе программирования оптимальные алгоритмы?



Новые знания

Графическое изображение или блок-схема алгоритма

Представление алгоритма в виде блок-схемы есть логическое отображение его структуры в графическом виде. Здесь каждый этап обработки информации отображается в виде соответствующей геометрической фигуры (блока).

Изображение алгоритма в виде блок-схемы относится к графическому методу. Части алгоритма в блок-схеме связываются направленными линиями. Таким образом, можно видеть, на каком этапе находится решение задачи (см. электронное приложение).

Очень важно при программировании задач составлять блок-схему алгоритмов, особенно для начинающих разработчиков. Важно знать, в какой последовательности нужно выполнять действия при использовании исходных данных для получения требуемого результата. Поэтому надо внимательно прочитать условия задач, подготовить тесты для каждого из условий, просчитать шаги алгоритма и только после этого приступить к написанию программы. Алгоритмы по структуре делятся на 3 вида.

Деление алгоритмов по структуре

- 1 **Линейный алгоритм** – алгоритмическая структура набора команд (указаний), выполняемых строго последовательно в порядке записи команд (*схема 1*).
- 2 **Разветвляющийся алгоритм** – алгоритмическая структура, в которой, в зависимости от условия, выполняется одна или другая последовательность действий (*схема 2*).

3 Циклический алгоритм – алгоритмические построения, которые должны повторяться указанное число раз, пока не выполнено заданное условие. С помощью циклов облегчается процесс создания программ (схема 3).

Схема 1. Линейное строение



Схема 2. Разветвляющееся строение

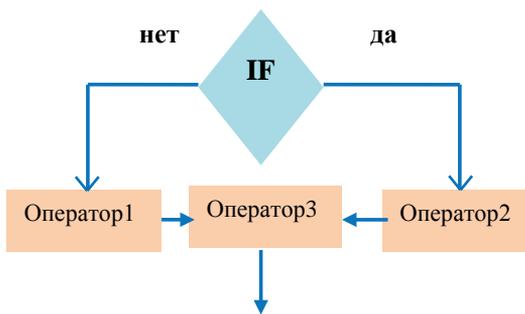
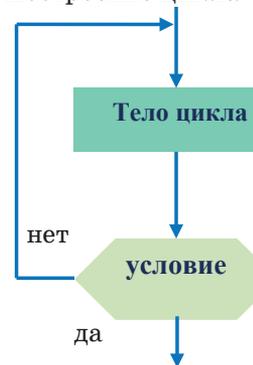


Схема 3. Построение цикла



Рассмотрим пример использования блок-схемы для решения поставленной задачи.

Пример. Дано двузначное число N. Является ли сумма цифр этого числа четной (нечетной)? Уточните. Если сумма нечетная, то в ответе запишите «1», если четная – «0».

Построим блок-схему задачи (Схема 4).

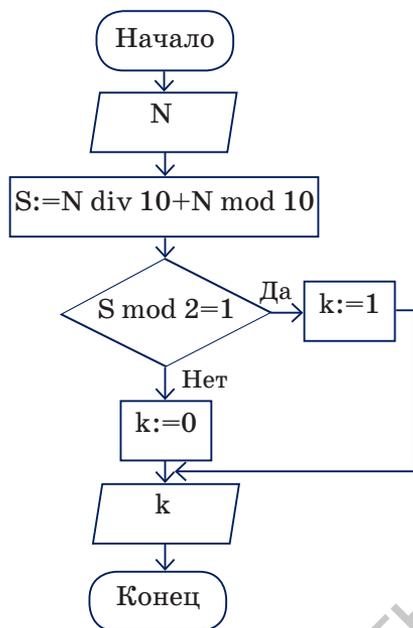


Схема 4

```

Var
s, n, k: integer;
begin
  readln(n);
  S:= N div 10 +N mod 10;
  if S mod 2 =1 then k:=1 else k:=0;
  write(k);
end.
    
```

Если сравнить блок-схему с кодом программы, то структура блок-схемы значительно облегчает понимание программирования разветвляющихся алгоритмов. Прежде чем записать код программы решения задачи, очень полезно для правильного решения задачи предварительно построить ее алгоритм.



Практическая работа

Проект «Поход в магазин»

Сегодня у Динары день рождения. К ней в гости придут друзья. Чтобы угостить гостей, Динаре нужно сходить в 2 магазина возле дома. Расстояние от дома Динары до 1-го магазина d_1 метров, до 2-го магазина – d_2 метров, расстояние между двумя магазинами d_3 метров. Помогите Динаре найти самый короткий путь, чтобы сходить в оба магазина и вернуться домой. Условие: d_1, d_2, d_3 ($1 \leq d_1, d_2, d_3 \leq 10^8$)

№	Пример	Результат
1	20 40 30	90
2	1 1 5	4



Рис. 1

Код программы проекта «Поход в магазин»

```
procedure TForm1.BitBtn1Click(Sender:
TObject);
var a,b,c:longint;
begin
a:=strtoint(edit1.text)+0;
b:=strtoint(edit2.text)+0;
c:=strtoint(edit3.text)+0;
if (a+b)<=c then edit4.text:=inttostr(2*a+2*b)
else if (b+c)<=a then edit4.
text:=inttostr(2*b+2*c)
else if (a+c)<=b then edit4.
text:=inttostr(2*a+2*c) else edit4.
text:=inttostr(a+b+c);
end;
```

До ввода кода программы проекта в компьютер приготовьте 5 видов теста, предусматривающих все возможные маршруты. На рисунке 1 вместе с формой проекта дан код программы. Введите код программы в компьютер.



Анализ



Проанализируйте форму проекта «Поход в магазин» и код программы. Протестируйте с помощью тестов и проверьте.



Синтез



Сконструируйте блок-схему проекта «Поход в магазин» и обоснуйте ее.



Оценка



Оцените важность перехода от составления алгоритма решения задач в схематичном виде к программированию разветвляющихся алгоритмов. Обсудите с одноклассниками, что лучше: написать программу сразу или сначала представить алгоритм решения задачи в схематичном виде.



Вопросы

1. Какие способы представления алгоритмов вы знаете?
2. Что такое графическое представление алгоритма?
3. Какие блок-схемы существуют для представления алгоритмов в графическом виде?
4. Какие бывают виды алгоритмов по структуре?
5. Как вы можете описать следующие алгоритмы – линейные, разветвляющиеся, циклические?
6. Как выбираются шаги выполнения алгоритмов для записи программы?



Задания

Проект «Квадратное уравнение»

Уравнение вида $ax^2+bx+c=0$ называется квадратным уравнением. Пути решения этого уравнения вы знаете из курса алгебры. Постройте проект «Квадратное уравнение» для нахождения всех решений квадратного уравнения, обоснуйте ответ.

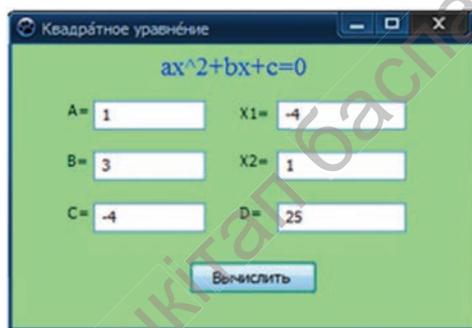


Рис. 2

Выполните следующие шаги:

- предложите блок-схему алгоритма;
- подготовьте 5 тестов на возможные решения;
- спланируйте форму проекта и используемые компоненты (рис. 2), форму проекта можно сделать по своему усмотрению;
- запишите код проекта;
- запустите проект, протестируйте его работу;
- проанализируйте результат.

Проект «Теплые дни»

В Казахстане в марте месяце дни становятся теплыми, весна начинает вступать в свои права. Но погода переменчива, на 3-4 теплых дня приходится 2-3 холодных. В марте месяце для N ($1 \leq N \leq 31$) дней даны значения температуры a ($-30 \leq a \leq 30$). Постройте проект для определения самых теплых дней месяца.

№	Пример	Результат
1	10 -1 2 3 -2 -3 2 4 6 8 -1	4

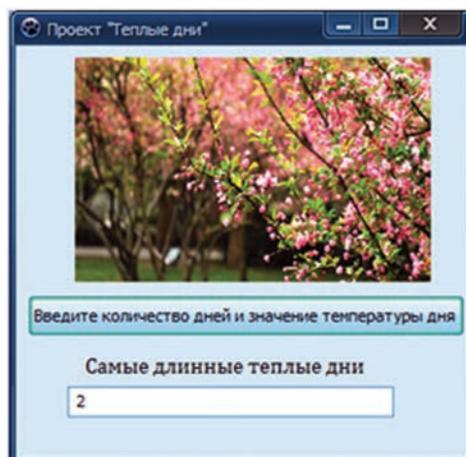


Рис. 3

Выполните следующие шаги:

- предложите блок-схему алгоритма;
- подготовьте 5 тестов на возможные решения;
- спланируйте форму проекта и используемые компоненты; (рис. 3), форму проекта можно сделать по своему усмотрению;
- напишите код проекта;
- запустите проект, протестируйте его работу;
- проанализируйте результат.

5.4 ТЕСТИРОВАНИЕ ПРОГРАММЫ



Как проверить и протестировать эффективность работы программы?



Подумай

- Что такое тестирование программы?
- Каким требованиям должны соответствовать тесты, предназначенные для проверки эффективной работы программы?



Новые знания

Тестирование программы

Тестирование – это испытание, проверка правильной работы программы или ее отдельных частей.

Отладка и тестирование – это два разных процесса и между ними нет ничего общего.

• При отладке выявляются и исправляются очевидные синтаксические и семантические ошибки.

Синтаксические ошибки – это ошибки в записи конструкций языка программирования (чисел, переменных, функций, выражений, операторов, меток, подпрограмм).

Семантические ошибки – это ошибки, связанные с неправильным содержанием действий и использованием недопустимых значений величин.

• В процессе тестирования проверяется способность работы программы без явных ошибок.

Тестирование выявляет наличие ошибок, а отладка выявляет причины неправильной работы программы.

Тест – это точное описание всех результатов некоторых наборов первичных данных и соответствие этим данным, которые должны быть получены во время работы программы.

Как бы тщательно ни была отлажена программа, решающим этапом определения работоспособности является ее выполнение непосредственно в системе тестирования.

Если при прохождении через специально отобранную систему тестов программа будет давать верные результаты в любых ситуациях, то в таком случае, она считается корректной.

Для реализации метода тестирования должны быть заранее подготовлены эталонные результаты тестов.

Эталонные результаты должны быть готовы до процесса тестирования.

Этапы процесса тестирования

Процесс тестирования можно разделить на три этапа.

1. Проверка при обычных условиях. Прогнозируется тестирование на основании первичных данных, соответствующих реальному состоянию работы программы.

2. Проверка в экстремальных условиях. В этом случае тестовые данные – это пограничные значения набора первичных данных. Этот набор состоит из первичных данных, предназначенных для правильной работы программы. Малые или очень большие значения чисел и информации являются типичными примерами таких проверок.

Известно, что все программы проверяются обработкой ограниченного набора первичных данных. Поэтому важно получить ответы на следующие вопросы:

1. Что случится, если по каким-либо причинам (например, ошибки ввода) программе, непредназначенной для обработки отрицательных и нулевых значений переменных, придется обрабатывать именно эти величины?
2. Как поведет себя программа, если элементы массива превысят количество, указанное в уведомлении?
3. Что произойдет, если обрабатываемое число будет очень маленьким или очень большим?

Если программа принимает неправильные данные за правильные и выдает неверные результаты, похожие на правильные, это считается самым худшим вариантом.

Программа обязана выдавать за неправильную любую информацию, которую сама не способна правильно обработать. Причиной неспособности программы пройти полное тестирование являются ошибки, допущенные при ее создании. Поэтому программист обязан знать и классифицировать все виды возможных ошибок.

Ошибки встречающиеся в программировании.

Следует помнить, что абсолютно на всех этапах разработки программы могут быть допущены ошибки, среди которых: неверная постановка задачи; неправильно разработанный алгоритм решения; неучтенные обстоятельства, при которых могут быть допущены логические, семантические и синтаксические ошибки; ошибки, допускаемые при выполнении операции и отправке данных; ошибки ввода-вывода и др.

Обычно синтаксические ошибки выявляются при **трансляции**. **Транслятор** не может обнаружить множество других ошибок, потому что не знает, о чем думал программист. Следовательно, отсутствие синтаксических ошибок не является достаточным условием корректности программы.

Трансляция программы – преобразование программы из одного языка программирования в другой. **Транслятор** – программа или техническое средство, выполняющее трансляцию программы.

Примеры синтаксических ошибок:

- Пропущенные знаки препинания;
- Неправильно поставленные скобки;
- Некорректная запись оператора;
- Неправильная запись служебного слова;
- Некорректная запись имени переменной;
- Отсутствие окончания условия цикла;
- Отсутствие описания массива и т.д.

Ошибки, которые не может обнаружить транслятор

Бывают ошибки, которые транслятор не может обнаружить, так как программа написана правильно, но ее модули неверно используются. Приведем примеры таких ошибок.

Логические ошибки:

- Условия проверены, но впоследствии происходит неправильный переход к выполнению ветви алгоритма;
- Неполный учет возможных условий;
- Пропущены один или несколько значений, необходимых для выполнения блоков алгоритма.

Ошибки, встречающиеся в циклах:

- Не указано начало цикла;
- Некорректно составлено условие окончания цикла;
- Неправильно определено число итераций;
- Бесконечный цикл.

Ошибки ввода-вывода и ошибки при работе с данными:

- Неправильно описаны типы данных;
- Некорректная организация ввода необходимого количества данных;
- Неправильная корректировка данных.

Ошибки, допускаемые при использовании переменных:

- Использование переменных без указания их начальных значений;
- Некорректное использование одной переменной вместо другой.

Ошибки, допускаемые при работе с массивами:

- Массивы не приведены заранее к нулю;
- Неправильное описание массивов;
- Массивы с некорректным указанием индексов.

Ошибки, допускаемые при выполнении арифметических операций:

- Неправильное использование типа переменной (например, вместо целой переменной используется вещественная переменная);
- Неточно указан порядок алгебраических действий;
- Деление на ноль;
- Извлечение квадратного корня отрицательного числа;
- Потеря разряда числа.

Все эти ошибки можно выявить на этапе тестирования программы.

Попробуем протестировать программу на практике

Рассмотрим задачу «Делители» (задача из районной олимпиады по информатике 2009 года) для ознакомления с тестированием программы.

«Делители»

Дано натуральное число b . Требуется определить его делители. Если данное натуральное число b делится без остатка на число a , то число a является делителем числа b . Например, у числа 15 есть 4 делителя: 1, 3, 5, 15. Для каждого данного числа b определите, четным или нечетным будет количество его делителей. Число b дано в количестве N ($1 < N \leq 10^5$). Для каждого из чисел b ($1 \leq b \leq 10^{18}$) выведите значение 0 – если будет четное количество делителей, и 1 – если количество делителей будет нечетным.

Пример	Результат	Пояснение
5		
5 8 16 11 9	0 0 1 0 1	У числа 8 количество делителей равно 4: 1, 2, 4, 8. Количество делителей – четное число. У числа 16 количество делителей равно 5: 1, 2, 4, 8, 16. Количество делителей – нечетное число.

Перейдем к тестированию задачи. Эти тесты необходимо приготовить до решения задачи. Общее количество примеров равно 10, но мы остановимся на 5 тестах. По сути, одинаковые типы тестов повторяются по 2 раза.

№	Тесты ввода	Результат
1	10 77 51 49 77 49 55 80 94 49 77	0 0 1 0 1 0 0 0 1 0
2	10 830 585 868 529 683 740 967 557 828 1147	0 0 0 1 0 0 0 0 0 0
3	10 501139 786769 976039 635209 902903 977269 557993 635209 654481 876831	0 1 0 1 0 0 0 1 1 0
4	10 50002071524071 50000214792889 50000348753931 50001431026237 50001060703878 50001009723014 50000483494043 50000950188029 50000119016331 50000702267865	0 1 0 0 0 0 0 0 0 0
5	10 500000001283061852 500000001394837861 499999937510787169 500000000548349143 500000000709382267 499999937510787169 500000002022608063 500000000107748979 50000000076465733 500000000770135069	



Почему так составлены тесты для проверки задачи? Какие затруднения встречаются при решении задачи?

Для определения делителя числа нам нужно использовать цикл с параметром, а если делители – это последовательности цифр, то необходимо добавить еще один цикл. Тогда для решения задачи мы должны использовать 2 вложенных цикла. А если вспомнить, что наша последовательность чисел составляет длину 100 000, то на каждый тест потребуется более 2 секунд. В этом случае предельное число $b=10^{18}$. Для определения делителя числа мы ограничимся половиной

чисел из последовательности (см. предыдущую главу). В то же время, невозможно ограничиться количеством циклов, сокращенным до половины. Это означает, что мы можем решить эту задачу по известному нам алгоритму только на 20 процентов. Иными словами, для ее решения мы должны рассмотреть другой подходящий алгоритм, который предполагает извлечение квадратного корня из данного числа – \sqrt{b} . Если квадратный корень равен целому числу, то количество делителей будет нечетным, если нет – четным.

Например, квадратный корень из 36 равен 6. Количество его делителей равно 9: 1, 2, 3, 4, 6, 9, 12, 18, 36. А корень из 40 = 6,324... Количество его делителей равно 8: 1, 2, 4, 5, 8, 10, 20, 40.

Тогда программа задачи будет выглядеть следующим образом:

Код программы «Делители»

```
var
i:longint;
b:int64;
begin
readln(n);
fori:=1 to n do   begin
read(b);
if sqrt(b)=int(sqrt(b)) then
write(1, ' ') else write(0, ' ');
end;
end.
```

Остановимся на данном тесте.

Тест № 1: для простого случая.

Тест № 2: среди чисел с единственным делителем, равным четному числу.

Тест № 3: для предельного случая 10^6 .

Тесты №№ 4, 5: значение **b** для тестирования 10^{18} чисел.

Как вы можете увидеть, тесты даны на простой и предельный случаи.



Вопросы

1. Что означает тестирование программы?
2. Из каких этапов оно состоит?
3. Какие ошибки можно выявить в процессе тестирования?
4. Сборник каких тестов разрабатывается для тестирования программы.



Задания

1. Дан ряд цифр из нулей и единиц. Необходимо найти самую длинную цепочку из единиц. Используя этапы тестирования задач, подготовьте 5 видов теста для определения корректности решения задачи.

№	Пример	Результат
1	01001111011	4

2. На рисунке показаны клетки шахматной доски. Определите, будут ли две заданные клетки одного цвета. Если обе клетки одного цвета, то следует вывести на экран значение «Да», если разного – значение «Нет». Используя этапы тестирования задач, подготовьте 5 видов теста для определения корректности решения задачи.

№	Пример	Результат
1	2 1 3 2	Да

	1	2	3	4	.	.
1	■	□	■	□	■	□
2	□	■	□	■	□	■
3	■	□	■	□	■	□
.	□	■	□	■	□	■
.	■	□	■	□	■	□

3. Рабочие должны за 5 часов сделать N деталей. Они за первый час делают X деталей, и далее на каждую последующую деталь тратят T минут. Условие: X, N и T ($X \leq N \leq 15, T \leq 240$). Сколько деталей рабочие успеют изготовить за 5 часов? Уточните. Приготовьте 5 соответствующих задаче тестов. Напишите программу.

№	Пример	Результат
1	7 12 45	12
2	3 10 60	7

4. Телефонная компания начала разрабатывать новый план по оплате своих услуг. Согласно новым расценкам, если клиент разговаривает в течение месяца A минут, то за каждую минуту он платит K тенге, а если более A минут, то за каждую дополнительную минуту будет платить B тенге. Дима разговаривал P минут в месяц. Сколько он заплатит телефонной компании? ($1 \leq P \leq 1000, 1 \leq A \leq 500, 1 \leq K \leq 50, 1 \leq B \leq 100$). Приготовьте 5 соответствующих задаче тестов. Напишите программу.

№	Пример	Результат
1	392 300 10 25	5300

5.5

ПРАКТИЧЕСКАЯ РАБОТА. ПОСТРОЕНИЕ БЛОК-СХЕМЫ ЗАДАНИЯ



Как при решении задач переходить от алгоритма блок-схемы к программированию?



Подумай

- Какие преимущества дает в процессе программирования составление алгоритма блок-схемы?



Практическая работа

Задания по линейным алгоритмам

Задание № 1. Время*

С начала суток прошло N секунд (N – целое число). Требуется преобразовать секунды в часы, минуты и секунды.

№	Пример	Результат
1	36482	10 часов 8 минут 2 секунды

Задание № 2. Площадь окружности*

Вычислить площади S_1 и S_2 двух окружностей с общим центром и радиусами R_1 и R_2 ($R_1 \geq R_2$), а также площадь S_3 кольца с внешним радиусом R_1 и внутренним радиусом R_2 .

$$S_1 = \pi \cdot R_1^2, S_2 = \pi \cdot R_2^2, S_3 = S_1 - S_2.$$

№	Пример	Результат
1	10 5	314 157 157

Задание № 3. Сопротивления**

Даны сопротивления проводников R_1 и R_2 . Найти общее сопротивление при последовательном и параллельном соединениях.

№	Пример	Результат
1	10 5	25 6

Задание № 4. Сложение цифр**

Дано трехзначное число. Требуется сложить разряды единиц и сотен.

№	Пример	Результат
1	497	11

Дескрипторы выполнения задания:

- читает задание, понимает смысл;
- создает 3 теста для проверки задания с различными значениями;
- строит блок-схемы по линейному алгоритму;
- пишет программу в среде Lazarus.

Примечание. Уровни: * – легкий, ** – средний, *** – сложный.

Задания с разветвляющимися алгоритмами**Задание № 5. Парты****

Формируется заказ на приобретение парт для 3-х кабинетов информатики. Необходимо определить наименьшее количество парт, если известно количество учеников, обучающихся в каждом кабинете. Количество парт $1 \leq N, P, K \leq 100$. Парты стандартные, т.е. за каждой партой сидят по 2 ученика.

№	Пример	Результат
1	21 27 30	40

Задание № 6. Конец урока***

Занятия в школе начинаются в 9:00. Урок длится 40 минут. После 1, 3, 5 и т.д. уроков перемена длится 5 минут, после 2, 4, 6 и т.д. уроков – 15 минут. По заданному количеству уроков (от 1 до 10) определите время окончания занятий в часах и минутах.

№	Пример	Результат
1	3	11:20

Задание № 7. Нумерация страниц***

В книге N страниц ($N \leq 1300$). Сколько всего цифр необходимо для нумерации страниц этой книги?

№	Пример	Результат
1	9	9
2	20	31

Задание № 8. Определение разряда числа*

Задано число X в интервале от 1 до 9999. Требуется найти количество разрядов числа X .

№	Пример	Результат
1	9	1
2	8596	4

Задание № 9. Целое число**

Задано число N ($N \leq 1000$). Определить целое это число или нет. Если целое, то вывести на экран значение 1, если вещественное, то вывести на экран значение 0.

№	Пример	Результат
1	112	1
2	4,5	0

Дескрипторы выполнения задания:

- читает задание, понимает смысл;
- создает 5 тестов для проверки задания;
- строит блок-схему алгоритма ветвления;
- выполняет проект в среде программирования Lazarus.

Задания с циклическими алгоритмами**Задание № 10. Цена книги***

Известно количество книг N ($N \leq 20$) и цена одной книги K тенге. Выведите на экран стоимость одной книги, двух, трех книг ...

№	Пример	Результат
1	3 500	1 книга 500 тенге 2 книги 1000 тенге 3 книги 1500 тенге

Задание № 11. Сложение четных цифр **

Дана последовательность чисел длиной N ($1 \leq N \leq 1000$). Необходимо найти сумму цифр в записи чисел, являющихся четными. Последовательность натуральных чисел не превышает 10^9 .

№	Пример	Результат	Пояснение
1	5 425 41 284 12 35	14 3	$2+8+4=14$ $1+2=3$

Задание № 12. Возвращение дробной части числа*

Задана последовательность дробных чисел длиной $N(1 \leq N \leq 1000)$. Записать дробную часть с точностью до 0,001. Члены ряда не превышают значение 10^5 . Воспользуйтесь оператором **frac**.

№	Пример	Результат
1	5 4.12312 3 45.01 2.1 7.00005	0.123 0.000 0.010 0.100 0.000

Задание № 13. Соседние простые числа ***

Дано число $X(1 \leq X \leq 10000)$. Найдите ближайщее простое число или выведите само число, если оно простое. Если же оба соседних числа простые, то выберите наименьшее из них.

№	Пример	Результат	Пояснение
1	31	31	2-й тест: рядом с 15, 13 и 17. По условию задания выводится 13
2	15	13	

Задание № 14. Найти член ряда ***

Вы знакомы с числами Фибоначчи. Требуется определить число по порядковому номеру. Условие: $(1 \leq X \leq 50)$. Последовательность: 0, 1, 1, 2, 3, 5, 8, 13.

№	Пример	Результат
1	11	89

Дескрипторы выполнения задания:

- читает задание, понимает смысл;
- создает 5 тестов для проверки задания;
- строит блок-схему циклического алгоритма;
- предлагает проект задания в среде Lazarus.

5.6

ПРИМЕНЕНИЕ ПРОЦЕДУРЫ И ФУНКЦИИ В ИНТЕГРИРОВАННОЙ СРЕДЕ РАЗРАБОТКИ ПРОГРАММ



Как можно использовать процедуры и функции в интегрированной среде разработки программ?



Подумай

- Каков смысл понятий «процедура» и «функция»?
- Как вы думаете, для чего применяются процедуры и функции в программировании?



Новые знания

Вы узнали, что объектно-ориентированное программирование – более усовершенствованный вид создания программы по сравнению со структурным программированием.

В определенный момент развития данной сферы сформировалось мнение о том, что для облегчения решения сложных задач их нужно разбивать на простые, внутренние задачи (подпрограммы).

Внутренние подпрограммы – это процедуры и функции, служащие для выполнения специальных операций. В среде Lazarus есть множество стандартных внутренних подпрограмм. Несмотря на это, иногда приходится создавать собственные подпрограммы, предназначенные для выполнения часто повторяющихся операций, в которых могут меняться фрагменты.

Использование любой из подпрограмм приведет к получению одинаковых результатов. Тем не менее, у них есть различия.

Процедура выполняет необходимые операции, может быть вызвана из различных частей программы по имени и списку параметров.

Функция имеет те же возможности, а также с ее помощью вычисляется значение, которое может быть использовано в выражении. Таким образом, функция является более универсальным объектом.

Описание внутренних подпрограмм начинается с ключевых слов **procedure** или **function**, после чего в скобках указывается список

параметров. Далее в записи **Функция** ставится двоеточие и указывается тип вычисляемого значения. В конце обязательно нужно ставить знак пунктуации «;». Код внутренней подпрограммы полностью записывается в операторной скобке. Для функции в коде должен быть указан тип вычисляемого значения. Запишем внутренние подпрограммы в общем виде:

```
procedure Процедура_имя(параметры);
begin
Код процедуры;
end;
```

```
function Функция_имя (параметры): результат_тип;
begin
Код функции;
Result: = результат;
end;
```

Описанные данным образом внутренние подпрограммы должны быть помещены в основную программу до того, как она будет выполнена. Если эти действия не будут выполнены, то во время компиляции на экран выводится сообщение «неизвестный идентификатор».

А проследить за этим сложно. Выход из данной ситуации заключается в том, что названия внутренних подпрограмм размещаются там же, где описаны все их данные.

Такие параметры – это список идентификаторов, отделенных друг от друга запятой, с указанием после двоеточия их типов.

Во внутренней подпрограмме параметров может и не быть. В таком случае они выполняют операции только с данными основной программы (**глобальные переменные**).

Теперь нужно ввести понятие **локальная переменная**. Локальная переменная – переменная основной программы и внутренней подпрограммы, используемая во время вызова данных и внутренних подпрограмм.

Локальные переменные процедуры и функции тоже могут описываться и использоваться только внутренними подпрограммами. Конечно, в среде программирования невозможно создавать разные проекты, используя только готовые процедуры, вы должны уметь

писать свои собственные процедуры и разрабатывать разные проекты. Для закрепления пройденного материала и более четкого понимания работы процедур и функций выполним практическую работу.

В процессе ее выполнения внимательно следите за исполнением процедур и функций, ваша задача – научиться правильно использовать их в среде программирования.



Практическая работа

Проект «Процедура и функция»

- Даны три натуральных числа. Найдите число с наибольшей суммой цифр и вычислите сумму цифр этого числа.

В процессе выполнения этого проекта, нужно вычислить сумму цифр каждого числа. В нашем примере вместо того, чтобы писать код для трех программ по вычислению суммы цифр числа, используем работу процедуры. На рисунке 1 приведен пример проекта. Конечно, это лишь образец, вы можете просчитать и выполнить его по-другому. В форме проекта используем кнопку выполнения внутренней подпрограммы (**компонент Button**) и разместим три строки для введения данных (**компонент Edit**). Также добавляем в проект два поля, в которых вычисляется число с наибольшей суммой цифр и сумма цифр этого числа. На рисунке 2 показано выполнение проекта.

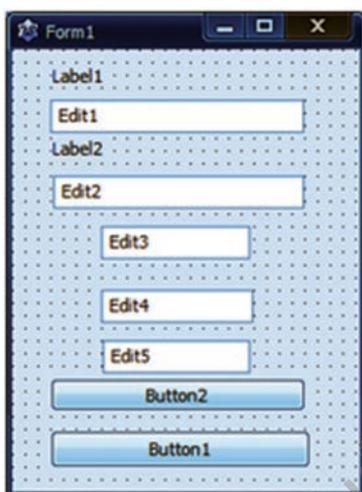


Рис. 1

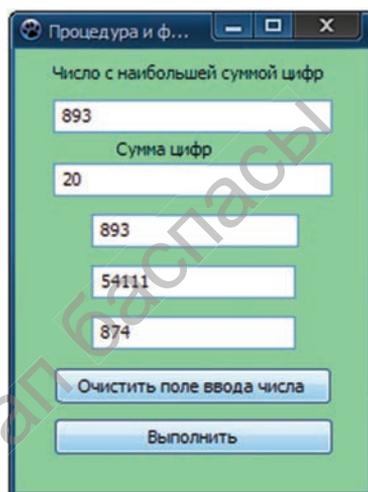


Рис. 2

Код проекта «Процедура и функции»

Код программы, описывающий работу процедуры.

```

var
Form1: TForm1;
s,k,t,a,c, b, max,result: Integer;
procedure sum(n: longint);
implementation{$R *.lfm}{ TForm1 }
procedure TForm1.Button1Click(Sender:
TObject);
begin
A:=StrToInt(Edit1.Text);
B:=StrToInt(Edit2.Text);
C:=StrToInt(Edit3.Text);
sum(a);sum(b);sum(c);
edit4.Text:=inttostr(result);
edit5.Text:=inttostr(max);
end;
procedure TForm1.Button2Click (Sender:
TObject);
begin
edit1.Text:="";edit2.Text:="";
edit3.Text:="";edit4.Text:="";
end;
procedure sum(n:longint);
begin
t:=n; s:=0;
repeat
k:=n mod 10;n:=n div 10;s:=s+k;
until(n=0);
if max < s then begin
max:=s; result:=t; end ;
end;
end.

```

Код программы, описывающий работу функции.

```

var
Form1: TForm1;
m,s,k,t, a,c, b,max: longint;
function sum(n: longint): longint;
implementation {$R *.lfm} { TForm1 }
procedure TForm1.Button1Click(Sender:
TObject);
begin
A:=StrToInt(Edit1.Text);
B:=StrToInt(Edit2.Text);
C:= StrToInt(Edit3.Text);
sum(a); sum(b); sum(c);
edit4.Text:=inttostr(m);
edit5.Text:=inttostr(max);
end;
procedure TForm1.Button2Click(Sender:
TObject);
begin
edit1.Text:=""; edit2.Text:="";
edit3.Text:=""; edit4.Text:="";
end;
function sum(n:longint): longint;
begin sum:=0; t:=n;
repeat
k:=n mod 10; n:=n div 10; sum:=sum+k;
until(n=0);
if max < sum then begin
max:=sum; m:=t; end ;
end;
end.

```



Анализ



Проанализируйте код проекта «Процедура и функции». Изучите различия в работе процедуры и функции. Какой вариант эффективнее? Почему? Приведите примеры, основанные на фактических данных.

**Вопросы**

1. Какова роль процедур и функций в программировании?
2. Как записывается общий вид процедуры?
3. Как записывается общий вид функции?
4. Как записываются операторы вызовов процедуры и функции?
5. Что такое локальная переменная?
6. В чем сходство и различия между процедурой и функцией?
7. Какие преимущества дает использование процедур и функций при написании программ?
8. Какие подпрограммы более удобны в применении: процедуры или функции?

**Задания**

1. Выведите на экран все натуральные числа, значение которых не больше N , и делящиеся на каждую из цифр своей записи. Например, цифра 12 делится на 1 и 2, а цифра 13 делится на 1, а на 2 не делится.

Пример	Результат
5	1 2 3 4 5

2. Даны вершины треугольника с координатами $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Все координаты – вещественные числа. Найдите периметр треугольника, используя формулу нахождения расстояния между двумя точками.

Пример	Результат
1 1 1 4 5 1	12

3. Найдите значение выражения $(2*4! + 3*8!)/(2! + 4!)$, в котором знак ! – факториал.

4. Определить наименьшее произведение цифр в записи чисел A и B .

Пример	Результат
121 43	121

5. Найдите наибольший общий делитель чисел A и B .

Пример	Результат
15 45	15

6. Выясните, сколько чисел находится между числами K и P , у которых хотя бы одна из цифр в их записи равна нулю ($1 \leq K \leq P \leq 1000$).

Пример	Результат
8 22	2

Создание проектов в среде программирования

Проект «Треугольник»

1. Даны координаты вершин треугольника: $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Проект должен включать следующие моменты:

- вычисление длины сторон треугольника;
- вычисление периметра треугольника;
- вычисление площади треугольника по формуле Герона;
- определение типа треугольника (равносторонний, равнобедренный, прямоугольный).

Проект «Числа»

2. Дана последовательность чисел длиной N ($1 \leq N \leq 30$). Создайте проект, который:

- выводит на экран последовательность простых чисел;
- выводит на экран сумму цифр в записи чисел, которая является четной;
- выводит на экран сумму цифр в записи числа, которая больше десяти.

Примечание. Если сложно выполнить в одном проекте все задания сразу, то можно выполнить каждое задание по-отдельности.

5.7 ТВОРЧЕСКО-ПРАКТИЧЕСКАЯ РАБОТА. СОЗДАНИЕ ПРОЕКТА



Как записывается ввод модели задачи в коде программы проекта в визуальной среде программирования?



Подумай

- Какие процессы из повседневной жизни можно смоделировать в виде компьютерной задачи?



Практическая работа

Задание № 1. «Астана – сердце Казахстана»

Наша Родина – Республика Казахстан. Мы с вами здесь родились и выросли. Казахская земля вскормила нас своим хлебом, напоила своими водами, и все здесь для нас родное и знакомое.

Много разных стран есть на планете, кроме Казахстана, но одна у человека родная мать, одна у него и Родина.

Создайте проект – альбом фотографий на тему «Астана – сердце Казахстана» в визуальной среде программирования. В вашем проекте должны быть отображены символы нашего независимого государства, а также главные достопримечательности столицы (рис. 1).

Дескрипторы для выполнения задания:

- выбирает компоненты проекта;
- использует свойства компонентов;
- вставляет фотографии города Астаны;
- записывает код программы проекта, запускает и получает результаты;
- продвигает идеи на основе выполненного задания.



Рис. 1

Задание № 2. «В дружной семье народов»

В Республике Казахстан единой, дружной семьей проживает более 18 миллионов человек, представителей более 100 национальностей.

На казахском, русском, узбекском, уйгурском, немецком, татарском и других языках издаются книги, газеты и журналы.



Рис. 2

В республике работают национальные театры. Всех нас, людей разных национальностей объединяет общая идея: Мы – народ Казахстана!

Создайте проект – альбом на тему «В дружной семье народов» в визуальной среде программирования.

В созданном вами проекте при нажатии кнопкой мыши на рисунке должна выводиться информация об истории, традициях и культуре национальностей (4-5), представители которых проживают в Казахстане (рис. 2).

Дескрипторы для выполнения задания:

- выбирает компоненты проекта;
- использует свойства компонентов;
- вставляет рисунки, изображающие культуру и традиции разных народов;
- записывает код проекта, запускает программу и получает результаты;
- продвигает идеи на основе выполненного задания.

Задание № 3. Создание проекта «Бой часов»***

Старые настенные часы каждые полчаса обозначают боем. В начале каждого часа количество ударов равно текущему времени, в дополнение к этому они совершают один удар при прохождении получасовой отметки. В час дня и в час ночи часы бьют по одному разу, в полночь и в полдень – по 12 раз.

Необходимо создать проект, рассчитывающий общее количество ударов часов в заданном промежутке времени в течение суток.

Заданы вводные числа **a**, **b**, **c** и **d**. Исходный час – **a**, исходная минута – **b**, конечный час – **c**, конечная минута – **d**. Условие: $(0 \leq a; c \leq 23; 0 \leq b; d \leq 59; b \neq 30; d \neq 30)$. Образец проекта представлен на рисунке 3.



Рис. 3

Пример	Результат
5 20 10 25	45
10 25 5 20	135

Задание №4. Создание проекта игры «Баран и козел»**

Руслан и Айя любят играть в различные логические игры. На этот раз они выбрали игру «Баран и козел» и переименовали ее по имени героев любимого мультфильма. Суть игры «Баран и козел» заключается в следующем (рис. 4). Руслан загадывает 4-значное число с неповторяющимися цифрами. Айя пробует его угадать. Руслан сообщает в ответ, сколько цифр угадано без совпадения с их позициями в тайном числе (количество баранов) и сколько цифр угадано вплоть до совпадения



Рис. 4

Пример	Результат
5671 7251	1 2
1425 1425	4 0

позиции (количество козлов). Айя вводит комбинации цифр одну за другой, пока не отгадает всю последовательность.

Дескрипторы заданий 3 и 4:

- правильно выбирает компоненты проекта;
- использует свойства компонентов;
- составляет алгоритм проекта;
- записывает код программы проекта, запускает и получает результат;
- обосновывает идеи проекта.

Задание №5. Подготовка проекта «Закон Ома и определение сопротивления проводника»

В разделе физики «Постоянный ток» значение закона Ома довольно велико. Пример проекта приведен на рисунке 5. Постройте проект для расчета цепи по закону Ома (сила тока) при данном напряжении и сопротивлении участков. Рассмотрите варианты соединения проводников – последовательное (рис. 6) и параллельное (рис. 7).

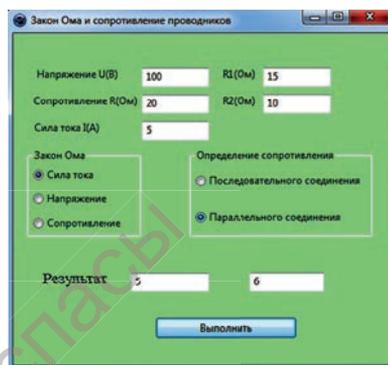


Рис. 5. Пример проекта

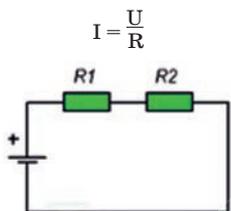


Рис. 6. Последовательное соединение

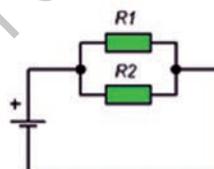


Рис. 7. Параллельное соединение

Дескрипторы для выполнения задания:

- правильно выбирает компоненты проекта;
- использует свойства компонентов;
- правильно применяет законы физики;
- составляет алгоритм проекта;
- записывает коды проекта, запускает и получает результаты;
- продвигает идеи на основе выполненного задания.

Задание № 6. Построение проекта «Времена года»

У каждого человека любимое время года. Кому-то нравится лето, кому-то – зима. Но не бывает плохих времен года. С каждым сезоном у нас связаны свои приятные воспоминания. Например, осенью – сбор урожая, зимой – катание на санях и т.д. Создайте проект, в котором при написании названия времени года появляется соответствующий рисунок в виде пейзажа или предмета. Образец проекта приведен на рисунках 8 и 9. При выполнении задания используйте свойства компонента Image. Для проекта достаточно использовать 4 рисунка с названиями времен года: «Лето», «Осень», «Зима» и «Весна».

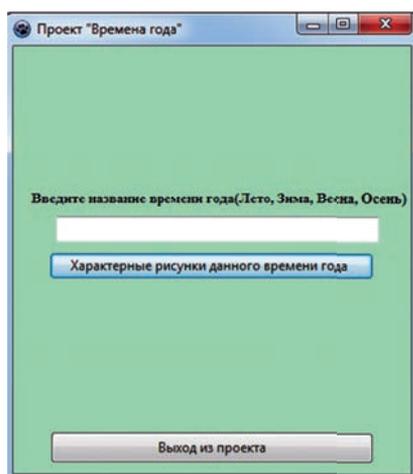


Рис. 8. Пример проекта

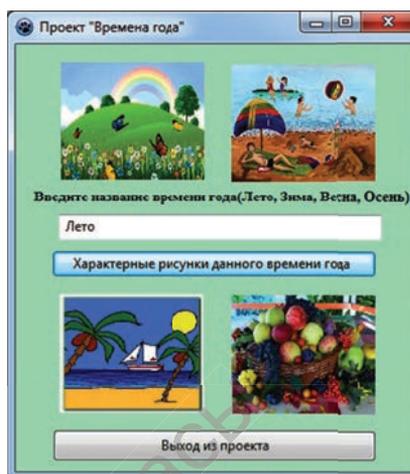


Рис. 9. Результат проекта

Дескрипторы для выполнения задания:

- правильно выбирает компоненты проекта;
- использует свойства компонентов;
- вставляет рисунок соответствующего времени года;
- записывает код проекта, запускает и получает результаты;
- продвигает идеи на основе выполненного задания.

Словарь – сөздік – dictionary

На русском языке	На казахском языке	На английском языке
Проблема	Мәселе	Problem
Вычисление	Есептеу	Calculating
Разработка алгоритма	Алгоритмді құру	Algorithm creation
Программирование	Программалау	Programming
Блок-схема	Блок-сызбалар	Flowchart
Линейный	Сызықтық	Linear
Разветвление	Тармақталу	Branching
Тест	Тест	Test
Ошибка	Қателіктер	Error
Транслятор	Транслятор	Assembler
Процедура	Процедура	Procedure
Функция	Функция	Function

ГЛОССАРИЙ

Абсолютные и относительные ссылки.

Абсолютные ссылки в формулах используются для указания фиксированного адреса ячейки. При перемещении или копировании формулы абсолютные ссылки не изменяются.

Относительные ссылки в формулах используются для указания адреса ячейки, вычисляемого относительно ячейки, в которой находится формула.

Алгоритм – это сформулированное правило, указывающее на действия, последовательное выполнение которых приводит от исходных данных к искомому результату.

Свойства алгоритма (отличающие его от любых других предписаний): понятность (для конкретного исполнителя); дискретность (команды последовательны, с точной фиксацией моментов начала и конца выполнения команды); точность (после выполнения каждой команды точно известно, завершено ли исполнение алгоритма или какая команда должна выполняться следующей); результативность (после конечного числа шагов задача решается или же становится ясно, что процесс решения не может быть продолжен); массовость (алгоритм одинаково применяется к любой конкретной формулировке задачи, для которой он разработан).

Алфавит – это множество символов, используемых для записи текста. Все множество используемых в языке символов традиционно называется алфавитом. Обычно под алфавитом понимают только буквы, но поскольку в тексте могут встречаться знаки препинания, цифры, скобки, то они тоже включены в алфавит. В алфавит также следует включить и пробел, т.е. пропуск между словами.

Аспект (от латинского *aspectus* – вид, облик, взгляд) – точка зрения, одна из сторон, с которой рассматривается какое-либо явление, понятие.

Булева функция – функция алгебры логики, аргументы которой принимают значения из двухэлементного множества (обычно $\{0, 1\}$).

Динамический ряд – совокупность однородных статистических величин, показывающих изменение какого-либо явления на протяжении определенного промежутка времени.

Информационная безопасность – это защищенность информации и поддерживающей инфраструктуры от случайных или преднамеренных воздействий естественного или искусственного характера, чреватых нанесением ущерба владельцам или пользователям информации и поддерживающей инфраструктуры.

Кибербуллинг – это травля, оскорбления или угрозы, высказываемые жертве с помощью средств электронной коммуникации (сообщений в социальных сетях, мгновенных сообщений, электронных писем и sms).

Коммутатор (жарг. свитч от англ. switch – переключатель) – устройство, предназначенное для соединения нескольких узлов компьютерной сети в пределах одного или нескольких ее сегментов.

Материнская плата (от англ. motherboard, МВ или англ. mainboard – главная плата) – печатная плата, являющаяся основой построения модульного устройства, например – компьютера.

Микропроцессор (CPU, Central Processor Unit – ЦПУ, или центральное процессорное устройство) – электронная схема, устройство обработки информации, выполненное в виде одной или нескольких больших интегральных схем. В компьютере микропроцессор выполняет функции управления и обрабатывает большую часть информации.

Мошенничество – хищение чужого имущества или приобретение права на чужое имущество путём обмана или злоупотребления доверием.

Мощность (размер) алфавита – это полное количество символов в алфавите. Мощность алфавита обозначается буквой N.

Операционная система (Operating system) – это комплекс взаимосвязанных системных программ, назначение которого – организация взаимодействия пользователя с компьютером и выполнение всех программ. В функции операционной системы входит:

- осуществление диалога с пользователем;
- ввод-вывод и управление данными;
- планирование и организация процесса обработки программ;
- распределение ресурсов (оперативной памяти и кеша, процессора, внешних устройств);

- запуск программ на выполнение;
- всевозможные вспомогательные операции обслуживания;
- передача информации между различными внутренними устройствами;
- программная поддержка работы периферийных устройств (монитора, клавиатуры, дисковых накопителей, принтера и др).

Параметр (от греч. *parametron* – отмеривающий) в математике – величина, числовые значения которой позволяют выделить определенный элемент (кривую) из множества элементов (кривых) того же рода. Например, в уравнении $x^2 + y^2 = r^2$ величина r является параметром окружности.

Прикладная программа (или приложение) – программа, предназначенная для выполнения определенных задач и рассчитанная на непосредственное взаимодействие с пользователем.

Прикладной программный интерфейс (application programming interface, API) – это описание способа, который позволяет какому-либо фрагменту программного обеспечения (ПО) обращаться к другой программе за получением сервиса. Этим сервисом может быть предоставление доступа к данным или выполнение конкретной функции.

Прогнозирование – специальное научное исследование конкретных перспектив дальнейшего развития какого-либо процесса. Точность любого прогноза обусловлена:

- объемом истинных (верифицированных) исходных данных и периодом их сбора;
- объемом неверифицированных исходных данных и периодом их сбора;
- свойствами системы, объекта, подвергающихся прогнозированию;
- методиками и подходами прогнозирования.

При возрастании совокупности факторов, влияющих на точность прогноза, он практически замещается рутинным расчетом с некоторой установившейся погрешностью. Прогнозы делятся (условно):

- по срокам – краткосрочные, среднесрочные, долгосрочные, дальнесрочные;
- по масштабу – частные, местные, региональные, отраслевые, страновые, мировые (глобальные);

- по ответственности (авторству) – личные, на уровне предприятия (организации), на уровне государственных органов.

К основным методам прогнозирования относят:

- статистические методы;
- экспертные оценки (например, метод Дельфи);
- методы моделирования;
- интуитивные (то есть выполненные без применения технических средств, экспромтом, «в уме» специалистом, имеющим опыт применения научных методов в данном типе прогнозов).

Программная система – это такая система, в которую входит программное обеспечение. В общем случае программная система помимо собственно программ содержит еще и аппаратное обеспечение, а также обычно рассматривается в окружении других программно-аппаратных систем.

Системная программа (System program) – программа, выполняемая вместе с прикладными программами, служит для управления ресурсами компьютера – центральным процессором, памятью, вводом-выводом, сетевым оборудованием.

Сопроцессор – специализированный процессор, расширяющий возможности центрального процессора компьютерной системы, но оформленный как отдельный функциональный модуль. Физически сопроцессор может быть отдельной микросхемой или может быть встроен в центральный процессор. Различают следующие виды сопроцессоров:

- математические сопроцессоры общего назначения, обычно ускоряющие вычисления с плавающей запятой;
- сопроцессоры ввода-вывода (например – Intel 8089), разгружающие центральный процессор от контроля за операциями ввода-вывода или расширяющие стандартное адресное пространство процессора,
- сопроцессоры для выполнения каких-либо узкоспециализированных вычислений.

Статистика (от латинского status – состояние дел) – отрасль знаний, наука, в которой излагаются общие вопросы сбора, измерения и анализа массовых статистических (количественных или качественных) данных; изучение количественной стороны массовых общественных явлений в числовой форме.

Тестирование (программного обеспечения) – процесс исследования, испытания программного продукта, имеющий две различные цели:

- продемонстрировать разработчикам и заказчикам, что программа соответствует установленным требованиям;
- выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.

Функция (от латинского *functio* – исполнение, осуществление) в программировании – фрагмент программного кода (подпрограмма), к которому можно обратиться из другого места программы. Функция может принимать параметры и должна возвращать некоторое значение, возможно пустое. Функции, которые возвращают пустое значение, часто называют процедурами. Функция должна быть соответствующим образом объявлена и определена.

Цикл (Cycle, loop) – разновидность управляющей конструкции в высокоуровневых языках программирования, предназначенная для организации многократного исполнения набора инструкций.

Цикл с постусловием – разновидность цикла, проверяющая условие после выполнения тела цикла. Цикл будет повторяться до тех пор, пока проверка данного условия будет давать результат «ложь» (false), то есть пока условие не выполнено. Даже если условие сразу окажется истинным, цикл выполнится хотя бы один раз.

Чипсет (англ. chipset) – набор микросхем, спроектированных для совместной работы с целью выполнения набора заданных функций. Так, в компьютерах чипсет, размещаемый на материнской плате, выполняет роль связующего компонента (моста), обеспечивающего взаимодействие центрального процессора (ЦП) с различными типами памяти, устройствами ввода-вывода, контроллерами и адаптерами ПУ, как непосредственно через себя (имея некоторые из них в своем составе), так и через другие контроллеры и адаптеры, с помощью многоуровневой системы шин.

IP-адрес (от англ. **I**nternet **P**rotocol **A**ddress) – уникальный идентификатор (адрес) устройства (обычно компьютера), подключенного к локальной сети или интернету.

Электромагнитные волны – процесс распространения электромагнитного поля в пространстве. Электромагнитная волна представляет собой процесс последовательного, взаимосвязанного изменения векторов напряженности электрического и магнитного полей, направленных перпендикулярно лучу распространения волны, при котором изменение электрического поля вызывает изменения магнитного поля, которые, в свою очередь, вызывают изменения электрического поля.

BIOS (Basic Input/Output System – базовая система ввода/вывода) – это программа, которая выполняет первоначальный запуск персонального компьютера, настройку оборудования и поддержку функций ввода/вывода.

Работа BIOS заключается в обеспечении начальной загрузки компьютера с последующим запуском операционной системы, а также в хранении аппаратной конфигурации компьютера.

Изменяя настройки BIOS, можно управлять работой компонентов компьютера.

Функции программы:

- проверка наличия и работоспособности найденного оборудования;
- поиск, загрузка и запуск операционной системы;
- установка начальных параметров оборудования.

В ситуациях, когда ошибочно были внесены неверные параметры или произошел сбой настроек, BIOS можно вернуть к исходным параметрам (обнулить).

Основные виды программы: Award BIOS (от компании Phoenix Technologies) и Ami BIOS (от American Megatrends). Они отличаются по своей структуре.

Через меню доступна настройка следующих функций BIOS:

- системные дата и время;
- выбор устройства загрузки операционной системы (актуально при установке самой ОС);
- включение или отключение поддержки USB-устройств;
- просмотр значений или изменение напряжения, подающегося на процессор и модули RAM-памяти (для увеличения их производительности);
- установка пароля на просмотр настроек BIOS.

СПИСОК ЛИТЕРАТУРЫ

1. В.Н. Корнеев, А.В. Куприянова. Microsoft Excel 2007. Просто о сложном (+ CD-ROM). Наука и техника, 2008. – 176 с.
2. В. Пташинский. Excel 2010 для начинающих. Эксмо, 2013. – 288 с.
3. Семенов В.П., Финкова М.А. Excel 2013 на примерах. Наука и техника, 2016. – 288 с.
4. В.В. Серогодский, М.А. Финкова, Р.Г. Прокди. Excel 2013. 2 в 1. Пошаговый самоучитель + справочник пользователя. Наука и техника, 2016. – 400 с.
5. Грицианова Т.Ю. Информатика. Программирование в примерах и задачах. Бином. Лаборатория знаний, 2017, 368 с.
6. Владимирович Н.А. Увлекательная информатика. 5–11 классы. Логические задачи, кроссворды, ребусы, игры, Учитель, 2017 г, 141 с.
7. Паронджанов В.Д. Учись писать, читать и понимать алгоритмы. Алгоритмы для правильного мышления. Основы алгоритмизации, ДМК Пресс, 2016 г., 520 с.
8. Кормен Томас Х. Алгоритмы. Вводный курс, Диалектика, 2014 г., 208 с.
9. Андреева Е.В. Программирование – это так просто. Программирование – это так сложно. Современный учебник программирования. МЦНМО, 2017, 184 с.
10. Алексеев Е. Р., Чеснокова О.В., Турбо Паскаль 7.0. М.: НТ Пресс, 2006. 320 с.
11. Е.Р. Алексеев, О.В. Чеснокова, Т.В. Кучер «Самоучитель по программированию на Free Pascal и Lazarus». Технопарк ДонНТУ УНИТЕХ, 2011, – 505 с.
12. Мансуров К.Т. «Основы программирования в среде Lazarus», Нобель Пресс, 2010, – 772 стр.
13. Алексеев Е.Р., Чеснокова О.В., Кучер Т.В. Free Pascal и Lazarus: Учебник по программированию / – М.: ALT Linux; Издательский дом ДМК-пресс, 2010. – 440 с.: ил.
14. Гуриков С.Р. Программирование в среде Lazarus для школьников и студентов: Учебное пособие, Инфра-М, Форум, 2017

СОДЕРЖАНИЕ

Введение	3
Раздел I. Технические характеристики компьютера и сетей	
1.1. Измерение информации. Вероятностный подход	4
1.2. Измерение информации. Алфавитный подход.....	8
1.3. Внутреннее устройство компьютера и периферийные устройства.....	12
1.4. Процессор и его характеристики	19
1.5. Компьютерные сети.....	25
Раздел II. Здоровье и безопасность	
2.1. Негативные аспекты использования компьютера.....	33
2.2. Безопасность в сети	38
Раздел III. Обработка информации в электронных таблицах	
3.1. Статистические данные. Использование абсолютных и относительных ссылок	44
3.2. Использование абсолютных и относительных ссылок в решении практических задач	50
3.3. Встроенные функции.....	55
3.4. Использование встроенных функций при решении практических задач	60
3.5. Анализ данных на основе имеющейся информации	64
3.6–3.7. Решение прикладных задач	70
3.8. Выполнение проектных работ в приложении MS EXCEL.....	78
Раздел IV. Интегрированные среды разработки программ	
4.1. Классификация программного обеспечения.....	82
4.2. Компоненты интегрированной среды разработки программ	86
4.3. Оператор выбора	91
4.4. Циклы с параметрами. Оператор FOR.....	97
4.5. Вложенные циклы.....	104
4.6. Цикл с постусловием. Оператор Repeat-until.	110
4.7. Цикл с предусловием. Оператор While.	116
4.8. Трассировка алгоритма	122
4.9. Творческо-практическая работа по теме «Циклы»	128
4.10. Создания проектных работ в среде программирования	132
Раздел V. Решение задач в интегрированной среде разработки	
5.1. Постановка задачи.....	136
5.2. Разработка алгоритма	142
5.3. Программирование алгоритма.....	149
5.4. Тестирование программы	154
5.5. Практическая работа. Построение блок-схемы задания.	161
5.6. Применения процедуры и функции в интегрированной среде разработки программ.....	165
5.7. Творческо-практическая работа. Создание проекта.	171
Глоссарий.....	176
Список литературы.....	182

Оқулық басылым Учебное издание

**Кадиркулов Роман Алауович
Нурмуханбетова Гулира Кенжебаевна**

ИНФОРМАТИКА ИНФОРМАТИКА

Жалпы білім беретін мектептің
8-сынып оқушыларына арналған оқулық

Учебник для учащихся 8 класса
общеобразовательной школы

Бас редактор *А.С. Диденко*
Спец. редактор *Б.А. Айтуллина*
Әдіскер *М.У. Кенжебаева*

Главный редактор *А.С. Диденко*
Спец. редактор *Б.А. Айтуллина*
Методист *М.У. Кенжебаева*

Көркемдеуші редактор *Т.В. Толыбекова*
Компьютерде беттеген *Р. Дюсенбаева*

Художественный редактор *Т.В. Толыбекова*
Компьютерная верстка *Р. Дюсенбаевой*

Басуға 11.06.2018 ж. қол қойылды.
Пішімі 70x100^{1/16}. Есептік баспа табағы 7,11.
Шартты баспа табағы 14,84. Офсеттік басылым.
Әріп түрі «DS SchoolBook». Офсеттік қағаз.
Таралымы 30 000 дана. Тапсырыс № 1794.

Подписано в печать 11.06.2018 г.
Формат 70x100^{1/16}. Уч.-изд.л. 7,11.
Усл.печ.л. 14,84. Печать офсетная.
Гарнитура «DS SchoolBook». Бумага офсетная.
Тираж 30 000 экз. Заказ № 1794

Сапасы жөнінде мына мекемеге хабарласыңыз:
Қазақстан Республикасы,
«Алматыкітап баспасы» ЖШС,
050012, Алматы қаласы, Жамбыл көшесі, 111-үй,
тел. (727) 250 29 58, факс: (727) 292 81 10.
e-mail: alkitap@intelsoft.kz

С претензиями по качеству обращаться:
Республика Казахстан,
ТОО «Алматыкітап баспасы»
050012, г. Алматы, ул. Жамбыла, 111,
тел. (727) 250 29 58; факс: (727) 292 81 10.
e-mail: alkitap@intelsoft.kz

Сапа және қауіпсіздік стандарттарына сай.
Сертификация қарастырылмаған.
Сақтау мерзімі шектелмеген.

Соответствует всем стандартам качества и безопасности.
Сертификация не предусмотрена.
Срок годности не ограничен.

Қытайда басылды / Отпечатано в Китае
WPS – Worldwide Printing Solutions. LTD
7/F KIN ON COMM BLDG 49, JERVOIS ST, SHEUNG WAN, HONG KONG

Приобрести книги можно в книжных магазинах ТОО «Алматыкітап баспасы»

г. Астана: ул. Иманова, 10, тел.: (7172) 53 70 84, 27 29 54;
пр. Б. Момышулы, 14, тел.: (7172) 42 42 32, 57 63 92; пр. Жеңіс, 67, тел.: (7172) 29 93 81; 29 02 12.
г. Алматы: пр. Абая, 35/37, тел.: (727) 267 13 95, 267 14 86;
ул. Гоголя, 108, тел.: (727) 279 29 13, 279 27 86; ул. Кабанбай батыра, 109, тел.: (727) 267 54 64, 272 05 66;
ул. Жандосова, 57, тел.: (727) 303 72 33, 374 98 59; пр. Гагарина, 76, тел. (727) 338 50 52;
пр. Аль-Фараби, 63/17, тел. (727) 387 17 24; ул. Майлина, 224 «А», тел.: (727) 386 15 19, 222 29 43.
ул. Толе би, 40/1, тел.: (727) 273 51 38, 224 39 37.
Управление складского учета: ул. Ангарская, 95 А, тел.: (727) 290 49 67, 290 49 66.

Интернет-магазин www.flip.kz
Отдел маркетинга, тел.: (727) 292 92 23, 292 57 20.
e-mail: sale1@almatykitap.kz

Об имеющихся книгах и новинках
вы можете узнать на сайте www.almatykitap.kz

