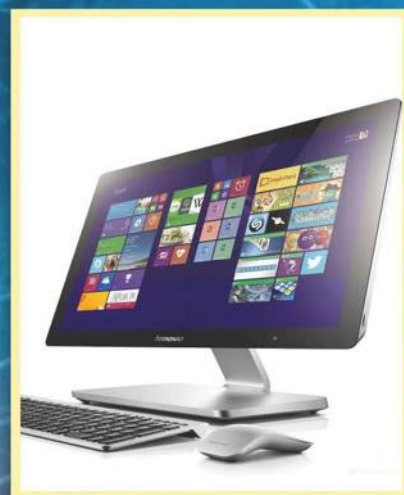
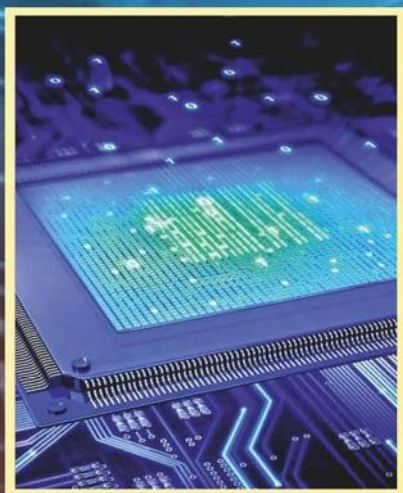


ИНФОРМАТИКА



8

С. Т. Мухамбетжанова, А. С. Тен, Н. Т. Маликова

ИНФОРМАТИКА

Учебник для 8 класса общеобразовательной школы

8

Рекомендовано Министерством образования и науки
Республики Казахстан



Алматы «Атамұра» 2018

УДК 373.167.1
ББК 32.973.202..я72
М 74

*Учебник подготовлен в соответствии
с Типовой учебной программой по учебному предмету
«Информатика» для 5–9 классов уровня основного среднего
образования по обновленному содержанию*

Условные обозначения учебника



Самое главное по изучаемой теме



Вопросы и задания для самообучения



Вопросы и задания для самоконтроля



Практикум



Исследование



Домашнее задание



Для тех, кто хочет знать больше



Индивидуальная работа



Парная работа



Групповая работа



Работа класса

Мухамбетжанова С.Т. и др.
М 74 Информатика: Учебник для 8 кл. общеобразоват. шк. / С. Т. Мухамбетжанова, А. С. Тен, Н. Т. Маликова – Алматы: Атамұра, 2018. – 192 с.

ISBN 978-601-331-158-6

УДК 373.167.1
ББК 32.973.202..я72

ISBN 978-601-331-158-6

© Мухамбетжанова С.Т., Тен А.С.,
Маликова Н.Т., 2018
© «Атамұра», 2018

Содержание

Раздел 1

Технические характеристики компьютера и сетей

1.1. Измерение информации	7
1.2. Процессор и его характеристики.....	12
1.3. Компьютерные сети	19
Дополнительные задания к первому разделу	25
Итоговый тест к первому разделу	25

Раздел 2

Здоровье и безопасность

2.1. Негативные аспекты использования компьютера	28
2.2. Безопасность в сети	34
Итоговый тест ко второму разделу	40

Раздел 3

Обработка информации в электронных таблицах

3.1. Статистические данные.....	42
3.2. Встроенные функции	52
3.3. Анализ данных на основе имеющейся информации	62
3.4. Решение прикладных задач	73
Дополнительные задания к третьему разделу.....	87
Итоговый тест к третьему разделу	89

Раздел 4

Интегрированные среды разработки программ

4.1. Классификация программного обеспечения.....	97
4.2. Компоненты интегрированной среды разработки программ	105
4.3. Оператор выбора.....	111
4.4. Цикл с параметром.....	121
4.5. Цикл с постусловием.....	127
4.6. Цикл с предусловием	133
4.7. Трассировка алгоритма	138
Дополнительные задания для программирования на языке C ++	147
Итоговый тест к четвертому разделу	147

Раздел 5

Решение задач в интегрированной среде разработки

5.1. Постановка проблемы.....	155
5.2. Разработка алгоритма	160
5.3. Программирование алгоритма.....	166
5.4. Тестирование программы.....	173
Итоговые задания к пятому разделу	179
Приложение	182
Глоссарий	186
Рекомендуемая литература	190
Рекомендуемые интернет-ресурсы	190

Дорогие восьмиклассники!

Учебник информатики для 8 класса написан в соответствии с Государственным общеобязательным стандартом образования. Основными целями изучения предмета информатики по обновленному содержанию образования являются:

- формирование базовых ИКТ-навыков для решения прикладных задач из различных предметных областей, позволяющих полученные знания, умения и навыки применять в окружающей среде, в практике;
- развитие компетенций: креативности, коммуникабельности и решение проблем при выполнении заданий исследовательского, экспериментального, проектного характера.

В процессе реализации целей обучения вы научитесь измерять информацию с помощью алфавитного подхода, рассчитывать пропускную способность сети, преобразовывать информацию с помощью компьютеров и т.д.

Прикладной характер учебника информатики позволит формировать и развивать навыки эффективного и правильного использования информационных технологий (электронной таблицы) как в учебной деятельности, так и в повседневной жизни.

Основное внимание в учебнике уделяется не только изучению методов и процессов преобразования информации с помощью компьютеров, но и умению соблюдать этикет и безопасность работы в Интернете, эффективно защищать себя от электромагнитного излучения компьютера и других электронных устройств.

Принцип «спиральности» учебника позволит рассматривать некоторые темы и цели обучения снова и снова. Но это будет происходить с постепенным углублением, усложнением и наращиванием объема знаний и навыков в течение учебного года. Так, например, изучение и решение задач в интегрированной среде разработки программ C++, начатые в 7 классе, будут продолжены и углублены в 8 классе.

Вы должны помнить, что без программирования были бы невозможны все передовые технологии – от квантовых компьютеров до космических программ. В реальной жизни не все вы станете программистами.

Но умение писать программы поднимет вас на более высокий уровень – творческий, интеллектуальный.

Весь учебный материал структурирован и разделен на разделы. Каждый раздел состоит из параграфов. К каждой теме приведены задания, ранжированные по шести разделам таксономии Б. Блума: знание, понимание, применение, анализ, синтез и оценивание. Разноуровневые задания в свою очередь распределены по уровням мышления и развития. **Низкий уровень** включает задания на «Знание и понимание». Задания на «Применение» и «Анализ» относятся к **среднему уровню**. Соответственно, **высокий уровень** включает задания из разделов «Синтез» и «Оценивание».

Прикладные задания помогут вам развивать функциональную грамотность, то есть критически работать с информацией (проблемой), представленной в таблицах, графиках, инфографике, быстро понимать ее, извлекать, анализировать, осмысливать, интерпретировать, аргументировать и оценивать.

Для систематической организации работы в учебнике предусмотрены обозначения: индивидуальная, парная, групповая работа и работа в классе.

Кроме того, рекомендации для выполнения домашних заданий позволят облегчить работу дома. Рубрика «Для тех, кто хочет знать больше» позволит познакомиться с дополнительным и интересным материалом для саморазвития.

Таким образом, учебник реализует связь обучения с жизнью и практикой, актуализирует знания в учебной деятельности и жизненных ситуациях.

Желаем вам успехов в изучении самой интересной и современной науки – информатики!

Авторы

Раздел 1

Технические характеристики компьютера и сетей

1.1. Измерение информации

Узнаете

как применять алфавитный подход при определении количества информации.

Ключевые слова

Алфавитный подход
Количество
информации

Ақпараттық
тәсілдеме
Ақпарат мөлшері

Alphabetical approach
Quantity of
information

Единица измерения информации была определена в науке, которая называется **теорией информации**. Для измерения различных величин применяют эталонные единицы измерения, как показано на рисунке 1.

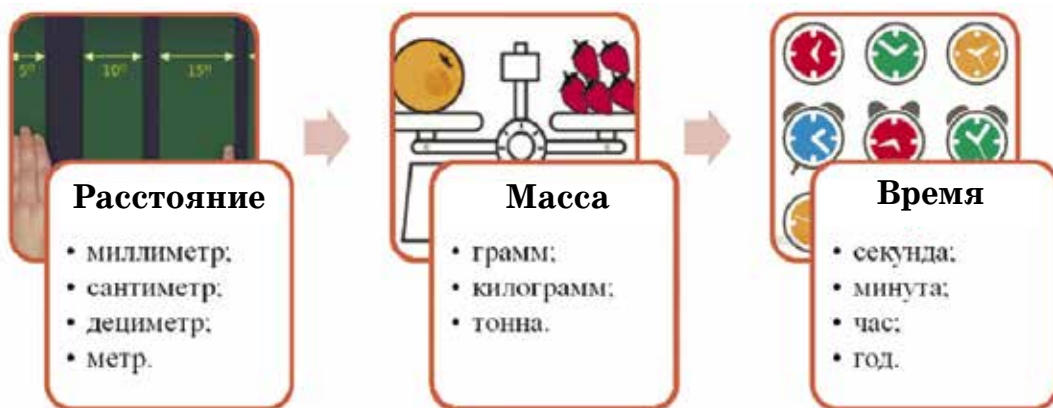


Рис. 1. Эталонные единицы измерения

Для измерения информации введена своя эталонная единица – **бит**. Восемь последовательных битов составляют **байт**. В одном байте можно закодировать значение одного символа из 256 возможных ($256 = 2^8$).



Какие более крупные единицы измерения количества информации используются наряду с битами и байтами?



Определить количество информации можно по-разному. Существуют два подхода к измерению информации: **содержательный** и **алфавитный** (рис. 2).

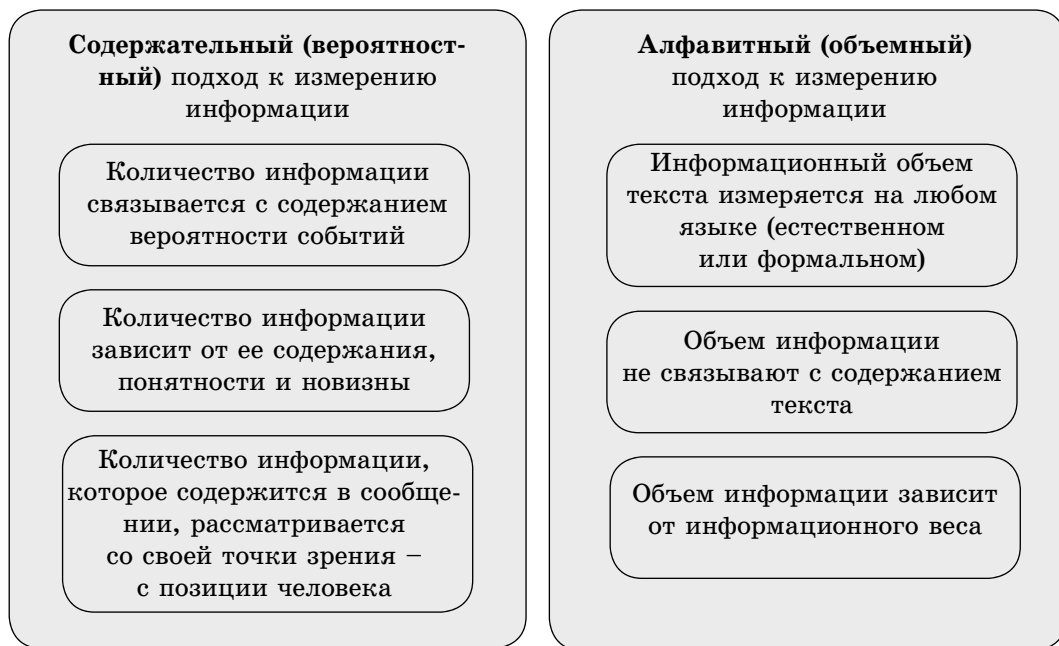


Рис. 2. Подходы к измерению информации

Алфавитный подход является объективным способом измерения информации, в отличие от субъективного, содержательного подхода.

Рассмотрим подробнее алфавитный подход. При нем количество информации в сообщении зависит от объема текста и от мощности алфавита, а не от содержания.



Что такое алфавит? Входят ли в алфавит, кроме букв, знаки препинания, цифры, скобки, пробелы и другие символы, используемые в тексте?



Множество используемых символов в языке называется **алфавитом**.

Количество символов в алфавите (его размер) называется **мощностью**.

Наименьшую мощность имеет алфавит, используемый в компьютере (машинный язык). Его называют **двоичным алфавитом**, и он содержит только два знака: **0** и **1**.

При алфавитном подходе считают, что каждый символ текста имеет информационный вес. Информационный вес символа зависит от мощности алфавита. С увеличением мощности алфавита увеличивается **информационный вес** каждого символа. Для измерения объема информации необходимо определить, сколько раз информация, равная 1 бит, содержится в определяемом объеме информации.



Для примера придумаем четырехзначный алфавит (рис. 3):



Четырехзначный алфавит				
Символ				
Порядковый номер	1	2	3	4
Двоичный код	00	01	10	11

Рис. 3. Четырехзначный алфавит

Для кодирования символов алфавита, мощность которого равна четырем, понадобятся два символа двоичного кода. Следовательно, каждый символ четырехзначного алфавита весит 2 бит.

Так как каждый символ несет i бит информации, то количество информации в одном символе можно определить из уравнения: $2^i = N$ (рис. 4).

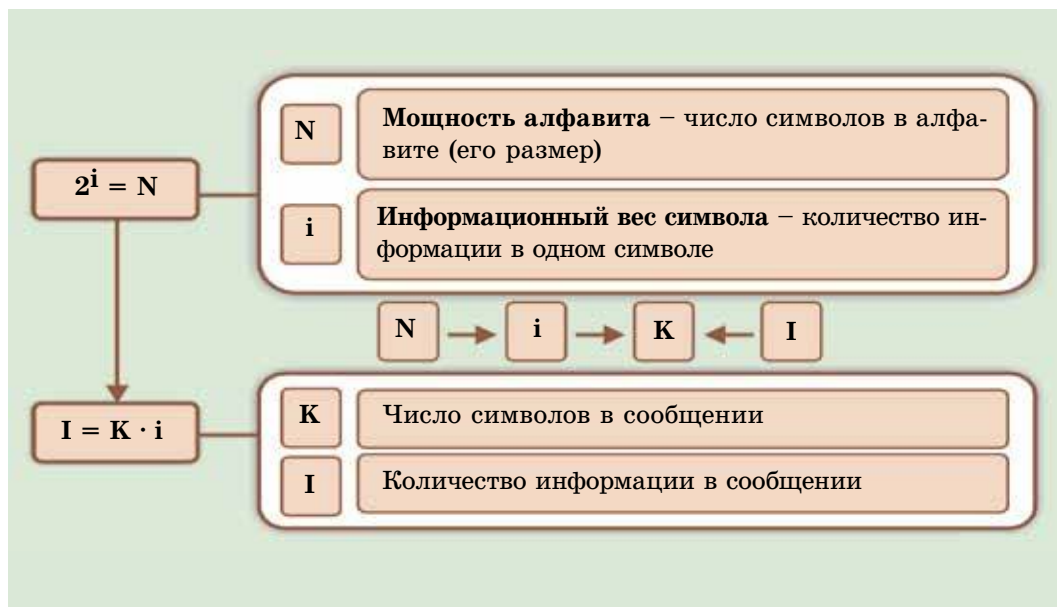


Рис. 4. Алфавитный подход к измерению информации



Рассмотрим пример расчета мощности алфавита. Текстовый файл содержит 15 Кбайт информации. Определите мощность алфавита.



Решение.

Объем информации будет равен: $15 \cdot 1024 = 15 \cdot 2^{10}$ (байт) $= 15 \cdot 2^{10} \cdot 2^3$ (бит). Тогда на 1 символ приходится $(15 \cdot 2^{10} \cdot 2^3) : (15 \cdot 2^{10}) = 2^3$ (бит) $= 8$ (бит). Мощность алфавита будет равна: $N = 2^8 = 256$.

Алфавит мощностью 256 символов несет в себе 8 бит информации (или 1 байт).

Применение алфавитного подхода удобно при использовании технических средств работы с информацией.



Знание и понимание



1. Какие два подхода к измерению информации вы знаете?
2. Что называется алфавитным подходом?
3. Что означает понятие «мощность алфавита»?
4. Как определяется информационный объем текста при использовании алфавитного подхода?
5. Где удобно применение алфавитного подхода?



Применение



1. Закодируйте с помощью двоичного кода каждый символ алфавита, мощность которого равна 8 (табл. 1.). Каков информационный объем каждого символа восьмизначного алфавита?

Таблица 1

Восьмизначный алфавит							
Порядковый номер	1	2	3	4	5	6	7
Двоичный код	000						

2. Закодируйте с помощью двоичного кода каждый символ алфавита, мощность которого равна 16. Какой вывод можно сделать?
3. Алфавит содержит 32 буквы. Какое количество информации несет одна буква?



Анализ



Сообщение, записанное буквами 32-символьного алфавита, содержит 140 символов. Какое количество информации оно несет?



Синтез



В статье, созданной с помощью текстового процессора Word, содержится 30 страниц, на каждой странице – 40 строк, в каждой строке – 50 символов. Какой объем информации содержит статья?

Рекомендации для выполнения домашнего задания

1. Узнайте, сколько на каждой странице символов.
2. Узнайте, сколько символов во всей статье.

	3. Вес каждого символа компьютерного алфавита равен 8 бит. 4. Вычислите информационный объем всей статьи. Оценивание  Первое письмо состоит из 50 символов 32-символьного алфавита, а второе – из 40 символов 64-символьного алфавита. Сравните объемы информации, содержащиеся в двух письмах.
	Эксабайт – единица измерения количества информации, равная 10^{18} байт. Аналитики из Калифорнийского университета утверждают, что человечеству потребовалось 300 тысяч лет, чтобы создать первые 12 эксабайт информации. Вторые 12 эксабайт были созданы всего за два года.

1.2. Процессор и его характеристики

Узнаете

о функциях процессора и его основных характеристиках.

Ключевые слова

Разрядность
Тактовая частота
Регистры памяти

Разрядтілігі
Тактілік жиілік
Жад регістрлері

Bit depth
Clock frequency
Memory registers

Современные процессоры – это сложнейшие устройства, которые изготавливает человек. Процессор является основной микросхемой компьютера и представляет собой небольшую плоскую полупроводниковую пластину сложной структуры (рис. 5).

Для любого компьютера, ноутбука, моноблока, смартфона, планшета основным источником системы является процессор. Именно процессор

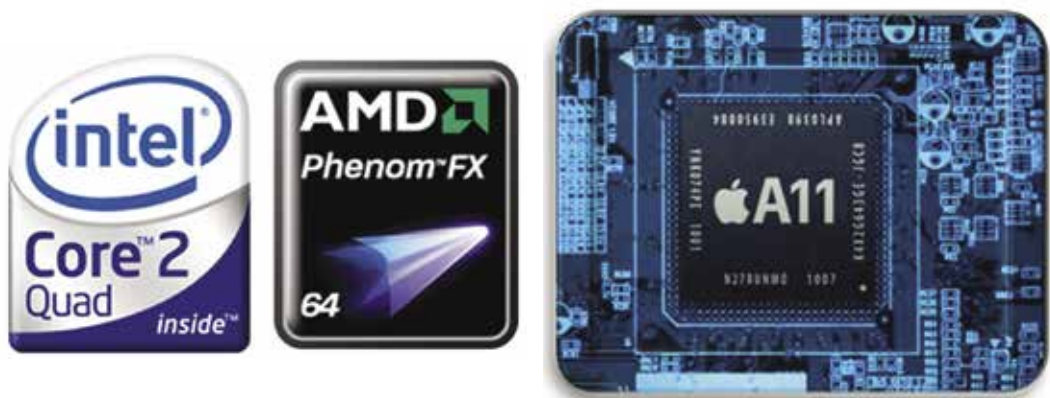


Рис. 5. Процессоры Intel, AMD, Apple

отвечает за выполнение задач, и от него зависит большая часть работы устройства. Современные компьютеры, состоящие из нескольких ядер центрального процессора, выполняют разные задачи независимо друг от друга.

В настоящее время лидерами по выпуску процессоров являются фирмы Intel (INTEgrated ELectronics) и AMD (Advanced Micro Devices). Известная компания Apple ведет разработку собственных графических процессоров, которые используются в мобильных устройствах: смартфонах iPhone и планшетах iPad.

Кроме перечисленных известных компаний, существуют другие, такие как Cyrix, Texas Instruments, конкурирующие с корпорацией Intel.



Процессор (центральный процессор, Central Processing Unit) – это микросхема, обрабатывающая информацию и координирующая работу всех устройств компьютера.

Для краткого обозначения центрального процессора принята аббревиатура ЦП. На английском языке принято обозначение **Central Processing Unit (CPU)**, которое переводится как «центральное обрабатывающее устройство».



Дорисуйте недостающие модули в функциональной схеме компьютера (рис. 6).



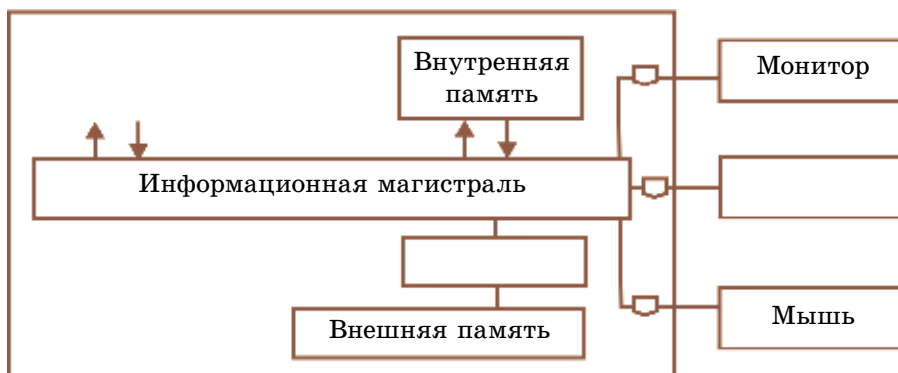


Рис. 6. Функциональная схема компьютера

Процессор состоит из десятков миллионов транзисторов. В основе внутренних схем процессора, образующих ядро процессора, лежат (рис. 7):

- устройство управления (координирует работу всех устройств компьютера);
- арифметико-логическое устройство (выполняет команды программы, находящиеся в оперативной памяти);
- внутренняя память (регистры памяти – ячейки, в которые по очереди помещаются команды программы для работы процессора);
- шины данных, команд и адресов (по этим магистралям происходит обмен данными между внутренними и внешними устройствами процессора);
- кэш-память (сверхоперативная память).

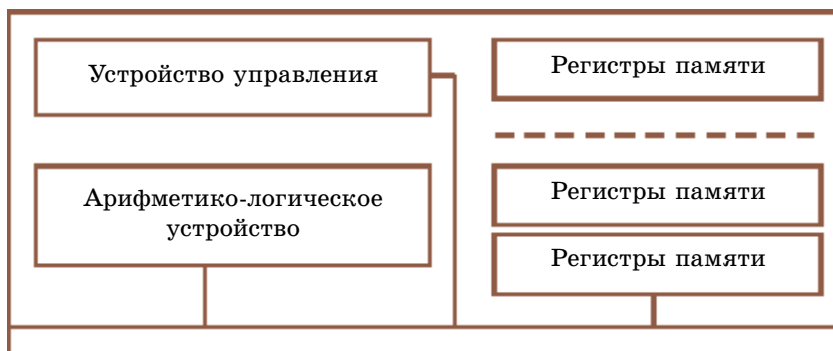


Рис. 7. Внутренняя схема процессора

Функционирование процессора происходит следующим образом: вся входящая информация, представленная в виде данных и команд, попадает из оперативной памяти в процессор через внешние шины. Обработка

данных происходит в арифметико-логическом устройстве в соответствии с командами. Результат выводится при помощи устройств вывода.

Скорость работы процессора зависит от основных характеристик:

- регистров – разрядности;
- тактовой частоты;
- адресного пространства.



Регистр – это ячейка процессора, в которой хранится машинное слово. Машинное слово представляет собой некоторое число (или команду), записанное в двоичном коде.

Существует много разнообразных процессоров, и у каждого свои регистры. Регистры различаются по размеру и по назначению. Бывают регистры 8-, 16-, 32-, 64-разрядные. Это значит, что в регистр помещаются соответственно 8, 16, 32, 64 бит двоичной информации. Размер регистра определяет одну из характеристик процессора – **разрядность**.



Разрядность процессора – это число одновременно обрабатываемых процессором битов. Поэтому процессор может быть 8-, 16-, 32-, 64-разрядным.

От разрядности центрального процессора зависит производительность компьютерной системы. Поэтому, чем больше разрядность процессора, тем большее количество информации обрабатывает процессор за единицу времени.

Первые модели процессоров были 16-разрядными и предназначались для персональных компьютеров IBM PC 80286. Следующими моделями процессора стали 32-разрядные. В настоящее время в производстве компьютеров используются 64-разрядные процессоры.

Кроме разрядности процессора, важную роль играет **тактовая частота**. Единицей измерения тактовой частоты является **мегагерц (МГц)**. Один мегагерц – это миллион тактов в секунду. Соответственно 1000 мегагерц (1 гигагерц) – это миллиард тактов в секунду.



Тактовой частотой называют количество тактов (операций), выполненных процессором в секунду. Чем выше тактовая частота процессора, тем выше его производительность.

Процессор не только исполняет машинные функции, но и управляет обменом данных между внешней и оперативной памятью.



С какой тактовой частотой работали первые модели процессоров Intel (i8088)? Какова тактовая частота у современных компьютеров?



Процессор управляет не только работой всех устройств компьютера, но и организывает обмен данными между внешней и оперативной памятью. Как происходит этот обмен? Для того чтобы найти в оперативной памяти нужные данные, процессор должен знать их адрес. А адрес к нему передается по **адресной шине**.

Если шина является N-разрядной, то по ней можно передать 2^N двоичных чисел. Именно такое количество адресов можно передать по этой шине.



Величина 2^N – это объем адресного пространства процессора, то есть объем оперативной памяти, физически адресуемой процессором.

Зная разрядность шины, можно легко вычислить величину адресного пространства процессора. Например, если адресная шина 24-разрядная, то адресное пространство равно $2^{24} = 16\,777\,216$ байт = 16 Мбайт. То есть процессор имеет адресуемый доступ к 16 Мбайт оперативной памяти.



Знание и понимание

1. Как устроен процессор?
2. Каково назначение процессора?
3. Как процессор управляет работой всех устройств персонального компьютера?
4. Каковы основные характеристики процессора?
5. Каким образом тактовая частота влияет на скорость работы процессора?
6. Как вы понимаете понятие «адресное пространство»?
7. По какой формуле можно вычислить адресное пространство?



Применение

1. Представьте, что вы работаете консультантом в компьютерном магазине. С каким процессором компьютер вы порекомендовали бы покупателю (табл. 2)?

Таблица 2

Вид деятельности	Процессор	Причина выбора
Распечатка текста, создание простейших рисунков, табличные расчеты		
Компьютерные игры		
Профессиональная работа с графикой и звуком (видеомонтаж)		

- Процессор Intel Core i7-6700K 4Ч4.0GHz (Skylake);
- Процессор Intel Core i5-6600K 3.5GHz LGA1151 OEM;
- Процессор Intel Core i3-7100 Kaby Lake 3.9 GHz LGA1151 OEM;
- Процессор Intel Celeron G1840 2.80GHz LGA1150 OEM;
- Процессор AMD FX-6300 BE Vishera 4.1;
- Процессор Intel Pentium MMX 233МГц;
- Процессор Intel Pentium II 400Гц;
- Процессор Intel Celeron 800МГц;
- Процессор Intel Pentium IV 3.5 МГц;
- Процессор Intel Pentium IV 2Гц.

2. Найдите в прайс-листе (по возможности используйте поисковые системы Интернета) характеристики процессоров, указанных в задании 1, и расшифруйте их.



Анализ

Проблемная ситуация: Как ускорить работу процессора?

Допустим, вы – рабочая группа и должны работать в помещении 1. Исходные данные и условие задачи находятся в помещении 2.

Выдача необходимой информации происходит очень медленно. В помещении 2 за порцией данных может ходить только один человек. Он приносит данные в помещение 1.

Рабочая группа обрабатывает данные и посылает за следующей порцией информации. Во время ожидания группа бездействует.

1. За счет чего можно ускорить этот процесс? Можно ли посылать за информацией в помещение 2 не одного человека, а несколько?

2. Смоделируйте процесс ускорения работы процессора.

Для этого обозначьте: помещение 1 – процессор, помещение 2 – оперативная память.



Синтез

Дополните таблицу 3 «Поколения процессоров» с 1978–2017 годы. (По возможности используйте данные из Интернета или других источников.)

Таблица 3

Процессоры

Год выпуска	Тип	Частота (МГц)	Шина данных	Адресное пространство
1978	8086	4–12	16	1 Мбайт
1982	80 286	8–20	16	16 Мбайт
1985				
1989	80 486	33–50	32	4 Гбайт
1993	Pentium	60–300	64	4 Гбайт
1997				
1999	Pentium III	450–1000	64	4 Гбайт
2000	Pentium 4	1000–2000	64	4 Гбайт
2008				
2011				
2014				
2015				
2017				
2018				



Оценивание

Постройте генеалогическое древо семейства персональных компьютеров в зависимости от характеристик процессора (используйте таблицу 3).

Рекомендации для выполнения домашнего задания

Генеалогическое древо – схематичное представление родственных связей в виде условно-символического «дерева». У «корней» указывается родоначальник. На «стволе» – представители основной (по старшинству) линии рода, на «ветвях» – различные линии родословия, известные его потомки – «листья».



Современные процессоры имеют возможность многоканальной работы с оперативной памятью, благодаря которым повышается их функциональный уровень.

У каждого процессора своя система команд, поэтому один и тот же код для разных процессоров может обозначать разные команды. Примером семейства процессоров являются все процессоры Intel. Их родоначальником является процессор Intel 8086.

Все процессоры можно разделить на:

- 1) процессоры с расширенной системой команд (CISC-процессоры);
- 2) процессоры с сокращенной системой команд (RISC-процессоры).

Первый тип процессоров характеризуется небольшим числом регистров, большим числом команд и медленной работой. Они используются в универсальных вычислительных системах и являются стандартом для микрокомпьютеров.

Второй тип процессоров характеризуется меньшим количеством команд и высоким быстродействием. Используются в специализированных вычислительных системах, ориентированных на выполнение однообразных операций.

Компьютерные технологии развиваются с большой скоростью. Что будет с компьютерами в будущем? Допустим, если производство микропроцессоров будет развиваться по закону Мура, то вычислительная мощность компьютеров должна удваиваться каждые два года. Тогда через 100 лет компьютеры будут в $1\,125\,899\,906\,842\,624$ раза мощнее, чем сегодня.

1.3. Компьютерные сети

Научитесь

определять пропускную способность сети.

Ключевое слово

Пропускная способность сети

Желінің өткізу қабілеті

Network bandwidth

Общая схема передачи информации включает в себя три объекта (рис. 8):



Рис. 8. Общая схема передачи информации

По **каналам передачи информации** происходит обмен данными. Каналы передачи информации могут использовать различные физические принципы. Например, при общении людей информация передается с помощью звуковых волн. При разговоре по телефону происходит распространение сигналов по линиям связи.



Канал связи – технические средства, позволяющие осуществлять передачу данных на расстоянии.

В компьютерной сети обмен информацией происходит с использованием каналов связи: **кабельных, оптоволоконных, радиоканалов** и т.д. Информация в сети передается с определенной скоростью. Что такое скорость передачи информации?



Скорость передачи информации – количество информации, передаваемое за единицу времени.

$$V = q \cdot t ,$$

где **V** – объем переданной информации, **q** – пропускная способность канала, **t** – время передачи информации.

Основной характеристикой каналов передачи информации является их **пропускная способность**.

Пропускная способность измеряется в битах в секунду (бит/с) и кратных единицах – **Кбит/с** и **Мбит/с**, а когда в качестве единицы измерения используется байт в секунду (байт/с), то – в кратных ему единицах – **Кбайт/с** и **Мбайт/с**.

Соотношения между единицами пропускной способности канала передачи информации такие же, как и между единицами измерения количества информации:

1 байт/с = 2^3 бит/с = 8 бит/с
1 Кбит/с = 2^{10} бит/с = 1024 бит/с
1 Мбит/с = 2^{10} Кбит/с = 1024 Кбит/с
1 Гбит/с = 2^{10} Мбит/с = 1024 Мбит/с



За 30 секунд по каналу связи передано 1500 байт информации. Чему равна пропускная способность канала?



Для решения воспользуемся формулой вычисления пропускной способности по каналу: $q = I/t$.

$$q = I/t = 1500/30 = 50 \text{ байт/с.}$$



Необходимо передать 200 страниц текста в 30 строк. В одной строке находятся 60 символов. Каждый символ кодируется 1 байтом. Сколько секунд потребуется модему, передающему сообщение с пропускной способностью 28 800 бит/с?



Решение.

Вычислим объем файла в битах:

$$V = 200 \cdot 30 \cdot 60 \cdot 8 \text{ (бит)} = 2\,880\,000 \text{ (бит)}.$$

Пропускная способность:

$$q = 28\,800 \text{ (бит/с)}.$$

Тогда время равно:

$$t = V/q = 2\,880\,000/28\,800 = 100 \text{ (секунд)}.$$

Ответ: время, которое потребуется модему, составляет 100 секунд, или 1 минуту 40 секунд.

Величина пропускной способности является важной характеристикой сети. Например, типичная сеть может быть спроектирована и реализована с полосой пропускания 100 Мбит/с. Но это не означает, что каждый пользователь сможет за одну секунду передавать данные объемом 100 Мбит/с. Такое возможно только в идеальной ситуации.

Под **пропускной способностью** понимается реальная полоса пропускания, измеренная в определенное время. Пропускная способность сети часто оказывается значительно меньше, чем максимально возможная цифровая полоса пропускания.

Ниже приведены основные факторы, определяющие реальную полосу пропускания, или пропускную способность сети (рис. 9).

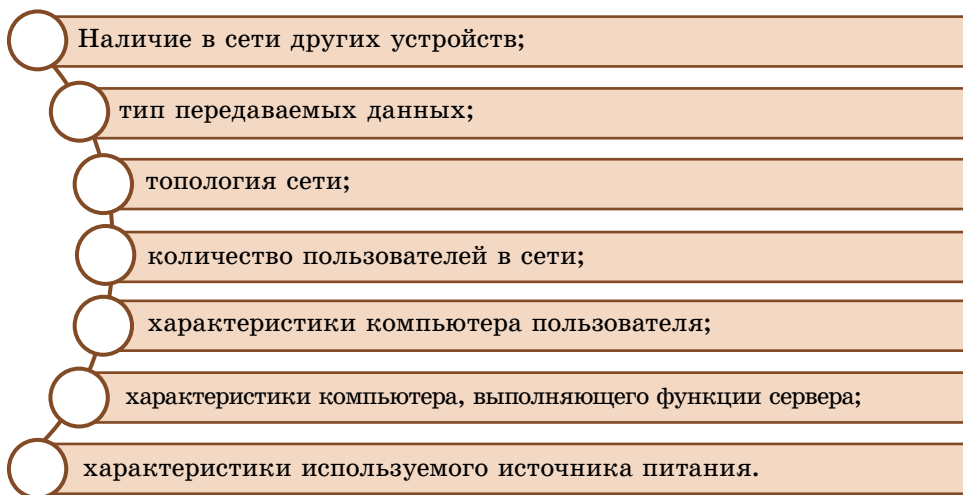


Рис. 9. Факторы пропускной способности сети



Знание и понимание



1. Какова общая схема передачи информации?
2. С помощью чего осуществляется передача данных на расстояние?
3. С помощью каких каналов связи происходит обмен информацией в компьютерной сети?
4. Что вы понимаете под пропускной способностью канала?
5. По какой формуле вычисляется объем передаваемой информации?
6. Каким образом можно измерить пропускную способность сети?
7. Назовите единицы измерения пропускной способности.
8. Перечислите основные факторы пропускной способности.



Применение



1. Скорость передачи данных через ADSL-соединение (вид модема) равна 128 000 бит/с. Через данное соединение передают файл размером 625 Кбайт. Определите время передачи файла в секундах.
2. Скорость передачи данных через ADSL-соединение равна 1 024 000 бит/с. Передача файла через данное соединение заняла 5 секунд. Определите размер файла в килобайтах.



Анализ

Текст подготовлен для передачи по сети и содержит 512 000 символов. Каждый символ кодируется 2 байт и во избежание искажений передается трижды. Время передачи текста составило 64 секунды. Какова скорость передачи в байтах в секунду?

Синтез

Скорость передачи данных через модемное соединение равна 512 000 бит/с. Передача текстового файла через это соединение заняла 10 с. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в 16-битной кодировке Unicode.

Оценивание

Нужно передать цветное растровое изображение размером 640×480 пикселей. Цвет каждого пикселя кодируется 3 байт. Сколько секунд потребуется модему, передающему сообщения со скоростью 28 800 бит/с?

Рекомендации для выполнения задания

1. Подсчитайте информационный объем информации, применив формулу: $I = K \cdot i$.
2. Из формулы вычисления пропускной способности канала $q = I/t$ найдите время (t), необходимое модему для передачи сообщения.



В результате реализации Государственной программы «Цифровой Казахстан» в 2021 году:

- доля пользователей сети Интернет составит 81%;
- уровень цифровой грамотности населения – 81,5%;
- рост производительности труда с ИКТ – 5,9%.

К 2020 году в Казахстане запланирован запуск беспроводной сети пятого поколения. С помощью сетей 5G Интернет проникнет во все сферы деятельности и быта человека.

Впервые в 2018 году новые технологии протестировали жители и гости Южной Кореи во время зимних Олимпийских игр. Через год тестовый контроль сетей 5G пройдет в Японии во время Летних игр.

5G – это стандарт беспроводной связи нового поколения. Он обеспечивает более высокую скорость, низкий уровень задержек и увеличенную пропускную способность.

Платформа нового поколения положит начало эпохе цифрового слияния. Это произойдет благодаря возможности подключения сотен миллиардов устройств и датчиков к Интернету.

Задание для проектной работы

Проект 1

Название проекта: Процессор и его характеристики.

Цель: развитие творческого мышления при систематизации знаний о процессоре и его характеристиках.

Задачи:

- систематизируйте информацию о работе процессора и его характеристиках;
- выявите особенности работы процессора и его характеристики;
- презентуйте проект.

Этапы проведения работы:

№	Содержание работы	Что делать?
1	Выявите тему и цель проекта	Обсудите тему с учителем и дополните
2	Найдите источники информации и обсудите	Закрепите план работы
3	Анализ информации и заключение	Анализ указанной работы
4	Презентуйте результаты	Презентация выполненной работы в группе
5	Рефлексия	Оценивание

Рекомендации к выполнению проекта: в виде «Генеалогического древа» начертите схему и характеристики процессора.

Требования к выполнению презентации:

1. Создайте «Генеалогическое древо» в программе Microsoft Power Point.
2. В корне «Генеалогического древа» укажите «основу».
3. В стволе – «объем памяти».
4. В стеблях – «принципы работы».
5. В листьях – вид процессора.
6. Вставьте соответствующие графические изображения, которые можно сфотографировать, нарисовать или применить готовые.
7. Оформите презентацию. Проявите творчество.

Дополнительные задания к первому разделу

1. Книга, набранная с помощью компьютера, содержит 150 страниц. На каждой странице – 40 строк, в каждой строке – 60 символов. Каков объем информации в книге в килобайтах и мегабайтах?

2. Сообщение, записанное буквами из 64-символьного алфавита, содержит 20 символов. Какой объем информации оно несет?

3. Сколько символов содержит сообщение, записанное с помощью 16-символьного алфавита, если его объем составил $1/16$ часть мегабайта?

4. Объем сообщения, содержащего 2048 символов, составил $1/512$ часть мегабайта. Каков размер алфавита, с помощью которого записано сообщение?

Итоговый тест к первому разделу

1. Укажите наиболее полный перечень основных устройств персонального компьютера:

- А) микропроцессор, сопроцессор, монитор;
- В) арифметико-логическое устройство, устройство управления, сопроцессор;
- С) процессор, оперативная память, устройства ввода-вывода;
- Д) процессор, оперативная память.

2. Тактовая частота процессора – это:

- А) число двоичных операций, совершаемых процессором в единицу времени;
- В) количество тактов, вырабатываемых за одну секунду;
- С) число возможных обращений процессора к оперативной памяти в единицу времени;
- Д) скорость обмена информацией между процессором и устройствами ввода-вывода;
- Е) обмен информацией между процессором и устройствами ввода-вывода.

3. Информационный объем информации при алфавитном подходе зависит от:

- А) суммы информационных весов составляющих его символов с учетом вероятности появления символа алфавита;
- В) суммы символов, составляющих текст;

- С) суммы информационных весов составляющих его символов;
- Д) информационного объема;
- Е) содержательного подхода к измерению информации.

4. Каков информационный вес одного символа компьютерного алфавита, мощность которого равна 256?

- А) 8 бит.
- В) 16 байт.
- С) 8 байт.
- Д) 2 байт.
- Е) 16 бит.

5. Что характерно для алфавитного подхода к измерению информации?

- А) Количество информации зависит от восприятия текста человеком.
- В) Количество информации зависит от информационного веса.
- С) Количество информации не зависит от восприятия текста человеком.
- Д) Количество информации зависит от смыслового содержания текста.
- Е) Количество информации зависит от позиции человека.

6. Нурдаулет отправил своему однокласснику SMS-сообщение объемом 20 480 бит. Позвонив однокласснику, Нурдаулет узнал, что свободного места для приема информации осталось только 3,125 Кбайт. Прочтет ли сообщение одноклассник?

- А) Да, так как $20\,480 \text{ бит} = 2,5 \text{ Кбайт}$, а свободного места – 3,125 Кбайт.
- В) Да, так как $2560 \text{ байт} = 2,5 \text{ Кбайт}$, а свободного места – 256 000 байт.
- С) Нет, так как $20\,480 \text{ бит} = 2,5 \text{ Кбайт}$, а свободного места – 3200 байт.
- Д) Нет, так как $20\,480 \text{ бит} = 2,5 \text{ Кбайт}$, а свободного места – 32 000 байт.
- Е) Нет, так как $20\,480 \text{ бит} = 3,5 \text{ Кбайт}$, а свободного места – 3,125 байт.

7. Сообщение объемом 2,5 Мбит проходит за 1/3 минуты. Рассчитайте пропускную способность канала (в Кбит/с).

- А) 120 Кбит/с.
- В) 128 Кбит/с.
- С) 8 Кбит/с.
- Д) 256 Кбит/с.

8. За какое время будет передано 900 Кбайт информации, если пропускная способность канала связи равна 512 байт/с?

- A) 10 мин.
- B) 100 с.
- C) 0,5 часа.
- D) 1/2.

9. Считая, что каждый символ кодируется одним байтом, оцените информационный объем стихотворения Абая Кунанбаева «Будь разборчив на пути своём...»: *Речи тех, кто учит детей, как зерно в земле прорастут.*

- A) 54 байт.
- B) 352 бит.
- C) 1024 байт.
- D) 8 бит.
- E) 1 байт.

10. Объем сообщения равен 11 Кбайт. Сообщение содержит 11 264 символа. Какова мощность алфавита?

- A) 33 символа.
- B) 24 символа.
- C) 256 символов.
- D) 11 символов.
- E) 16 символов.

Раздел 2

Здоровье и безопасность

2.1. Негативные аспекты использования компьютера

Научитесь

приводить примеры влияния различных электронных устройств на организм человека и эффективно использовать методы защиты.

Ключевые слова

Электронные
устройства
Электромагнит-
ное излучение

Электрондық
құрылғылар
Электромагниттік
сәулелену

Electronic devices
Electromagnetic
radiation



Как вы считаете, электромагнитное излучение есть только в электронных устройствах? Приведите примеры.



В настоящее время электромагнитное излучение преследует нас везде. Источниками электромагнитного излучения являются электрические сети, вся бытовая техника, транспорт (троллейбусы, трамваи), уличное освещение. К ним также относятся телевизоры, мобильные телефоны, гаджеты и другие электрические приборы (рис. 10). То есть любое электронное устройство, которое производит или потребляет электроэнергию, создает электромагнитное излучение.

В наше время компьютеризация в Казахстане принимает широкий размах. Сотни тысяч людей проводят большую часть свободного време-



Рис. 10. Электрические приборы

ни и рабочего дня перед монитором. Несмотря на пользу применения компьютерной техники, она может негативно воздействовать на здоровье человека.



Электромагнитное излучение – это вид энергии, представляющей электромагнитные волны, возбуждаемые различными излучающими объектами.

Комфорт и удобства от использования современных технологий – главная причина не думать о вреде и негативном воздействии их на здоровье. Компьютерная техника (ноутбук, планшет, смартфон и т.д.) является неотъемлемым элементом практически каждого дома. Человек, находясь в непосредственной близости от компьютера, подвергается электромагнитному излучению. Это излучение в 500 раз превышает допустимое значение, если учитывать, что предельно допустимая доза электромагнитного излучения для человека – 0,2 мкТл (микротесла).

Воздействие компьютера на организм человека достаточно серьезное. Это связано с тем, что:

- 1) монитор и системный блок являются источниками электромагнитного излучения;
- 2) часто устройства размещаются на небезопасном расстоянии;
- 3) часто не регулируется длительность работы за компьютером;
- 4) большое количество компьютеров концентрируется в одном месте.

На рисунке 11 представлены основные проблемы, связанные с электромагнитным излучением, и возможности минимизации их последствий.



Рис. 11. Вредные воздействия компьютера на человека

При работе за компьютером следует предпринимать определенные меры предосторожности. Необходимо эффективно использовать методы защиты от вредного воздействия электромагнитного излучения. Если выполнять несложные правила, то можно в значительной мере снизить воздействие магнитных полей на свой организм.



Какие правила техники безопасности при работе за компьютером и защиты от негативного влияния электронных устройств вы знаете?



Как известно, уровень электромагнитного фона Земли превышает естественный уровень в 200 000 раз. Это становится реальной угрозой всему живому. Электромагнитное загрязнение окружающей среды происходит из-за автомобильных, портативных и ручных радиостанций, компьютерной и бытовой электротехники и т.д. Чтобы избежать или минимизировать электромагнитное излучение при работе с электронными устройствами, необходимо соблюдать технику безопасности. Приведем некоторые способы защиты от негативного влияния электронных устройств (табл. 4).

Таблица 4

Электронные устройства	Защита человека от негативного воздействия
Кондиционеры, принтеры, проекторы, неоновые лампы, электропроводка, источники бесперебойного питания и т.д.	Если расстояние менее 1,5 метра, зона считается опасной. Постарайтесь находиться от них на расстоянии более чем 1,5 метра
При работе за компьютером	Самым уязвимым является место справа, под углом или сзади компьютера
Несколько компьютеров в компьютерном классе повышают общий электромагнитный фон помещения	Чтобы уменьшить источник электромагнитного излучения, необходимо: – делать на перемене влажную уборку в кабинете; – запрещается очищать от пыли и загрязнения электрооборудование, находящееся под напряжением; – размещать компьютеры по периметру кабинета, оставляя центр помещения свободным; – выключать компьютер по окончании работы
Процессор генерирует СВЧ-излучение	Молодой организм более подвержен влиянию сверхчастотного излучения. Поэтому необходимо придерживаться определенного времени работы за компьютером. Систематически выполнять упражнения для снятия напряжения с глаз и мышц тела
Мобильные устройства	Эксперты рекомендуют конкретные меры предосторожности: – не прижимать мобильный телефон к голове, а использовать гарнитуру (наушники) или громкую связь; – передавать большие объемы данных или смотреть видео при хорошей беспроводной связи устройства; – выключать точки доступа, когда они не используются, так как устройство все равно посылает сигналы

Для беспроводного подключения к Интернету широко внедряется Wi-Fi-система. Опасны ли для здоровья человека сети Wi-Fi? Исследования показывают, что сети Wi-Fi действуют на той же частоте, что и сверхвысокочастотная печь (СВЧ-печь). Для человека такая частота вредна. Поэтому точку доступа к Wi-Fi нужно размещать не ближе, чем на 1 метр от места, где человек находится длительное время (кровать, стол, диван, место для игр и т.д.).



Какие методы защиты от вредного воздействия электромагнитного излучения при работе за компьютером вы еще можете предложить?



Знание и понимание



1. Расскажите о вредных воздействиях электромагнитного излучения при работе с электронными устройствами.
2. Все ли электронные устройства дают электромагнитное излучение?
3. Какие основные проблемы, связанные с электромагнитным излучением, могут возникнуть у человека?
4. Какие возможности существуют для минимизации электромагнитного излучения?
5. Перечислите причины воздействия компьютера на организм человека.



Применение



Примените изученный материал в конкретных условиях, соедините теорию и практику. Создайте презентацию.

Для теоретиков: Изучите влияние негативных факторов на здоровье человека. На слайдах информацию можно представить в любой форме: в виде таблиц, рисунков, видео и т.д. В отчете представить как минимум шесть негативных факторов, влияющих на здоровье человека.

Для практиков: Изучите способы профилактики заболеваний, вызываемых вредными факторами. На слайдах информацию можно представлять в любой форме: в виде таблиц, рисунков, видео и т.д. В отчете представить как минимум шесть заболеваний и их профилактику.



Анализ



Проведите социологический опрос по вопросам из таблицы 5. По итогам опроса составьте диаграмму в электронной таблице Excel. Проанализируйте данные и составьте краткий отчет.

Таблица 5

№	Вопросы	Ответы		
		Да	Нет	Иногда
1	Сможет ли человек обойтись без компьютера?			
2	Дает ли компьютер электромагнитное излучение?			
3	Может ли компьютер влиять на ухудшение зрения, памяти, психики и т.д.?			
4	Можно ли за компьютером работать без перерыва?			
5	Возможно ли оградить себя от электромагнитного излучения при работе за компьютером?			
6	Делаете ли вы физкультминутки или динамические паузы во время работы за компьютером?			
7	Соблюдаете ли вы технику безопасности при работе с электронными устройствами?			

Синтез

Напишите эссе «Компьютер – друг мой, враг мой». В эссе должна быть новизна.

Рекомендации для выполнения домашнего задания

Эссе – литературный жанр, прозаическое сочинение небольшого объема и свободной композиции.

Оценивание

Систематизируйте изученный материал. Составьте рекомендации для эффективной защиты здоровья от электромагнитного излучения электронных устройств. Для представления информации примените инфографику.

Инфографика – это способ подачи информации, данных и знаний с помощью графического изображения.

Можно воспользоваться онлайн-сервисами для создания инфографики или нарисовать на бумаге.

Онлайн-сервисы для иконографии:

- Piktochart.com;
- Easel.ly;
- Creately.com;
- Infogr.am.





Дополнительные устройства, подключенные к компьютеру, также несут опасность для здоровья. Чтобы защитить себя от их вреда, соблюдайте меры безопасности, приведенные ниже.

Наушники, беспроводные гарнитуры представляют опасность, потому что всегда надеваются на голову.

Колонки создают вокруг себя большое электромагнитное поле. От них нужно находиться на расстоянии не менее 50 сантиметров.

Простой домашний **принтер** держите на расстоянии 50 сантиметров.

Магнитные поля **роутеров, модемов, маршрутизаторов** простираются вокруг на много метров. Располагайте их на расстоянии не менее 35 сантиметров от себя.

2.2. Безопасность в сети

Научитесь

соблюдать правила обеспечения безопасности пользователя в сети (мошенничество и агрессия в Интернете).

Ключевые слова

Безопасность в сети

Фишинг

Спам

Кибербуллинг

Троллинг

Желідегі қауіпсіздік

Фишинг

Спам

Кибербуллинг

Троллинг

Security in the network

Phishing

Spam

Cyberbullying

Trolling



Какие достоинства и недостатки Интернета вы можете назвать?



Все чаще мы пользуемся Интернетом. Пользуются им не только для игр и общения с друзьями, но и для поиска учебных материалов. Современные технологии, Интернет, ставшие неотъемлемой частью нашей жизни, таят в себе немало опасностей.

В предыдущих классах вы научились защищать информацию от вирусов, вредоносных программ с использованием антивирусных программ. Какие еще правила безопасности необходимо соблюдать в Сети, чтобы не навредить и защитить себя?

Всем известно, что дети легко интегрируются в мир интернет-технологий. Но они являются уязвимыми к виртуальной агрессии сверстников, чаще подвергаются краже личных данных, взлому аккаунтов, агрессии и другим видам мошенничества.

Как избежать опасностей, которые подстерегают вас в сети Интернет? Для этого познакомьтесь с существующими специальными терминами видов угроз (рис. 12).

Фишинг – это сетевое мошенничество

- спам, вредоносные веб-сайты;
- почтовые и мгновенные сообщения;
- кража личных данных.

Троллинг – способ виртуального общения

- провоцирование конфликта;
- насмешки, оскорбления, провокации.

Кибербуллинг – это травля, оскорбления или угрозы с помощью средств электронной коммуникации

- сообщения в социальных сетях;
- мгновенные сообщения;
- электронные письма и СМС.

Спам – рассылка навязчивой, нежелательной рекламной информации

- электронные послания;
- рекламные письма и т.п., рассылаемые отдельными формами по Интернету или электронной почте.

Рис. 12. Специальные термины угроз

Аккаунт (англ. *account*) – учетная запись пользователя. Он представляет собой набор данных о пользователе, введенных им и хранящихся на каком-либо сайте или интернет-сервисе.

Кража личных данных – это один из наиболее распространенных видов виртуальных преступлений. Ниже приведены **правила защиты личных данных** от посторонних лиц.



1. Не сообщайте пароли от своих персональных страниц и электронной почты.
2. Придумывайте сложные пароли, включающие комбинации букв и цифр. Не используйте в качестве пароля дату и год рождения.
3. Придумывайте разные логины и пароли для своей электронной почты, аккаунтов в социальных сетях и регистрации на различных сайтах. Один и тот же пароль позволит мошенникам взломать и другие ваши персональные страницы.



4. Закройте свой аккаунт, если использовали чужой компьютер для входа в него.
5. При взломе аккаунта смените пароль.

Чтобы свести опасность попадания в неприятные ситуации к минимуму, каждый пользователь должен знать и соблюдать правила безопасности в Интернете. На рисунке 13 приведены некоторые правила поведения во Всемирной информационной сети.

- 1 Не указывайте свою личную информацию: адрес, телефон, место жительства, номер школы и т.д.
- 2 Осторожно публикуйте личные фотографии. Не выкладывайте фотографии чужих людей без их согласия.
- 3 Не вся информация в Интернете достоверная, поэтому не доверяйте незнакомым людям.
- 4 Не переходите по сомнительным ссылкам, например, обещающим выигрыши. Не посещайте сомнительные сайты.
- 5 Помните об авторских правах и незаконном использовании чужой интеллектуальной собственности.
- 6 Хорошо подумайте, прежде чем встречаться в реальной жизни с людьми, с которыми познакомились в Интернете.
- 7 Помните, что в виртуальном мире действуют те же правила этикета, что и в реальной жизни.
- 8 Не отправляйте СМС, чтобы получить какую-либо услугу или выиграть приз.
- 9 Обращайтесь за советом к взрослым при малейшем подозрении или сомнении.

Рис. 13. Правила поведения в сети Интернет

Интернет – это часть нашей жизни. В виртуальной жизни вы не должны вести себя плохо: грубить, проявлять агрессию в адрес других людей, нецензурно выражаться. Очень важно соблюдать **правила вежливости при общении**:



1. Если кто-то проявляет агрессию в ваш адрес, не отвечайте грубостью на грубость.
2. Не выкладывайте в сеть информацию, оскорбляющую и унижающую достоинство других людей.
3. Регистрируясь на сайтах и форумах, обязательно ознакомьтесь с правилами общения.
4. Играя в онлайн-игры, уважительно относитесь к другим участникам игры.

Для безопасной работы в Интернете желательно устанавливать «детские» браузеры, программы родительского контроля и фильтры поисковых систем.



Пользуетесь ли вы «детским браузером», стандартными фильтрами, специализированными контролирующими приложениями фильтрации сайтов и т.д.? Поделитесь опытом.



Знание и понимание



1. Какие опасности подстерегают пользователей сети Интернет?
2. Что означают специальные термины «троллинг», «фишинг», «кибербуллинг», «спам»?
3. Перечислите основные правила поведения в Сети.
4. Каковы правила защиты личных данных?
5. Какие правила вежливого общения в Интернете нужно соблюдать?



Применение



1. К каждой группе угроз приведите примеры:
 - 1) Угроза заражения вредоносным программным обеспечением (ПО).
 - 2) Доступ к нежелательному содержанию.



3) Контакты с незнакомыми людьми с помощью чатов или электронной почты.

4) Поиск развлечений (например, игр) в Интернете.

5) Неконтролируемые покупки.

2. Существуют ли специальные программы для защиты от угроз сети Интернет?

3. Найдите в Интернете сайты, которые защитят компьютер от опасной информации.

4. Зарегистрируйтесь в чате и пообщайтесь, используя правила безопасности.

Анализ

Сочините сказку о «золотых правилах» безопасной работы в сети Интернет.

Синтез

Нарисуйте рисунки на тему «Безопасный Интернет в картинках».

Рекомендации для выполнения домашнего задания

Для создания рисунков используйте графические редакторы. По желанию можете выбрать растровый или векторный редактор.

Оценивание

Составьте сводную таблицу «Азбука безопасного Интернета». Напишите в первом столбце заголовок «Нужно», во втором столбце – заголовок «Нельзя». В строчках запишите 7 самых важных правил безопасности поведения подростков в сети Интернет.

Интернет сегодня влияет на нашу жизнь, а иногда даже контролирует ее. Вот несколько интересных фактов о сети Интернет.

1. Социальная сеть Facebook находится на 3-м месте по численности пользователей. Это равносильно тому, что Facebook можно считать третьим по численности государством в мире. Ежемесячно пользователи социальной сети Facebook выкладывают 3 миллиарда фотографий.

2. На первом месте по распространению доступа в Интернет находятся Бермудские острова.

3. Технологически развитая страна Япония находится на 22-м месте по доступу к сети Интернет.

4. 80% населения мира не пользуется сетью Интернет.

5. Объем информации удваивается каждые 2 года.

6. 247 миллиардов электронных писем посылаются каждый день, среди них 81% – спам.

Задание для проектной работы

Проект 2

На сегодняшний день самым ценным активом любой организации в Казахстане и в любой другой стране является информация. А в основе всех протекающих процессов находятся информационно-коммуникационные технологии.

В Концепции кибербезопасности «Киберщит Казахстана» отмечено, что «за период с 2010 по 2016 год плотность пользователей Интернета увеличилась с 36,1% до 75%, а количество пользователей мобильного Интернета с 3 миллионов 694 тысяч практически утроилось и достигло 10 миллионов 567 тысяч. Такое экспоненциальное увеличение числа пользователей Интернета повышает критичность и делает более ощутимыми последствия в случае отказов или вредоносного воздействия на технические средства».

Как в этих условиях грамотно выстроить защиту данных? К каким последствиям может привести недостаточная защищенность информации и технических средств?

Придумайте название мини-проекту, проявите творчество при оформлении проекта, проведите исследование, основываясь на информации из Концепции кибербезопасности «Киберщит Казахстана».

Советы безопасности в Сети для детей в возрасте от 13 до 17 лет

1. Используйте средства блокирования нежелательной информации.
2. Никогда не встречайтесь лично с друзьями из сети Интернет.
3. Никогда не выдавайте личную информацию средствами электронной почты, чатов, систем мгновенного обмена сообщениями, регистрационных форм, личных профилей и при регистрации на конкурсы в Интернет.
4. Не загружайте программы из подозрительных источников, так вы можете случайно загрузить вирусы или другое нежелательное программное обеспечение.
5. Сообщайте о любых угрозах или тревогах, связанных с Интернетом.
6. Не выдавайте в Интернет свой электронный адрес, не отвечайте на нежелательные письма и используйте специальные почтовые фильтры.
7. Приучитесь не использовать сеть для хулиганства, распространения сплетен или угроз другим людям.
8. Обсудите с одноклассниками проблемы сетевых азартных игр и их возможный риск.

При оформлении проектной работы примените шаблон, представленный в проекте 1: тема проекта, цель, задачи, этапы проведения, рекомендации.

Требования к оформлению проекта:

1. Используйте текстовый процессор Microsoft Word.
2. Информацию и картинки можно нарисовать, сфотографировать или сохранить из Интернета.
3. Оформите проект в виде книжки, буклета (буклет – это издание, отпечатанное на листе формата А-3 или А-4 и сложенное один или несколько раз).
4. Напишите мини-эссе.

Итоговый тест ко второму разделу

1. Электромагнитное излучение – это:

- А) вид энергии, представляющий волны, возбуждаемые различными излучающими объектами;
- В) вид энергии, представляющий электромагнитные волны, возбуждаемые различными излучающими объектами;
- С) вид энергии, представляющей волны, возбуждаемые различными неизлучающими объектами;
- Д) электромагнитные волны, возбуждаемые различными излучающими объектами.

2. Какие устройства компьютера дают наибольшее электромагнитное излучение?

- А) Монитор.
- В) Принтер.
- С) Системный блок.
- Д) Колонки.

3. Фишинг – это:

- А) сетевое мошенничество;
- В) общение;
- С) спам;
- Д) троллинг.

4. Травля, оскорбления или угрозы с помощью средств электронной коммуникации – это:

- А) фишинг;
- В) кибербуллинг;
- С) спам;
- Д) троллинг.

5. Рассылка навязчивой, нежелательной рекламной информации называется:

- A) фишинг;
- B) кибербуллинг;
- C) спам;
- D) троллинг.

6. Отметьте вредные факторы, влияющие на состояние здоровья людей, работающих за компьютером.

- A) Проблемы, связанные с электромагнитным излучением.
- B) Проблемы зрения.
- C) Проблемы, связанные с мышцами и суставами.
- D) Стресс, депрессия и другие нервные расстройства, вызванные влиянием компьютера на психику человека.
- E) Негативное влияние на кругозор.

7. Отметьте опасные угрозы Сети.

- A) Вредоносные программы.
- B) Кража компьютера.
- C) Хакерские атаки.
- D) Спам.
- E) Финансовое мошенничество.

8. Что не дает хакерам проникать в компьютер и просматривать файлы и документы?

- A) Применение брандмауэра.
- B) Обновления операционной системы.
- C) Антивирусная программа.
- D) Блокирование спама.

9. Какую информацию нельзя разглашать в Интернете?

- A) Свои увлечения.
- B) Свой псевдоним.
- C) Домашний адрес.
- D) Свой электронный адрес.

10. Действуют ли правила этикета в Интернете?

- A) Интернет – пространство, свободное от правил.
- B) В особых случаях.
- C) Да, как и в реальной жизни.
- D) Нет, не действуют правила этикета.

Раздел 3

Обработка информации в электронных таблицах

3.1. Статистические данные

Узнаете

- как использовать абсолютную и относительную ссылки;
- как использовать различные форматы данных для решения задач в электронных таблицах.

Ключевые слова

Относительная ссылка	Салыстырмалы сілтеме	Relative cell reference
Абсолютная ссылка	Абсолютті сілтеме	Absolute cell reference
Статистические данные	Статистикалық ақпарат	Statistical data



Вспомните, для чего используется электронная таблица Excel. Что находится на пересечении строки и столбца? С чего начинается ввод формул?



Одной из функций электронной таблицы Excel является работа с формулами. В формулах указываются не значения, а адреса ячеек, в которых находятся значения. Для быстрого пересчета значений в ячейке со ссылками нужно нажать клавишу **F9**.

Почему нужны абсолютные и относительные адреса ячеек?

Любая ячейка, которая содержит формулу, называется **зависимой**. От метода адресации ссылок зависит, что будет с ними происходить при копировании формулы из одной ячейки в другую.

Поэтому очень важно отличать **относительную ссылку** на адрес ячейки от **абсолютной ссылки**. **Ссылка** – это указание на адрес ячейки.

По умолчанию ссылки на ячейки в формулах рассматриваются как относительные.

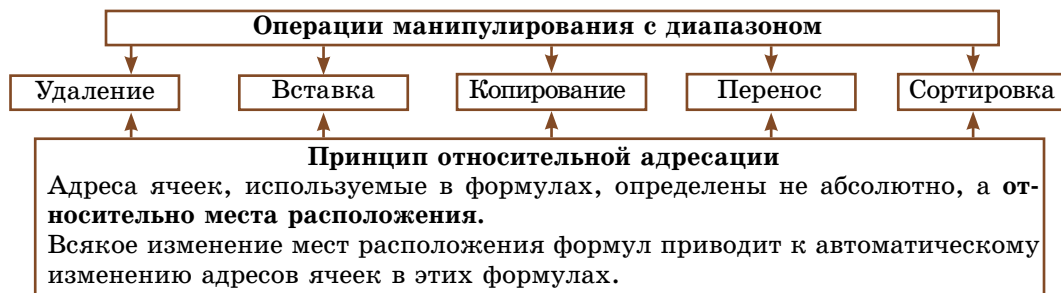


Если при копировании формулы адрес ячейки изменяется, то он называется **относительным**. Относительный адрес имеет вид: **A1, B1, C1** и т.д. При копировании такой формулы вправо или влево будет изменяться заголовок столбца в имени ячейки, а при копировании вверх или вниз – номер строки.

Абсолютный адрес применяется при использовании значения, которое не будет меняться в процессе вычислений. Абсолютный адрес записывают **\$C\$1**. Значение = **\$C\$1** всегда будет постоянным при копировании формулы в другие ячейки рабочего листа. Для быстрой установки в ячейке абсолютного адреса нужно нажать на клавишу **F4**.

Кроме относительной и абсолютной адресации используется **смешанная адресация**. При задании смешанной адресации применяется комбинация относительной и абсолютной адресаций. Одна из составляющих имени ячейки не меняется при копировании. Примеры такой адресации: **\$A1, A\$1**.

При работе в электронной таблице вы часто будете выполнять операции с диапазоном ячеек.



Например:

	D4	=	B4	*	C4
1	A	B	C	D	E
2				Курс доллара	=321
3	Товар	Цена, тенге	Продажи, шт	Продажи, тенге	Продажи, доллар
4	товар1	1000	130	=B4*C4	=D4/\$E\$2
5	товар2	2000	150	=B5*C5	=D5/\$E\$2
6	товар3	3000	500	=B6*C6	=D6/\$E\$2
7	товар4	2500	200	=B7*C7	=D7/\$E\$2

Рис. 14. Пример операции манипулирования с диапазоном ячеек

Ссылки на ячейки **B4, B5, B6, B7** в примере на рисунке 14 являются **относительными**, так как при копировании формулы меняются адреса ячеек.

Ссылка на ячейку **\$E\$2** является **абсолютной**. Она применяется для указания фиксированного значения и адреса ячейки, которые при копировании и перемещении формулы не меняются.

Используя ссылки в формулах, можно быстро провести сложные расчеты для большого количества данных. В электронной таблице Excel средства статистической обработки данных используются для проведения статистического анализа данных.

Внешне **статистическая таблица** представляет собой систему, построенную из горизонтальных строк и вертикальных столбцов. В такой таблице имеются общий заголовок, заголовки столбцов и строк, на пересечении которых записываются **статистические данные**.

В зависимости от задачи применяются различные форматы представления данных. Например, для представления чисел могут использоваться несколько различных форматов: **числовой** (100,72), **экспоненциальный** (для больших чисел: 500000000 – 5,00E + 08), **дробный** и **процентный**. Существуют специальные форматы для хранения дат (например, 01.09.2017) и **времени** (например, 13:30:55), а также **финансовый** и **денежный** форматы.



Постройте таблицу умножения Пифагора, используя относительные, абсолютные ссылки в формулах.



Шаг за шагом

Создайте таблицу умножения от 1 до 9, выполнив действия по шагам:

1. Введите данные в ячейках с B1 по J1 – 1, 2, ..., 9. Для заполнения числами 1, 2, ... примените **Маркер автозаполнения**. Установите **Формат ячеек** – **Числовые форматы** – **Общий формат** (рис. 15).

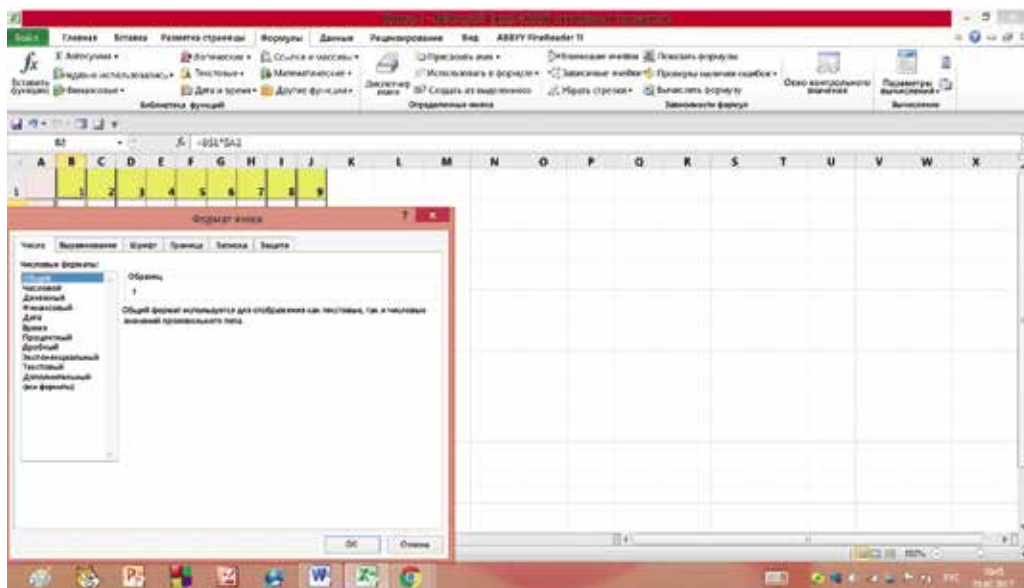


Рис. 15. Формат ячеек

2. Выполните аналогичные действия для ячеек с **A2** по **A10**.

3. Выделите столбцы от **A** до **J**. Установите ширину столбца 5,00 (или 40 пикселей). Для строк установите высоту 30,00 (или 40 пикселей).

4. Выделите диапазон ячеек **A1:J10**. Установите **Границы – Все границы**.

5. А теперь попробуйте в ячейку **B2** ввести формулу: **=B\$1*\$A2**. Зафиксируйте строку 1 и столбец A, поставив перед ними знак \$ (примените клавишу F4).

6. Скопируйте формулу на всю таблицу. В итоге вы увидите таблицу умножения (рис. 16).

7. Проанализируйте полученную таблицу умножения (рис. 17). Каким образом меняются относительные и абсолютные ссылки в таблице?

8. Поменяйте данные в строке заголовка, например, 10, 11, ..., 19. При этом изменятся данные в столбцах. Что произойдет с таблицей умножения?

	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1									
3	2									
4	3									
5	4									
6	5									
7	6									
8	7									
9	8									
10	9									

Рис. 16. Таблица умножения от 1 до 9

	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1	=B\$1*\$A2	=C\$1*\$A2	=D\$1*\$A2	=E\$1*\$A2	=F\$1*\$A2	=G\$1*\$A2	=H\$1*\$A2	=I\$1*\$A2	=J\$1*\$A2
3	2	=B\$1*\$A3	=C\$1*\$A3	=D\$1*\$A3	=E\$1*\$A3	=F\$1*\$A3	=G\$1*\$A3	=H\$1*\$A3	=I\$1*\$A3	=J\$1*\$A3
4	3	=B\$1*\$A4	=C\$1*\$A4	=D\$1*\$A4	=E\$1*\$A4	=F\$1*\$A4	=G\$1*\$A4	=H\$1*\$A4	=I\$1*\$A4	=J\$1*\$A4
5	4	=B\$1*\$A5	=C\$1*\$A5	=D\$1*\$A5	=E\$1*\$A5	=F\$1*\$A5	=G\$1*\$A5	=H\$1*\$A5	=I\$1*\$A5	=J\$1*\$A5
6	5	=B\$1*\$A6	=C\$1*\$A6	=D\$1*\$A6	=E\$1*\$A6	=F\$1*\$A6	=G\$1*\$A6	=H\$1*\$A6	=I\$1*\$A6	=J\$1*\$A6
7	6	=B\$1*\$A7	=C\$1*\$A7	=D\$1*\$A7	=E\$1*\$A7	=F\$1*\$A7	=G\$1*\$A7	=H\$1*\$A7	=I\$1*\$A7	=J\$1*\$A7
8	7	=B\$1*\$A8	=C\$1*\$A8	=D\$1*\$A8	=E\$1*\$A8	=F\$1*\$A8	=G\$1*\$A8	=H\$1*\$A8	=I\$1*\$A8	=J\$1*\$A8
9	8	=B\$1*\$A9	=C\$1*\$A9	=D\$1*\$A9	=E\$1*\$A9	=F\$1*\$A9	=G\$1*\$A9	=H\$1*\$A9	=I\$1*\$A9	=J\$1*\$A9
10	9	=B\$1*\$A10	=C\$1*\$A10	=D\$1*\$A10	=E\$1*\$A10	=F\$1*\$A10	=G\$1*\$A10	=H\$1*\$A10	=I\$1*\$A10	=J\$1*\$A10

Рис. 17. Таблица умножения в формулах

9. Вставьте строки над таблицей и введите заголовок. Сохраните файл.



В ячейку **A10** ввели формулу **=(A5+A9)/\$D\$1**. Затем эту формулу скопировали в ячейку **A15**. Какая формула содер­жится в ячейке **A15**? Проверьте на практике правиль­ность выполнения.



Знание и понимание

1. Где используются адреса ячеек?
2. Какие адреса ячеек встречаются в формулах?
3. В чем отличие относительной ссылки от абсолютной? Приведите примеры.
4. Могут ли в адресе ячейки формулы использоваться и относительная, и абсолютная адресация?
5. Какие форматы данных чаще всего используют в электронных таб­лицах?
6. Установите соответствие:

1	Относительный адрес	А	Перечисление имени столбца и номера строки, на пересечении которых находится ячейка
2	Абсолютный адрес	Б	Это адрес ячеек в формулах, который при ко­пировании изменяется относительно места расположения формулы
3	Адрес ячейки	В	Это адрес ячеек в формулах, который при копи­ровании меняется частично: или только номер строки, или только имя столбца
4	Смешанный адрес	Г	Это адрес ячеек в формулах, который при копи­ровании не изменяется (фиксирование адреса)

7. Установите соответствие:

1	R5, G23, J122	А	Смешанный адрес ячейки в формуле
2	\$R5, G\$23, \$J122	Б	Абсолютный адрес ячейки в формуле
3	\$R\$5, \$G\$23, \$J\$122	В	Относительный адрес ячейки в формуле



Применение

Примените теоретические знания в практической ситуации.

1. Оформите на листе электронной таблицы Excel расписание школьных звонков. За исходные данные возьмите начало уроков и продолжительность урока (рис. 18).

Установите курсор в ячейку **B8** и введите формулу **=C7+D7** (к концу урока прибавляем переменную).

Формулы из ячеек **B8** и **C7** скопируйте.

Получите таблицу «Расписание звонков» (рис. 18).

Расписание звонков	
начало урока	
продолжительность урока	
начало урока	конец урока
D3	=B7+\$D\$4
C7+D7	=B8+\$D\$4
C8+D8	=B9+\$D\$4
C9+D9	=B10+\$D\$4
C10+D10	=B11+\$D\$4
C11+D11	=B12+\$D\$4
C12+D12	=B13+\$D\$4
C13+D13	=B14+\$D\$4
C14+D14	=B15+\$D\$4
C15+D15	=B16+\$D\$4

Расписание звонков			
	начало урока		8:30
	продолжительность урока		0:40
№ урока	начало урока	конец урока	перемена
1	8:30	9:10	0:05
2	9:15	9:55	0:15
3	10:10	10:50	0:05
4	10:55	11:35	0:20
5	11:55	12:35	0:05
6	12:40	13:20	0:20
7	13:40	14:20	0:05
8	14:25	15:05	0:15
9	15:20	16:00	0:05
10	16:05	16:45	

Рис. 18. Таблица **Расписание звонков** (в виде формул и данных)

2. На рисунках 19, 20 даны две таблицы «Расписание уроков». Заполните ячейки, залитые желтым цветом, формулами.

Расписание уроков

Прямая задача			
Урок	Начало урока	Конец урока	Перемена после урока
1	8:30	9:15	
2	9:20	10:05	
3	10:35	11:20	
4	11:45	12:30	
5	12:35	13:20	
6	13:40	14:25	
7	14:40	15:25	
8	15:30	16:15	

Рис. 19. Прямая задача

Расписание уроков

Обратная задача			
Продолжительность перемен после урока	Урок	Начало урока	Конец урока
0:10	1	8:30	
0:15	2		
0:05	3		
0:20	4		
0:15	5		
0:10	6		
0:05	7		
	8		

Рис. 20. Обратная задача

Анализ

В ячейке **B2** (табл. 6) записано количество информации в байтах. Получите в ячейке **B3:B5** количество информации в единицах измерения, указанных в соответствующих ячейках столбца **A**. Проанализируйте полученные данные.

Таблица 6

	A	B	C
1			
2	Количество информации в байтах:		
3	Количество информации в битах:		
4	Количество информации в килобайтах:		
5	Количество информации в мегабайтах:		

Синтез

Информация о распределении суши и воды на земном шаре приведена в таблице 7.

Таблица 7

Поверхность земного шара	Северное полушарие		Южное полушарие		Земля в целом	
	в млн кв. км	в %	в млн кв. км	в %	в млн кв. км	в %
Суша	100,41		48,43			
Вода	154,64		206,62			
Всего						

Используя электронную таблицу, заполните данными пустые клетки приведенной таблицы. Обобщите полученную таблицу. Предложите альтернативу формулам.



Оценивание

По таблице 8 рассчитайте сумму оплаты электроэнергии за каждый месяц. Стоимость 1 кВт/ч равна 16,65 тенге.

Оцените счет при изменении стоимости 1 кВт/ч.

Таблица 8

Счет на оплату электроэнергии

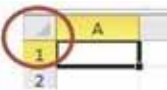
Стоимость 1 кВт/ч

16,65 тенге

Месяц	Дата	Показания счетчика	Расход кВт/ч	Сумма (тенге)
Январь	30.01.2017	540	540	
Февраль	28.02.2017	1026	486	
Март	29.03.2017	1480	454	
Апрель	30.04.2017	2158	678	
Май	28.05.2017	2996	838	
Июнь	29.06.2017	3458	462	
Июль	28.07.2017	4021	563	
Август	30.08.2017	4684	663	
Сентябрь	25.09.2017			
Октябрь	26.10.2017			
Ноябрь	30.11.2017			
Декабрь	25.12.2017			



Выделение всех ячеек

Чтобы выделить все ячейки, нажмите **Ctrl + A** или специальную кнопку  в углу листа.

Быстрый ввод данных в несколько ячеек

Чтобы сэкономить время при вводе новых данных (даты, строки, числа, формулы) или при замене уже существующих, выделите ячейки (можно несмежные), введите данные и нажмите **Ctrl + Enter**.

Как восстановить файл, который вы не сохранили

Для восстановления несохраненного файла в Excel 2010 нажмите на **Файл**, выберите **Последние** (рис. 21). В правом нижнем углу экрана появится опция **Восстановить несохраненные книги**.

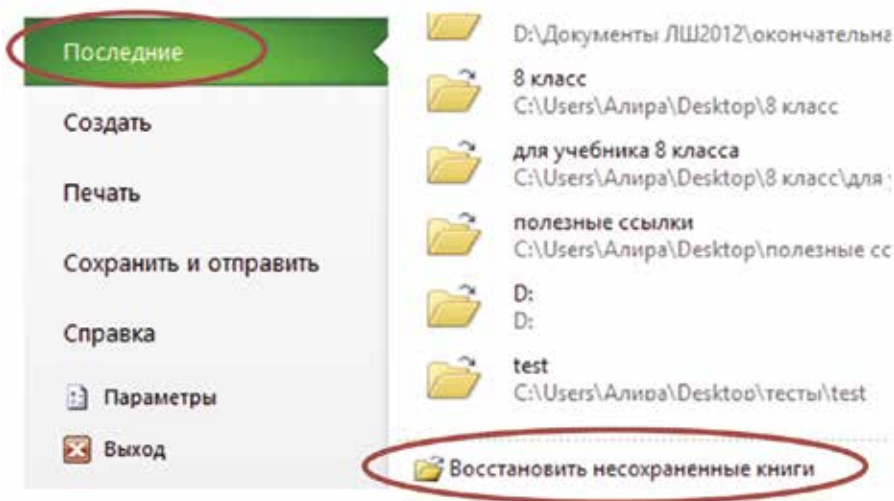


Рис. 21. Восстановление несохраненного файла в Excel 2010

В версии 2013 года путь такой: **Файл – Сведения – Управление версиями**.

Супертайный лист

Допустим, вы хотите скрыть часть листов в электронной таблице Excel от других пользователей, работающих над книгой. Это можно сделать классическим способом – щелкните по ярлычку листа правой кнопкой

мыши и в контекстном меню выберите команду **Скрыть**. В этом случае имя скрытого листа все равно будет видно другому человеку. Чтобы сделать его абсолютно невидимым, нужно действовать так:

1. Нажмите **Alt + F11**.
2. Выберите лист в **VBAProject** (рис. 22).
3. Слева появится окно **Microsoft Visual Basic for Applications**.
4. Найдите свойство **Visible** и сделайте его **xlSheetVery Hidden**.
5. Лист будет скрыт.

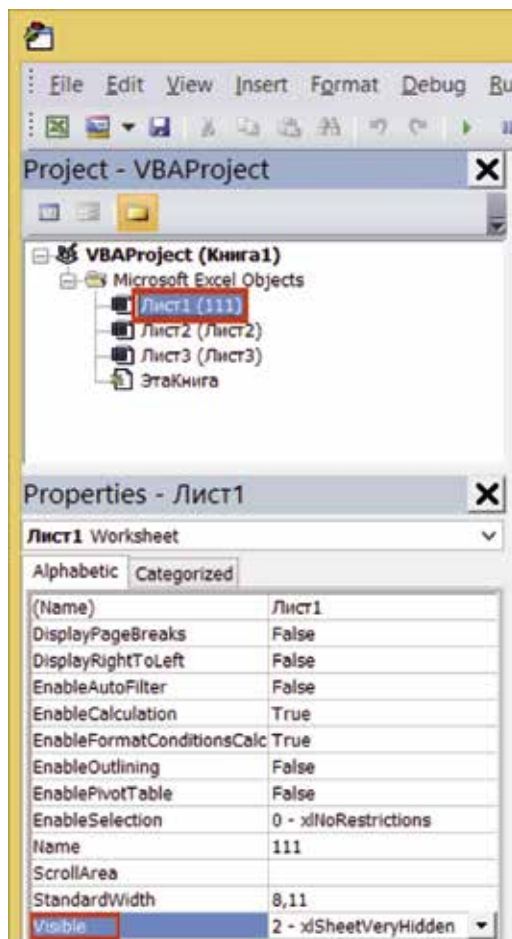


Рис. 22. Окно Microsoft Visual Basic for Applications

3.2. Встроенные функции

Научитесь

использовать встроенные функции для решения задач с применением электронных таблиц.

Ключевое слово

*Встроенные
функции*

*Kipicirilgen
funkciyalar*

*Builtin
functions*

Кроме простых арифметических операций, электронная таблица Excel содержит достаточно большой набор **встроенных функций**. Функции в Excel упорядочены по категориям в зависимости от функциональной области. Функций несколько сотен.



Функция – встроенный инструмент Excel, способный вернуть значение в зависимости от переданного ему параметра. Она предназначена для расчетов, вычислений и анализа данных. Каждая функция может включать в себя **константу, оператор, ссылку, имя ячейки (диапазона ячеек) и формулы**.

Имя – это краткое обозначение, представляющее ячейку, диапазон ячеек, формулу, таблицу или значение константы. Можно создавать свои собственные определенные имена. С помощью поля **Имя** можно быстро находить и выделять конкретные ячейки или **диапазоны**. Для этого в поле, расположенное слева от строки формул,  нужно ввести имена или ссылки на ячейки.

Диапазоном называются две или более **ячеек** листа. Диапазон рассматривается как единое целое. Прямоугольный блок ячеек называется **смежным диапазоном** и обозначается при помощи двух ссылок, разделенных двоеточием. Первая ссылка указывает на **ячейку** в левом верхнем углу диапазона, а вторая – на **ячейку** в правом нижнем углу.

Константа представляет собой число либо текст. Она не меняется в процессе расчетов. Операторы бывают трех видов:

1. **Арифметический оператор** предназначен для выполнения арифметических действий. Он возвращает числовое значение.

2. **Оператор сравнения** предназначен для сравнения данных. Он возвращает логическое значение ИСТИНА или ЛОЖЬ (TRUE или FALSE).

3. **Текстовый оператор** применяется для объединения данных (в Excel им является амперсанд – &).

Каждая встроенная функция Excel имеет название или уникальное имя. Синтаксис встроенной функции:

- **Название функции (Аргументы)**, где Название функции – уникальное имя функции.

- **Аргументы** – список аргументов функции или его входные данные. Встроенные функции Excel работают по принципу:

- задать функции входным данным;

- функция вычисляет результат или возвращает выходные данные.

Имена функций могут быть набраны как строчными, так и заглавными буквами. В зависимости от функции ее аргументы могут представлять собой число, строку, логическое значение, ссылку и т.д. В качестве аргументов функции можно также использовать другие функции и формулы.

Чтобы вставить формулу с функцией, нужно щелкнуть по кнопке с изображением символа функции

слева от строки формул. Можно также воспользоваться специальной кнопкой



Вставить функцию

на вкладке **Формулы**. В каждом из этих случаев открывается диалоговое окно **Мастер функций**, представленное на рисунке 23.

В раскрывающемся списке **Категория** (рис. 24) можно выбрать ту категорию, к которой относится искомая функция. Функцию также можно найти с помощью поисковой строки **Поиск функции...**. Для этого введите в поле первые несколько букв ее названия или слово. Можно выбрать элемент из **Полного алфавитного перечня**. После того как функция выбрана, открывается диалоговое окно **Аргументы функции**, где в поля вводятся аргументы функции.

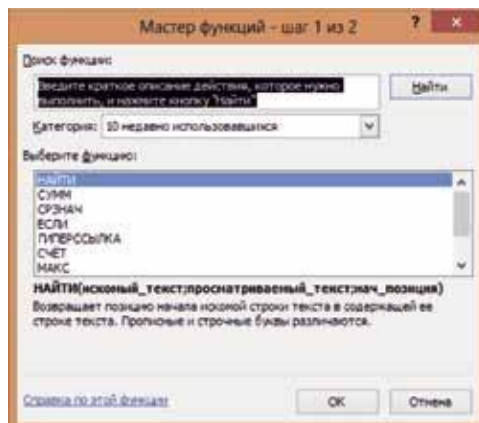


Рис. 23. Мастер функций



Рис. 24. Встроенные функции



Рассмотрим простейший пример вычисления среднего значения данных в диапазоне ячеек **C1:C10**.



Для вычисления среднего арифметического значения можно использовать формулу:

$$=(C1 + C2 + C3 + C4 + C5 + C6 + C7 + C8 + C9 + C10)/10.$$

Но проще применить функцию из категории **Статистические** **=СРЗНАЧ(C1:C10)**.

Применим на практике функции из категории **Текстовые**: **СИМВОЛ** и **КОДСИМВ**. Как вам известно, система ASCII (коды для представления цифр, букв и других символов) использует числа от 0 до 127 (в некоторых системах до 255). Функции **СИМВОЛ** и **КОДСИМВ** тесно связаны с системой ASCII. Например, функция **СИМВОЛ** возвращает символ, который соответствует числовому коду ASCII.

Синтаксис: **=СИМВОЛ(число)**

Функция **КОДСИМВ** возвращает код ASCII для первого символа ее аргумента.

Синтаксис: **=КОДСИМВ(текст)**

Например, формула =СИМВОЛ(83) возвратит букву S.



Формула =КОДСИМВ(«S») возвратит значение 83.

Аналогично, если первым символом в ячейке A1 является буква S, формула =КОДСИМВ(A1) возвратит код 83. Поскольку цифры также являются символами, аргумент КОДСИМВ может быть числовым. Например, формула =КОДСИМВ(«8») возвращает 56, то есть код ASCII символа 8.



1. При помощи функции КОДСИМВ узнайте коды, принадлежащие буквам A, D, Y, L, R, N, M.
2. При помощи функции СИМВОЛ определите символы, закодированные кодами 59, 67, 70, 78.



В одну из ячеек будет введено число. В другой ячейке нужно получить ответ на вопрос, является ли введенное число отрицательным. Если число отрицательное, то «Да», иначе «Нет».



Для решения данной задачи рассмотрим функцию ЕСЛИ() из категории **Логические**. Эта функция является одним из мощных инструментов в мире функций.

Синтаксис логической функции:

ЕСЛИ (лог_выражение; значение_если_истина),

где лог_выражение – логическое выражение, возвращающее значение ИСТИНА или ЛОЖЬ.

Логические выражения строятся с помощью операций отношения (<, >, <= (меньше или равно), >= (больше или равно), =, <> (не равно)) и логических операций (логическое И, логическое ИЛИ, логическое отрицание НЕ).

На рисунке 25 представлено решение задания об отрицательном числе. Проанализируйте введенную логическую функцию ЕСЛИ.

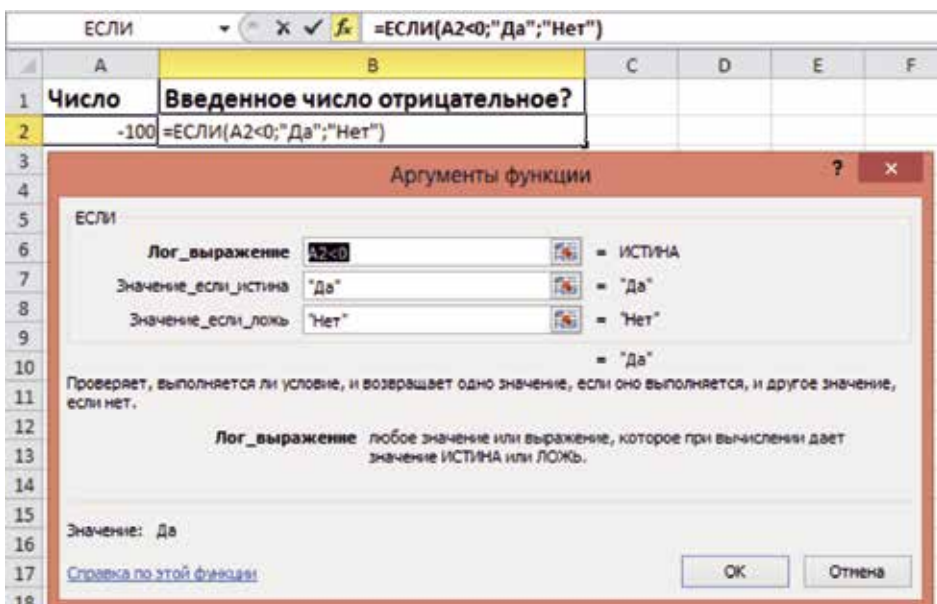


Рис. 25. Функция ЕСЛИ()

При проведении множественных логических проверок используется функция **ЕСЛИ()** внутри другого **ЕСЛИ()**, функции **И()** и функции **ИЛИ()**.

Существуют особенности записи логических операций. Сначала записывается имя логической операции (**И**, **ИЛИ**, **НЕ**), а затем в круглых скобках перечисляются логические операнды. Например:

= **ЕСЛИ(И(A5 > D1; B3 < 5); «Да»; «Нет»)**

Содержимое ячейки **A5** больше содержимого ячейки **D1**, а содержимое ячейки **B3** меньше 5. Если условие выполняется, то в выбранной клетке будет отображаться «Да», в противном случае – «Нет».



Нужно протабулировать функцию $y = f(x)$ с шагом 0,2:

$$y = \begin{cases} 0, & \text{если } x < 0 \\ x, & \text{если } x \in [0,1] \\ 1, & \text{если } x > 1 \end{cases}$$



Табулирование функции – это вычисление значений функции при изменении аргумента от некоторого начального значения до некоторого конечного значения с определенным шагом.

На рисунке 26: x – аргумент, y – результат функции, шаг = 0,2. Каким образом меняются значения в столбце x ? Проанализируйте формулу логической функции вложенного ЕСЛИ. Соответствует ли результат функции условиям функции $y = f(x)$?

	B2		fx =ЕСЛИ(A2<0;0;ЕСЛИ(A2>1;1;A2))				
	A	B	C	D	E	F	G
1	x	y	шаг				
2	-0,4	0	0,2				
3	-0,2	0					
4	0	0					
5	0,2	0,2					
6	0,4	0,4					
7	0,6	0,6					
8	0,8	0,8					
9	1	1					
10	1,2	1					
11	1,4	1					
12	1,6	1					

Рис. 26. Вложенное ЕСЛИ



Знание и понимание

1. Каким образом упорядочены встроенные функции?
2. Как вы понимаете понятие «встроенная функция»?
3. Каков синтаксис встроенной функции?
4. По какому принципу работают встроенные функции?
5. Можно ли имя функции набрать строчными буквами?
6. Каким образом можно вставить встроенную функцию?
7. Перечислите, какие категории входят во встроенные функции.
8. Каков синтаксис логической функции ЕСЛИ()?
9. С помощью каких логических операций строится логическое выражение?
10. Что означает вложенное ЕСЛИ()?
11. Каким образом происходит табулирование функции?



Применение

Чтобы получить допуск к экзамену, учащиеся должны успешно сдать зачет. Результаты класса занесите в таблицу с графами: **Фамилия, имя; Зачет; Экзамен** (рис. 27). Составьте функцию **ЕСЛИ()** в ячейке **С2** и скопируйте с помощью маркера заполнения.

	А	В	С
1	Фамилия, имя	Зачет	Экзамен
2	Абенов В.	зачет	допущен
3	Ан М.	зачет	допущен
4	Алимов А.	незачет	не допущен
5	Крузе Н.	зачет	допущен
6	Сахарова Т.	незачет	не допущен

Рис. 27. Таблица зачетов



Анализ

Создайте таблицу для анализа успеваемости учащихся. Если ученик получит 5 баллов – «отлично», 4 – «хорошо», 3 – «удовлетворительно». Составьте вложенное условие **ЕСЛИ** (рис. 28) в ячейке **С5** и скопируйте в остальные ячейки.

	А	В	С
4	Фамилия, имя	Баллы	Результат
5	Абенов В.	5	отлично
6	Ан М.	5	отлично
7	Алимов А.	2	неудовлетворительно

Рис. 28. Таблица Анализ успеваемости



Синтез

На рисунке 29 представлена таблица «Заказ на мягкую мебель». Проведите расчеты в столбцах с заголовками **ВСЕГО, ИТОГО**. Найдите минимальную, максимальную и среднюю стоимость заказа. Обобщите данные в таблице.

Заказ на мягкую мебель				
№	Тип салона мебели	Цена за 1 комплект	Количество комплектов	ВСЕГО
1	МЕРЕЙ	T408 000,0	2	
2	ЖАННА	T218 000,0	4	
3	БРАВО	T245 000,0	3	
4	БЕЛОРУССКАЯ МЕБЕЛЬ	T305 000,0	5	
5	RED APPLE	T230 800,0	4	
6	ARMADA	T439 000,0	6	
7	ДЕТСКАЯ	T110 870,0	10	
8	ZETA	T315 000,0		
ИТОГО:				
Минимальная стоимость заказа				
Максимальная стоимость заказа				
Средняя стоимость заказа				

Рис. 29. Заказ на мягкую мебель

Рекомендации к выполнению задания

1. Формат ячеек – Денежный – Казахский.
2. Для расчета **ВСЕГО** примените формулу «Цена за 1 комплект умножить на количество комплектов».
3. Для расчета **ИТОГО** используйте математическую функцию СУММ (число1; число2;...) – сумма аргументов.
4. Для расчета минимальной, максимальной и средней стоимости заказа используйте статистические функции МИН, МАКС, СРЗНАЧ.

Статистические функции:

МИН(число1;число2;...) возвращает минимальное значение из списка аргументов.

МАКС(число1;число2;...) возвращает максимальное значение из списка аргументов.

СРЗНАЧ(число1;число2;...) возвращает среднее арифметическое аргументов.



Оценивание

1. Проведите исследование, используя данные, приведенные в таблице 9. Для этого выбраны популярные национальные блюда:
 - английское «Ростбиф с овощами»;
 - казахское «Бешбармак»;

- русское «Пюре с котлетой»;
- японское «Ролл».

Известно, что суточная потребность калорий в день в белках, жирах, углеводах и энергии для подростков от 11 до 13 лет составляет:

Подростки	Белки, г	Жиры, г	Углево- ды, г	Энергетическая ценность, ккал
Мальчики	93	93	370	2700
Девочки	85	85	340	2450

2. Какие расчеты можно выполнить в таблице с помощью встроенных функций?
3. Почему ингредиент «соль» отсутствует во всех блюдах в таблице?
4. Какие из перечисленных блюд по энергетической ценности в день наиболее подходят подросткам?
5. Какие проблемы могут возникнуть, если растущий организм не насыщается необходимыми питательными веществами?
6. Сделайте заключение, учитывая рекомендации о здоровом питании для подростков.

Таблица 9

Энергетическая ценность популярных национальных блюд

№	Национальная кухня	Белки	Жиры	Углеводы	Калории
1	Английская				
	<i>Ростбиф с овощами</i>				
	говядина	18,5	16	0	218
	цветная капуста	2,5	0,3	5,4	30
	морковь	1,3	0,1	9,3	34
	картофель	2	0,4	18,1	80
2	Казахская				
	<i>Бешбармак</i>				
	говядина	18,5	16	0	218
	мука	10,3	1,1	69	334
	лук	1,4	0	10,4	41
	картофель	2	0,4	18,1	80
	морковь	1,3	0,1	9,3	34

3	Русская				
	<i>Пюре с котлетой</i>				
	говядина	18,5	16	0	218
	лук	1,4	0	10,4	41
	сливочное масло	0,8	72,5	1,3	661
	чеснок	6,5	0	6	46
	ржаной хлеб	6	1,2	41,8	181
	яйцо	12,7	10,9	0,7	155
	картофель	2	0,4	18,1	80
4	Японская				
	<i>Ролл</i>				
	рис	0,6	7	73,7	330
	осетр	10,9	16,4	0	100
	огурец	0,8	0,1	3,8	14
	редька	1,9	0,2	8	35



77-летний художник из Японии Тацуо Хориучи с помощью Microsoft Excel создает красивые произведения. Его картины – сложные и изысканные естественные японские пейзажи, усложненные культурными мотивами (рис. 30А, Б).

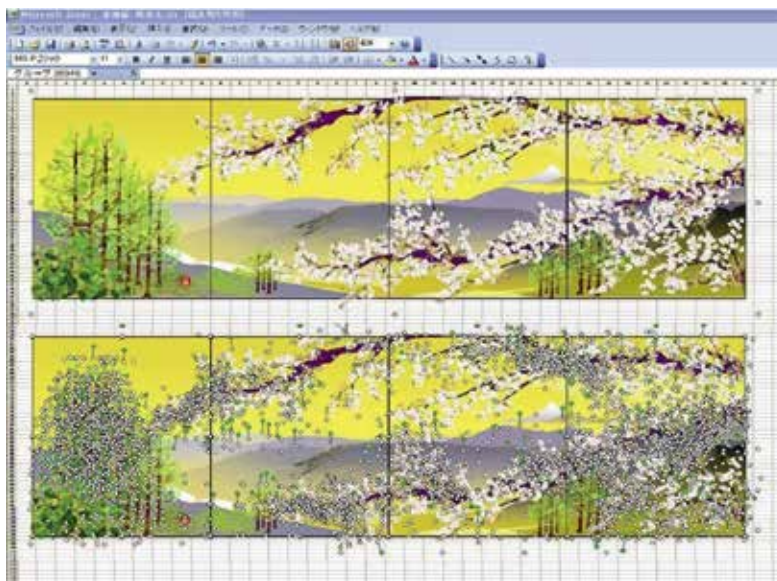


Рис. 30А. Картины, созданные в Microsoft Excel

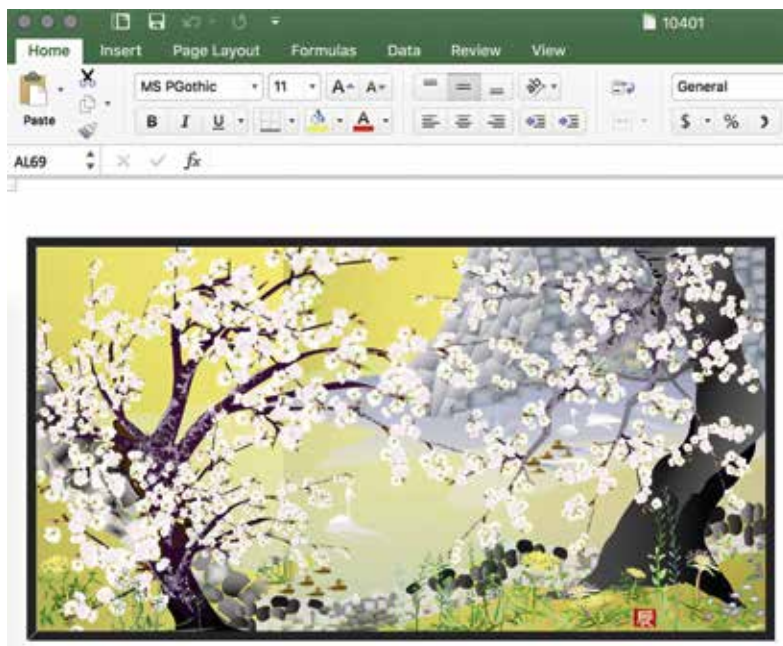


Рис. 30Б. Картины, созданные в Microsoft Excel

3.3. Анализ данных на основе имеющейся информации

Научитесь

использовать встроенные функции для решения задач с применением электронных таблиц.

Ключевые слова

Анализ данных
Спарклайн
Сводная таблица
Итоги

Деректерді
талдау
Спарклайн
Жиынтық кестесі
Нәтижелері

Data analysis
Sparkline
Summary table
Results

Электронная таблица Excel – эффективное средство для анализа больших объемов данных. Данные в электронной таблице можно вычислять, сортировать и фильтровать, создавать профессионально оформленные диа-

граммы. Кроме этого, можно применять стиль таблицы, создавать сводные таблицы, быстро добавлять итоги, применять условное форматирование, средства сортировки и фильтрации и т.д. Мы рассмотрим некоторые из них.

Наиболее подходящий и быстрый способ анализа данных в электронной таблице Excel – это построение диаграмм и графиков, миниатюрных графиков – **спарклайнов**. От качества их построения зависит точность анализа данных. Рассмотрим некоторые способы анализа.



Предприятие по производству мебели выпускает несколько видов продукции: шкафы (трех видов), столы (пяти видов) и стулья (двух видов).
В таблице 10 приведены данные об объемах продаж за первый квартал текущего года. На основе этих данных нужно проанализировать, какие виды продукции пользуются большим спросом, и принять решение об увеличении/уменьшении объемов производства. Вспомните, как создавать гистограмму.



Таблица 10

Данные из условия задачи

	Продукция	Объем продаж (шт.)	Суммарный объем
Шкафы	шкаф 1	135	533
	шкаф 2	80	
	шкаф 3	318	
Стол	стол 1	200	481
	стол 2	55	
	стол 3	73	
	стол 4	90	
	стол 5	63	
Стулья	стул 1	505	805
	стул 2	300	

Чтобы решить данную проблему, нужно представить данные по всем видам продукции с помощью **гистограммы** (рис. 31).



Рис. 31. Данные о продажах, представленные с помощью гистограммы



Рассмотрите вторую альтернативу – представьте данные по всем видам продукции с помощью **линейчатой диаграммы**. Она похожа на гистограмму, только столбцы расположены горизонтально.



Проанализируйте диаграммы, полученные в первом и во втором решениях. Какой из видов продукции пользуется наибольшим спросом, а какой – наименьшим? Какие решения необходимо принять об увеличении/уменьшении объемов производства?



Рассмотрите третью альтернативу – представьте данные по всем видам продукции с помощью **круговой диаграммы**.



Что показывает третье решение – круговая диаграмма? Какая часть от общего объема продаж составляет объем продажи каждого из видов продукции?

Какие функции анализа можно еще использовать? Рассмотрим **спарклайны**.

Спарклайны – это отслеживание тенденции в рядах данных (колебание цен, объем продаж). Спарклайны располагаются внутри ячеек и представляют собой мини-графики. Они наглядно отображают поведение данных, максимальные и минимальные значения и другие особые точки.

Отличие спарклайнов от диаграмм в мгновенной скорости их создания. Создание спарклайна происходит следующим образом:

1. Выберите вкладку **Вставка** – группы **Спарклайны** – кнопка **График** (рис. 32).

2. В появившемся окне **Создание спарклайнов** выберите место для размещения спарклайнов. Обычно диапазон размещения указывают рядом с данными.

3. Нажмите **ОК**.

Если выделить ячейку со спарклайном, появится вкладка для работы с ним.

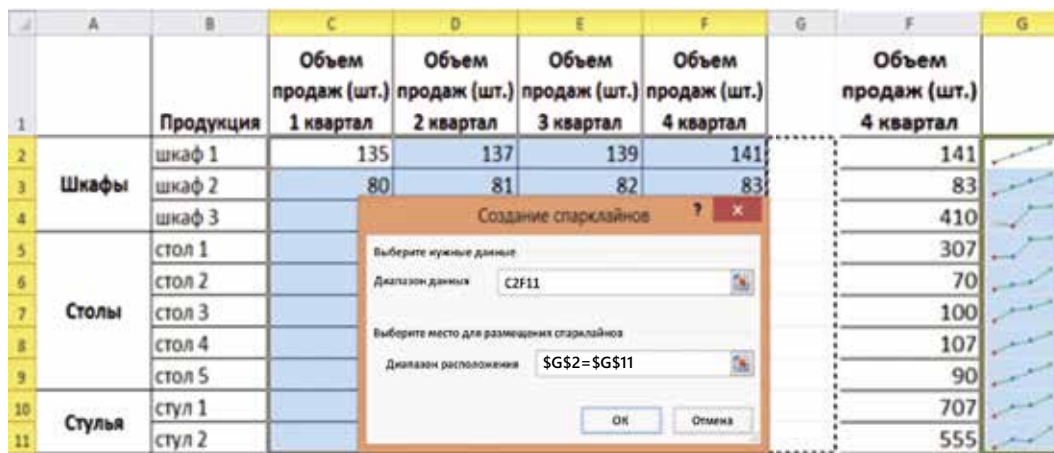


Рис. 32. Создание спарклайнов



Поэкспериментируйте: измените минимальные и максимальные точки, стиль, цвет спарклайна, цвет маркера, тип спарклайна и т.д.



Альтернативой спарклайнам является **Условное форматирование** с правилом **Цветовые шкалы** (рис. 33).

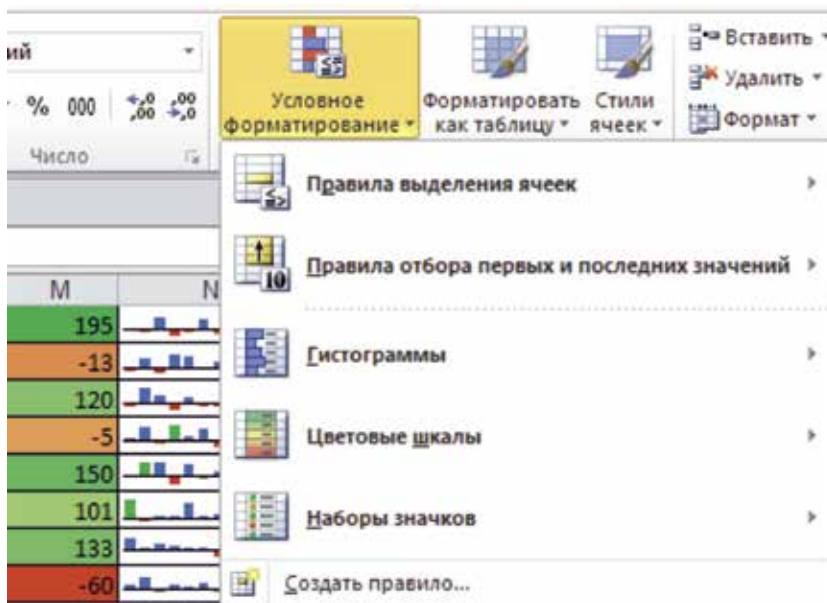


Рис. 33. Условное форматирование

Анализировать данные в Excel можно также с помощью встроенных функций и других инструментов для анализа данных, представленных на рисунке 34:

	Группировка данных – объединение диапазона ячеек
	Консолидация данных – объединение нескольких диапазонов данных в один диапазон

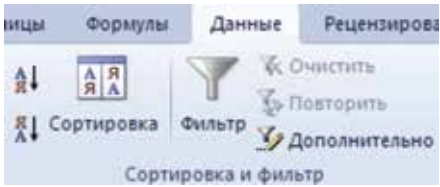
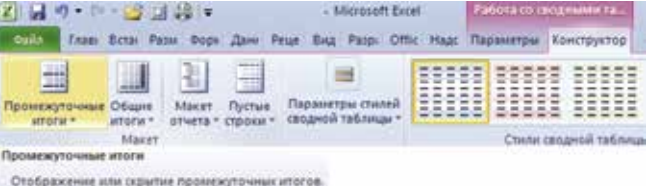
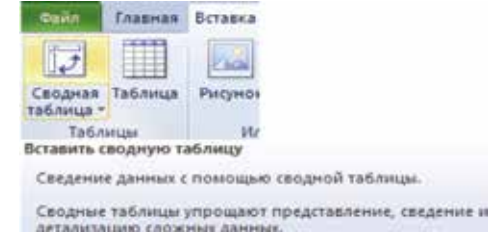
	Сортировка и фильтрация изменяют порядок строк
	Промежуточные итоги используются при работе со списками
	Сводная таблица – анализ больших данных

Рис. 34. Инструменты для анализа данных



При анализе большого объема данных в таблице применяется сводная таблица. **Сводная таблица** – это отчет, построенный на основе табличных данных.



Создадим сводную таблицу для анализа больших данных в таблице.



Шаг за шагом

1. Постройте простую таблицу «Продажа канцелярских товаров».
2. Щелкните на ячейке **A1** для определения данных, с которыми нужно будет работать.
3. На вкладке **Вставка** выберите из группы **Таблицы** – **Сводная таблица**. Откроется окно **Создание сводной таблицы** (рис. 35).
4. Укажите вариант **Выбрать таблицу** или **диапазон**.

5. Укажите диапазон и место расположения на текущем листе, например **F1**.

6. Появится форма сводной таблицы и **Список полей сводной таблицы**. На панели находятся:

1) **Выберите поля для добавления в отчет.** Для того чтобы добавить/убрать поле в сводную таблицу, поставьте/снимите галочку напротив названия необходимого поля или перетащите поля данного списка.

2) **Фильтр отчета.** В этот список надо перемещать поля, по которым будут фильтроваться данные отчета. За счет этого поля создается многомерность отчета.

3) **Названия столбцов.** В данный список перемещают поля, значения которых будут располагаться в названиях столбцов сводной таблицы.

4) **Названия строк.** В данный список перемещают поля, значения которых будут располагаться в названиях строк сводной таблицы.

5) **Значения.** Сюда помещают поля, значения которых будут отображаться на пересечении строк и столбцов сводной таблицы. Для отображения данных указывают формулу – сумму, среднее, количество и т.д.

На рисунке 35 выбраны только поля для отчета. Попробуйте поэкспериментировать со строками и столбцами.

	A	B	C	D	E	F	G
1	Фамилия	Кварталы	Дата продажи	Сумма		Названия строк	Сумма по полю Сумма
2	Алиева А.	тетради	02.06.2017	5000		Алиева А.	23000
3	Сатыбалдиева К.	ручки	03.06.2017	3000		02.06.2017	5000
4	Петров П.	дневники	04.06.2017	4000		06.07.2017	8000
5	Алиева А.	тетради	06.07.2017	8000		01.08.2017	10000
6	Сатыбалдиева К.	ручки	07.07.2017	10000		Петров П.	32000
7	Петров П.	дневники	08.07.2017	15000		04.06.2017	4000
8	Алиева А.	тетради	01.08.2017	10000		08.07.2017	15000
9	Сатыбалдиева К.	ручки	02.08.2017	18000		03.08.2017	13000
10	Петров П.	дневники	03.08.2017	13000		Сатыбалдиева К.	31000
11						03.06.2017	3000
12						07.07.2017	10000
13						02.08.2017	18000
14						Общий итог	86000

Рис. 35. Создание сводной таблицы

В качестве исходных данных для построения отчета мы взяли небольшую таблицу продаж. Но она могла бы состоять из большого количества строк и столбцов. В электронной таблице можно также быстро посчитать и мгновенно подготовить отчет.



Знание и понимание



1. Какие инструменты анализа данных вы знаете?
2. Как с помощью гистограмм и диаграмм можно анализировать данные?
3. Для чего применяются спарклайны?
4. Что является альтернативой для спарклайнов?
5. С какой целью применяются сводные таблицы?
6. Какие еще встроенные функции применяются для анализа данных в таблице?
7. Как с помощью встроенных функций можно проанализировать данные?



Применение



Сравните данные о доле потребления некоторых видов продуктов детьми и взрослыми (рис. 36). Данные в таблице указаны в процентах. На рисунке 37 представлена диаграмма.

	А	В	С
1	Продукты питания	Дети	Взрослые
2	Конфеты	73	-25
3	Мороженое	68	-30
4	Мясо	50	-50
5	Картофель	43	-57
6	Сок	80	-20
7	Пиво	0	-100

Рис. 36. Доля потребления некоторых видов продуктов детьми и взрослыми



Рис. 37. Диаграмма сравнения данных о потреблении продуктов детьми и взрослыми

Шаг за шагом

1. Наберите таблицу (рис. 36).
2. Выберите на вкладке **Вставка – Диаграммы – Линейчатая – Линейчатая с накоплением**.
3. Щелкните правой кнопкой мыши по **Области диаграммы**. В контекстном меню выберите **Выбрать данные**.
4. Появится диалоговое окно **Выбор источника данных**. Нажмите кнопку **Добавить**. В поле **Имя ряда**: укажите адрес ячейки с названием **Дети**, щелкнув по ней левой кнопкой мыши. В поле **Значения**: укажите диапазон всех значений столбца **Дети**, выделив их. Нажмите кнопку **ОК**.
5. В окне **Выбор источника данных** нажмите кнопку **Добавить**. Напротив поля **Имя ряда**: щелкните левой кнопкой мыши по ячейке с названием столбца **Взрослые**. Напротив поля **Значения**: укажите диапазон всех значений в столбце **Взрослые**. Нажмите кнопку **ОК** (рис. 38).

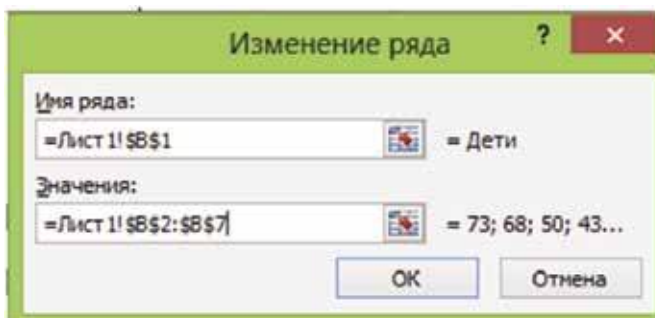


Рис. 38. Изменение ряда

6. Напротив **Подписи к горизонтальной оси** нажмите кнопку **Изменить**. Выделите названия продуктов в столбце **Продукты питания** и нажмите кнопку **ОК**. Нажмите кнопку **ОК** в окне **Выбор источников данных**.
7. На вкладке **Макет** выполните **Легенда – Добавить легенду сверху**.
8. Выделите горизонтальную ось, щелкнув по ней. Чтобы открыть окно настроек, нажмите одновременно клавиши **Ctrl + 1**, появится окно **Формат оси**. В разделе настроек **Параметры оси** для параметра **минимальное значение** выберите вариант **фиксированное** и укажите значение – 100. Для параметра **Вертикальная ось пересекает** укажите вариант **Значение оси** и введите значение – 100. Нажмите **Заккрыть**.

9. Чтобы на горизонтальной оси слева и справа от нуля были положительные числа, зайдите в окно настроек горизонтальной оси. В разделе **Число** для параметра **Числовые форматы** укажите значение **Все форматы**, в поле **Код формата** укажите **0;0** и нажмите **Добавить**. Затем выберите добавленное значение из списка **Тип** и нажмите кнопку **Заккрыть**. Вы должны получить диаграмму как на рисунке 37.



Анализ

На рисунке 39 представлены данные о 17-и озерах.

	A	B	C
1	Название	Площадь, кв. км	Наибольшая глубина, м
2	Алаколь	2650	54
3	Аральское море	51 000	68
4	Байкал	31 500	1620
5	Балкаш	18 300	26
6	Ильмень	982	10
7	Имандра	876	67
8	Иссык-Куль	6280	702
9	Каспийское море	371 000	1025
10	Ладожское	17 700	230
11	Онежское	9720	127
12	Севан	1360	86
13	Таймыр	4560	26
14	Телецкое	223	325
15	Топозеро	986	56
16	Ханка	4190	11
17	Чаны	1990	9
18	Чудское с Псковским	3550	15

Рис. 39. Данные о 17-и озерах

Подготовьте лист для получения ответа на вопрос: «Верно ли, что площадь самого большого озера превышает площадь самого маленького озера более чем в N раз?» (Значение N задается в отдельной ячейке.) Постройте гистограмму и проанализируйте данные. Выделите максимальную и минимальную площади озера.



Синтез

Для расчета коммунальных затрат на обслуживание жилых домов электроэнергией, водой и газом подготовлен следующий лист (рис. 40):

	A	B	C	D	E	F
1	Вид обслуживания		Январь	Февраль	Март	Апрель
2	Электроэнергия	Расход				
3		Тариф				
4	Вода	Расход				
5		Тариф				
6	Газ	Расход				
7		Тариф				

Рис. 40. Расчет коммунальных услуг

Расход будет указываться в соответствующих единицах измерения (электроэнергия – кВт/час, вода – литр, газ – куб. м). Тариф – это стоимость единицы расхода в тенге.

Подготовьте лист для определения:

- а) среднесрочного расхода электроэнергии, воды и газа;
- б) общих за 4 месяца затрат на каждый из трех видов обслуживания;
- в) общих затрат на все виды обслуживания за каждый месяц;
- г) общих затрат на обслуживание домов за 4 месяца.

Сравните данный лист с квитанцией по оплате коммунальных услуг в реальности. Сделайте выводы.



Оценивание



Оцените возможности анализа данных в электронной таблице Excel и напишите эссе, проанализировав получившуюся таблицу.

Рекомендации для выполнения домашнего задания

Сформулируйте свое мнение по проведению мгновенного анализа в электронной таблице Excel.



С помощью функций **ПРОПИСН** и **СТРОЧН** можно менять регистр текстовой информации в Excel. Функция **ПРОПИСН** делает все буквы прописными, а **СТРОЧН** – строчными. **ПРОПНАЧ** делает прописной только первую букву в каждом слове.

При внесении информации с нулями в начале, например, число 00785, Excel автоматически превратит его в 785. Для того чтобы оставить нули в начале, поставьте перед числом апостроф «'».

3.4. Решение прикладных задач

Научитесь

- использовать встроенные функции для решения задач с применением электронных таблиц;
- использовать различные типы и форматы данных для решения задач в электронных таблицах;
- строить графики функций, заданных в таблице.

Ключевое слово

*Прикладная
задача*

Қолданбалы есеп

Application task

Усиление прикладной направленности курса информатики – это осуществление связи теории с практикой. Поэтому прикладные задачи позволяют осознать сущность будущей профессиональной деятельности.



Прикладная задача – задача, поставленная вне информатики, но решаемая методами использования информационно-коммуникационных технологий. В прикладной задаче прослеживается взаимосвязь других знаний (математики, физики, техники и т.д.) и видов деятельности, связанных с информатикой. Приемы решения таких задач приближены к профессиональной деятельности.



По метеорологическим наблюдениям (измерениям прогнозов погоды за три года) произведите расчеты (табл. 11–14).



Шаг за шагом

1. Заполните формулами пустые ячейки. Для всех данных установлен формат ячеек:

- числовой формат – числовой;
- число десятичных знаков – 1.

2. Для расчета количества выпавших осадков, которые меньше 15 мм – «засуха», воспользуйтесь статистической функцией **ЕСЛИСЧЕТ**.

3. Заполните столбец **Прогноз**, применив вложенную функцию **ЕСЛИ**:
 - **засуха**, если количество осадков < 15 мм;
 - **дождливо**, если количество осадков > 70 мм;
 - **нормально** в остальных случаях.
4. Представьте данные таблицы **Количество осадков** (мм) графически, расположив диаграмму на Листе 2. Выберите тип диаграммы и элементы оформления по своему усмотрению.
5. Переименуйте Лист 1 в **Метео**, Лист 2 – в **Диаграмма**. Удалите лишние листы рабочей книги.
6. Установите ориентацию листа – **альбомная**.
7. Сохраните таблицу под именем **Метео**.

Таблица 11

Количество осадков

Название месяца	2015	2016	2017
январь	37,2	38,3	
февраль	11,4	67,8	
март	16,5	20,7	
апрель	19,5	26,8	
май	12,8	45,5	
июнь	130,1	71,8	
июль	57,8	161	
август	57	98,8	
сентябрь	83,5	78,3	
октябрь	87	15,9	
ноябрь	15,4	23	
декабрь	27,5	25,8	

Таблица 12

Данные за 2015–2017 годы

Максимальное количество осадков за 3 года (мм)	
Минимальное количество осадков за 3 года (мм)	
Суммарное количество осадков за 3 года (мм)	
Среднемесячное количество осадков за 3 года (мм)	
Количество засухливых месяцев за 3 года	

Таблица 13

Данные за 2017 год

№	Название месяца	2017	Прогноз
1	январь	10,7	засуха
2	февраль	2,5	засуха
3	март	14,1	засуха
4	апрель	11,8	засуха
5	май	68,8	нормально
6	июнь	70,7	дождливо
7	июль	100,3	дождливо
8	август	155,8	дождливо
9	сентябрь	78,5	дождливо
10	октябрь	71,3	дождливо
11	ноябрь	53,2	нормально
12	декабрь	13,7	засуха

Таблица 14

Данные за 2015–2017 годы

Минимальное количество осадков (мм)	2,5
Суммарное количество осадков (мм)	651,4
Среднемесячное количество осадков (мм)	54,28
Количество засушливых месяцев	5

Следующая задача применяется при анализе данных, когда нужно их отнести к тому или иному относительному диапазону. Такая задача возникает, например, при количественной оценке качества продукции, при распределении премии, при выставлении оценок и т.д.



Представим, что учащиеся на уроках технологии изготовили некоторое количество поделок. Нужно оценить их труд в зависимости от объема поделок:

- учащиеся, попавшие в верхний 20%-й участок диапазона изготовленных поделок, получают оценку «отлично»;
- учащиеся, попавшие в последние 10%, получают оценку «удовлетворительно»;
- остальные – оценку «хорошо».



Шаг за шагом

Построим функцию, реализующую цели задания. Выработку числа поделок обозначим буквой **X**. Иллюстрация техники вычисления **X_{хор}** и **X_{отл}**, являющиеся рубежами диапазонов «хорошо» и «отлично», представлена на рисунке 41.

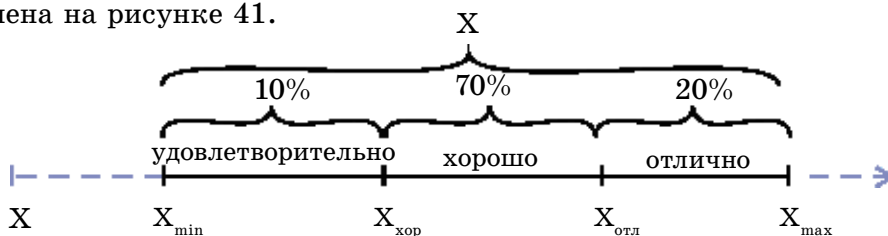


Рис. 41. Разбиение диапазона данных

Отсюда функция для ячейки C2 будет выглядеть так: $C2 = \text{ЕСЛИ}(B2 > (\text{МАКС}(B\$2:B\$10) - \text{МИН}(B\$2:B\$10)) * 90\%; \text{«отлично»}; \text{ЕСЛИ}(B2 > \text{МИН}(B2:B10) + 20\% * (\text{МАКС}(B\$2:B\$10) - \text{МИН}(B\$2:B\$10)); \text{«хорошо»}; \text{«удовлетворительно»}))$.

The screenshot shows the Microsoft Excel interface. The active worksheet contains a table with the following data:

	A	B	C
1	Имя	Поделки	Оценка
2	Тимур	25	хорошо
3	Станислав	30	хорошо
4	Мурат	35	хорошо
5	Максим	50	отлично
6	Костя	45	отлично
7	Кайрат	20	хорошо
8	Ертостик	15	удовлетворительно
9	Артем	40	отлично
10	Айдар	10	удовлетворительно

A sorting dialog box is open, showing the range A2:D10. The sort key is set to 'Имя' (Name) in ascending order. The dialog box also includes options for sorting by color or icons, and a note about the effect of sorting on the data.

Рис. 42. Таблицы оценок



Выполните сортировку по возрастанию по полю **Имя** (рис. 42). Что произошло? Поменялся ли список имен? Какой тип диаграммы лучше применить для таблицы?





Знание и понимание



1. Какие задачи называют прикладными?
2. Приведите примеры прикладных задач.
3. Что такое разбиение диапазона данных?
4. Рассмотрите таблицу расчета заработной платы (табл. 15). Заполните в ячейках используемые функции для расчета заработной платы.

Таблица 15

Курс доллара:

Расчет заработной платы

№	ФИО	Начислено (тенге)	Налоги		К выдаче в тенге	Эквива- лент в \$
			по- доход. 5%	пенс. 10%		
1	Майтбасов А.Т.	208 000				
2	Ержанов А.В.	205 000				
3	Закирова Е.Н.	235 000				
	Итого					
	Максимум					
	Минимум					
	Средняя					



Применение

Создайте кроссворд, где задание указывается в примечаниях, а правильно вставленная буква приводит к окрашиванию ячейки. Если слово вставлено правильно, то это отмечается каким-либо символом и появляется сообщение.

Примечание. При переходе от одного уровня к другому процесс создания кроссворда будет усложняться. Примените творчество.

Шаг за шагом

1. Запустите Excel.
2. Переименуйте лист:
 - Откройте контекстное меню, щелкнув правой кнопкой мыши по имени листа. Выберите команду **Переименовать** и введите имя **Кроссворд**.

3. Удалите остальные листы:
 - В контекстном меню выберите пункт **Удалить**.
4. Введите все слова кроссворда. Каждая буква вводится в отдельную ячейку таблицы.
5. Выделите все буквы:
 - Сначала выделите мышкой одно слово, затем, удерживая клавишу **Ctrl**, выделите остальные слова кроссворда.
6. Установите границы:
 - Командой **Границы** из группы **Шрифт** вкладки **Главная** выберите из списка границ пункт **Все границы** (рис. 43).

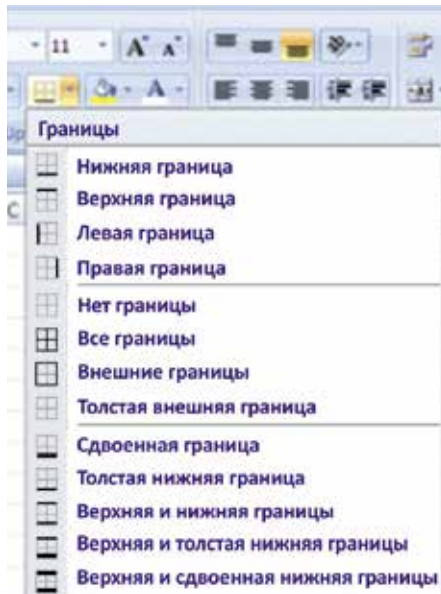


Рис. 43. Установка границы кроссворда

7. Сделайте ячейки кроссворда в виде квадратиков. Для этого выделите столбцы, где находятся слова кроссворда:
 - Установите курсор на имя столбца **D**. Нажмите на левую кнопку мыши и выделите область до столбца **K**.
8. Поставьте курсор между любыми буквами **D** и **E** (можно взять любые другие буквы выделенного диапазона) так, чтобы появился крестик .
9. С нажатой левой кнопкой мыши передвиньте крестик, чтобы уменьшить ширину столбца на 40 пикселей.
10. Увеличьте высоту строк аналогично уменьшению ширины столбцов, чтобы получилась ячейка в виде квадрата.

	А	В	С	Д	Е	Р	О	И	Т	К
1										
2										
3				И	Н	Т	Е	Р	Н	Е
4					Е					
5					Х					
6					Н					
7				Ш	К	О	П	А		
8					Л					
9					О					
10					Г					
11					И					
12					Я					

Рис. 44. Отформатированный кроссворд

11. Выровняйте буквы по центру ячеек:

- Выделите весь кроссворд.

- Выполните команду **Выровнять по центру** из группы **Выравнивание** вкладки **Главная** (рис. 44).



командой

12. Вставьте придуманные задания в примечания:

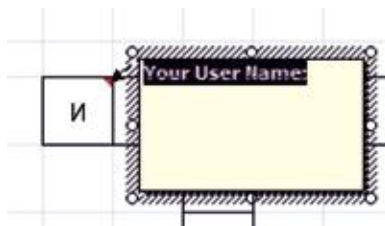


Рис. 45. Вставить примечание

- Активируйте ячейку с буквой **И (D3)**.

- В контекстном меню ячейки выберите пункт **Вставить примечание** (рис. 45).

- В появившемся окне с красным треугольником замените текст на задание (например, введите определение Интернета – это *Глобальная компьютерная сеть*).

- Так же вставьте примечания в ячейки с буквами **Ш (D7)** и **Т (F3)** – *Организация образования*.

13. Задайте условное форматирование ячейке для ввода нужной буквы.

- Выделите букву **И (D3)**.

- Выполните команду **Условное форматирование – Правила выделения ячеек – Равно** из группы **Стили** вкладки **Главная** (рис. 46).

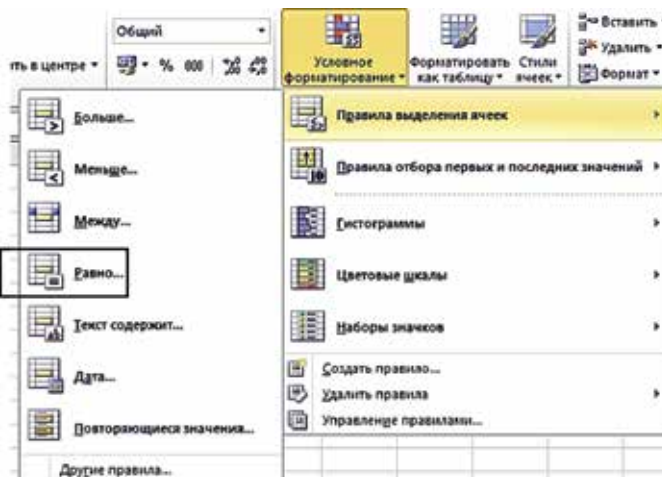


Рис. 46. Условное форматирование

- В появившемся окне **Равно** (рис. 47) вставьте букву **И**, затем выберите формат и нажмите **ОК**.
- То же сделайте с остальными буквами кроссворда.

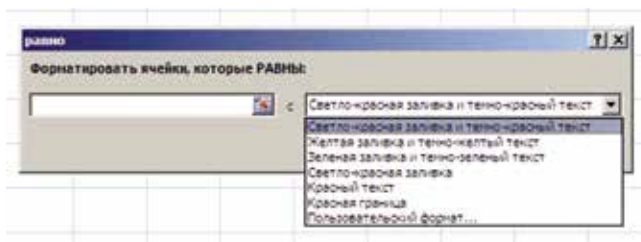


Рис. 47. Условное форматирование Равно

14. В ячейки слева или выше слова поставьте какой-нибудь символ, который будет отображаться при правильном наборе слова.

- Например, поставьте курсор в ячейку **М3**. Выберите функцию **ЕСЛИ** на вкладке **Формулы**, нажав на команду **Вставить функцию** (или **SHIFT + F3**).

- В появившемся списке выберите **Другие функции...**
- В окне **Мастер функций** выберите из категории **Логические** функцию **ЕСЛИ**.

- В окне **Аргументы функции** (рис. 48) поставьте курсор в область **«Значение_если_истина»** и наберите любой символ, например, **γ**. Данный символ будет выводиться, если ответ верный.

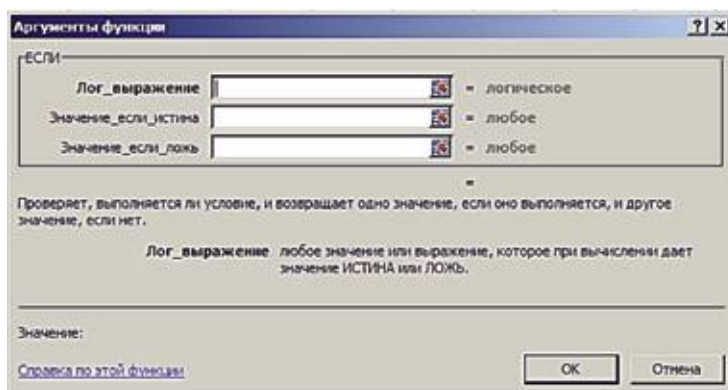


Рис. 48. Аргументы функции

- Поставьте курсор в область **Значение_если_ложь** и наберите кавычки «».
- Поставьте курсор в область **Лог_выражение** и выберите функцию **И** (рис. 49).
- Слева в командной строке около функции **ЕСЛИ** откройте список, нажав на черную стрелку **М**, выберите **Другие функции**.
- В окне **Мастер функций** выберите логическую функцию **И**.

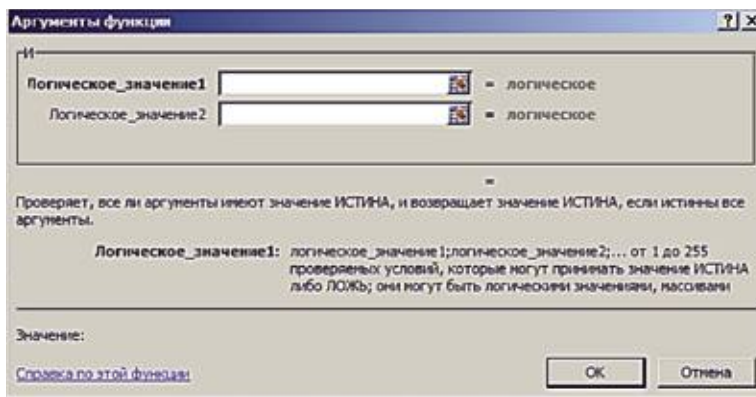


Рис. 49. Аргументы функции

- Поставьте курсор в область **Логическое_значение1**, выделите ячейку с буквой **И**. В области появится адрес ячейки, допишите «**И**».
- Перенесите курсор в следующую область **Логическое_значение2**, выделите ячейку с буквой **Н**. В области появится адрес ячейки, допишите «**Н**» и т.д. (рис. 50).

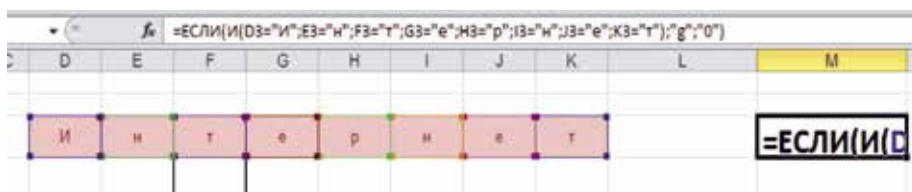


Рис. 50. Ввод формулы в ячейке М3

Рекомендации для выполнения задания

Составьте кроссворд в бумажном варианте. Например, в замечание введите вопрос кроссворда: 1) *Организация образования* – школа; 2) *Глобальная компьютерная сеть* – Интернет и т.д. Количество вопросов в кроссворде – от 5 до 7.



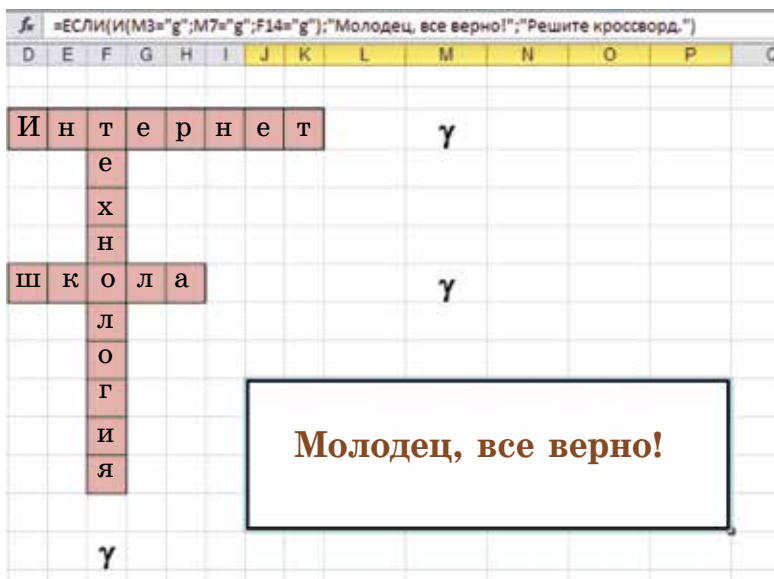
Анализ

Проанализируйте данные в кроссворде. Запишите для них функцию **ЕСЛИ**.

– В ячейках с символом измените шрифт – **Symbol**. В формуле будет отображаться символ **g**, в ячейке – **γ**.

ЕСЛИ(И(D3=«И»;E3=«н»;F3=«т»;G3=«е»;H3=«р»;I3=«н»;J3=«е»;K3=«т»);«g»;«0»)

15. Выведите информацию о работе над кроссвордом (рис. 51):



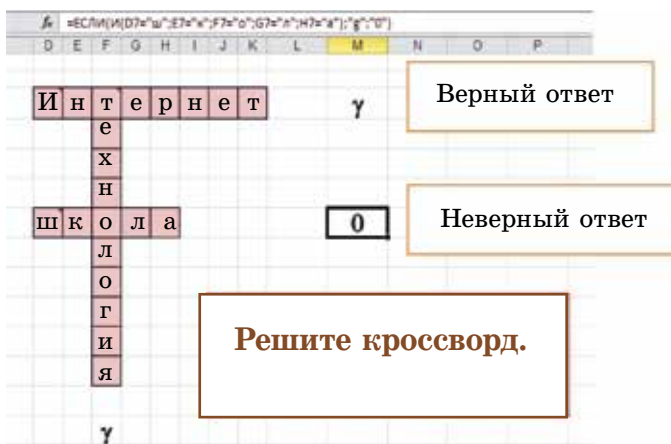


Рис. 51. Результаты решения кроссворда



Синтез

Для очистки кроссворда создайте макрос. Макрос – автоматизация повторяющихся задач.

- Выделите область ячеек **M11:S14**.
 - Введите формулу аналогично описанному выше.
 - =ЕСЛИ(И(M3=«g»;M7=«g»;F14=«g»);«Молодец, все верно!»;«Решите кроссворд.»)
 - Выполните условное форматирование области ячеек **M11:S14**.
16. Уберите сетку. Для этого нужно убрать галочку с **Сетка** в группе **Показать** или скрыть на вкладке **Вид**.
17. Очистите кроссворд (рис. 52):
- Выделите область ячеек **A1:R14**.
 - Нажмите на клавишу **Delete**.

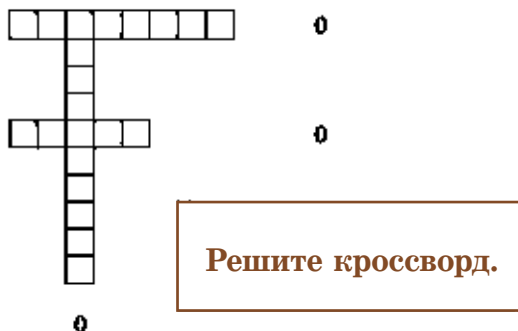


Рис. 52. Очистка кроссворда

18. Создайте кнопку для автоматической очистки кроссворда:

- Создайте **Макрос** для очистки кроссворда. Выполните команду **Макросы – Запись макроса** из группы **Макросы** вкладки **Вид** (рис. 53).



Рис. 53. Окно Макросы

- В появившемся окне **Запись макроса** впишите **Имя макроса – Очистка**, нажмите **ОК** (рис. 54).

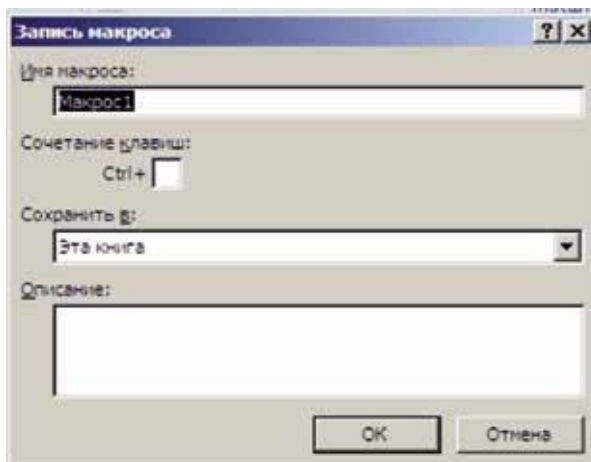


Рис. 54. Окно записи макроса

- На листе **Кроссворд** выделите область кроссворда (D3:K12) и клавишей **Delete** очистите область.
- Выполните команду **Макросы – Остановить запись** из группы **Макросы** вкладки **Вид**.



Оценивание

Как нарисовать кнопку на листе, чтобы очистить кроссворд?

19. Добавьте на **Ленту** вкладку **Разработчик**. Меню **Файл**, пункт **Параметры – Дополнительно**. Поставьте галочку **Разработчик**. Нажмите кнопку **ОК**.

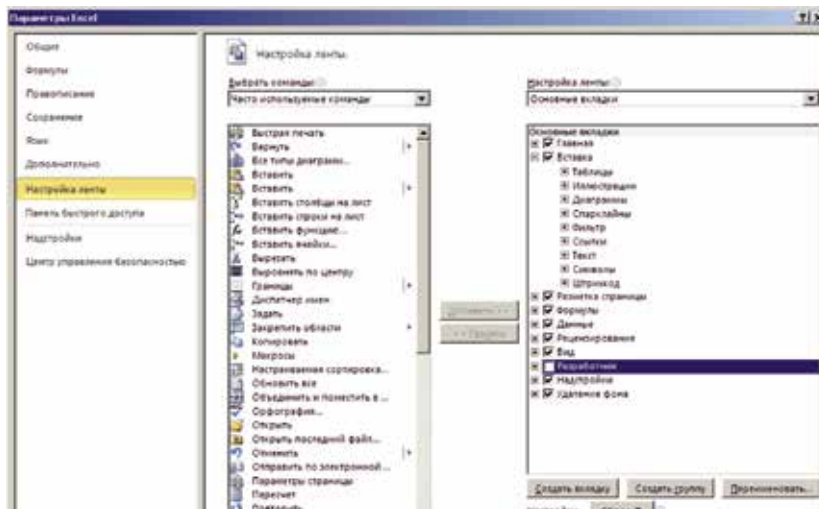


Рис. 55. Параметры Excel

20. Нарисуйте кнопку для очистки кроссворда:

– Щелкните по **Кнопке** (элемент управления формы), вкладка **Разработчик**, группа **Элемент управления**, команда **Вставить** (рис. 56).

– Переведите курсор на лист и нажмите на левую кнопку мыши. Нарисуйте кнопку.

– В появившемся окне **Назначить макрос объекту** выберите нужный **Макрос** – **Очистка**, **ОК** (рис. 57).

– Выделите кнопку и измените надпись на ней.

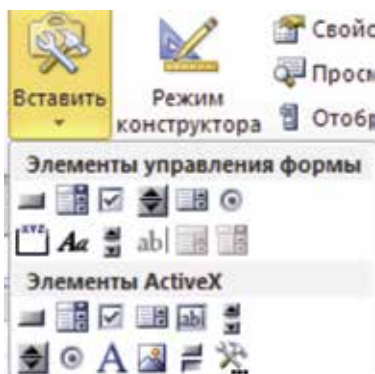


Рис. 56. Элементы управления

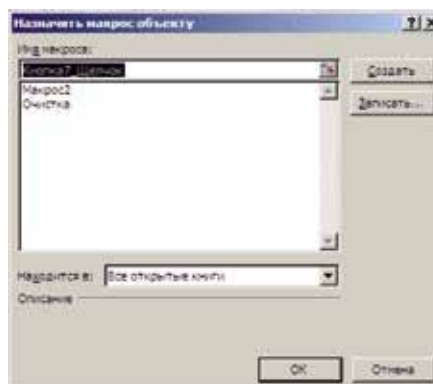


Рис. 57. Назначение макроса

21. Сохраните документ: команда **Сохранить как...** из меню **Файл**.
22. Рассмотрите пример готового кроссворда (рис. 58).

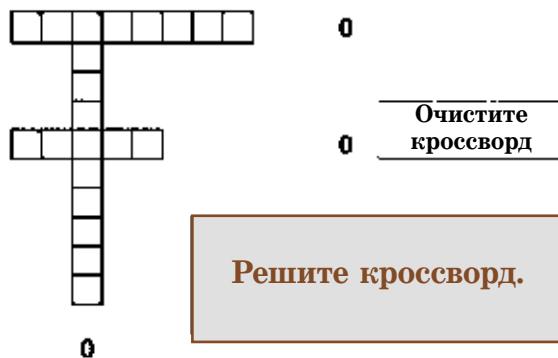


Рис. 58. Готовый кроссворд

23. Оцените кроссворд.
24. Рекомендуйте кроссворд.



Инструмент **Камера** или как перевернуть текст в ячейке?

Инструмент **Камера** по умолчанию недоступен во всех версиях Excel. Рассмотрим алгоритм вывода **Камеры** на **Панель быстрого доступа** в электронной таблице Excel (версии после 2003).

Выберите **Настройка панели быстрого доступа** из команды **Все команды – Камера – Добавить – ОК**.

Выделите ячейку с данными, которую необходимо скопировать. Лево́й кнопкой мыши кликните на значок инструмента **Камера**. Появится окантовка выделенной ячейки, то есть ячейка станет активной как при обычном копировании.

Лево́й кнопкой мыши укажите место на листе, куда необходимо вставить снимок. Снимок автоматически вставляется в ячейку. Проверните его на 180 градусов.

С получившимся снимком можно работать как с обычным рисунком, внедренным на лист. Можно изменять его размеры, свойства, форму, устанавливать прозрачность, рамки и т.д.

Поэкспериментируйте. Можно ли одновременно «фотографировать» несколько ячеек/диапазонов или несмежные диапазоны?



Рис. 59. Поворот рисунка

Задание для проектной работы

Проект 3

Создайте таблицу по теме «Статистические данные по областям Республики Казахстан» и проанализируйте ее. На основе статистических данных составьте краткий отчет. (Отчет – это сообщение или документ, который предоставляет информацию и описывает суть исследования применительно к определенной ситуации. Отчет основывается на привлечении документальных данных.)

Требования к оформлению проекта:

1. Используйте для обработки информации электронную таблицу Microsoft Excel.
2. Создайте таблицу, в которой отобразите статистические данные о численности населения, проживающего в областях Казахстана (казахи, русские и другие национальности).
3. Вставьте соответствующие рисунки людей в национальных костюмах.
4. Задайте таблице различное оформление.
5. Отобразите данные на различных типах диаграмм. Задайте диаграмме заголовок, названия осей, подписи данных.
6. Добавьте краткий отчет.

Дополнительные задания к третьему разделу

1. Решите в электронной таблице Excel квадратные уравнения: $y = x^2$; $y = x^2 + 1$; $y = x^2 + x + 1$ (рис. 60). При этом значения x меняются в интервале от -5 до 5 . Постройте в одной координатной плоскости три графика функции.

x	$y = x^2$	$y = x^2 + 1$	$y = x^2 + x + 1$
-5	25	26	21
-4	16	17	13
-3	9	10	7
-2	4	5	3
-1	1	2	1
0	0	1	1
1	1	2	3
2	4	5	7
3	9	10	13
4	16	17	21
5	25	26	31

Рис. 60. Квадратные уравнения

2. Числа Фибоначчи – это ряд чисел, в котором каждое последующее число равно сумме двух предыдущих: 1, 1, 2, 3, 5, 8, 13 и т.д.

Формула:

$$F_0 = 1$$

$$F_1 = 1$$

$$F_2 = F_1 + F_0 = 1 + 1 = 2$$

$$F_3 = F_2 + F_1 = 2 + 1 = 3$$

$$F_4 = F_3 + F_2 = 3 + 2 = 5$$

$$F_5 = F_4 + F_3 = 5 + 3 = 8$$

.....

$$F_n = F_{n-1} + F_{n-2}$$

В электронной таблице Excel проведите расчет десяти чисел Фибоначчи. Постройте два типа диаграмм: гистограмму и график с маркерами. Сравните их. Какой тип диаграммы лучше отражает последовательность чисел Фибоначчи?

3. Составьте в электронной таблице Excel журнал класса по информатике за одну четверть, в котором выведены баллы от 1 до 10. Диапазон баллов дан в десятичной системе оценивания: если количество баллов меньше 5, то выводится оценка «2», от 5–6 баллов – оценка «3», от 7–8 – оценка «4», от 9–10 – оценка «5». Для вывода оценок используйте вложенную функцию ЕСЛИ. Установите для данных: **формат данных – числовой формат – общий**.

4. На стыке двух континентов Европы и Азии расположена Республика Казахстан. Казахстан занимает площадь, равную 2724,9 тысячам квадратных километров (тыс. км²). Протяженность его территории с запада на восток превышает 3000 км, с юга на север – 1700 км. Казахстан граничит на севере с Россией – 6467 км, на востоке с Китаем – 1460 км, на юге с Кыргызстаном – 980 км, Узбекистаном – 2300 км и Туркменистаном – 380 км, на западе омывается Каспийским морем. Общая протяженность границ – 12 187 км. Заполните представленные данные (взяты с сайта Комитета индустрии туризма Министерства туризма и спорта Республики Казахстан) в электронной таблице Excel. Посчитайте границы РК по Каспию в километрах. Какую диаграмму по этим данным лучше построить?

Итоговый тест к третьему разделу

Вариант 1

1. Табличные процессоры — это:

- А) прикладные программы, предназначенные для работы с электронными таблицами;
- В) прикладные программы, предназначенные для работы с базами данных;
- С) прикладные программы, предназначенные для работы с текстом;
- Д) таблицы в Word.

2. В ячейке D7 записана формула $(C3+C5)/D6$. Как она изменится при переносе этой формулы в ячейку E8?

- А) $(C3+C5)/D6$.
- В) $(C3+C5)/E6$.
- С) $(D4+D6)/E7$.
- Д) $(C4+C6)/D7$.

3. Следствием принципа абсолютной адресации является правило:

- А) при перемещении формулы в другую ячейку адреса ячеек не изменяются;
- В) при перемещении формулы в другую ячейку изменяется только номер строки в адресах ячеек;
- С) при перемещении формулы в другую ячейку изменяются адреса ячеек (номер строки и буква столбца);
- Д) при перемещении формулы в другую ячейку изменяется только имя столбца (буква) в адресах ячеек.

4. Дана электронная таблица:

	A	B	C	D
1	5	3	4	
2	6	1	5	$=A1+ \$B\$2 * C\$1$
3	8	3	-4	

В ячейку D2 введена формула $(A1+* \$B\$2 * C\$1)$ и перемещена в ячейку D3. Какое значение появится в ячейке D3?

- А) 9.
- В) 21.
- С) 10.
- Д) 20.

5. Фрагмент или блок таблицы – это:

- А) часть строки;
- В) часть столбца;
- С) любая ее прямоугольная часть;
- Д) ячейка.

6. Форма графического представления числовых значений, позволяющая облегчить восприятие и интерпретацию числовых данных, называется:

- А) чертежом;
- В) картой;
- С) блок-схемой;
- Д) диаграммой.

7. График функции можно создать в Excel при помощи:

- А) Строки формул;
- В) Мастера Функций;
- С) Мастера Шаблонов;
- Д) Мастера Диаграмм.

8. Диаграмма – это:

- А) карта местности;
- В) форма графического представления числовых значений, которая позволяет облегчить интерпретацию числовых данных;
- С) красиво оформленная таблица;
- Д) обычный график.

9. Круговая диаграмма – это диаграмма:

- А) которая представлена в виде круга, разбитого на секторы, и в которой допускается только один ряд данных;
- В) в которой отдельные ряды данных представлены в виде закрашенных разными цветами областей;
- С) в которой используется система координат с тремя координатными осями, что позволяет получить эффект пространственного представления рядов данных;
- Д) в которой отдельные значения представлены точками в декартовой системе координат.

10. На основе чего строится любая диаграмма?

- A) Книги Excel.
- B) Графического файла.
- C) Текстового файла.
- D) Данных таблицы.

11. При построении диаграммы она получилась пустой. Почему это произошло?

- A) Не хватило оперативной памяти. Следует закрыть ненужную программу.
- B) Выбран неверный путь диаграммы.
- C) Не выделен блок ячеек с данными.
- D) Диаграмма не помещается на экране, так как слишком большая.

12. Абсолютные ссылки – это такие ссылки, которые при копировании формулы из одной ячейки в другую:

- A) меняются, если только эти ячейки расположены рядом;
- B) не меняются;
- C) меняются, причем ссылки в новой формуле имеют такое же взаимное расположение ячеек, участвующих в формуле, как и в формуле-оригинале;
- D) не копируются.

13. В ячейке F6 записана формула =МИН(B2:D5). Эту формулу скопировали в ячейку G6. Укажите формулу, которая запишется в ячейку G6.

- A) =МИН(C2:D5).
- B) =МИН(C2:E5).
- C) =МИН(E2:F5).
- D) =МИН(B2:D5).

14. В ячейках записаны числа: в A1 – 5, в A2 – 5, в A3 – 5, в A4 – 5, в B1 – 100, в B2 – 1000, в B3 – 10000, в C1 – 5000. В ячейке A6 записана формула =МИН(A1:B2)+\$C\$1. Какой результат в ней будет?

- A) 5120.
- B) 1120.
- C) 5005.
- D) 5000.

15. В ячейках записаны числа: в A1 – 5, в A2 – 5, в A3 – 5, в A4 – 5, в B1 – 100, в B2 – 1000, в B3 – 10000. В ячейке A6 записана формула =СУММ(A1:B2). Какой результат в ней будет?

- A) 1020.
- B) 1110.
- C) 1120.
- D) 1010.

16. Круговая диаграмма построена для трех ячеек с числами 30, 30, 40. На ней возле секторов указаны значения 30%, 30%, 40%. Какие значения были бы указаны у секторов, если бы в этих ячейках содержались числа 60, 60, 80?

- A) 60%, 60%, 80%.
- B) 30%, 30%, 40%.
- C) 25%, 25%, 50%.
- D) значений не будет, так как диаграмма не будет построена, поскольку $60 + 60 + 80 > 100$.

17. Какой отчет называется сводной таблицей?

- A) Сводная таблица – это отчет, построенный на основе символьных данных.
- B) Сводная таблица – это отчет, построенный на основе табличных данных.
- C) Сводная таблица – это отчет, построенный на основе формул.
- D) Сводная таблица – это отчет, построенный на основе статистических функций.

18. Табулирование функции – это:

- A) вычисление значений функции от некоторого начального значения до некоторого конечного значения;
- B) вычисление значений функции при изменении аргумента;
- C) вычисление значений функции при изменении аргумента от некоторого начального значения до некоторого конечного значения;
- D) вычисление значений функции при изменении аргумента от некоторого начального значения до некоторого конечного значения с определенным шагом.

Вариант 2

1. В электронной таблице выделили группу из 4 ячеек. Это могут быть ячейки:

- A) A1:B4;
- B) A1:C2;

- C) A1:B2;
- D) A2:C4.

2. Дана электронная таблица:

	A	B	C	D
1	5	2	4	
2	10	1	6	=A2*B1+C1

В ячейку D2 введена формула: $(A2*B1+C1)$. Какое значение появится в ячейке D2?

- A) 6.
- B) 14.
- C) 16.
- D) 24.

3. В ячейке B1 записана формула $=2*\$A1$. Какой вид приобретет формула, после того как ячейку B1 скопируют в ячейку C2?

- A) $=2*\$B1$.
- B) $=2*\$A2$.
- C) $=3*\$A2$.
- D) $=3*\$B2$.

4. Деловая графика представляет собой:

- A) совокупность графиков функций;
- B) графические иллюстрации;
- C) график совещания;
- D) совокупность программных средств, позволяющих представить в графическом виде закономерности изменения числовых данных.

5. Гистограмма наиболее пригодна для:

- A) отображения вклада каждого значения в общую сумму;
- B) отображения динамики изменения данных;
- C) сравнения различных членов группы;
- D) отображения удельных соотношений различных признаков.

6. Диаграмма в электронных таблицах — это:

- A) качественно оформленная числовая таблица;
- B) график, отображающий зависимость между всеми числами таблицы;
- C) средство наглядного графического изображения информации, пред-

назначенное для сравнения одной или нескольких величин или нескольких значений одной величины;

D) зависимость между числовыми значениями.

7. В каком разделе меню табличного процессора Excel находится Мастер Диаграмм?

A) Правка.

B) Вставка.

C) Сервис.

D) Данные.

8. Гистограмма – это диаграмма, в которой:

A) отдельные значения представлены полосами различной длины, расположенными горизонтально вдоль оси ОХ;

B) отдельные значения представлены вертикальными столбцами различной высоты;

C) используется система координат с тремя координатными осями;

D) используются параллелепипеды, размещенные вдоль оси ОХ.

9. Диаграмма, в которой отдельные значения представлены точками в декартовой системе координат, называется:

A) линейчатой;

B) точечной;

C) круговой;

D) гистограммой.

10. Линейчатая диаграмма – это диаграмма:

A) в которой отдельные значения представлены вертикальными столбиками различной высоты;

B) в которой отдельные значения представлены точками в декартовой системе координат;

C) в которой отдельные значения представлены полосами различной длины, расположенными горизонтально вдоль оси ОХ;

D) которая представлена в виде круга, разбитого на секторы, и в которой допускается только один ряд данных.

11. Чтобы отформатировать какой-либо элемент диаграммы, необходимо:

A) дважды щелкнуть по нему курсором мыши;

B) выполнить команды **Диаграмма, Исходные данные**;

- С) выполнить команды **Диаграмма, Добавить данные**;
- Д) все вышеперечисленные ответы верны.

12. Укажите неверное высказывание относительно следующей фразы. Вместо формулы $=A2+A3+A4+A5+A6+A7+A8+A9+A10$ предпочтительнее пользоваться формулой $=СУММ(A2:A11)$.

- А) Вторая формула более удобна для чтения и записи.
- В) При вставке новой ячейки в диапазон A2:A10 формула будет откорректирована автоматически и получится $=СУММ(A2:A11)$.
- С) Сумма может быть подсчитана только по первой формуле.
- Д) Первую формулу можно написать мышкой.

13. Относительные ссылки – это такие ссылки, которые при копировании формулы из одной ячейки в другую:

- А) меняются, если только эти ячейки расположены рядом;
- В) не меняются;
- С) меняются, причем ссылки в новой формуле имеют такое же взаимное расположение ячеек;
- Д) не копируются.

14. В ячейке E8 записана формула $=МИН(B2:D5)+\$A\2 . Эту формулу скопировали в ячейку F8. Укажите формулу, которая запишется в ячейку F8.

- А) $=МИН(C2:D5)$.
- В) $=МИН(C2:E5)+\$A\2 .
- С) $=МИН(E2:F5)+\$B\2 .
- Д) $=МИН(B2:D5)+\$A\2 .

15. В ячейках записаны числа: в A1 – 5, в A2 – 5, в A3 – 5, в B1 – 100, в B2 – 1000, в C1 – 0, в C2 – 0, в D1 – 0. В ячейке A6 записана формула $=МИН(A1:B2)+\$D\1 . Эту формулу скопировали в ячейку B6. Какой результат в ней будет?

- А) 5005.
- В) 0.
- С) 6000.
- Д) 5100.

16. Круговая диаграмма построена для трех ячеек с числами 40, 60, 100. Какие значения будут указаны у секторов на диаграмме?

- А) 40%, 60%, 100%.
- В) 30%, 30%, 40%.

- С) 20% , 30% , 50% .
D) значений не будет, так как диаграмма не будет построена, поскольку $60 + 60 + 80 > 100$.
E) 20% , 20% , 60% .

17. Какие функции в Excel называются встроенными?

- A) Это функции, которые предназначены для расчетов, вычислений и анализа данных.
B) Это функции, которые способны вернуть значение.
C) Это функции, которые способны вернуть значение в зависимости от переданного им параметра и предназначены для расчетов, вычислений и анализа данных.
D) Это функции, которые не способны вернуть значение в зависимости от переданного им параметра и предназначены для расчетов, вычислений и анализа данных.

18. Прикладная задача — задача, поставленная вне информатики, но решаемая методами использования:

- A) математики;
B) ИКТ;
C) анализа данных;
D) физики.

Раздел 4

Интегрированные среды разработки программ

4.1. Классификация программного обеспечения

Узнаете

как различать системное, прикладное программное обеспечение и системы программирования.

Ключевое слово

*Программное
обеспечение*

*Бағдарламалық
қамтамасыз ету*

Software



Что вы понимаете под программным обеспечением? Обладает ли компьютер сам по себе знаниями для различных областей применения?



Программное обеспечение (ПО или Software) – это совокупность всех программ, обеспечивающих функционирование аппаратных средств, разработку, отладку и выполнение задач пользователя.

Программное обеспечение – неотъемлемая часть компьютерной системы. Оно является продолжением технических средств. В зависимости от сферы применения конкретного компьютера определяется его программное обеспечение.

Сам по себе компьютер не обладает знаниями ни в одной области применения. Все эти знания сосредоточены в выполняемых программах, записанных на компьютер. Программное обеспечение современных компьютеров, смартфонов, планшетов включает миллионы программ – от игровых до научных.

Все программы, работающие на компьютере, можно условно разделить на три категории (рис. 61). При построении классификации программного обеспечения нужно учитывать факты стремительного развития компью-

терной техники. Расширение сферы применения компьютеров ускорило процесс эволюции программного обеспечения.

Даже классическое программное обеспечение, например, операционные системы, непрерывно развивается. Оно наделяется такими интеллектуальными функциями, которые раньше относились к интеллектуальным возможностям человека.

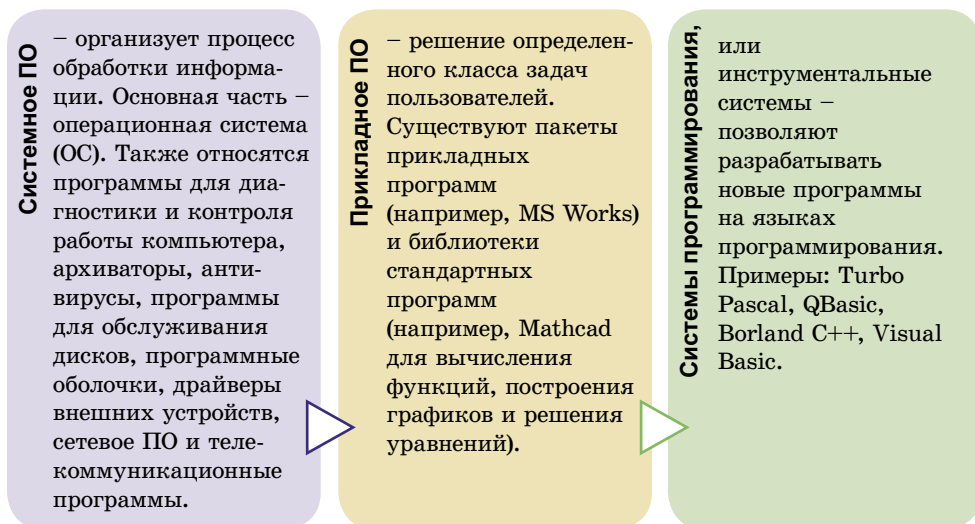


Рис. 61. Категории программного обеспечения



Системные программы выполняются вместе с прикладными и служат для управления ресурсами компьютера – центральным процессором, памятью, вводом-выводом.

Системные программы – это программы общего пользования. Они предназначены для всех пользователей компьютера.

Операционная система занимает особое место среди системных программ. Она обеспечивает управление ресурсами компьютера с целью их эффективного использования.

Компоненты операционной системы делятся на 2 класса: **системные** и **прикладные**. К прикладным компонентам относятся **текстовые редакторы, компиляторы, отладчики, системы программирования, программы графического вывода информации, коммуникационные программы** и т.д.

К системным компонентам относятся **ядро системы**, обеспечивающее взаимодействие всех компонентов, **загрузчик программ, подсистемы**, обеспечивающие диалог с человеком – **оконная система, интерпретатор команд и файловая система**. Именно системные компоненты операционной системы определяют ее основные свойства.



Прикладная программа – это любая конкретная программа, которая способствует решению какой-либо задачи в пределах данной программной области.

На прикладные программы возложены задачи как индивидуального, так и общего характера. Например, контроль за финансовой деятельностью, подготовка платежных ведомостей – это задачи индивидуального характера. Составление и печатание документов – задача общего характера.

Прикладные программы могут использоваться автономно, то есть решать задачи без помощи других программ либо в составе программных комплексов или пакетов. Рассмотрите классификацию программ прикладного уровня (рис. 62).

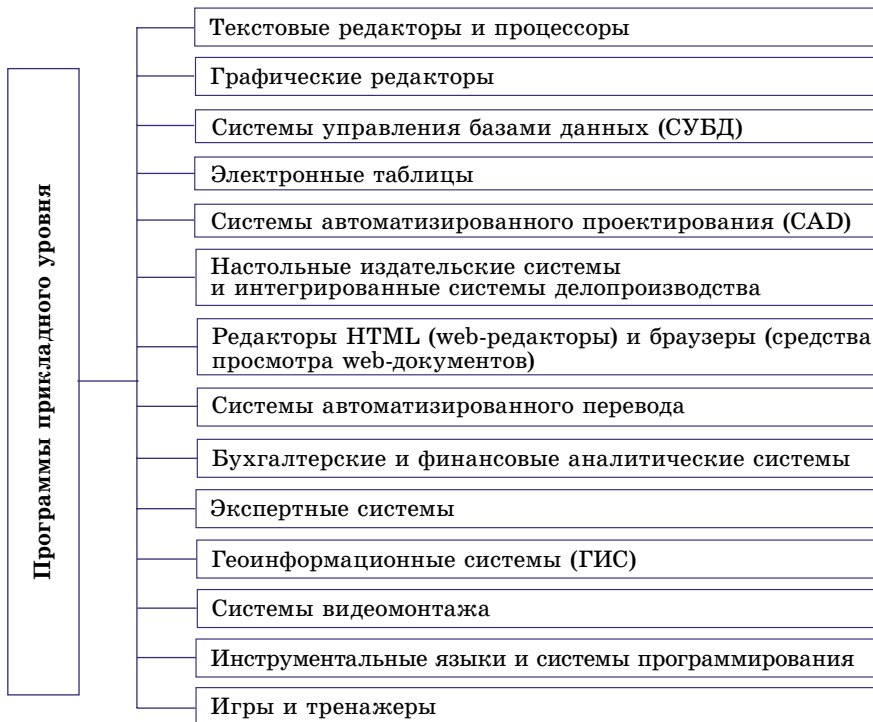


Рис. 62. Классификация прикладных программ



Система программирования – это система для разработки новых программ на конкретном языке программирования.

Современные системы программирования обычно предоставляют пользователям мощные и удобные средства разработки программ.

К ним относятся:

- компилятор или интерпретатор;
- интегрированная среда разработки;
- средства создания и редактирования текстов программ;
- библиотеки стандартных программ и функций;
- отладочные программы, помогающие находить и устранять ошибки в программе;
- «дружественная» к пользователю диалоговая среда;
- многооконный режим работы;
- графические библиотеки;
- утилиты для работы с библиотеками;
- встроенная справочная служба;
- другие специфические особенности.

Раньше отладка программы была длительным, трудоемким и утомительным процессом. В настоящее время разработаны и успешно используются системы программирования, представляющие собой единую инструментальную среду. В рамках одного программного пакета осуществляются все операции. Кроме того, пакет обычно снабжается удобными средствами отладки программ, системой контекстной помощи и рядом дополнительных сервисных возможностей.



Инструментальная среда – это интегрированная система, которая используется в ходе разработки, корректировки, редактирования, отладки, запуска программы на выполнение, не выходя из системы.

Системы программирования включают в себя языки программирования и трансляторы, позволяющие разрабатывать как системное, так и прикладное программное обеспечение. Современные системы программирования для персональных компьютеров обычно предоставля-

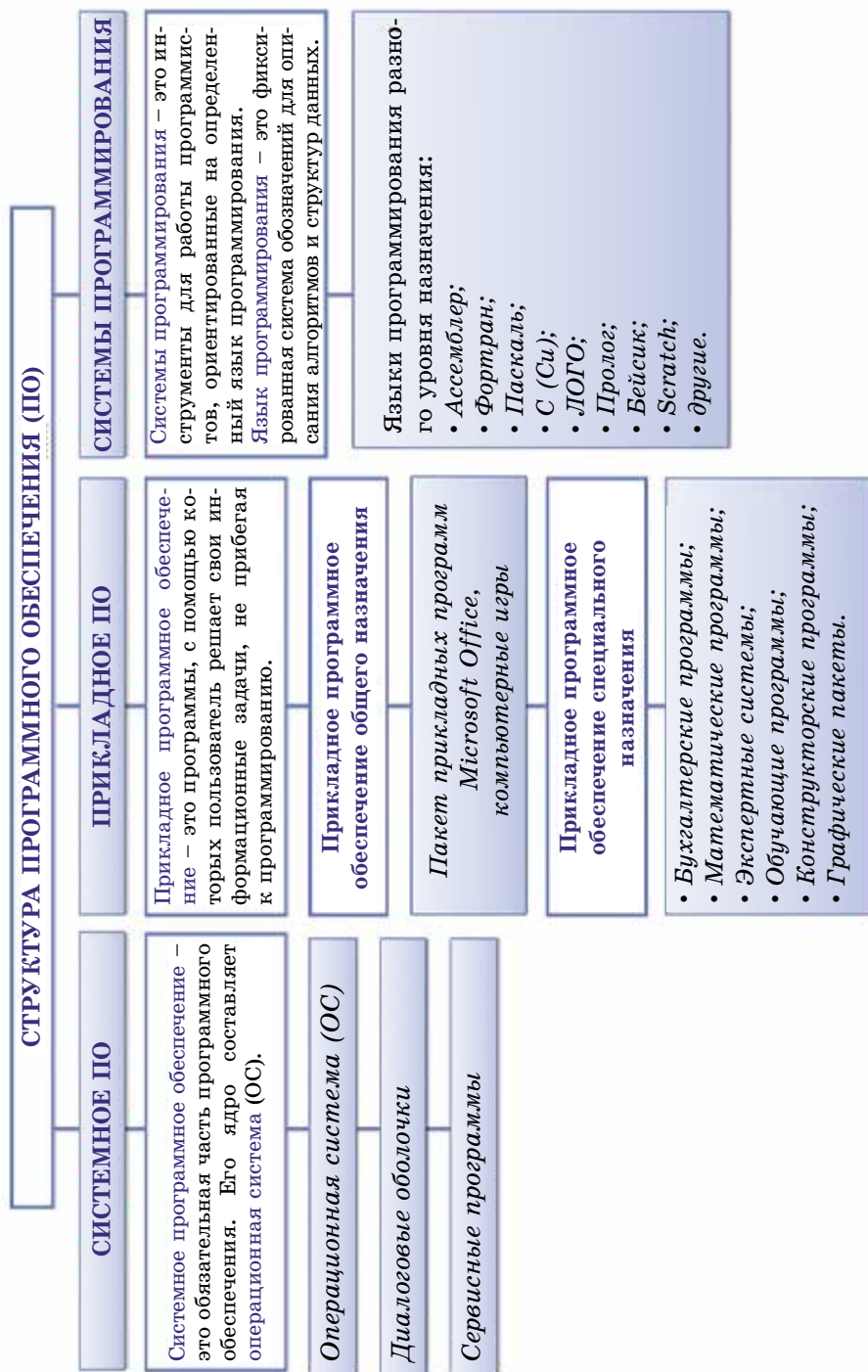


Рис. 63. Структура программного обеспечения

ют пользователю весьма мощные и удобные средства для разработки программ.

Для разработки приложений средствами визуального программирования применяются инструментальные среды C++, Visual Basic, Java, Delphi. Система имитационного моделирования MatLab, системы моделирования бизнес-процессов BpWin и баз данных ErWin используются для моделирования различных процессов.

Для обобщения информации на рисунке 63 представлена структура программного обеспечения. Эта классификация не является полной, но она более или менее наглядно отражает направления совершенствования и развития программного обеспечения.



Знание и понимание

1. Для чего используется программное обеспечение?
2. Перечислите три категории программного обеспечения.
3. Что понимается под системным программным обеспечением?
4. Какие программы относятся к системному программному обеспечению?
5. К какой категории программного обеспечения относится операционная система?
6. Перечислите программы, которые относятся к прикладным.
7. Чем отличается системное программное обеспечение от прикладного?
8. Какие средства разработки программ представляют современные системы программирования?
9. Проанализируйте структуру программного обеспечения, показанную на рисунке 63.



Применение

Что пропущено в программном обеспечении на рисунке 64? Нарисуйте схему в текстовом редакторе Word и заполните ее, записав вместо вопросительных знаков ответы в прямоугольниках.

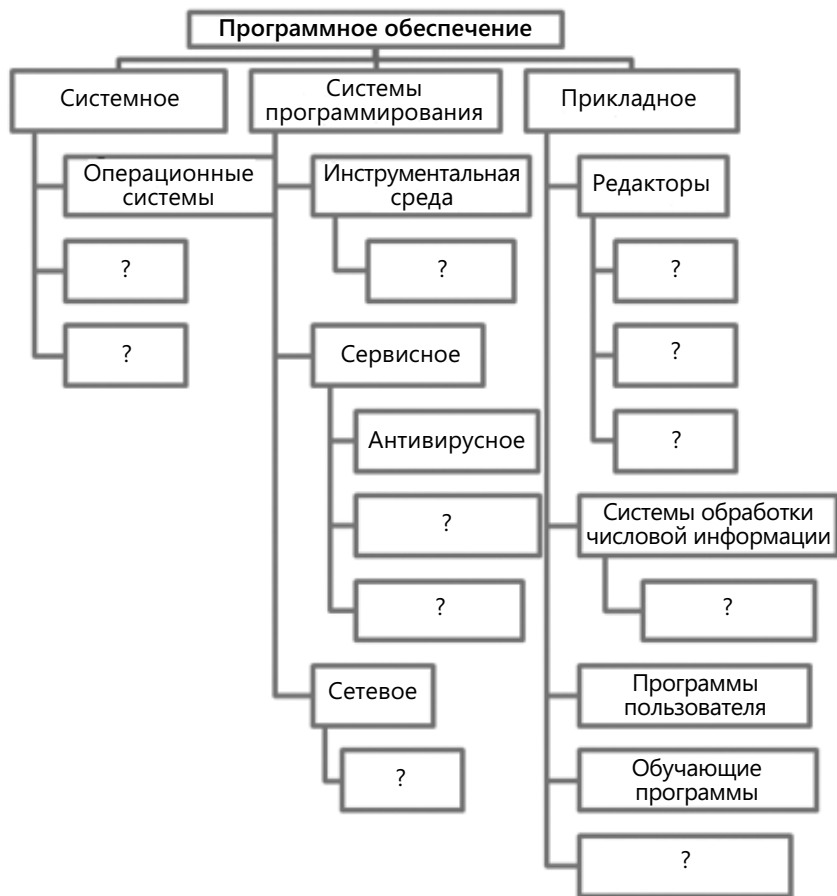


Рис. 64. Программное обеспечение

Анализ

Укажите, какое программное обеспечение необходимо людям в следующих ситуациях (заполните таблицу 16):

Таблица 16

Ситуация	Системное ПО	Прикладное ПО	Системы программирования
Ландшафтные дизайнеры создают проект нового городского ландшафта			

Профессиональный программист пишет компьютерную программу по заказу крупной фирмы			
Ученые научно-исследовательского института расшифровывают записи, переданные марсоходом			
Учащиеся 8 класса готовят фотоальбом и собирают воспоминания о своей школьной жизни			
Web-дизайнер создает сайт известной фирмы			
Школьник играет в компьютерную игру			
Создатели нового мобильного телефона пробуют различные варианты дизайна			
Учитель пишет компьютерный тест по своему предмету			
Конструкторы проектируют модель новой подводной лодки			

Рекомендации для выполнения домашнего задания

Для выполнения задания используйте рисунок 63.



Синтез

Оценивание

Проведите исследование программного обеспечения вашего компьютера. Классифицируйте программы по уровням: системные, прикладные и инструментальные. Дайте краткую характеристику программам. Обоснуйте ответ. Для визуализации информации можете применить интеллект-карту, кластер и т.д.



К основным тенденциям развития программного обеспечения относятся:

- ориентация на объектно-ориентированное проектирование и программирование программных средств;
- интеллектуализация интерфейса пользователя, обеспечение его интуитивной понятности, использование средств мультимедиа;
- широкое использование методов искусственного интеллекта, позволяющее делать приложения более «умными» и решать все более сложные задачи;
- ориентация на совместную групповую работу пользователей при решении той или иной проблемы при помощи сетевого программного обеспечения (предназначено для организации совместной работы группы пользователей на разных компьютерах);
- внедрение программного обеспечения в аппаратную часть, что повышает его надежность;
- и другие.

4.2. Компоненты интегрированной среды разработки программ

Узнаете

как использовать компоненты интегрированной среды разработки программ.

Ключевое слово

*Интегрированная
среда разработки
программ*

*Бағдарламалық
қамтамасыз
етудің кірікті-
рілген ортасы*

*Integrated
software
development
environment*



Вспомните, какую интегрированную среду вы начали изучать.



Четкого определения термина «интегрированная среда разработки» не существует. Рассмотрим определение из Википедии.



Интегрированная среда разработки ИСР/IDE (англ. *Integrated development environment*) – комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО).

Современные интегрированные среды разработки предлагают программистам большое количество возможностей. Например, многие среды разработки являются визуальными. Они позволяют создавать интерфейс программы без написания программного кода.

Интегрированные среды разработки, не являющиеся визуальными, требуют от программиста писать специальный код. Этот код отвечает за создание пользовательского интерфейса программы.

В зависимости от того, для каких платформ можно писать программы и на каких платформах работает сама среда разработки, они разделяются на:

- **кроссплатформенные**, которые поддерживают работу с разными платформами;
- **платформенно-зависимые**, которые работают только с одной платформой.

Кроссплатформенными можно назвать большинство современных высокоуровневых языков программирования. Например, С, С++, Free Pascal, Eclipse CDT. Для этих языков существуют компиляторы под различные платформы. Среда программирования Delphi относится к платформенно-зависимой.

Мы продолжим изучать кроссплатформенную среду программирования Code::Blocks. Это бесплатная среда разработки. Она позволяет писать не только на языке программирования С/С++, но и обеспечивает поддержку других языков программирования.

Для каждой из интегрированных сред разработки характерно наличие следующих **компонентов**:

- **единая интерактивная оболочка** используется для обеспечения вызова всех других компонентов, не выходя из среды, с широким использованием функциональных клавиш;
- **текстовый редактор** для набора и редактирования исходных текстов программ;
- **система поддержки сборки (build)** предназначена для компиляции проектов из исходных кодов. Она включает компилятор с исходного реализуемого языка и компоновщик (linker);

– **отладчик (debugger)** для отладки программ в среде. К нему относится набор команд: установить контрольную точку остановки, остановиться в заданной процедуре, визуализировать значения переменных.

На рисунке 65 представлены базовые компоненты интегрированной среды разработки.



Рис. 65. Базовые компоненты интегрированной среды разработки

Кроме базовых компонентов, в интегрированной среде разработки присутствуют еще и другие компоненты – это система управления версиями, инструменты для конструирования графического интерфейса программы, инспектор объектов.

Очень часто возникают вопросы по выбору интегрированной среды разработки. Можно ли устанавливать несколько сред? Да, можно устанавливать несколько сред разработки. Основные концепции интегрированной среды разработки должны работать в любой среде.

Интегрированная среда разработки Code::Blocks имеет открытый исходный код. Для работы в Windows применяется версия с Minimalist GNU for Windows (MinGW) в комплекте. MinGW – это набор программ и библиотек, скомпилированных для ОС Microsoft Windows.

Новые возможности стандарта C++11 – многопоточность. Это означает, что выполняются несколько процессов на разных процессорах или ядрах. Чтобы настроить версию компилятора C++11 в Code::Blocks, нужно перейти в **Settings – Compiler**. Затем поставить галочку напротив пункта **Have g++ follow the C++11 ISO C++ language standard [-std=c++11]** (рис. 66) и **ОК**.

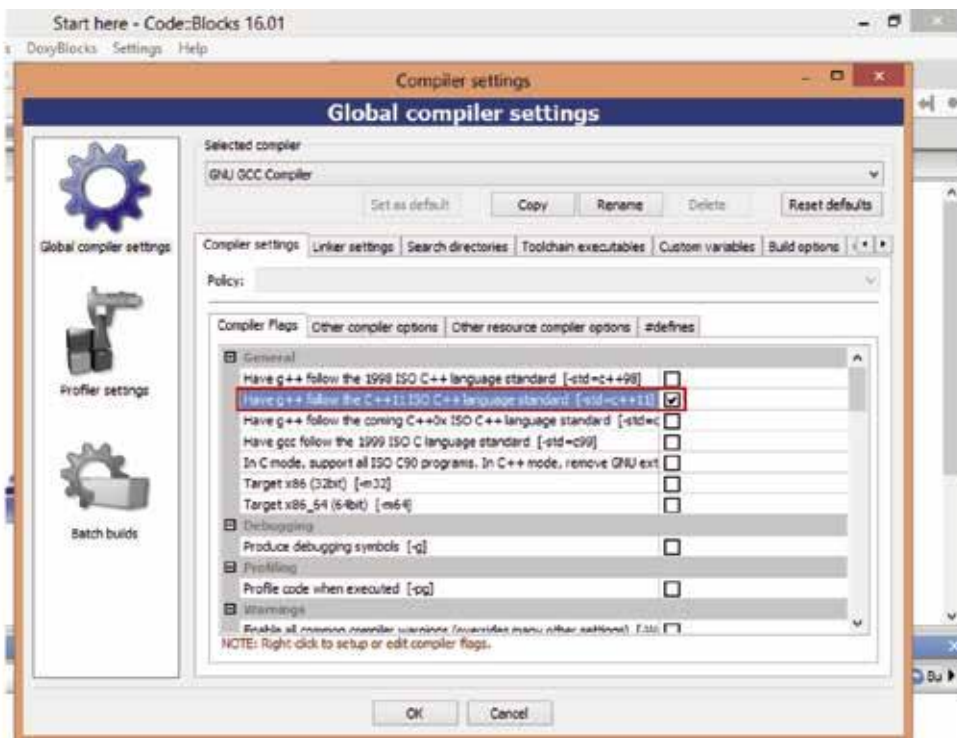


Рис. 66. Настройка версии компилятора

Интегрированная среда разработки Code::Blocks написана на C++ и имеет открытую архитектуру. Она поддерживает языки программирования C, C++ и т.д. Эта среда имеет следующие возможности:

- подсветка синтаксиса;
- автодополнение кода;
- планировщик под несколько пользователей;
- поддержка всех популярных отладчиков и компиляторов.



Знание и понимание



1. Что такое интегрированная среда разработки?
2. Из каких базовых компонентов состоит интегрированная среда разработки?
3. Какие интегрированные среды разработки используются для написания программы C++?

4. Что такое редактор с подсветкой синтаксиса?
5. Для чего нужен компилятор?
6. С какой целью применяется отладчик?
7. Перечислите особенности интегрированной среды разработки для C++.



Применение

Разгадайте кроссворд (рис. 67), применив полученные данные по ИСР.

Слова по горизонтали:

1. Трансляция программы на машинный язык называется ...
5. Аббревиатура «Интегрированная среда разработки» – это ...

Слова по вертикали:

2. Какой компонент интегрированной среды разработки используется для ввода программного кода?
3. Компонент для отладки программ называется ...
4. Среда, поддерживающие работу с разными платформами, называются ...
6. Среда, предназначенные для создания интерфейса программы без написания программного кода, называются ...

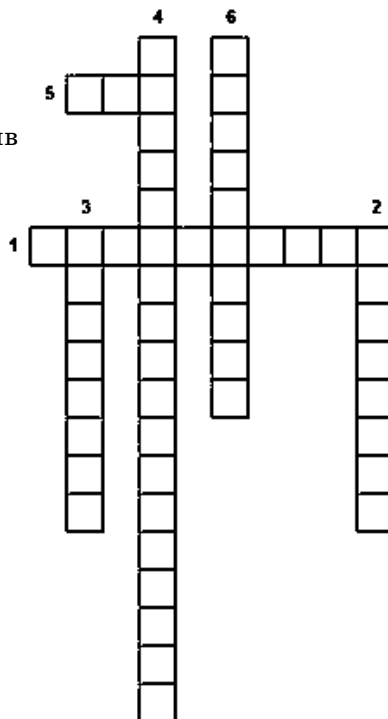


Рис. 67. Интегрированная среда разработки

Анализ

Проведите исследовательскую работу по теме «Интегрированная среда разработки: достоинства и недостатки».

Рекомендации для выполнения домашнего задания

Для поиска информации используйте различные источники информации.

Синтез

Свяжите воедино знания по ИСР из 7 и 8 классов.





Оценивание

Составьте свои рекомендации по использованию компонентов интегрированной среды разработки программ Code::Blocks.



Microsoft Visual Studio

- разработка консольных приложений, приложений с графическим интерфейсом, веб-сайты, веб-приложения и веб-службы;
- для операционных систем: Windows/Linux/Mac OS X



Eclipse CDT

- свободная интегрированная среда разработки модульных кроссплатформенных приложений;
- для операционных систем: Windows/Linux/Mac OS X



NetBeans

- бесплатная интегрированная среда разработки для создания приложений на языках программирования Java, Python, C, C++ и другие



CodeLite

- простой и понятный интерфейс;
- для операционных систем: Windows/7/8/8.1/10 и другие



Code::Blocks

- бесплатная среда разработки;
- для операционных систем: Windows/Linux/Mac OS X

4.3. Оператор выбора

Узнаете

как использовать операторы выбора и циклов
в интегрированной среде разработки программ.

Ключевое слово

Оператор выбора

Таңдау операторы

Selection operator



Вспомните синтаксис ветвления **if**. Приведите примеры ветвления из жизни.



Мы продолжим изучать оператор выбора **if**. Часто возникает ситуация, когда программа должна выбрать, какую операцию ей выполнить в зависимости от определенного условия.



На рисунке 68 представлена схема работы оператора выбора **if**. Нарисуйте схему работы упрощенного варианта условного оператора.

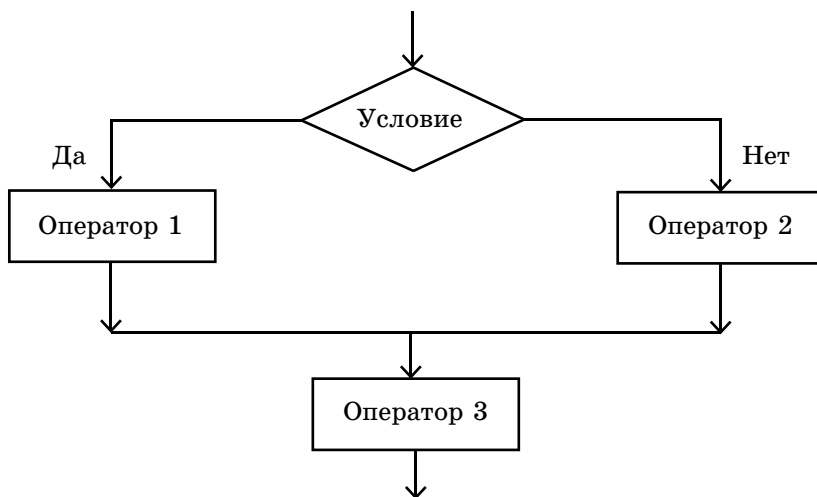


Рис. 68. Оператор выбора **if**



К примеру, введем с клавиатуры целое число. Если оно положительное, то программа должна выполнить одно действие, иначе – другое. Решим задачу с оператором выбора.



В программе объявляются две переменные x и y типа **double**. Значения этих переменных определяются пользователем (считываются с клавиатуры). Если значение переменной y отлично от нуля, вычисляется деление переменных x/y . На экран будет выводиться соответствующее значение. Иначе на экран будет выводиться текстовое сообщение «**Division by zero!**» (**Деление на ноль!**). Для реализации такого алгоритма в программе используется условный оператор.

```
#include <iostream>
using namespace std;
int main(){
    double x,y;
    cout<<"x=";
    cin>>x;
    cout<<"y=";
    cin>>y;
    if(y!=0) cout<<"x/y ="<<x/y<<"\n";
    else cout<<"Division by zero!\n";
    return 0;
}
```

```
1  #include <iostream>
2  using namespace std;
3  int main(){
4  double x,y;
5  cout<<"x=";
6  cin>>x;
7  cout<<"y=";
8  cin>>y;
9  if(y!=0) cout<<"x/y ="<<x/y<<"\n";
10 else cout<<"Division by zero!\n";
11 return 0;
12 }
13
```

В C++ любое ненулевое значение интерпретируется как **true**, а нулевое – как **false**.

Ветка **else** может отсутствовать, это называется **неполным ветвлением**, или **неполной условной инструкцией**.

Рассмотрим пример применения оператора выбора из учебного предмета «Математика».



Решим квадратное уравнение вида:

$$ax^2 + bx + c = 0, \text{ где } a \neq 0.$$



Шаг за шагом

Решение квадратного уравнения приведем через нахождение дискриминанта. Обозначим дискриминант буквой **D**. Дискриминант вычисляется по формуле $D = b^2 - 4 \cdot a \cdot c$.

Существует несколько условий для дискриминанта:

- если $D > 0$, то решение имеет 2 различных вещественных корня;
- если $D = 0$, то оба вещественных корня равны;
- если $D < 0$, то уравнение не имеет корней.

Алгоритм решения довольно прост. Вычисляется дискриминант. Если он больше или равен 0, то корни вычисляются по следующей формуле:

$$x_{1,2} = \frac{-b \pm \sqrt{D}}{2a}.$$

Объединив две предыдущие формулы в одну, получим:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Приведем алгоритм решения квадратного уравнения на языке программирования C++. Для этого применим вышеприведенный алгоритм.

1. **Определим переменные.** Нам понадобятся 4 переменные: a , b , x , c . Корни и коэффициенты всегда вещественного типа, поэтому переменные a , b , x , c будут иметь тип **double**.

2. **Определим интерфейс.** Пользователь будет вводить с консоли значения для переменных a , b , c . На экране будут выводиться корни уравнения (x_1 , x_2) либо сообщение « $D < 0$, нет корней», то есть «дискриминант меньше 0, нет корней».

3. **Определим в программе в функции main() переменные:**

```
double a;  
double b;  
double c;  
double x;
```

4. **Вывод и ввод** будут осуществляться с консоли, поэтому подключаем заголовок **#include <iostream>**.

5. Для работы с математическими функциями подключаем **#include <cmath>** и область **using namespace std;**

```
#include <iostream>  
#include <cmath>  
using namespace std;
```

6. **Ввод** пользователем значения переменных на экране:

```
cout << «Vvod a=: »;  
cin >> a;  
cout << «Vvod b=: »;  
cin >> b;  
cout << «Vvod c=: »;  
cin >> c;
```

7. **Проверка условия.** Если дискриминант больше или равен 0, то вычисляются корни и выводится на экран:

```
if((b*b - 4*a*c) >= 0) // Если дискриминант больше или равен 0
{
    x = (-b + sqrt(b*b - 4*a*c)) / (2 * a);
    cout << «x1 = » << x << endl;
    x = (-b - sqrt(b*b - 4*a*c)) / (2 * a);
    cout << «x2 = » << x << endl; }
```

в противном случае выводится сообщение:

```
else {cout << «D<0, net korney» << endl; }
```

8. **Компиляция программы – Ctrl + F9.**

9. **Запуск и проверка – Ctrl + F10 (или компиляция и выполнение – F9):**

а) ввод данных, чтобы **D** был меньше 0. Введите значения **a** = 2, **b** = 3, **c** = 3. Дискриминант меньше 0, корни не вещественные. Для закрытия данного окна нажмите <Enter>;

б) следующая проверка: ввод данных **a** = 2, **b** = 6, **c** = 4. Первый корень равен -1. Второй корень равен -2.

Для закрытия данного окна нажмите <Enter>.

Программа работает правильно.

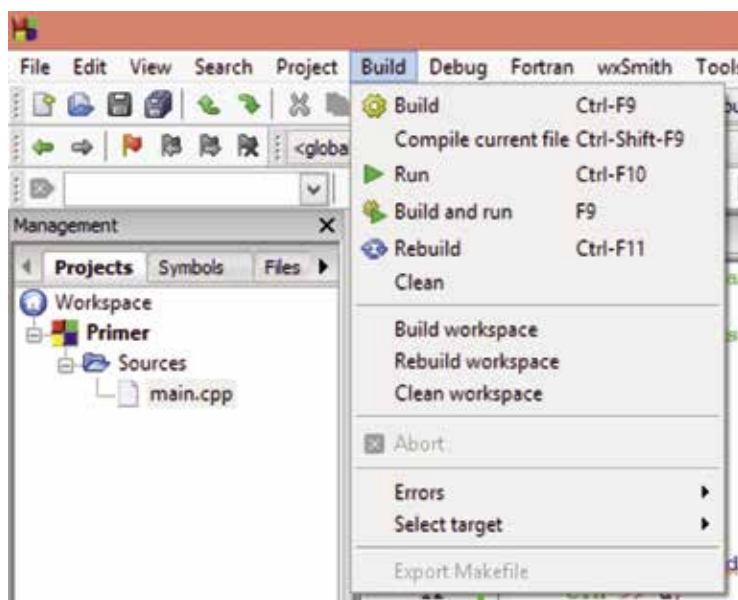


Рис. 69. Компиляция и выполнение

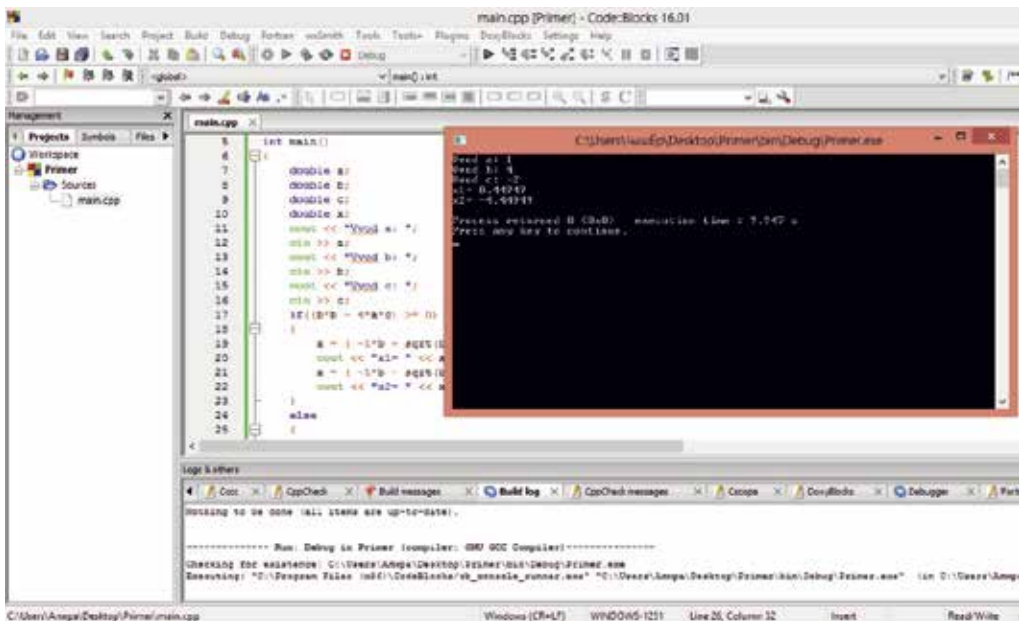


Рис. 70. Решение квадратного уравнения

Ниже представлен весь листинг программы для нахождения корней квадратного уравнения на языке программирования C++:

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    double a;
    double b;
    double c;
    double x;
    cout << «Vvod a: »;
    cin >> a;
    cout << «Vvod b: »;
    cin >> b;
    cout << «Vvod c: »;
    cin >> c;
    if((b*b - 4*a*c) >= 0) //Если дискриминант больше или равен 0
    {
```

```

    x = (-b + sqrt(b*b - 4*a*c)) / (2 * a);
    cout << «x1 = » << x << endl;
    x = (-b - sqrt(b*b - 4*a*c)) / (2 * a);
    cout << «x2 = » << x << endl;
}
else
{
    cout << «D<0, net korney» << endl;
}
return 0;
}

```

Для выбора одного из нескольких вариантов в зависимости от значения некоторой переменной можно использовать оператор **switch**.



Синтаксис **switch** выглядит следующим образом:

```

switch (i)
{
case 0:
case 1:
    ... // последовательность операторов
    break;
case 2:
    ... // последовательность операторов
    break;
default:
    ...
}

```

Здесь **i** – выражение-селектор, которое должно быть целым типом. Каждая ветвь исполнения начинается с ключевого слова **case**, за ним следует значение выражения, при котором должна выполняться данная ветвь.

Для выхода из оператора выбора **switch** используется специальная команда **break**. Если пропустить **break**, то компилятор не выдаст ошибки. Программа будет выполняться неверно.

Ветвь **default** выполняется тогда, когда среди других ветвей не нашлось подходящей.

Приведем пример организации выбора из нескольких альтернатив (рис. 71).

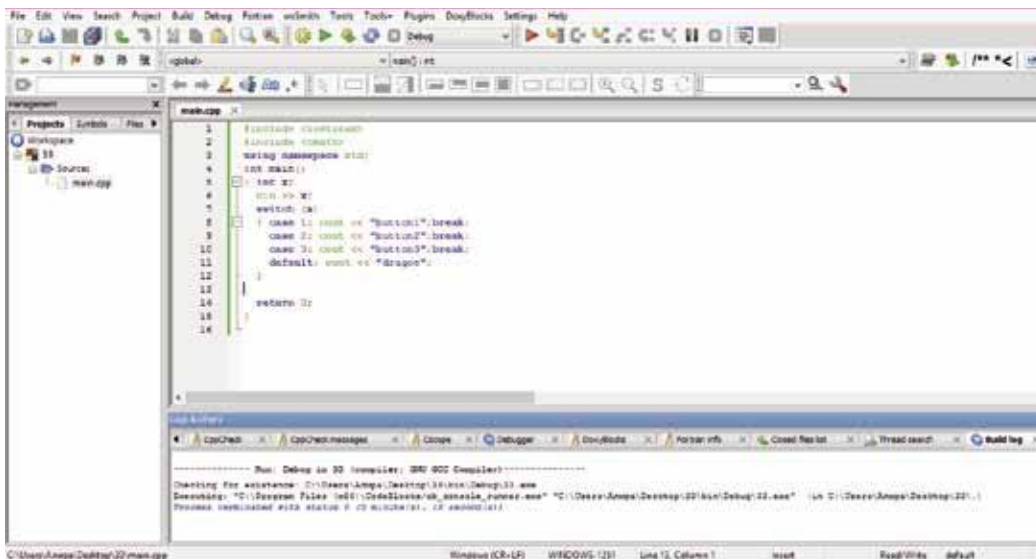


Рис. 71. Выполнение оператора выбора switch

```

#include <iostream>
#include <cmath>
using namespace std;
int main()
{ int x;
  cin >> x;
  switch (x)
  { case 1: cout <<«button1»;break;
    case 2: cout <<«button2»;break;
    case 3: cout <<«button3»;break;
    default: cout <<«drugoe»;
  }
  return 0;
}

```

В этой программе относительно переменной x предполагается, что она может принимать несколько значений. Если эта переменная примет значение 1, 2 или 3, то будет выведено соответствующее сообщение. Если же значение переменной x будет какое-то иное, то будет выведено сообщение «drugoe».

Оператор **break** необходим для прерывания последовательности выполнения операторов в операторе **switch**. Без него управление передается на соответствующий **case**. Затем выполняются все операторы, следующие ниже.



Знание и понимание



1. В каком случае применяется оператор выбора **if**?
2. Когда предпочтительнее применять оператор выбора **switch**?
3. Объясните, как работает структура оператора выбора **switch**.
4. Приведите примеры из повседневной жизни, когда необходимо применить оператор выбора.
5. Почему вложенный оператор выбора **if** лучше заменить **switch**?



Применение



Дан код программы (рис. 72):

```
main.cpp x main.cpp x
1      #include <iostream>
2      using namespace std;
3      int main()
4      {
5          setlocale(LC_ALL, "rus");
6          int variable1 = 0;
7          int variable2 = 0;
8          cout << "Введите первое число: ";
9          cin >> variable1;
10         cout << "Введите второе число: ";
11         cin >> variable2;
12         if (variable1 > variable2)
13             cout << "\n Результат сравнения: Первое число больше чем второе \n";
14         else if (variable1 < variable2)
15             cout << "\n Результат сравнения: Второе число больше чем первое\n";else
16             cout << "\n Результат сравнения: Числа равны\n ";
17         return 0;
18     }
19 }
```

Рис. 72. Код программы

Напишите к каждой строчке комментарий. Какой оператор выбора используется в программе?



Анализ

Синтез



1. Заполните схему работы выбора **switch** на рисунке 73.

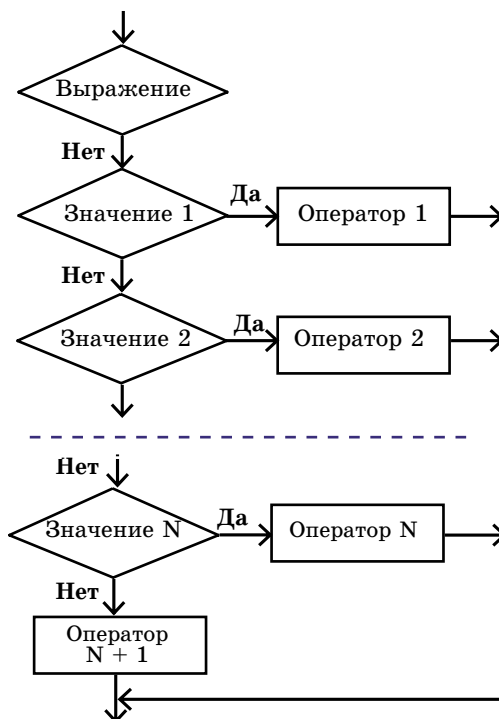


Рис. 73. Схема работы оператора выбора **switch**

2. Проанализируйте представленную программу с оператором выбора **if** и создайте программу с помощью оператора множественного выбора **switch**.

```

#include <iostream>
using namespace std;
int main()
{
    setlocale(0, «Rus»);
    int den;
    cout << «Введите цифрами день недели: »;
    cin >> den;
    if (den >= 1 && den <= 7) {
        if (den == 6 || den == 7)
            cout << «Выходной день» << endl;
        else
            cout << «День учебы» << endl;
    }
    else
        cout << «Ошибка. Введено неправильное значение» << endl;
}
  
```

Рекомендации для выполнения домашнего задания

Данную задачу решите аналогично программе на рисунке 71.



Оценивание

Критически оцените программы, представленные на рисунках 74 и 75. Какие ошибки допущены в них? Обоснуйте ответ.

```
main.cpp x main.cpp x main.cpp x main.cpp x
1      #include <iostream>
2      using namespace std;
3      int main()
4      { int mark;
5      cout<<"Enter mark - ";
6      cin>>mark;
7      if (mark==5)
8      cout<<"Excellent\n";
9      else
10     if (mark==4) cout<<"Good\n";
11     else
12     if (mark==3) cout<<"Satisfy\n";
13     else cout<<"Bad\n";
14     return 0; }
```

Рис. 74. Программа с использованием вложенных конструкций if-else

```
1      #include <iostream>
2      using namespace std;
3      int main() {
4      int mark;
5      cout<<"Enter mark - ";
6      cin>>mark;
7      switch (mark) {
8      case 5: cout<<"Excellent\n"; break;
9      case 4: cout<<"Good\n"; break;
10     case 3: cout<<"Satisfy\n"; break;
11     default: cout<<"Bad\n";
12     }
13     return 0;}
14 }
```

Рис. 75. Программа с использованием оператора switch



Каждый день мы сталкиваемся с плодами технического прогресса. Кто-то, не задумываясь об этом, просто нажимает на кнопки и клавиши. Другие, осознанно действуя теми же кнопками и клавишами, незаметно «правят» нашим миром, создавая правила игры и инструкции для всех остальных. Некоторые – это программисты, интеллектуальная элита общества.

Программисты на самом деле придумывают наше завтра, определяют картину будущего мира и раздвигают границы мироздания. Сегодня без них сложно обойтись в любой сфере человеческой деятельности. А в будущем без них невозможно будет жить. Если говорить о самых перспективных и востребованных профессиях, то программист – это очень перспективная профессия! А если говорить о самых популярных языках программирования, то C++ – это один из пяти самых используемых языков информационно-коммуникационного мира.

C++ является мощным инструментом для создания различных приложений, от небольших до достаточно объемных и трудоемких. Главным преимуществом этого языка является сочетание в нем возможности работы на низком и высоком уровнях. Таким образом, код, создаваемый программистом, может быть максимально адаптирован к системным возможностям той платформы, на которой он будет работать – компьютера или мобильного телефона, игровой приставки.

Среди приложений, написанных на C++ или содержащих код на C++, можно увидеть проекты таких крупных компаний, как Apple, Facebook, Google (поисковая система), Microsoft, MySQL Server и т.д.

Как видно из перечня крупнейших компаний, языку C++ доверяют самое сложное, самое ответственное.

4.4. Цикл с параметром

Узнаете

**как использовать операторы выбора и циклов
в интегрированной среде разработки программ.**

Ключевое слово

*Цикл
с параметром*

Параметри́цикл

*Loop with
parameter*



В повседневной жизни мы часто сталкиваемся с тем, что одно и то же действие нужно повторять несколько раз. А задумывались ли вы над тем, как все эти многократные действия заменить одним?



Операторы цикла многократно выполняют серии однотипных действий. Действия выполняются до тех пор, пока не будет выполнено некоторое условие.

Если мы знаем точное количество действий (итераций) цикла, то можем использовать цикл **for**. **Итерацией цикла** называется один проход этого цикла.

Общий синтаксис вызова оператора **for** следующий:



```
for (счетчик = значение; счетчик < значение; шаг цикла) {  
    тело цикла;  
}
```

Схема выполнения оператора цикла представлена на рисунке 76.

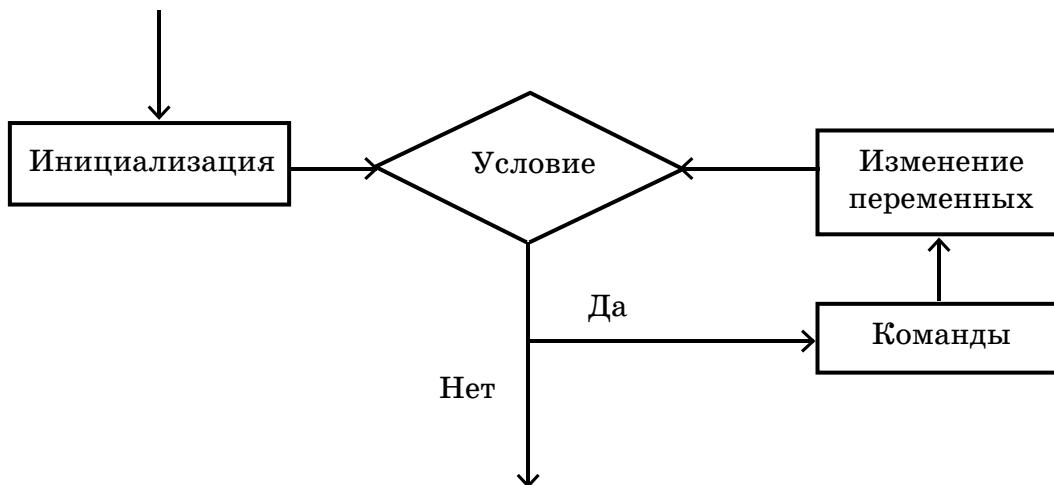


Рис. 76. Схема работы оператора цикла **for**

Опишем синтаксис **for**:

1. Сначала присваивается первоначальное значение счетчику, после чего ставится точка с запятой. **Счетчик цикла** – это управляющая переменная, в которой хранится количество проходов данного цикла.

2. Затем задается конечное значение счетчика цикла. После того как значение счетчика достигнет указанного предела, цикл завершится. Снова ставим точку с запятой.

3. Задаем шаг цикла. **Шаг цикла** – это значение, на которое будет увеличиваться или уменьшаться счетчик цикла при каждом проходе.



Рассмотрим пример по применению цикла с параметром **for**. Напишем программу, которая будет считать сумму всех чисел от 1 до 1000.



Шаг за шагом

```
#include <iostream>
using namespace std;
int main()
{
    int i; // счетчик цикла
    int sum = 0; // сумма чисел от 1 до 1000.
    setlocale(0, «»);
    for (i = 1; i <= 1000; i++) // задаем начальное значение 1, конечное
1000 и задаем шаг цикла – 1.
    {
        sum = sum + i;
    }
    cout << «Сумма чисел от 1 до 1000 = » << sum << endl;
    return 0;
}
```

Как видно на рисунке 77, скомпилированный и выполненный код программы показал ответ: 500500. Это и есть сумма всех целых чисел от 1 до 1000. Если считать это вручную, понадобится очень много времени и сил. Цикл выполнил всю работу моментально.

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int i; // счетчик цикла
6      int sum = 0; // сумма чисел от 1 до 1000.
7      setlocale(0, "");
8      for (i = 1; i <= 1000; i++) // задаем начальное значение 1,
9      {
10         sum = sum + i;
11     }
12     cout << "Сумма чисел от 1 до 1000 = " << sum << endl;
13     return 0;
14 }
15
```

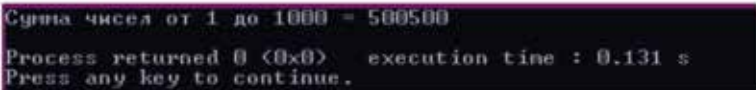


Рис. 77. Пример цикла с параметром

Будьте очень внимательны при задании конечного значения счетчика. Применяйте нестрогое неравенство ($\leq$ – меньше либо равно). Если поставить знак меньше, то цикл произведет 999 итераций, т.е. на одну меньше, чем требуется. Значение шага цикла равно единице. Выражение $i++$ равносильно выражению $i = i + 1$.

В теле цикла при каждом проходе программа увеличивает значение переменной **sum** на **i**. В начале программы переменной **sum** обязательно нужно присвоить значение **ноль**.

Управляющая переменная может как увеличиваться, так и уменьшаться. Это зависит от того, какая задача поставлена перед программистом, например, можно записать: **for (int i = 100; i > 0; i--)**.

Управляющая переменная всегда должна иметь целочисленный тип данных. Если переменная была объявлена в цикле, по завершении цикла эта переменная будет уничтожена.



Знание и понимание

1. Расскажите о синтаксисе цикла с параметром в языке C++.
2. Что такое итерация?
3. Опишите работу структур с параметром цикла **for**.
4. Приведите примеры из жизни с параметром цикла.
5. На что необходимо обратить внимание при задании конечного значения цикла **for**?

6. Какому выражению равносильно выражение $i = i + 1$?
7. Можно ли цикл `for(int i = 0; i < 100; i++)` записать следующим образом?

```
int i=0;
for( ; ; )
{
    // ...
    if(i>=100)
        break;
    i++;
}
```



Применение

Введите программу в интегрированную среду разработки Code::Blocs.



```
#include <iostream>
using namespace std;
int main()
{
    for (int i = 1; i <= 20; i++)
    {
        cout << «*» << endl;
    }
    return 0;
}
```

1. Каковы начальное значение, приращение и конечное значение параметра цикла `for`?
2. Что будет выведено на экран?
3. Упростился ли синтаксис цикла с параметром?



Анализ



На рисунке 78 приведен листинг программы. Проанализируйте код программы. Как работает цикл с параметром `for`? Можно ли его назвать вложенным? Как меняется рисунок на экране в зависимости от вводимого значения?

На основании данной программы попробуйте предложить собственный вариант программы.

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      setlocale(LC_ALL, "rus");
6      int height = 0; // высота треугольника
7      cout << "Введите высоту равнобедренного треугольника: ";
8      cin >> height;
9
10     for (int i = 0; i < height; i++)
11     {
12         for (int j = 1; j < height - i; j++)
13         {
14             cout << " ";
15         }
16         for (int j = height - 2 * i; j <= height; j++)
17         {
18             cout << "+ ";
19         }
20         cout << endl;
21     }
22     return 0;
23 }
```

Рис. 78. Листинг программы

Рекомендации для выполнения домашнего задания

Введите данную программу в интегрированную среду разработки Code::Blocs. Откомпилируйте и выполните программный код на C++. По результатам выполнения программы сделайте анализ программного кода.



Синтез

Приведены несколько примеров заголовков цикла. Запишите, как будет меняться управляющая переменная в приведенных примерах. Будет ли иметь смысл выражение **(int i=1; i<=10; i++)** без **for**?

Пример:

Изменение управляющей переменной от 1 до 100 с шагом 1: **for(int i = 1; i <= 100; i++)**.

1. _____
for(int i = 100; i > 0; i--)
2. _____
for(int i = 7; i <= 77; i += 7)
3. _____
for(int i = 20; i >= 2; i -= 2)
4. _____
for(int j = 2; j <= 20; j += 3)
5. _____
for(int j = 99; j >= 0; j -= 11)

	Оценивание 1. Составьте программу для вывода всех четных чисел от 0 до N, вводимых пользователем. 2. Решите программу для вывода всех чисел, которые делятся на 4 или на 9, от 0 до N, вводимых пользователем.
	1. Язык программирования C++ является расширением языка программирования C для поддержки объектно-ориентированной парадигмы. В C++ может реализовываться как процедурный подход, так и объектно-ориентированный. 2. Программа C++ содержит один метод main(), выполнение которого отождествляется с выполнением программы. 3. Все операторы, используемые в C++, можно разбить на четыре группы: арифметические, логические, операторы сравнения и побитовые операторы. 4. При вычислении выражений в C++ используется автоматическое приведение типов. Можно также явно описывать типы. 5. В C++ используется ряд управляющих инструкций, посредством которых в программах реализуются ветвления и выполняются циклические повторения однотипных команд.

4.5. Цикл с постусловием

Узнаете

как использовать операторы выбора и циклов
в интегрированной среде разработки программ.

Ключевое слово

*Цикл
с постусловием*

*Keiingi шартты
цикл*

*Loop with
postcondition*

Как вам известно, циклы используются для многократного повторения некоторого участка программного кода. Вы познакомились с часто используемым циклом **for**. Каждый из этих циклов имеет свое применение. Поэтому вы должны знать, когда применяется тот или иной тип цикла.

Когда мы не знаем, сколько итераций должен произвести цикл, применяются циклы **do-while** (постусловие) или **while** (предусловие).

Рассмотрим подробнее тип цикла **do-while**. Особенность цикла с пост-условием в том, что тело цикла обязательно должно выполниться хотя бы один раз. И это независимо от условия, которое проверяется в конце итерации.

Синтаксис цикла **do-while**:

```
do
{
    оператор 1;
    оператор 2;
    ...
    оператор n;
}
while (условие выполнения цикла);
```

} Тело цикла

Схема действия оператора цикла **do-while** проиллюстрирована на рисунке 79.

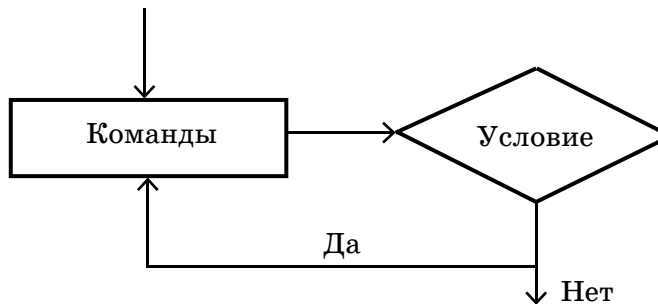


Рис. 79. Схема действия оператора цикла **do-while**



Рассмотрим пример с применением цикла **do-while**. Напишем программу, которая будет считать сумму всех целых чисел от 1 до 1000 (рис. 80).



Шаг за шагом

```
#include <iostream>
using namespace std;
int main()
{
    setlocale (LC_ALL, «RUS»);
    int i = 0; // инициализируем счетчик цикла.
    int sum = 0; // инициализируем счетчик суммы.
```

```

do { // выполняем цикл.
    i++;
    sum += i;
} while (i < 1000); // пока выполняется условие.
cout << «Сумма чисел от 1 до 1000 =» << sum << endl;
return 0;
}

```

```

1
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      setlocale(LC_ALL, "RUS");
7      int i = 0; // инициализируем счетчик цикла.
8      int sum = 0; // инициализируем счетчик суммы.
9      do { // выполняем цикл.
10         i++;
11         sum += i;
12     } while (i < 1000); // пока выполняется условие.
13     cout << "Сумма чисел от 1 до 1000 = " << sum << endl;
14     return 0;
15 }
16
17

```

Сумма чисел от 1 до 1000 = 500500

Process returned 0 (0x0) execution time : 0.646 s

Press any key to continue.

Рис. 80. Сумма чисел с циклом **do-while**

В результате вычисления суммы чисел от 1 до 1000 с помощью цикла **do-while** нет отличия от цикла **for**. Но если переменной *i* присвоить значение большее, чем 1000, то цикл все равно выполнит хотя бы один проход.



Поэкспериментируйте с умножением чисел. Проанализируйте каждую строчку выполненного программного кода на рисунке 81. Что означает выражение **n!=25**?



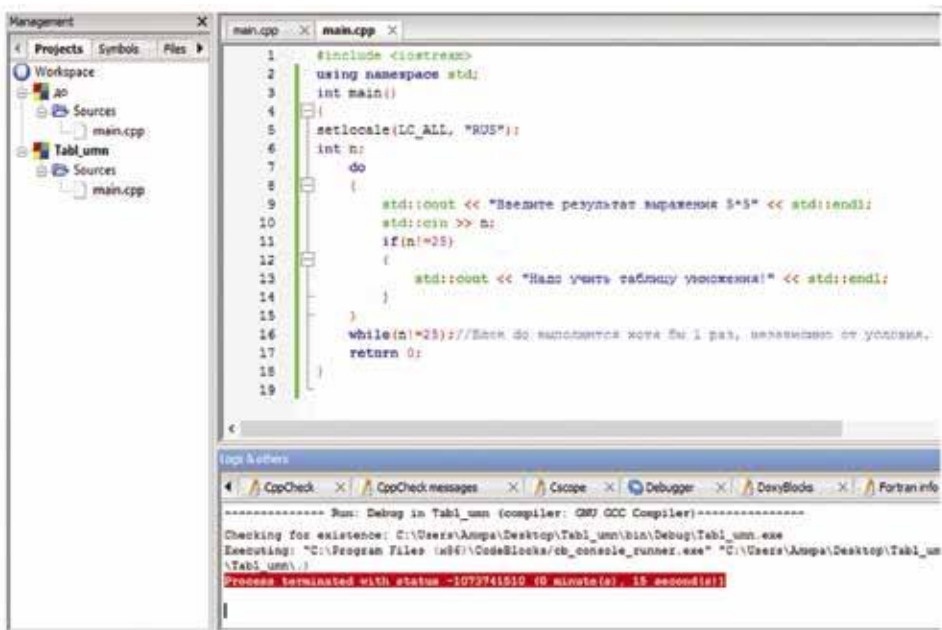


Рис. 81. Программный код «Таблица умножения»

В точке останова отладчика можно посмотреть и изменить значения переменных. Можно также изменить исходный код, перекомпилировать, а затем перезапустить программу. Можно даже изменить исходный код посреди пошагового выполнения программы.



Знание и понимание



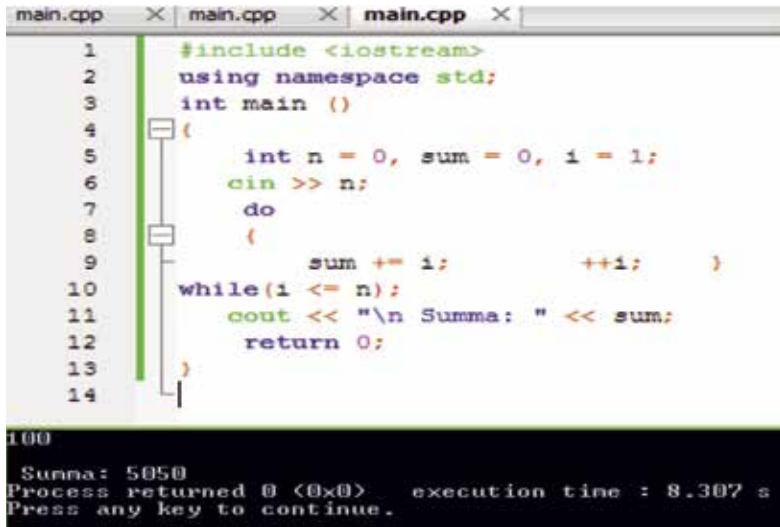
1. Перечислите три типа циклов.
2. Какой цикл используется чаще других?
3. С какой целью используется в программе цикл **do-while**?
4. Объясните синтаксис цикла с постусловием.
5. Где происходит проверка условия цикла с постусловием?
6. Сколько раз выполняется цикл с постусловием?
7. Чем отличается цикл **for** от цикла **do-while**?



Применение



1. На рисунке 82 дана программа, вычисляющая сумму чисел от 1 до n . Значение n вводится с клавиатуры.



```
1  #include <iostream>
2  using namespace std;
3  int main ()
4  {
5      int n = 0, sum = 0, i = 1;
6      cin >> n;
7      do
8      {
9          sum += i;      ++i;      }
10     while(i <= n);
11     cout << "\n Summa: " << sum;
12     return 0;
13 }
14
```

100
Summa: 5050
Process returned 0 (0x0) execution time : 8.307 s
Press any key to continue.

Рис. 82. Вычисление суммы чисел от 1 до n

2. Ответьте на вопросы:

- 1) Почему переменная `sum = 0`?
- 2) Что произойдет, если выражение в `while` ($i \leq n$) поменять на «`==` - равно», или «`>=` - больше или равно»?
- 3) В какой переменной будет находиться сумма?



Анализ

1. Проанализируйте программу. Что будет выведено на экран после ее выполнения?

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    a = 0;
    do {
        // «Privet!» будет напечатано хотя бы 1 раз, даже, если a = 0
        cout<<«Privet!\n»;
    } while ( a != 0 );
    cin.get();
}
```

2. Попробуйте самостоятельно составить любую задачу с применением цикла с постусловием.



Синтез

Любое целое четырехзначное число можно представить в виде: (a, b, c, d) – цифры числа, причем $a = 1000$).

Например: $1742 = 1 * 1000 + 7 * 100 + 4 * 10 + 2$.

Как можно записать на C++ то, что цифры числа не должны совпадать, в виде условия: _____?

Условие на разность чисел, составленных из цифр числа:

$a * 10 + b - (c * 10 + d) = a + b + c + d$.

Как можно с помощью цикла **for** записать выполняемую часть программ?



Оценивание

Решите задачу «Угадай число», используя цикл **do-while** в C++.



Настройка редактора кода

Конфигурация редактора кода отвечает за аспекты набора текста: шрифт, подсветку синтаксиса, проверку орфографии в комментариях и многое другое. Эти параметры каждый может настраивать как ему удобно. Важно правильно настроить кодировку по умолчанию. Это зависит от версии Windows и системного языка. Но рекомендуется установить ее в UTF-8 из выпадающего списка во вкладке **Encoding settings** раздела **General Settings** пункта меню **Settings – Editor** (рис. 83).

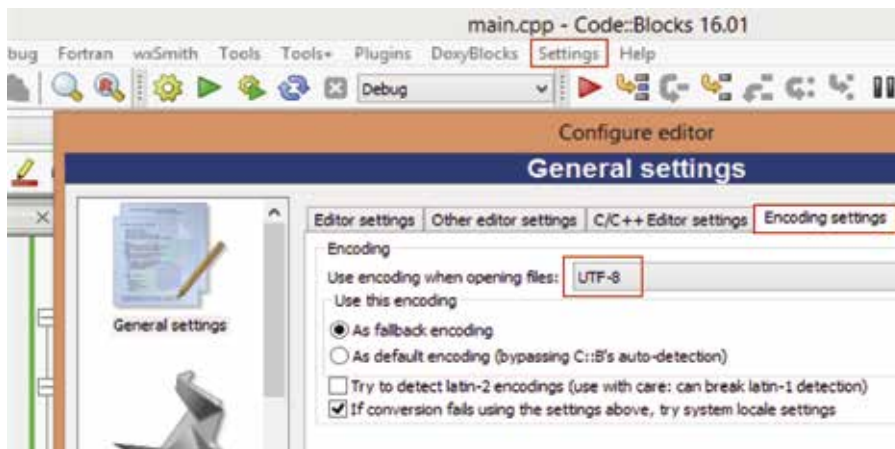


Рис. 83. Настройка Редактора кода

4.6. Цикл с предусловием

Узнаете

как использовать операторы выбора и циклов
в интегрированной среде разработки программ.

Ключевое слово

*Цикл
с предусловием*

*Алгы шартты
цикл*

*Loop with
a precondition*

Помимо операторов цикла **for** и **do-while**, широко используется цикл **while**.

Синтаксис цикла **while** в C++ выглядит следующим образом:

```
while (условие выполнения цикла)
{
    оператор 1;
    оператор 2;
    ...
    оператор n;
}
```

} Тело цикла

Данный цикл будет выполняться, пока условие, указанное в круглых скобках, является **истиной**. Схема действия оператора цикла **while** проиллюстрирована на рисунке 84. В цикле с предусловием тело цикла может ни разу не выполниться, если условие начала цикла не выполняется.

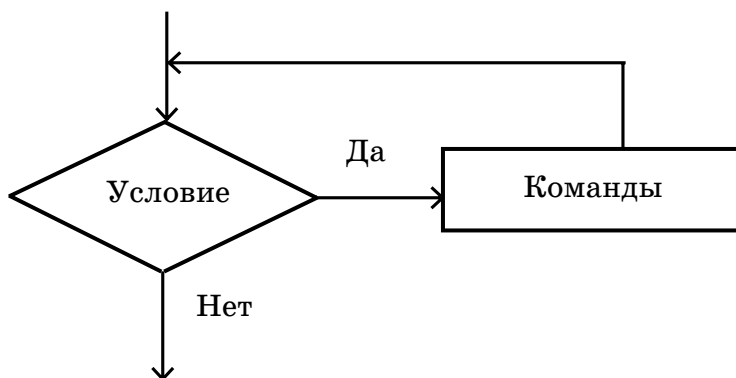


Рис. 84. Схема выполнения оператора цикла **while**



Напишем программу, которая будет считать сумму всех целых чисел от 1 до 1000 с помощью цикла **while**.



Ниже приведен исходный код программы, считающей сумму всех целых чисел от 1 до 1000.

```
#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL, «RUS»);
    int i = 0; // инициализируем счетчик цикла.
    int sum = 0; // инициализируем счетчик суммы.
    while (i < 1000)
    {
        i++;
        sum += i;
    }
    cout << «Сумма чисел от 1 до 1000 = » << sum << endl;
    return 0;
}
```

После компиляции программа выдаст результат, аналогичный результату работы предыдущей программы (рис. 85).

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      setlocale(LC_ALL, "RUS");
6      int i = 0; // инициализируем счетчик цикла.
7      int sum = 0; // инициализируем счетчик суммы.
8      while (i < 1000)
9      {
10         i++;
11         sum += i;
12     }
13     cout << "Сумма чисел от 1 до 1000 = " << sum << endl;
14     return 0;
15 }
16
```

Сумма чисел от 1 до 1000 = 500500
Process returned 0 (0x0) execution time : 0.395 s
Press any key to continue.

Рис. 85. Программа с циклом **while**

Почему в условии цикла **while** задано **строгое неравенство** и счетчик цикла **i** инициализирован нулем? Это связано с тем, что в цикле **while** выполняется на одну итерацию больше. Цикл будет выполняться до тех пор, пока значение счетчика перестанет удовлетворять условию выполнения.

Допустим, если поставить **нестрогое неравенство**, то цикл закончится, когда переменная **i** станет на одну итерацию больше и будет равна 1001.

Инициализация счетчика цикла и переменной, хранящей сумму чисел, происходит в самом начале программы. Затем описываем условие цикла **while (i < 1000)**. Это означает, что «пока переменная **i** меньше 1000, выполняй цикл». При каждой итерации цикла значение переменной-счетчика **i** увеличивается на единицу внутри цикла.

Когда выполнится 1000 итераций цикла, счетчик станет равным 999 и следующая итерация уже не выполнится, поскольку 1000 не меньше 1000.

Для суммирования чисел применяется выражение **sum += i**. Это выражение является укороченной записью **sum = sum + i**. После окончания выполнения цикла выводим сообщение с ответом.



Попрактикуйтесь, поэкспериментируйте с умножением чисел. Проанализируйте каждую строчку выполненного программного кода на рисунке 86. Что будет выводиться на экран, если введенный результат будет верный? Иначе?



```
main.cpp x main.cpp x main.cpp x *main.cpp x
1  #include <iostream>
2  using namespace std;
3  int main ()
4  {
5      setlocale(LC_ALL, "RUS");
6      int n;
7      std::cout << "Введите результат выражения 8*8" << std::endl;
8      std::cin >> n;
9      while(n!=64)//Цикл будет выполняться до тех пор, пока выражение в скобках истинно.
10     {
11         std::cout << "Учите таблицу умножения!" << std::endl;
12         std::cout << "Попробуйте снова!" << std::endl;
13         std::cin >> n;
14     }
15     std::cout << "Вы справились!"<< std::endl;
16     return 0;
17 }
18
```

Рис. 86. Программа умножения с циклом **while**



1. Операторы, заключенные в фигурные скобки {}, выполняются до тех пор, пока не выполнится условие завершения цикла.
2. В программе обязательно должно быть условие выхода из цикла, чтобы не было заикливания.
3. В языке программирования C++ возможно увеличение значения переменной на единицу с помощью оператора инкремента (++).

Таким образом, все три цикла – **for**, **while** и **do-while** – похожи. Только в цикле **for** все необходимое для его нормальной работы собрано в одной строке. Это объявление и инициализация управляющей переменной, проверочное условие, изменение управляющей переменной.

Принципиальная разница между операторами цикла **while** и **do-while** состоит в том, что в первом случае сначала проверяется условие. Если условие верно, то выполняются команды. Во втором случае сначала хотя бы один раз выполняются команды, а затем проверяется условие.

При этом программа должна быть составлена корректно, чтобы не получить бесконечный цикл.

В циклах **while** и **do-while** управляющая переменная объявлена до цикла, а ее изменение происходит ниже в теле.

Какую конструкцию повторения использовать, будет зависеть от вас. Чаще циклы **while** и **do-while** применяются тогда, когда неизвестно изначально, сколько будет повторений. Стоит еще помнить, что в отличие от циклов **while** и **do-while**, управляющая переменная цикла **for** не видна за пределами его тела (то есть при выходе из цикла она уничтожается).



Знание и понимание

1. Почему цикл с предусловием назван именно так?
2. Объясните синтаксис цикла с предусловием.
3. Каким должно быть условие для выхода из цикла с предусловием?
4. Какое должно быть условие, чтобы цикл с предусловием не заикливался?
5. В каком случае цикл с предусловием не выполнится ни разу?



Применение



Напишите программу с помощью **while**, которая будет выводить на экран 10 звездочек (*).



Анализ



На рисунке 87 приведена программа. Напишите комментарии к ней. Какова постановка задачи (условие задачи)?

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      setlocale(LC_ALL, "rus");
6      double payment = 0;
7      int exit = 1; // управляющая переменная
8      while (exit == 1)
9      {
10         cout << "Введите сумму для пополнения счета: ";
11         cin >> payment;
12         cout << "Сплатить и выйти - нажмите 0. Изменить сумму - нажмите 1: ";
13         cin >> exit; // изменения управляющей переменной
14     }
15     cout << "\n\n-----\n\n";
16     cout << "Ваш счет пополнен на " << payment << "$ Спасибо! До свидания!";
17     cout << "\n\n-----\n\n";
18     return 0;
19 }
20
```

Рис. 87. Что выполняет программа?

Рекомендации для выполнения домашнего задания

Проанализировав программный код, можно решить обратную задачу, то есть написать условие задачи.



Синтез



Как организовать непрерывный ввод чисел с клавиатуры, пока пользователь не введет 0? Как после ввода нуля показать на экране количество чисел, которые были введены, их общую сумму и среднее арифметическое?

Подсказка: необходимо объявить переменную-счетчик, которая будет считать количество введенных чисел, и переменную, которая будет накапливать общую сумму чисел.



Оценивание

Решите, сколько слагаемых должно быть в сумме $2 + 4 + 6 + 8 + \dots$, чтобы эта сумма оказалась больше некоторого данного натурального числа N .



Если пропала панель сообщений об ошибках, то вывести ее можно, выбрав вкладку **View – Perspectives – GDB/CDB debugger:Default** (рис. 88).

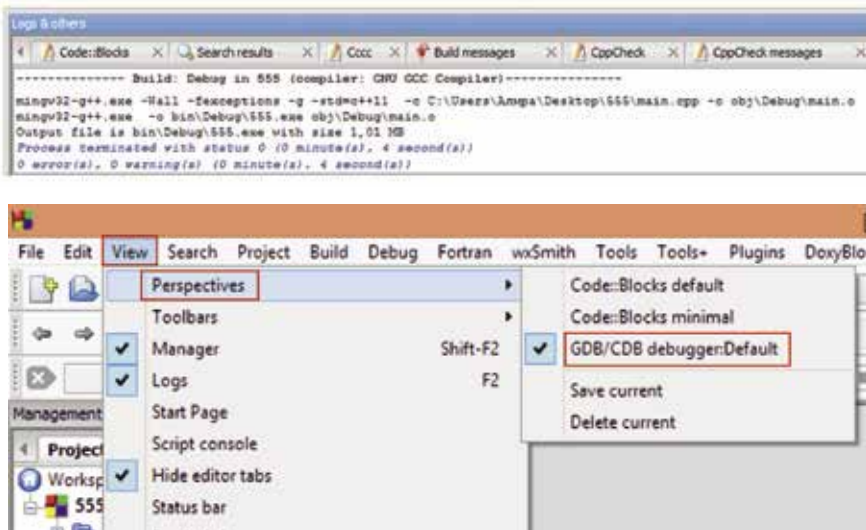


Рис. 88. Установка панели сообщений об ошибках

4.7. Трассировка алгоритма

Научитесь

осуществлять трассировку алгоритма.

Ключевое слово

*Трассировка
алгоритма*

*Алгоритм
трассировкасы*

*Algorithm
trace*



Отладка – это процесс поиска и исправления ошибок в программе, препятствующих корректной работе программы.

Отладчик позволяет выполнять трассировку программы, наблюдать значения переменных, контролировать выводимые программой данные.

Во время работы программы ее инструкции выполняются одна за другой со скоростью работы процессора компьютера. В случае неправильной работы программы необходимо видеть реальный порядок выполнения инструкций. Это можно сделать, выполнив **трассировку** программы.



Что такое трассировка? Можно ли выполнить трассировку вручную?



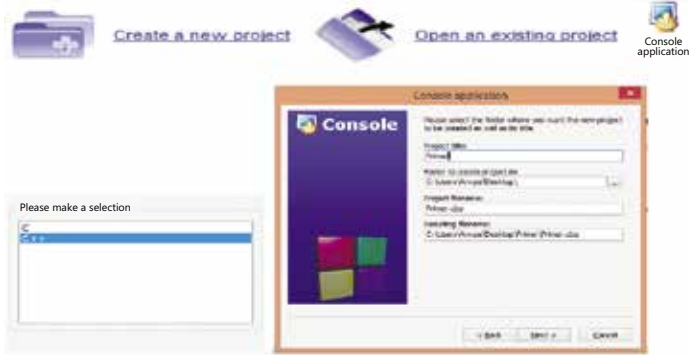
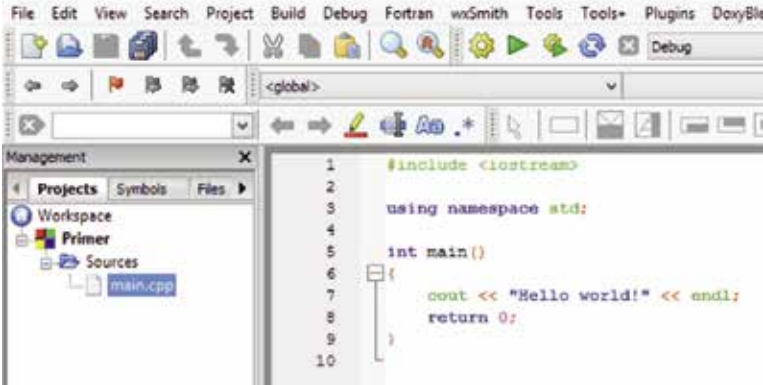
Трассировка – это процесс выполнения программы по шагам (step-by-step), инструкция за инструкцией. Во время трассировки программист дает команду выполнить очередную инструкцию программы.

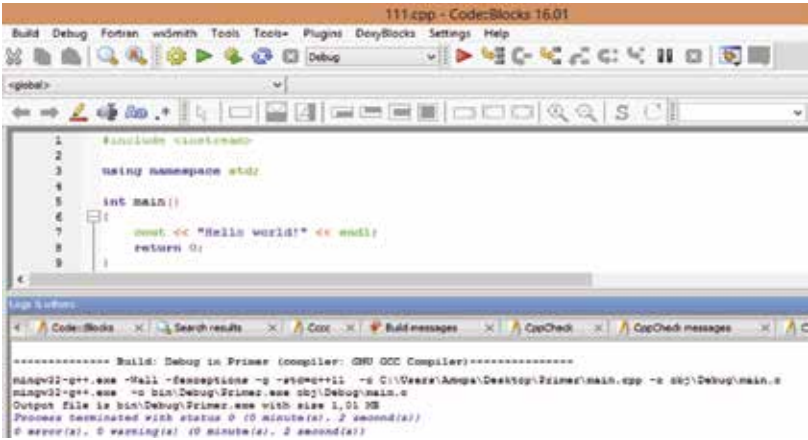
Трассировка применяется для проверки правильной работы программы. Она по шагам показывает изменение всех переменных после каждого выполнения оператора.

Как проводится трассировка в интегрированной среде разработки Code::Blocks? Для этого рассмотрим основные этапы выполнения программы в таблице 17.

Таблица 17

Этап выполнения	Порядок выполнения
1. Загрузка ИСП Code::Blocks	Одним из способов: 1. С помощью ярлыка на рабочем столе. 2. С помощью системного меню. 3. Выполнением загрузочного файла C:\Program Files(x86)\CodeBlocks\codeblocks.exe.

2. Создание или открытие файлов проекта	 Рис. 89. Последовательность создания проекта
3. Ввод и редактирование программы (текстового файла)	Загрузка файла программы в окно редактора: дважды щелкните по main.cpp, откроется заготовка программы «Hello world!», где можно вводить или редактировать новый проект (рис. 90).  Рис. 90. Окно редактора Можно выполнить команду меню Файл/Открыть файл или проект, или нажать комбинацию клавиш Ctrl + O. Откроется диалоговое окно Открыть файл. Выберите файл и нажмите кнопку Открыть
4. Сохранение файла активно-го окна редактора	Выполните команду меню Файл/Сохранить как/. Откроется диалоговое окно Сохранить как. Можно задать новое имя файла или оставить существующее. Нажмите кнопку Сохранить

5. Сохранение изменений в файле	Выполните команду меню Файл/Сохранить или нажмите комбинацию клавиш Ctrl + S. Команда доступна после внесения изменений в файл
6. Запуск программы на компиляцию	Выберите один из способов: 1. Выполните команду меню Скомпилировать или нажмите комбинацию клавиш Ctrl + F9. После успешной компиляции запускается компоновщик для получения .exe файла.  Рис. 91. Окно компиляции Появится окно Compiler с протоколом компиляции. 2. Команда Выполнить или комбинация клавиш Ctrl + F10. Запускается .exe – файл на выполнение. 3. Или все это можно заменить командой меню Скомпилировать и Выполнить клавишей F9. После успешной компиляции запускается компоновщик. После успешной компоновки запускается полученный .exe – файл на выполнение. Если возникли ошибки во время компиляции, первая строка, в которой возникла ошибка, выделяется другим цветом. Открывается окно Компилятор с сообщениями об ошибках. Выделение в окне Компилятор строк с сообщениями об ошибках синхронно сопровождается высвечиванием строк с ошибками в тексте программы. При нажатии клавиши Enter активизируется окно редактора, и курсор устанавливается на строку с ошибкой

7. Запуск программы на пошаговое выполнение, строка исходного текста за строкой или трассировка программы	Программа предварительно должна быть скомпилирована. Для пошагового выполнения необходимо установить точку останова, поставив курсор на нужную строку. Для начала отладки необходимо выбрать пункт меню Отладка/Отладка или нажать клавишу F8 (рис. 92). Программа выполнится до точки останова. Затем можно отлаживать программу в пошаговом режиме. Нажатием комбинации клавиш Shift + F7 или выбором команды меню Отладка/Шаг внутрь вызывается выполнение кода, соответствующего одной строке программы. Запустить программу на выполнение до строки, в которой в данный момент расположен текстовый курсор, можно командой меню Отладка/Выполнить до курсора или нажатием клавиши F4	
8. Просмотр результатов выполнения программы	После запуска программы на выполнение открывается окно. В нем можно просмотреть вывод на экран, который осуществляет программа. Можно ввести запрашиваемые данные, а также просмотреть результаты выполнения программы. После завершения работы программы окно автоматически закрывается	
9. Завершение работы ИСР	Выполните команду меню Файл/Выход или нажмите комбинацию клавиш Ctrl + Q	

Рис. 92. Отладчик программы

В таблице 17 трассировка проводится на седьмом этапе, после компиляции программы. Во время трассировки могут быть обнаружены ошибки следующих видов:

- 1) ошибки этапа компиляции;

- 2) ошибки этапа выполнения;
- 3) логические ошибки.

Чтобы представить, как проводится трассировка в программе, попробуем рассмотреть пример ручной трассировки. Для этого приведем пример суммирования чисел от 1 до 10. В таблице 18 слева приведена блок-схема программы, а справа – программа на C++. В таблице 19 дана трассировка цикла с параметром **for**. В результате выполнения трассировки программы сумма чисел от 1 до 10 будет равна 55.

Пошаговая трассировка помогает вникать в логику работы программы. Проверка на каждом шаге позволяет определить, правильны ли наши рассуждения и решение поставленной задачи.

Таблица 18

Блок-схема	Программа на C++
<pre> graph TD Start([Начало]) --> Sum0[sum = 0] Sum0 --> Loop{i = 1 ... 10} Loop --> SumAdd[sum = sum + i] SumAdd --> Loop Loop --> Output[/Вывод sum/] Output --> End([Конец]) </pre>	<pre> #include <iostream> using namespace std; int main() { int i; int sum = 0; setlocale(LC_ALL, «RUS»); for (i = 1; i <= 10; i++) { sum = sum + i; } cout << «Сумма чисел от 1 до 10 = » << sum << endl; return 0; } </pre>

Таблица 19

Трассировка программы

Оператор	Условие цикла	I	Sum	Примечание
sum:=0			0	
for (i = 1; i <= 10; i++)	Да	1		
sum = sum + i			1	0 + 1

for (i = 2; i <= 10; i++)	Да	2		
sum = sum + i			3	1 + 2 = 3
for (i = 3; i <= 10; i++)	Да	3		
sum = sum + i			6	3 + 3 = 6
for (i = 4; i <= 10; i++)	Да	4		
sum = sum + i			10	6 + 4 = 10
for (i = 5; i <= 10; i++)	Да	5		
sum = sum + i			15	10 + 5 = 15
for (i = 6; i <= 10; i++)	Да	6		
sum = sum + i			21	15 + 6 = 21
for (i = 7; i <= 10; i++)	Да	7		
sum = sum + i			28	21 + 7 = 28
for (i = 8; i <= 10; i++)		8		
sum = sum + i			36	28 + 8 = 36
for (i = 9; i <= 10; i++)	Да	9		
sum = sum + i			45	36 + 9 = 45
for (i = 10; i <= 10; i++)	Да	10		
sum = sum + i			55	45 + 10 = 55
for (i = 11; i <= 10; i++)	Нет	???		Значение счетчика цикла не определено
Сумма чисел от 1 до 10 =55				



Знание и понимание



1. Для чего нужен отладчик программы?
2. С какой целью проводится трассировка программы?
3. На каком этапе проводится трассировка?
4. Как проводится трассировка в интегрированной среде разработки Code::Blocks?
5. Какие виды ошибок существуют при отладке программы?
6. Если допущена логическая ошибка, то каким образом лучше ее устранить?



Применение

Составьте и оформите программу, которая выводит таблицу степеней двойки от нулевой до десятой. Ниже приведен рекомендуемый вид экрана программы.

Таблица степеней двойки

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

Выполните трассировку программы и заполните таблицу 20.

Таблица 20

№	Оператор	Условие цикла	i	i * i	Примечание
1					
2					



Анализ

Создайте и оформите программу, которая выводит таблицу значений функции $y = -2,4x^2 + 5x - 3$ в диапазоне от -2 до 2 с шагом $0,5$. Ниже приведен рекомендуемый вид экрана программы (рис. 93).

X	Y
-2	-22.60
-1.5	-15.90
-1	-10.40
-0.5	-6.10
0	-3.00
0.5	-1.10
1	-0.40
1.5	-0.90
2	-2.60

Рис. 93. Рекомендуемый вид экрана программы

Выполните трассировку программы.



Синтез

Составьте блок-схему и таблицу трассировки для задачи, которая выводит на экран таблицу значений функции $y = 2x^2 - 5x - 8$ в диапазоне от -4 до 4 . Шаг изменения аргумента равен $0,5$.

Рекомендации для выполнения домашнего задания

Для переменных присвойте значения, указанные в диапазоне:

$x1 = -4$;

$x2 = 4$;

$dx = 0,5$;

Используйте цикл **while** ($x < x2$) для проверки условия.



Оценивание

Выберите проблемную ситуацию, для которой необходимо написать проект для автоматизации расчета данных с использованием цикла. Сформулируйте проблему в виде задачи. Для решения задачи определите исходные данные. Выберите оптимальный вариант цикла.



Дан код программы:

```
unsigned int i;  
for (i = 100; i >= 0; --i)  
    printf(«%d\n», i);
```

В коде есть две ошибки. Попробуем найти их.

Первая ошибка заключается в том, что используется тип **unsigned int**, который работает только со значениями, большими или равными нулю. Поэтому условие цикла **for** всегда будет истинно, и цикл будет выполняться бесконечно.

Корректный код, выводящий значения всех чисел от 100 до 1, должен использовать условие $i > 0$. Если на самом деле нужно вывести нулевое значение, то следует добавить дополнительный оператор **printf** после цикла **for**.

Вторая ошибка – вместо **%d** следует использовать **%u**, поскольку мы выводим целые значения без знака.

```
unsigned int i;  
for (i = 100; i > 0; --i)  
    printf(«%u\n», i);
```

Теперь этот код правильно выведет список чисел от 100 до 1 в убывающем порядке.

Дополнительные задания для программирования на языке C++

1. С помощью оператора цикла **for** рассчитайте сумму натуральных чисел от 1 до n .

2. Напишите программу по распечатке натуральных чисел от 1 до 15 по пяти столбикам. Например:

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

3. С помощью оператора цикла **while** рассчитайте сумму натуральных чисел от 1 до n .

4. Напишите программу, которая реализует простой калькулятор. Пользователь вводит поочередно числа и символы операций с этими числами (сложение, вычитание, умножение, деление). Операции выполняются до тех пор, пока пользователь не вводит знак равенства.

5. Модифицируйте программу «Калькулятор» так, чтобы, кроме четырех арифметических операций (сложение, вычитание, умножение, деление), выполнялось еще и возведение в степень.

Итоговый тест к четвертому разделу

1. Укажите правильное определение функции **main** в соответствии со спецификацией стандарта ANSI.

- A) `int main(void).`
- B) `void main().`
- C) `void main(void).`
- D) `int main().`

2. Укажите обозначение цикла с предусловием.

- A) `do-while.`
- B) `while.`
- C) `for.`
- D) `if.`

3. Какое значение по умолчанию возвращает программа операционной системе в случае успешного завершения?

- A) `-1.`
- B) `0.`
- C) Программа не возвращает значение.
- D) `1.`

4. Какому зарезервированному слову программа передает управление в случае, если значение переменной или выражения оператора switch не совпадает ни с одним константным выражением?

- A) contingency.
- B) other.
- C) all.
- D) default.

5. Программа, переводящая входную программу на исходном языке в эквивалентную ей выходную программу на результирующем языке, называется:

- A) транслятор;
- B) интерпретатор;
- C) сканер;
- D) компилятор.

6. Оператор вывода на экран:

- A) cin;
- B) read;
- C) write;
- D) cout.

7. Тело любого цикла выполняется до тех пор, пока его условие:

- A) у цикла нет условия;
- B) ложно;
- C) истинно;
- D) равно 0.

8. Общий формат оператора множественного выбора – switch.

- A)
1 switch (switch_expression)
2 {
3 case constant1, case constant2: statement1; [break;]
4 case constantN: statementN; [break;]
5 [default: statement N+1;]
6 }

- B)
1 switch (switch_expression)
2 {

```

3 case constant1: statement1; [break;]
4 case constant2: statement2; [break;]
5 case constantN: statementN; [break;]
6 [else: statement N+1;]
7 }

```

C)

```

1 switch (switch_expression)
2 {
3 case constant1: statement1; [break;]
4 case constant2: statement2; [break;]
5 case constantN: statementN; [break;]
6 [default: statement N+1;]
7 }

```

9. До каких пор будут выполняться операторы в теле цикла while ($x < 100$)?

- A) Пока x больше 100.
- B) Пока x строго меньше 100.
- C) Пока x меньше или равен 100.
- D) Пока x равен 100.

10. Какими знаками заканчивается большинство строк кода в C++?

- A) ; (точка с запятой).
- B) , (запятая).
- C) . (точка).
- D) : (двоеточие).

11. Что будет напечатано в программе?

```

1 int main()
2 {
3 for (int i = 0; i < 4; ++i)
4 {
5 switch (i)
6 {
7 case 0 : std::cout << «0»;
8 case 1 : std::cout << «1»; continue;
9 case 2 : std::cout << «2»; break;
10 default : std::cout << «D»; break;
11 }

```

```
12 std::cout << «.»;  
13 }  
14 return 0;  
15 }
```

- A) 0.1.2.
- B) Ошибка компиляции в строке 10.
- C) 01.2.D.
- D) 0112.D.
- E) 011.2.D.

12. Какой из перечисленных типов данных не является типом данных в C++?

- A) float.
- B) real.
- C) int.
- D) double.

13. Укажите обозначение цикла с постусловием.

- A) for.
- B) while.
- C) do-while.
- D) if.

14. Какая из следующих записей – правильный комментарий в C++?

- A) ** Комментарий **.
- B) */ Комментарий */.
- C) {комментарий}.
- D) /* комментарий */.

15. Структура объявления переменных в C++.

- A) [=], < идент. 2>.
- B) [=];< идент. 2>.
- C) [:=], < идент. 2>.
- D) [==]; < идент. 2>.

16. Какой служебный знак ставится после оператора case?

- A) - (тире).
- B) ; (точка с запятой).
- C) . (точка).
- D) : (двоеточие).

17. Какой оператор не допускает перехода от одного константного выражения к другому?

- A) Stop.
- B) Точка с запятой.
- C) End.
- D) Break.

18. Какой из следующих операторов – оператор сравнения двух переменных?

- A) =.
- B) equal.
- C) ==.
- D) :=.

19. Какую функцию должны содержать все программы на C++?

- A) system().
- B) start().
- C) main().
- D) program().

20. Укажите правильную форму записи цикла do-while.

A)

```
1 // форма записи оператора цикла do-while:
2 do // начало цикла do-while
3 {
4 /*блок операторов*/;
5 }
6 while {/*условие выполнения цикла*/} // конец цикла do-while
```

B)

```
1 // форма записи оператора цикла do-while:
2 do // начало цикла do-while
3 {
4 /*блок операторов*/;
5 }
6 while (/*условие выполнения цикла*/); // конец цикла do-while
```

C)

```
1 // форма записи оператора цикла do-while:
2 do // начало цикла do-while
```

```

3 {
4 /*блок операторов*/;
5 }
6 while (/*условие выполнения цикла*/) // конец цикла do-while

```

21. Какие служебные символы используются для обозначения начала и конца блока кода?

- A) begin end.
- B) < >.
- C) { }.
- D) ().

22. Какие простые типы данных в C++?

- A) Целые – int, вещественные – float или double, символьные – string.
- B) Целые – int, вещественные – float или double, символьные – char.
- C) Целые – bool, вещественные – float или double, символьные – string.
- D) Целые – int, вещественные – float или real, символьные – char.

23. Чтобы подключить заголовочный файл в программу на C++, например iostream, необходимо написать:

- A) #include <> с iostream внутри скобок;
- B) #include <>; с iostream.h внутри скобок;
- C) include #iostream,h;
- D) include (iostreamh).

24. Какой из перечисленных ниже операторов не является циклом в C++?

- A) for.
- B) do-while.
- C) repeat until.
- D) while.

25. Что будет выведено на экран после выполнения условного оператора if?

```

1 int a=7;
2 if (! (a==5)) cout << "hello"

```

- A) 5.
- B) 7.

- C) Hello.
- D) hello.

26. Язык программирования C++ разработал:

- A) Кен Томпсон;
- B) Никлаус Вирт;
- C) Дональд Кнут;
- D) Бьерн Страуструп.

27. Чему будет равна переменная *a* после выполнения кода `int a; for(a = 0; a < 10; a++) {}`?

- A) 9.
- B) 10.
- C) 1.
- D) 11.

28. Выберите правильный вариант объявления константной переменной в C++, где *type* – тип данных в C++ *variable* – имя переменной *value* – константное значение.

- A) `const type variable := value;`
- B) `const variable = value;`
- C) `const type variable = value;`
- D) `const type variable = value`

29. Каков результат работы следующего фрагмента кода?

```
1 int x = 0;
2 switch(x)
3 {
4 case 1: cout << «Один»;
5 case 0: cout << «Нуль»;
6 case 2: cout << «Привет мир»;
7 }
```

- A) Нуль.
- B) Нуль. Привет мир.
- C) Один.
- D) Привет мир.

30. В приведенном коде измените или добавьте один символ, чтобы код напечатал 20 звездочек – *.

```
1 int i, N = 20;
2 for(i = 0; i < N; i--)
3 printf(«*»);
```

A)

```
1 int i, N = 20;
2 for(i = 19; i < N; i--)
3 printf(«*»);
```

B)

```
1 int i, N = 20;
2 for(i = 20; i < N; i--)
3 printf(«*»);
```

C)

```
1 int i, N = 40;
2 for(i = 0; i < N; i--)
3 printf(«*»);
```

D)

```
1 int i, N = 20;
2 for(i = 0; i < N; N--)
3 printf(«*»);
```

Раздел 5

Решение задач в интегрированной среде разработки

5.1. Постановка проблемы

Узнаете

как создавать модели задач в интегрированной среде разработки программ.

Ключевое слово

*Постановка
проблемы*

*Мәселенің
қойылымы*

*Formulation
of the problem*



Проблема – форма научного отображения проблемной ситуации. Проблема формулируется как выражение необходимости изучения и практических действий, направленных на выявление причин и на их разрешение.

В процессе постановки проблемы выдвигают следующие этапы (рис. 94):

1. Формулирование проблемы – это выдвижение центрального вопроса проблемы и предположительного описания ожидаемого результата.

2. Оценка проблемы. В оценку проблемы входит определение всех необходимых для ее решения условий и средств:

- методы исследования;
- источники информации;
- виды научного обсуждения программы и методик исследования;
- конечные и промежуточные результаты и другие.

3. Обоснование проблемы. В детальное обоснование проблемы входят следующие компоненты:

- поиск аргументов в пользу необходимости ее решения, научной или практической ценности ожидаемых результатов;
- выдвижение исследователем возможных возражений против существования проблемы, возможности ее разрешения научным или практическим путем.



Рис. 94. Постановка проблемы

4. Структурирование проблемы. Процесс структурирования начинается с «расщепления» проблемы на второстепенные вопросы. Ответы на них позволят получить ответ на основной проблемный вопрос.



Какие задачи можно решать с помощью моделей? Как создавать модели задач в интегрированной среде разработки программ, применяя постановку проблем?



Вернемся к теме «Моделирование», которую изучили в предыдущих классах. Моделирование является эффективным средством для решения различных задач из окружающего мира. А модель – это условный образ объекта в виде компьютерных диаграмм, таблиц, схем, изображений, анимационных фрагментов. И, естественно, алгоритм, программа отображают структуру и взаимосвязи между элементами объекта.

Для правильного выбора той или иной задачи приведем их классификацию по предметным областям. Для моделирования можно выделить следующие типы задач (рис. 95):

Задачи с практическим содержанием:

- для них известны математические формулы;
- физические законы.

Задачи, для решения которых неизвестны точные методы решения:

- расчет площадей;
- решение уравнений;
- дискретизация непрерывных процессов.

Задачи из различных областей деятельности человека:

- экономики, экологии, лингвистики и т.д.;
- актуализация знаний по информатике для решения задач: экологические, шифровка текста и т.д.

Задачи информационного моделирования:

- организация данных;
- алгоритмы функционирования модели;
- записная книжка;
- модель исполнителя и т.д.

Рис. 95. Классификация типов задач

Для решения задачи первоначально надо поставить (сформулировать) проблему, затем с помощью анализа информации об изучаемом явлении или объекте выделить **исходные данные**. Затем необходимо определить, что будет результатом и какова будет связь между **исходными данными и результатом**.



Для постановки проблемы надо ответить на следующие вопросы:

1. Что мы хотим сделать?
2. Для чего?
3. Какой должна быть программа?



Отвечая на эти вопросы, необходимо четко сформулировать идею задачи. Например, приведем несколько примеров задач:

1. Хочу написать программу, которая будет вычислять число x^6 с помощью трех операций умножения.

2. Хочу написать программу, которая будет вводить двузначное число и находить число десятков в нем.

3. Хочу написать программу, которая будет генерировать 2D-лабиринт и позволит пользователю перемещаться по нему.

4. Хочу написать программу, которая будет выводить на экран визитную карточку о пользователе.

Задача, которая решается с помощью моделей, обычно связана с областью применения объекта, процесса или явления. Для одного объекта может быть множество моделей, и, наоборот, разные объекты могут описываться одной моделью. Например, в механике различные материальные тела (от планеты до песчинки) могут рассматриваться как материальные точки.

Никакая модель не может заменить сам объект. У каждого объекта имеется большое количество информационных аспектов. При решении конкретной задачи вас должны интересовать только определенные свойства изучаемого объекта. И тогда модель оказывается полезным, а иногда и единственным инструментом исследования.



Знание и понимание

1. Как вы понимаете понятие «постановка проблемы»?
2. Перечислите этапы постановки проблемы.
3. Приведите классификацию задач.
4. Перечислите задачи, которые вы решали.
5. Объясните схему решения задач в интегрированной среде разработки на языке C++ в Code::Blocks.
6. На какие вопросы нужно ответить при постановке проблемы?
7. Приведите примеры задач с четко сформулированной идеей.

Применение

Составьте разные виды задач:

- с практическим содержанием;
- для решения которых неизвестны точные методы и которые содержат сложные расчеты;
- из различных областей деятельности человека;
- информационного моделирования.



Анализ



К составленной задаче из предыдущего задания напишите постановку задачи без применения компьютера. Составьте математическую модель задачи.

Рекомендации для выполнения домашнего задания

Математическая модель – это система математических соотношений – формул, уравнений, неравенств и т.д., отражающих существенные свойства объекта или явления.

Создавая математическую модель для решения задачи, нужно:

- выделить предположения, на которых будет основываться математическая модель;
- определить, что считать исходными данными и результатами;
- записать математические соотношения, связывающие результаты с исходными данными.



Синтез



Подумайте, какие проблемные ситуации сложились в вашем городе, поселке, селе, школе. Какая из них представляет для вас наибольшую актуальность? Опишите ее, выделите основные противоречия и сформулируйте проблему в соответствии с этапами постановки проблемы.



Оценивание

Подумайте о возможных способах решения проблемы, выявленной в задании из предыдущего уровня.

Сформулируйте тему, цель и задачи проекта; определите целевую группу, участников, партнеров; составьте календарный план проекта, рассчитайте примерный бюджет и т.д.



Все модели можно разбить на два больших класса: модели **предметные (материальные)** и модели **информационные**.

Предметные модели воспроизводят геометрические, физические и другие свойства объектов в материальной форме. К ним относятся глобус, модели кристаллических решеток, макеты зданий и сооружений и другие.

Информационные модели представляют объекты и процессы в образной или знаковой форме. **Образные модели** (рисунки, фотографии и др.) представляют собой зрительные образы объектов, зафиксированные на каком-либо носителе информации (бумаге, фото- и киноплёнке и др.).

Широко используются образные информационные модели в образовании и науках. Например, учебные плакаты по различным предметам, стенды и другие.

5.2. Разработка алгоритма

Узнаете

- как создавать модели задач в интегрированной среде разработки программ;
- как осуществлять трассировку алгоритма.

Ключевое слово

*Разработка
алгоритма*

*Алгоритмді
құру*

*Algorithm
development*



С разработкой алгоритмов вы уже знакомы. Приведите примеры алгоритмов.



Вам известно, что **алгоритм** – это точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

При разработке алгоритма необходимо помнить о его основных свойствах:

- **понятности** для исполнителя;
- **дискретности** (последовательного выполнения простых шагов);

- **определенности** (четкости, однозначности);
- **результативности** (решения задачи за конечное число шагов);
- **массовости**.

Алгоритм решения задачи разрабатывается в общем виде. Для алгоритмов существуют разные формы представления:

- **словесная** (записи на естественном языке);
- **графическая** (изображения из графических символов);
- **псевдокоды** (полуформализованные описания алгоритмов на условном алгоритмическом языке);
- **программная** (тексты на языках программирования).

При написании алгоритма для решения задачи необходимо в первую очередь ответить на вопрос «Как?». Как правило, хорошие решения имеют следующие характеристики:

- простота;
- документирование;
- построение по модульному принципу. Это означает, что любая часть программы может быть повторно использована или изменена;
- полезные сообщения об ошибках.

Таким образом, прежде чем разработать алгоритм, мы должны выполнить этап постановки проблемы, затем – этап постановки задачи.



1. Этап постановки задачи:
 - сбор информации о задаче;
 - формулировка условия задачи;
 - определение конечных целей решения задачи;
 - определение формы выдачи результатов;
 - описание данных.
2. Анализ и исследование задачи, модели.
3. Разработка алгоритма.

При разработке алгоритма умение составлять блок-схемы является основой. Например, необходимо вывести на экран решение уравнения $a + x = b$ (значения a и b пользователь должен ввести с клавиатуры).

Составим алгоритм решения задачи.

1. Прочитайте задачу.
2. Выпишите известные и неизвестные переменные из задачи в виде:

Дано: a, b

Найти: x – корень уравнения.

3. Составьте математическую модель: $x = b - a$.
4. Составьте блок-схему (рис. 96).
5. Запишите алгоритм на языке программирования C++ (рис. 97):

Блок-схема

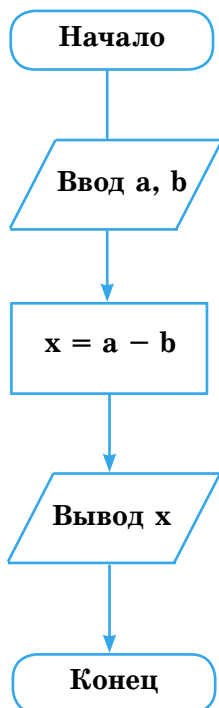


Рис. 96. Блок-схема

Программа на языке программирования C++

The image shows a code editor window titled 'main.cpp' with the following C++ code:

```

1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      double a, b;
6      cout << "Input a: ";
7      cin >> a;
8      cout << "Input b: ";
9      cin >> b;
10     double x = b - a;
11     cout << "x=" << x << endl;
12     return 0;
13 }
14

```

To the right of the code editor is a terminal window showing the program's execution:

```

Input a: 8
Input b: 5
x=-3

Process returned 0 (0x0)
Press any key to continue.

```

Рис. 97. Код программы

При составлении программы необходимы:

- 1) объявление переменных a и b – вещественного типа двойной точности;
- 2) запись числа, введенного пользователем, в переменные a и b ;
- 3) объявление переменной x с одновременным присвоением ей значения;
- 4) вывод строки, вещественного числа и перевод строки.

Для проверки правильности работы программы рекомендуется провести трассировку программы (табл. 21).

Таблица 21

a	b	$x = b - a$	Примечание
8	5	-3	
7	4	-3	
9	3	6	



Знание и понимание



1. Какие свойства нужно учитывать при составлении алгоритма?
2. Какие формы представления алгоритма вы знаете?
3. Что такое блок-схема? Почему блок-схема является наиболее распространенной формой представления алгоритма?
4. Каких этапов при решении задач нужно придерживаться?
5. Почему при разработке алгоритма обязательно проводить трассировку?



Применение



Скорость первого автомобиля – V_1 км/ч, второго – V_2 км/ч, расстояние между ними – S км. Какое расстояние будет между ними через T часов, если автомобили движутся в разные стороны? Значения V_1 , V_2 , T и S задаются с клавиатуры.

Решите, применяя алгоритм решения задачи:

Дано: V_1 , V_2 , S , T

Найти: S_1

Математическая модель: $S_1 = (V_1 + V_2) * T + S$.



Анализ



По блок-схеме, представленной на рисунке 98, решите обратную задачу, определите условие задачи. Проведите трассировку в трассировочной таблице.

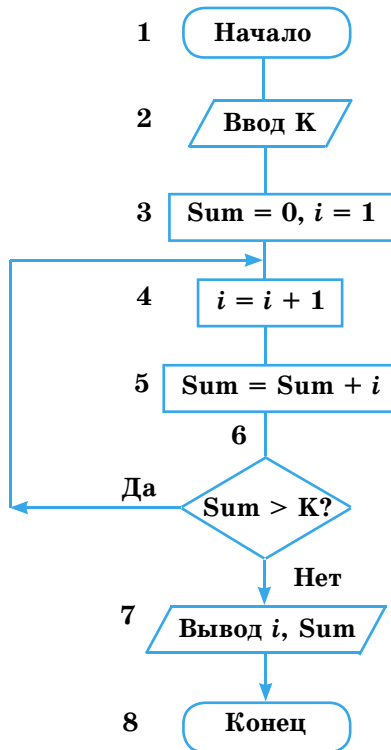


Рис. 98. Блок-схема Сумма чисел



Синтез

Составьте блок-схему для задачи нахождения площади треугольника со сторонами a , b , c по формуле Герона:

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

где a , b , c – длины сторон, а p – полупериметр.

Полупериметр – это сумма длин всех сторон, деленная на два.

$$p = \frac{a + b + c}{2}$$

Рекомендации для выполнения задания

Для проверки правильности решения алгоритма проведите трассировку.



Оценивание

Дано трехзначное число. Найдите сумму его цифр. Реализуйте задачу, применяя словесный алгоритм. Проведите трассировку программы.



Составление алгоритмов решения задач – это работа творческая. Нет универсального способа, позволяющего без особого труда составлять любые алгоритмы.

Основной целью программирования является построение надежной, легко читаемой и модифицируемой программы, решающей поставленную задачу. Для этого программа должна иметь более простую структуру.

Рассмотрим основные методы и технологии разработки алгоритмов и программ.

Структурное программирование – метод создания простых, понятных и легко читаемых программ.

Процедурное программирование – метод построения программы как совокупности ее функциональных частей: процедур или функций.

Модульное программирование – организация программы в виде совокупности независимых частей (модулей) со строгим порядком их взаимодействия.

Объектно-ориентированное программирование – метод программирования, основанный на использовании в программе совокупности объектов, каждый из которых содержит некоторые данные и методы их обработки.

Перечисленные методы реализуют одну из возможных технологий современного программирования: **нисходящую** или **восходящую**.

Нисходящее проектирование – технология разработки программ, при которой на каждом шаге проектирования задача разбивается на более мелкие подзадачи.

Восходящее проектирование – технология разработки программ, при которой сначала проектируются и отлаживаются подпрограммы для выполнения простых операций, после чего они связываются в единую программу.

5.3. Программирование алгоритма

Узнаете

- как создавать модели задач в интегрированной среде разработки программ;
- как использовать компоненты интегрированной среды разработки программ;
- как использовать операторы выбора и циклов в интегрированной среде разработки программ.

Ключевое слово

*Программирование
алгоритма*

*Алгоритмді
программалау*

*Programming
the algorithm*



Вернемся к вопросу: «Как создавать модели задач в интегрированной среде разработки программ?».



После постановки проблемы, разработки алгоритма переходим к решению задачи с помощью компьютерного моделирования. **Компьютерная модель** – это условный образ объекта в виде программы, отображающей структуру и взаимосвязи между элементами объекта.



Компьютерное моделирование – это метод решения задачи анализа или синтеза сложной системы на основе использования ее модели.

Основная задача компьютерного моделирования заключается в получении количественных и качественных результатов на основе имеющейся модели. На основе созданной компьютерной модели проводится серия вычислительных экспериментов в интегрированной среде разработки. Этап исследования свойств объектов или физических процессов с использованием компьютера позволяет найти оптимальные параметры, режимы работы и уточнить модель.

Рассмотрим процесс разработки программы в интегрированной среде разработки. Для того чтобы выполнить программу, необходимо пройти несколько этапов.

На рисунке 99 приведена упрощенная графическая схема решения задачи в интегрированной среде разработки.

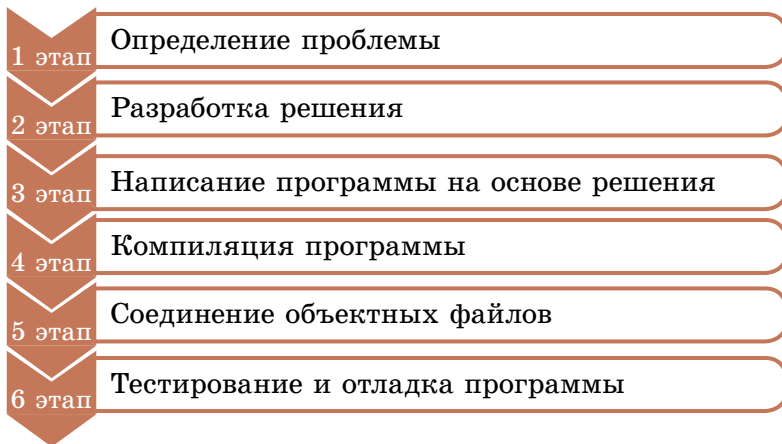


Рис. 99. Схема решения задачи в интегрированной среде разработки

Для программирования алгоритма мы использовали компилятор C++ и свободную кроссплатформенную среду разработки Code::Blocks. Существуют также и другие среды разработки и компиляторы:

- среда Eclipse (Windows/Linux, бесплатная);
- среда Dev-C++ (Windows, бесплатная);
- среда Microsoft Visual C++ (Windows, коммерческая);
- компилятор GCC (Windows/Linux, бесплатный);
- компилятор Intel C (Windows/Linux, в Linux, бесплатный).

Подготовка программы в среде разработки Code::Blocks проходит по упрощенной схеме (рис. 100):



Рис. 100. Упрощенная схема подготовки программы

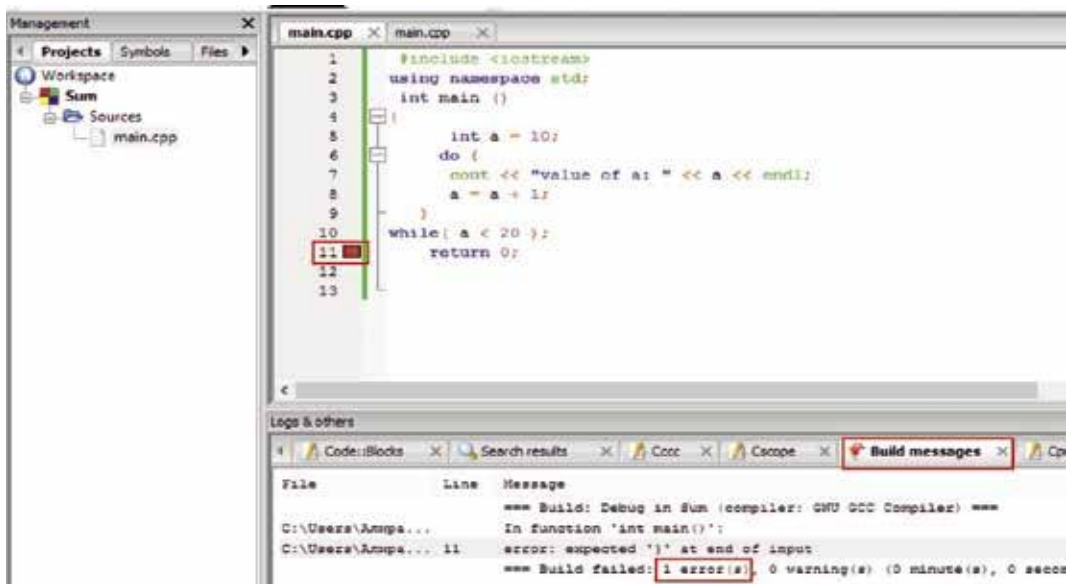


Рис. 101. Интегрированная среда разработки Code::Blocks

При работе в интегрированной среде разработки Code::Blocks надо соблюдать некоторые особенности редактора (рис. 101). Это такие, как:

1. **Нумерация строк** – функция полезна, когда компилятор выдает сообщения об ошибках. Типичная ошибка будет включать в себя текст «ошибка, строка 5».

2. **Подсветка синтаксиса и выделение** изменяют цвет различных частей программы и кода, чтобы было легче увидеть общую картину всей программы и ее структуру.

3. **Специальный шрифт**, позволяющий решить такие ситуации, которые возникают между числом 0 и буквой O, между числом 1, буквой l (нижний регистр L) или буквой I (верхний регистр i). Специальный шрифт предотвращает случайное использование одного символа вместо другого.

При сохранении файла следует обязательно указывать расширение файла – **cpp**: **hello-world.cpp**.

Большинство конструкций в C++ завершаются символом «;». Символ служит для того, чтобы отделять друг от друга различные операции.

Конструкция `#include <iostream>` означает, что необходимо поместить код из заголовочного файла **iostream** в программу, перед тем как создать исполняемый файл.

Подключение к программе заголовочного файла позволит получить доступ к множеству различных функций, которые можно использовать в программе. Например, оператору `cout` требуется `iostream`. Строка `using namespace std;` сообщает компилятору, что нужно использовать группу функций, которые являются частью стандартной библиотеки `std`. В том числе эта строка позволяет программе использовать такие операторы, как `cout`. Функция `cin >>` считывает значение в переменные, которые будет вводить пользователь. Язык C++ довольно простой, но к нему подключается большая библиотека дополнительных компонентов. Эти компоненты находятся в стандартной библиотеке C++ и используются в программе (библиотеки графики, звука).

Очень важно использовать правильный тип переменной, так как некоторым переменным требуется больше памяти. Добавление комментариев делает программу понятней как для себя, так и для других. Чтобы создать комментарий, используйте `//` или `/*` и затем `*/`.

Для компиляции программы используется компилятор. Работа компилятора состоит из двух частей.

1. Проверка программы на соответствие правилам языка C++, то есть **проверка синтаксиса**. Если проверка будет неудачной, то компилятор выдает сообщение об ошибках, которые нужно будет исправить.

2. Конвертация каждого исходного файла с кодом в объектный файл на машинном языке. Объектные файлы, как правило, имеют названия `name.o` или `name.obj`. Имя должно быть такое же, как имя исходного `.cpp` файла. Если программа имеет три `.cpp` файла, то компилятор сгенерирует три объектных файла.



Знание и понимание

1. Для чего применяется компьютерное моделирование?
2. Какие этапы включает решение задач в интегрированной среде разработки?
3. Приведите упрощенную схему решения задачи в интегрированной среде разработки.
4. Для чего нужен компилятор?
5. Какие компиляторы, кроме C++, вы знаете?
6. Какие особенности надо соблюдать при написании программы в интегрированной среде разработки?
7. Из каких двух частей состоит работа компилятора?



Применение

Заполните алгоритмические кроссворды. Придумайте задания для приведенных блок-схем (рис. 102). Напишите к ним программы. Сделайте трассировку программ.

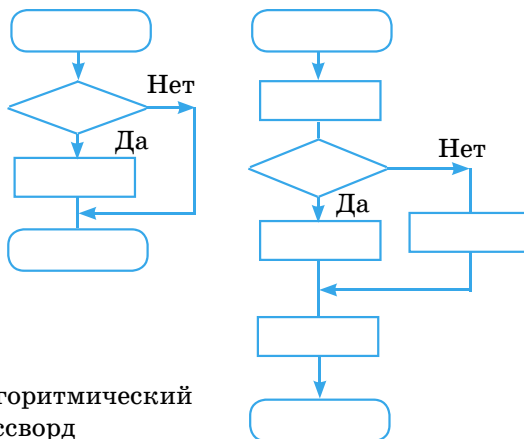


Рис. 102. Алгоритмический кроссворд



Анализ

Для графика функции $y = x^2$ (рис. 103) приведен алгоритм на школьном алгоритмическом языке. С помощью данного алгоритма составьте программу на языке программирования C++. Выполните трассировку программы.

Школьный алгоритмический язык

```
алг
нач вещ x, y
ввод x, y
если y <= x * x
то
    если x >= -1
    то
        если y >= 1
        то
            вывод "принадлежит"
        иначе
            вывод "не принадлежит"
        все
    все
    все
кон
```

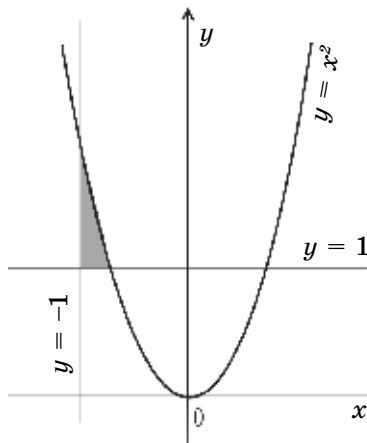


Рис. 103. График функции $y = x^2$



Синтез

Задача «Дни недели»

1. По введенному номеру дня недели выведите его название, используя оператор множественного выбора **switch**. Нумерация дней недели начинается с: 1 – понедельник, 2 – вторник, 3 – среда, 4 – четверг, 5 – пятница, 6 – суббота, 7 – воскресенье.

Вид экрана:

Введите номер: 2

Вторник

2. Разработайте блок-схему задачи, программу, проведите трассировку.



Оценивание

На рисунке 104 дан программный код. Сформулируйте постановку задачи. Составьте блок-схему программы и таблицу трассировки.



```
1      #include <iostream>
2      using namespace std;
3      int main ()
4      {
5          { int n,a;
6              cin >> n; // ввод
7              a=1;
8              for (int i=1;i<=n;i++) a=a*i; // цикл
9              cout<<a; //вывод
10         }
11
12         return 0;
13     }
14
15
```

Рис. 104. Программный код

Рекомендации для выполнения домашнего задания

Для определения постановки задачи разберите каждый оператор. Проведите трассировку программы.



Интересные задачи

1. Переворот числа. Вводится число. Преобразовать его в другое число, цифры которого будут следовать в обратном порядке по сравнению с введенным числом.
2. Дано число. Найти сумму и произведение его цифр.
3. Числа Фибоначчи. Вывести на экран ряд чисел Фибоначчи, состоящий из n элементов.
Числа Фибоначчи – это элементы числовой последовательности 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ..., в которой каждое последующее число равно сумме двух предыдущих.
4. Вывод квадратов натуральных чисел. Вывести все квадраты натуральных чисел, не превосходящие данного числа N .
Пример: $N = 50$ | 1 4 9 16 25 36 49
5. Написать программу, подсчитывающую количество четных и нечетных цифр в числе.
6. Кубы чисел от A до B . Вывести на экран кубы чисел от A до B , которые вводит пользователь.
7. Вывод на экран таблицы ASCII. ASCII – код для обмена информацией. В настоящее время чаще используется 8-битный вариант (под каждый символ выделяется 1 байт памяти). Проще говоря, получается таблица из 256 символов, соответствующих числам от 0 до 255.
8. Отгадать целое число, которое «загадал» компьютер в определенном диапазоне.
9. Определить количество дней в году, которые вводит пользователь. В високосном году 366 дней, тогда как в обычном их 365. Високосными годами являются все годы, делящиеся нацело на 4, за исключением столетий, которые не делятся нацело на 400.
10. Определить принадлежность точки кругу. Даны координаты точки и радиус круга с центром в начале координат. Определить, принадлежит ли данная точка кругу.

5.4. Тестирование программы

Научитесь

- создавать модели задач в интегрированной среде разработки программ;
- осуществлять трассировку алгоритма.

Ключевое слово

*Тестирование
программы*

*Программаны
тестілеу*

*Testing
the program*

Необходимым условием для проверки правильности выполнения программы является **проведение тестирования**. На тестах проверяется реализация программы запланированного алгоритма.



Тестирование программы – это процесс выполнения программы с целью обнаружения ошибки в программе на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этих программ. Указанный набор данных называется **тестовым**, или просто **тестом**.



Процесс поиска и исправления ошибок называется **отладкой**.

Отладку можно представить в виде многократного повторения трех этапов:

1. На первом проводится тестирование программы и обнаружение ошибки.
2. На втором – поиск места ошибки в программе и документации.

3. На третьем – редактирование программы и документации с целью устранения обнаруженной ошибки.

К отладке программы нужно подходить ответственно. С ее помощью отыскиваются и устраняются ошибки, обнаруженные только при тестировании.

Очень важно подготовить такой набор тестов и применить к ним программу, чтобы обнаружить в нем по возможности большее число ошибок.

Но чем дольше продолжается процесс тестирования (и отладки в целом), тем большей становится стоимость программы.

Для тестирования программы необходимо заранее подготовить набор тестов. Эти тесты должны при заданном интервале времени, отведенном на тестирование, обнаруживать большее число ошибок.

Таким образом, при тестировании и отладке можно столкнуться со схемой, указанной на рисунке 105:

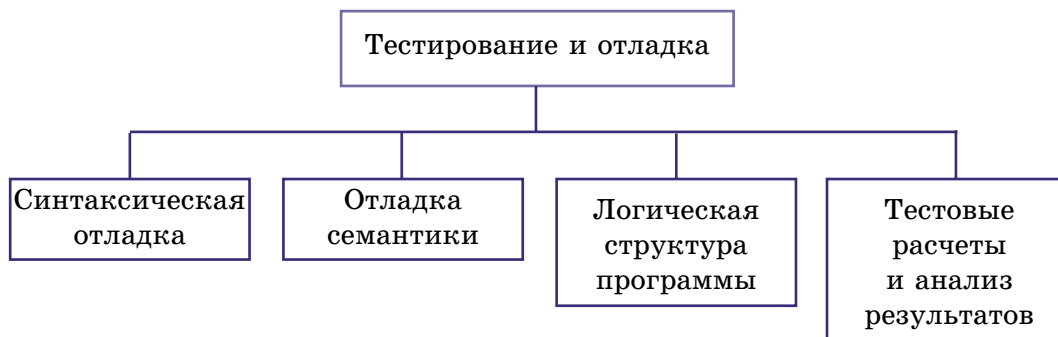


Рис. 105. Схема тестирования и отладки

После отладки программы следует ее **совершенствование**. Для этого нужно запустить исполняемый файл и проанализировать результат, предполагаемый изначально. Если ответ отрицательный, то необходимо продолжить поиск ошибок.

В интегрированной среде разработки Code::Blocks имеется встроенный отладчик.

Большое преимущество интегрированной среды разработки заключается в том, что она собирает и объединяет все используемые отдельные программы в одно целое. В среде также автоматически генерируются параметры для компиляции и связки программы в исполняемый файл.

Например, на экран нужно вывести одностороннюю пирамиду из звездочек высотой n (рис. 106).

Ввод	Вывод	Программный код
3	* ** ***	<pre>#include <iostream> using namespace std; int main() { int n = 0; cin >> n; for(int i = 1; i <= n; ++i) { for(int s = 0; s < i; ++s) cout << '*'; cout << endl; } return 0; }</pre>
5	* ** *** **** *****	

Рис. 106. Пирамида звездочек



Проанализируйте сообщение об ошибках после отладки (рис. 107). Какая ошибка допущена в программе?



```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6
7  {
8      int n = 0;
9      cin >> n;
10     for( int i = 1; i <= n; ++i ) {
11         for( int s = 0; s < i; ++s )
12             cout << '*';
13         cout << endl;
14     }
15     return 0;
16 }
```

Build messages:

```

=== Build: Debug in piramida (compiler: GNU GCC Compiler) ===
C:\Users\Ampa... 7 error: expected initializer before 'int'
C:\Users\Ampa... 8 error: 'cin' does not name a type
C:\Users\Ampa... 9 error: expected unqualified-id before 'for'
C:\Users\Ampa... 9 error: 'i' does not name a type
C:\Users\Ampa... 9 error: expected unqualified-id before '++' token
C:\Users\Ampa... 14 error: expected unqualified-id before 'return'
C:\Users\Ampa... 15 error: expected declaration before '}' token
=== Build failed: 7 error(s), 0 warning(s) (0 minute(s), 1 second(s)) ===

```

Рис. 107. Сообщение об ошибках



Знание и понимание

1. С какой целью проводится тестирование программы?
2. Как вы понимаете термин «отладка»? Для чего она применяется?
3. Какие типы ошибок существуют?
4. Как можно представить отладку?
5. Что входит в схему тестирования и отладки?
6. Если в программе есть логическая ошибка, то выполнится ли программа?



Применение

С клавиатуры вводятся числа, последнее число – 0. Найдите сумму этих чисел. Какие ошибки допущены в программе? Устраните ошибки. Введите программный код в редактор. Откомпилируйте и выполните программу. Проверьте правильность выполнения программы с помощью ниже приведенных тестов:

Ввод

1 2 3 4 5 6 7 8 9 0
100500 100500 0

Вывод

55
201000

```
#include <iostream>
using namespace std;
int main ()
{
    int sum = 1, a = 0;
    do {
        cin >> a;
        sum = sum + a;
    }
    while (a = 0);
    cout << sum;
    return 1;
}
```



Анализ

Даны три типа циклов. Какой цикл будет выполняться быстрее? Каким образом это можно узнать?

Цикл с предусловием на C++:

```
int i = 0;
while (i < a) { //условие продолжения
    cout << i; //тело цикла
    i = i + 1;
}
```

Цикл с постусловием на C++:

```
int i = 0;
do {
    cout << i; //тело цикла
    ++i; // То же, что и i = i + 1; но короче
}
while (i < a); //условие
```

Цикл for:

```
for (int i = 0; i < 10; ++i) {
    //тут тело цикла
}
```

Рекомендации для выполнения задания

Допишите программы и введите программный код в редактор.

В отладчике найдите сообщение о том, сколько секунд выполняется программа.

Синтез

На рисунке 108 приведен кроссворд со словами. Выполните обратную задачу: напишите определения к данным словам (по горизонтали и вертикали).

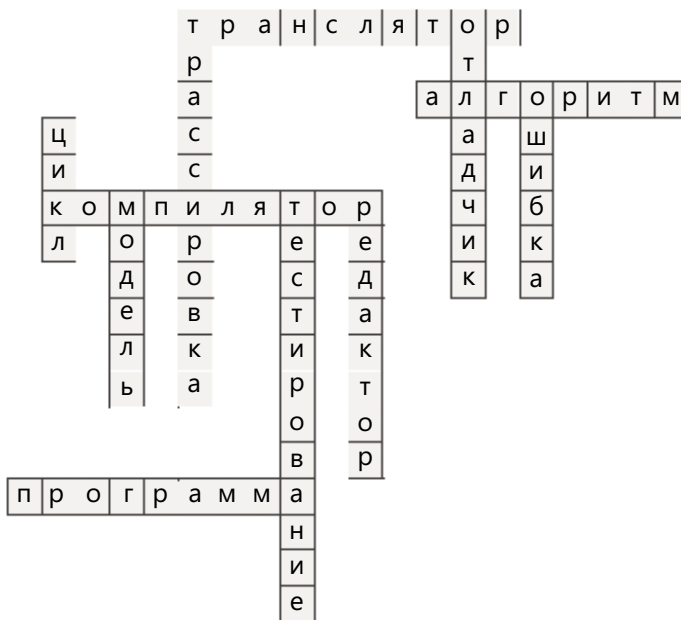


Рис. 108. Кроссворд



Оценивание

Проведите исследование по выбранной вами проектной идее. Выполните следующие этапы:

1. Постановка проблемы.
2. Постановка задачи.
3. Создание математической модели.
4. Разработка алгоритма.
5. Разработка программы.
6. Отладка и тестирование.
7. Компиляция и выполнение.
8. Документация к алгоритму, программе.
9. Презентация проекта.
10. Защита проекта.



Программирование – один из ключевых навыков, которому нужно обучать с младших классов. Компьютерный код – тот же иностранный язык, только он позволяет разговаривать с компьютером, ставить ему задачи и контролировать их выполнение. Знание этого языка сегодня так же необходимо, как знание английского и математики.

Компьютеры занимают очень важное место в современном мире. Однако мы можем не только использовать эти технологии. Научившись программировать, мы можем развивать их, создавать собственные произведения цифрового искусства.

Многие считают, что программирование – одно из наиболее важных умений в XXI веке. Учиться программировать очень интересно. Это отличная возможность для творчества. Это первая область науки, со вмещающая искусство, логику, сочинительство и бизнес.

Кроме того, умение программировать очень пригодится в жизни. Оно развивает логику и интеллект, которые важны в самых разных областях. Количество вакансий, где нужно умение программировать, будет со временем только расти.

Итоговые задания к пятому разделу

1. Установите соответствие (табл. 22):

Таблица 22

1) Трансляция	А) Переводит программу целиком, а затем ее выполняет
2) Интерпретатор	В) Перевод программы на язык, понятный компьютеру
3) Компилятор	С) Переводит и выполняет программу строка за строкой

2. Дополните. *Перевод программы на язык, понятный компьютеру, называется ...*

3. Дополните. *... переводит и выполняет программу строка за строкой.*

4. Дополните. *... переводит программу целиком, а затем ее выполняет.*

5. Укажите правильный ответ. *Величиной целого типа `Int` является:*

- А) количество мест в зрительном зале;
- В) рост человека;
- С) площадь государства;
- Д) оценка в дневнике.

6. Установите соответствие между номерами (рис. 109) и их названиями, обозначенными латинскими буквами.

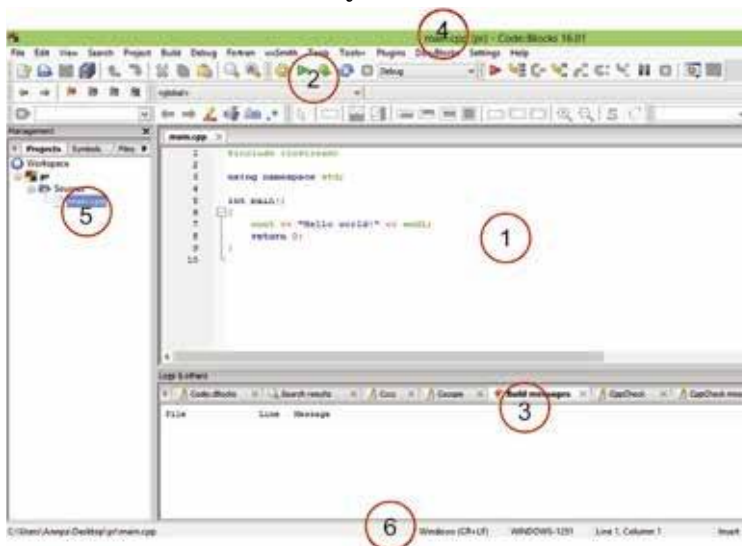


Рис. 109. Интерфейс UCP Code::Blocks

- А) Главное меню.
- В) Имя файла.
- С) Редактор.
- Д) Строка состояния.
- Е) Отладчик.
- Ф) Рабочее пространство.

7. Установите соответствие (табл. 23).

Таблица 23

1) Символьный тип данных	а) int
2) Вещественный	б) char
3) Целый тип данных	в) double

8. Дополните. *Постановкой проблемы называется ...*

9. Дополните. *Тестирование программы – это ...*

10. Установите соответствие между номерами и их названиями, обозначенными латинскими буквами (табл. 24).

Таблица 24

1) {}	А) включить в текст программы файл iostream
2) cin	В) тело любой функции
3) <<	С) то, что нужно вывести
4) endl	Д) приводит к переводу курсора
5) #include <iostream>	Е) оператор вывода
6) cout	Ф) оператор ввода
7) return 0;	Г) 0 – признак того, что программа завершилась успешно

11. Укажите правильный ответ. *В конце оператора ставится:*

- А) точка;
- В) точка с запятой;
- С) двоеточие;
- Д) восклицательный знак.

12. Укажите правильный ответ. Оператор присваивания обозначается:

- A) :=
- B) =
- C) :
- D) = =

13. Установите правильную последовательность действий.

1. Проведение расчетов.
2. Получение результатов.
3. Составление программы.
4. Построение алгоритма.
5. Постановка задачи.
6. Отладка и тестирование.

14. Допишите. Разработка алгоритма – это ...

15. Допишите. Трассировка программы – это ...

16. Запишите ответ на вопрос. Что сообщает строка «using namespace std» компилятору?

17. Из каких двух частей состоит работа компилятора?

18. Нужны ли комментарии к программе? Ответ обоснуйте.

19. Какие основные свойства нужно помнить при составлении алгоритма?

20. В чем особенности операторов циклов for, while, do-while?

Интегрированная среда разработки Code::Blocs

После создания проекта в интегрированной среде разработки **Code::Blocs** появляется главное окно с закрытым кодом (рис. 1).

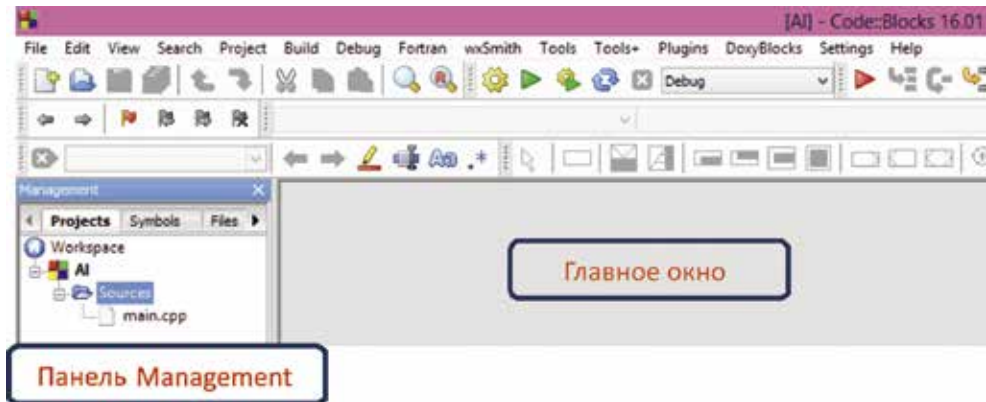


Рис. 1. Главное окно и панель **Management**

Слева будет отображаться панель **Management** (рис. 2). На ней отображается в виде дерева иерархическая структура проекта на вкладке **Project**. Проект будет принадлежать рабочему пространству **Workspase** и находиться в каталоге **Sources** с именем файла **main.cpp**. Имя проекта будет таким, какое вы дадите при создании.

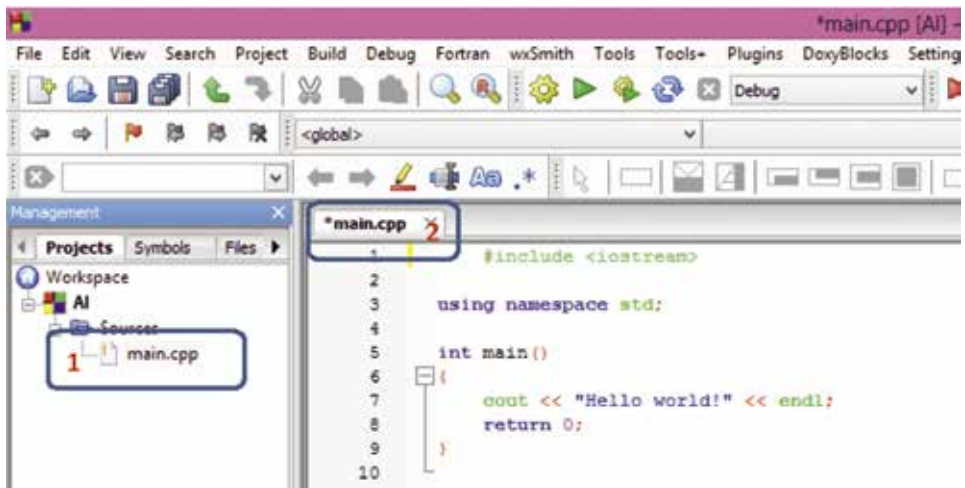


Рис. 2. Открытый файл **main.cpp**

Кроме того, в левой части строки состояния будут отображаться путь и имя открытого в данный момент файла. Для нормальной работы программы должна быть установлена кодировка **UTF-8**, находящаяся правой части строки состояния.

Для настройки кодировки в **Code::Blocs** нужно выбрать пункт **Editor** в меню **Settings**. В окне **Configure editor** настроек текстового редактора кода надо выбрать кодировку **UTF-8** (рис. 3), нажать на кнопку **OK** и перезапустить все открытые файлы.

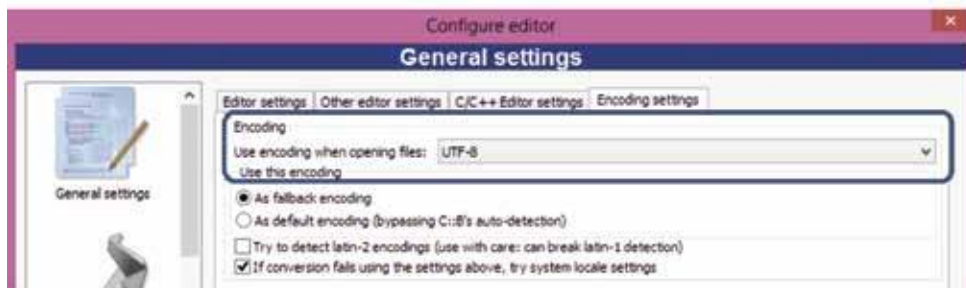


Рис. 3. Настройка кодировки текстового редактора

Для того чтобы вывести или спрятать панели инструментов в Code::Blocs, применяют пункт **Toolbars** из меню **View** (рис. 4).

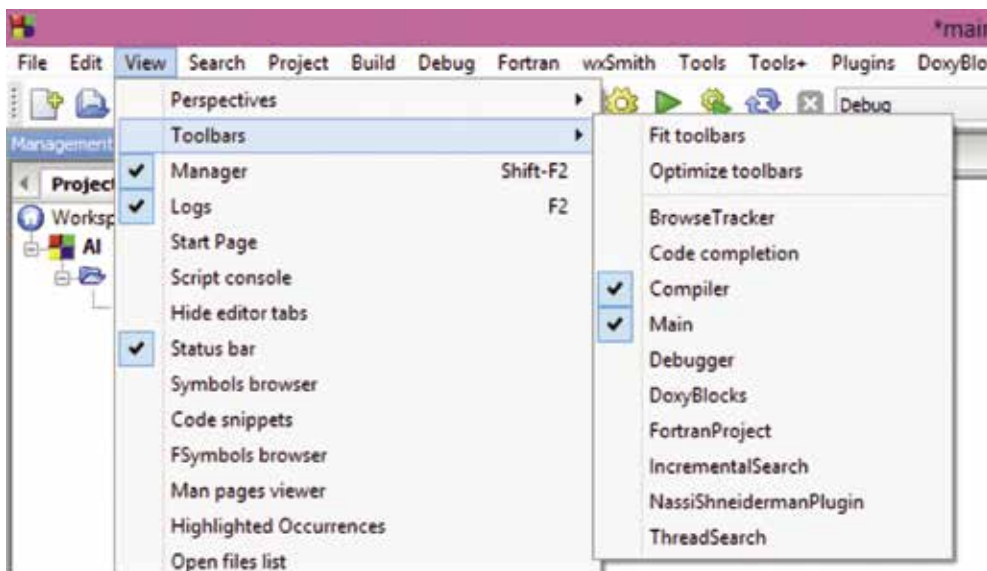
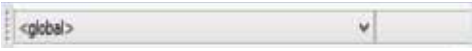


Рис. 4. Настройка внешнего вида Code::Blocs

Например, в таблице 1 представлены часто используемые панели в интегрированной среде разработки программ.

Таблица панелей в ИСР Code::Blocs

Таблица 1

Main — главная панель инструментов, на которой отображаются основные действия по работе с проектом	
Code completion — эта панель позволяет просматривать объекты кода	
Compiler — на данной панели находятся кнопки управления компиляцией приложения	
Debugger — панель, на которой находятся кнопки управления отладкой приложения	

Для сборки проекта, компиляции и запуска приложения необходимо нажать клавишу **F9** или пункт **Build and Run** в меню **Build** (рис. 5).

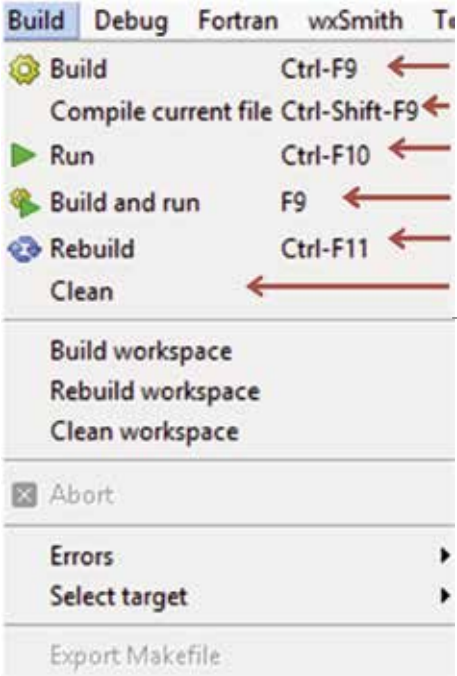
	Сборка всего приложения Компиляция файла, открытого в данный момент Запуск откомпилированного приложения Собрать и запустить Пересобрать Очистить проект от скомпилированных и временных файлов
--	---

Рис. 5. Меню Build

Для того чтобы только скомпилировать проект без запуска, нужно выбрать пункт меню **Build** или нажать на сочетание клавиш **Ctrl + F9**. Для удаления временных файлов проекта нужно выбрать пункт меню **Clean**.

Для поиска ошибок в программе используется отладчик **Gdb**. Чтобы запустить отладчик, выберите пункт меню **Debug – Start** (рис. 6).

Debug Fortran wxSmith Tools Tools+ Plugir Active debuggers			
▶ Start / Continue	F8	←	Запуск отладки
Break debugger			
☒ Stop debugger	Shift-F8	←	Остановка отладки
🏠 Run to cursor	F4	←	Выполнить программу до курсора
🏠 Next line	F7	←	Переход к следующей строке
🏠 Step into	Shift-F7	←	Войти в функцию
🏠 Step out	Ctrl-F7	←	Выйти из функции
🏠 Next instruction	Alt-F7		
🏠 Step into instruction	Alt-Shift-F7		
Set next statement			
Toggle breakpoint	F5	←	Установить точку останова
Remove all breakpoints			
Add symbol file			
Debugging windows			
Information			
Attach to process...			
Detach			
Send user command to debugger			

Рис. 6. Меню Debug

Глоссарий

Алгоритм – это точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

Быстродействие – показатель скорости работы ЭВМ и ее производительности в единицу времени (число операций в секунду).

Вероятность – это степень возможности наступления некоторого события.

Группировка данных – объединение диапазона ячеек.

Инструментальная среда – это интегрированная система, которая используется в ходе разработки, корректировки, редактирования, отладки, запуска программы на выполнение, не выходя из системы.

Интегрированная среда разработки ИСР/IDE (англ. Integrated development environment) – комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО).

Инфографика – это способ подачи информации, данных и знаний с помощью графического изображения.

Итерация цикла – один проход цикла.

Информационные модели – это объекты и процессы в образной или знаковой форме.

Информационный вес символа – количество информации в одном символе.

Канал связи – технические средства, позволяющие осуществлять передачу данных на расстоянии.

Кибербуллинг – это травля, оскорбления или угрозы, высказываемые жертве с помощью средств электронной коммуникации, в частности, сообщений в социальных сетях, мгновенных сообщений, электронных писем и СМС.

Компьютерная модель – это условный образ объекта в виде программы, отображающий структуру и взаимосвязи между элементами объекта.

Компьютерное моделирование – это метод решения задачи анализа или синтеза сложной системы на основе использования ее модели.

Консолидация данных – объединение нескольких диапазонов данных в один диапазон.

Математическая модель – это система математических соотношений – формул, уравнений, неравенств и т.д., отражающих существенные свойства объекта или явления.

Мощность алфавита (N) – количество символов в алфавите (его размер).

Образные модели (рисунки, фотографии и др.) – зрительные образы объектов, зафиксированные на каком-либо носителе информации (бумаге, фото- и киноплёнке и др.).

Отладка – это процесс поиска и исправления ошибок в программе, препятствующих корректной работе программы.

Относительная ссылка – это обозначение ячейки, составленное из номера столбца и строки.

Предметные модели воспроизводят геометрические, физические и другие свойства объектов в материальной форме. К ним относятся глобус, модели кристаллических решеток, макеты зданий и сооружений и другие.

Прикладная задача – задача, поставленная вне информатики, но решаемая методами использования ИКТ. В прикладной задаче прослеживается взаимосвязь других знаний (математики, физики, техники и т.д.) и видов деятельности, связанных с информатикой. Приемы решения таких задач приближены к профессиональной деятельности.

Прикладная программа – это любая конкретная программа, которая способствует решению какой-либо задачи в пределах данной программной области.

Программное обеспечение (ПО или Software) – это совокупность всех программ, обеспечивающих функционирование аппаратных средств, разработку, отладку и выполнение задач пользователя.

Пропускная способность канала – максимальная скорость передачи информации по каналу связи в единицу времени.

Процессор (центральный процессор, Central Processing Unit) – это микросхема, обрабатывающая информацию и координирующая работу всех устройств компьютера.

Разрядность процессора – это число одновременно обрабатываемых процессором битов.

Регистр – это ячейка процессора, в которой хранится машинное слово. Машинное слово представляет собой некоторое число (или команду), записанное в двоичном коде.

Сводная таблица – это отчет, построенный на основе табличных данных.

Скорость передачи информации – количество информации, передаваемое за единицу времени.

Спам – рассылка навязчивой, нежелательной рекламной информации.

Счетчик цикла – это управляющая переменная, в которой хранится количество проходов данного цикла.

Табулирование функции – это вычисление значений функции при изменении аргумента от некоторого начального значения до некоторого конечного значения с определенным шагом.

Тактовая частота – скорость передачи информации между устройствами компьютера.

Тестирование программы – это процесс выполнения программы с целью обнаружения ошибки в программе на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этих программ.

Трассировка – это процесс выполнения программы по шагам (step-by-step), инструкция за инструкцией. Во время трассировки программист дает команду выполнить очередную инструкцию программы.

Спарклайны – это отслеживание тенденции в рядах данных (колебание цен, объем продаж).

Троллинг – форма социальной провокации или издевательства в сетевом общении.

Фишинг – это сетевое мошенничество.

Функция – встроенный инструмент Excel, способный вернуть значение в зависимости от переданного ему параметра.

Цикл – это многократное выполнение одного и того же кода программы, пока условие истинно.

Шаг цикла – это значение, на которое будет увеличиваться или уменьшаться счетчик цикла при каждом проходе.

Эксабайт – единица измерения количества информации, равная 10^{18} байт.

Электромагнитное излучение – это вид энергии, представляющей электромагнитные волны, возбуждаемые различными излучающими объектами.

Рекомендуемая литература

1. *Васильев А.Н.* С++ с задачами и примерами. Учебное пособие. – СПб: Наука и техника, 2010.
2. *Златопольский Д.М.* 1700 заданий по Microsoft Excel. – СПб: БХВ-Петербург, 2003.
3. *Литвиненко Н.А.* Технология программирования на С++. – СПб: БХВ-Петербург, 2005.
4. *Молодцов В.А., Рыжикова Н.Б., Головки Т.Г.* Репетитор по информатике. – Ростов-на-Дону, 2004.
5. *Мухамбетжанова С.Т., Тен А.С., Даутова Т.К., Сагымбаева А.Е.* Информатика: учебник для 7 класса общеобразовательных школ. – Алматы: Атамұра, 2015.
6. *Семакин И.Г., Вараскин Г.С.* Структурированный конспект базового курса. Приложение к учебнику: Информатика. Базовый курс 7–9 классы. – М.: Лаборатория базовых знаний, 2001.
7. Основы компьютерных сетей. Учебное пособие. – Алматы: Microsoft Corporation, 2008.
8. *Попов А.* Excel: Практическое руководство. – М., 2000.
9. *Романов Е.Л.* Си++. От дилетанта до профессионала. – Новосибирск, 2014.
10. *Стефан Р. Дэвис.* С++ для чайников. – Москва. Санкт-Петербург: Диалектика, 2003.

Рекомендуемые интернет-ресурсы

1. <https://bilimland.kz/ru>.
2. <http://on-line-teaching.com/excel/index.html>.
3. http://www.lessons-tva.info/edu/e-inf2/m2t2_1.html.
4. http://www.lessons-tva.info/edu/e-inf2/m1t1_4.html.
5. <http://www.informatik.kz/>.
6. <https://rcdo.kz/>.
7. <http://daryn.kz/>.
8. <http://www.c-cpp.ru/>.
9. <http://cppstudio.com/cat/285/>.

10. <https://tproger.ru/digest/competitive-programming-practice/>.
11. <http://programm.ws/kat.php?id=14>.
12. metod-kopilka.ru.
13. <http://www.openclass.ru/>.
14. <https://www.osp.ru/school/archive>.
15. <http://assembly.kz>.
16. <http://ypk.yspu.yar.ru/tolerance/3.htm>.
17. enet.kz.
18. it-n.ru.
19. <http://schools.keldysh.ru/sch444/MUSEUM/>.
20. <http://www.computer-museum.ru/index.php>.
21. <http://school-collection.edu.ru>.
22. <http://www.edusite.ru>.
23. <http://www.videoyroki.info/>.

Учебное издание

**Мухамбетжанова Сауле Талапеденовна
Тен Алира Сунтаковна
Маликова Найла Токтамуратовна**

ИНФОРМАТИКА

Учебник для 8 класса общеобразовательной школы

Зав. редакцией *Н. Жиенгалиев*

Редактор *Л. Холина*

Художественный редактор *А. Беккожанова*

Технический редактор *О. Рысалиева*

Корректоры *С. Тайтелиев, Г. Туленова*

Компьютерная верстка *С. Тулегеновой*

ИБ № 076

Сдано в набор 21.01.2018. Подписано в печать 30.05.2018. Формат 70х90 ¹/₁₆.
Бумага офсетная. Гарнитура «Школьная». Печать офсетная. Усл.печ.л. 14,04.
Уч.-изд.л. 9,36. Тираж 15 000 экз. Заказ № 3484.

ТОО «Корпорация «Атамұра», 050000, г. Алматы, пр. Абылай хана, 75.

Полиграфкомбинат ТОО «Корпорация «Атамұра», Республика Казахстан,
050002, г. Алматы, ул. М. Макатаева, 41

