

Лабораторная работа №1

Знакомство с Java

Цель работы: изучить основные понятия, синтаксис языка Java и общую структуру программ, получить практические навыки программирования на языке Java.

Требования к выполнению лабораторных работ

Все создаваемые классы должны лежать в пакете:

ru.bstu.группа.ваша_фамилия.labномер_лабораторной

Например, для студента группы ИТ-31 Петрова К.В., сдающего 1-ю лабораторную пакет будет называться

ru.bstu.it31.petrov.lab1

При защите лабораторной работы необходимо предоставить:

- Отчет (титульный лист, задание на работу, необходимые пояснения, исходный текст программы, скриншоты);
- Тестовые данные и результаты работы приложений на них.
- Откомпилированные классы программы в виде отдельных файлов .class, либо архива .jar;
- Для запуска откомпилированного приложения:
 - Windows BAT-файл
 - Linux Shell-файл
- Тексты должны содержать комментарии разработчика для всех методов и классов.

Основные понятия

Java – строго типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный **байт-код**, поэтому они могут работать на любой компьютерной архитектуре, с помощью **виртуальной Java-машины** – специальной программы, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина.

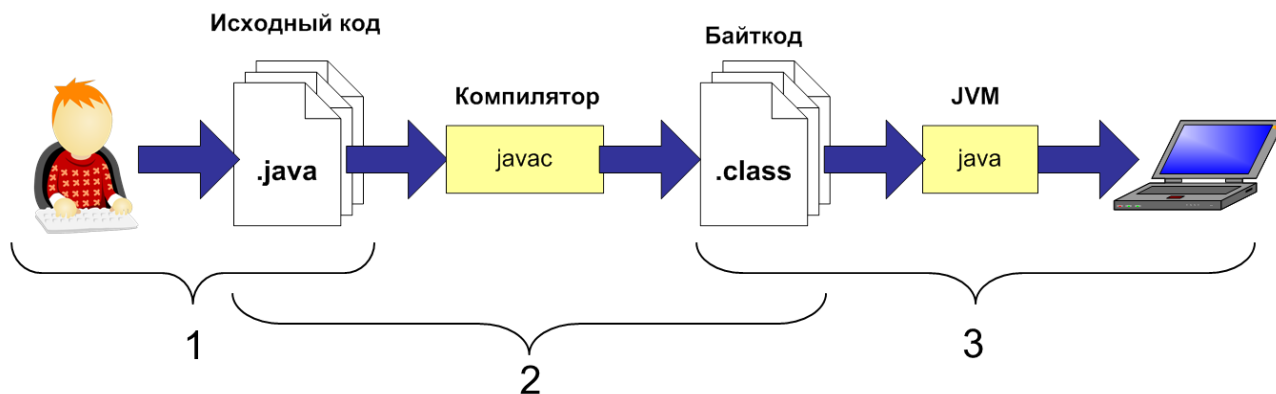


Рисунок 1. Этапы создания приложения Java.

Java Virtual Machine (JVM) – виртуальная машина Java, представляющая собой специальную программу, зависящую от платформы и операционной системы. Она исполняет байт-код, предварительно созданный из исходного текста Java-программы компилятором Java (javac – java compiler). Так как виртуальные машины Java доступны для многих аппаратных и программных платформ, один и тот же байт-код может исполняться на различных операционных системах.

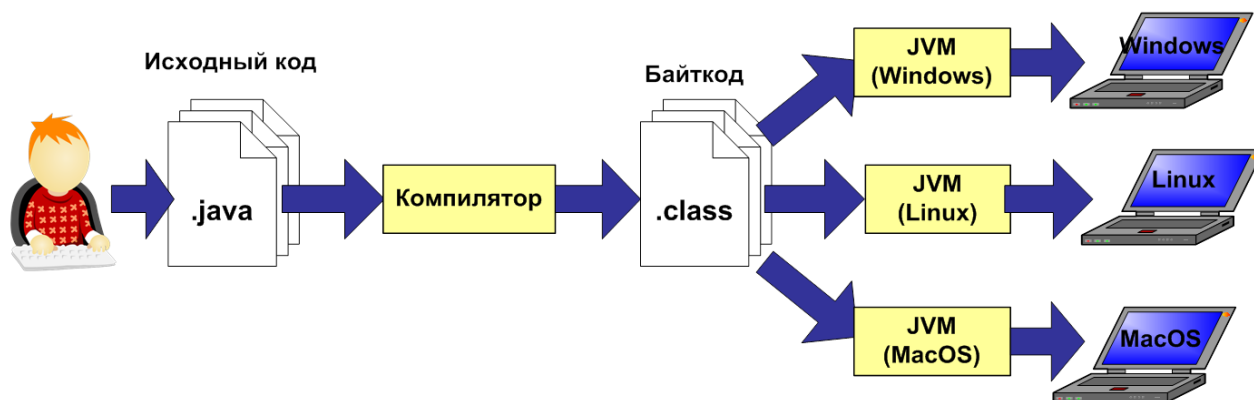


Рисунок 2. Кроссплатформенность приложения Java.

JVM является частью среды выполнения **Java Runtime Environment (JRE)**, куда входит также набор стандартных библиотек Java.

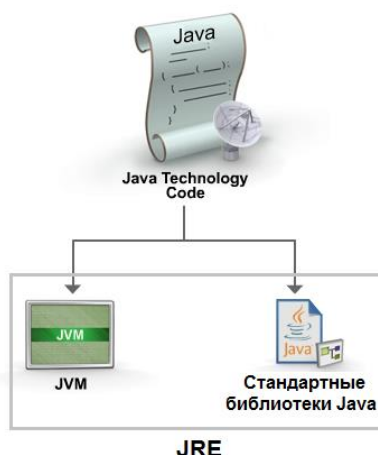


Рисунок 3. Java Runtime Environment.

Java Development Kit (JDK) — бесплатно распространяемый компанией Oracle Corporation комплект разработчика приложений на языке Java, включающий в себя компилятор Java (javac), стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную систему Java (JRE).

Таким образом, чтобы исполнять программы, написанные на языке Java достаточно установить на ПК комплект JRE. Для разработки же программ потребуется инсталляция JDK.

Платформы Java

- Standart Edition (JSE) – для создания и исполнения приложений для рабочих станций (или использования в масштабах малого предприятия);
- Enterprise Edition (JEE) – для создания коммерческих приложений для крупных и средних предприятий;
- Micro Edition (JME) – для разработки приложений для устройств, ограниченных в ресурсах (сотовых телефонов, карманных персональных компьютеров и т.п.).



Рисунок 4. Платформы Java.

Версии платформы Java SE:

Год	JDK	Платформа
1996	1.0	1
1997	1.1	1
1998	1.2	2
2000	1.3	2
2002	1.4	2
2004	1.5	5
2006	1.6	6
2011	1.7	7
2014 (март)	1.8	8

Рисунок 5. Версии платформы Java.

Среда разработки программ

Для написания программ на языке Java достаточно использовать самый простой текстовый редактор, однако применение специализированных средств разработки IDE (Eclipse, IntelliJ IDEA, NetBeans) предоставляет большой набор полезных и удобных функций. Существует несколько способов компиляции и запуска на выполнение программ, написанных на языке Java: из командной строки или из другой программы, например, интегрированной среды разработки. Для компиляции программы из командной строки необходимо вызвать компилятор, набрав команду `javac` и указав через пробел имена компилируемых файлов:

```
javac MyClass.java
```

При успешном выполнении этапа компиляции в директории с исходными кодами появятся файлы с расширением `.class`, которые являются java байт-кодом. Запустить данную программу на выполнение можно выполнив команду:

```
java MyClass
```

Структура программы

Программа на языке Java должна состоять, как минимум из одного, объявления класса (ключевое слово `class`) и может содержать:

- Директивы `import` и `package`.
- Объявления и инициализации атрибутов класса.

- Объявления методов.
- Комментариев.

Исходный код программы хранится в текстовом файле с расширением *.java, а имя файла соответствует названию класса т.е. для класса MyClass файл будет называться MyClass.java. В одном файле может существовать только один публичный (обозначенный модификатором public) класс и его название должно соответствовать имени файла. Код класса ограничен парой фигурных скобок {}. Если вы создаете исполняемую программу, то этот класс должен иметь метод **main()**, с которого начинается выполнение программы. Его декларация должна всегда выглядеть следующим образом:

```
public static void main(String[] args) {
    /* код программы */
}
```

Классы, не имеющие метода main(), не могут быть запущены на выполнение как самостоятельные программы, однако могут быть скомпилированы и подключены к другим программам на языке Java.

Комментарии

- Строчные


```
int a=1; // Это строчный комментарий
int b=2; // Это тоже строчный комментарий
```
- Блочные (многострочные)


```
int a=1; /* Это
блочный
комментарий*/ int b=2;
```
- Документация


```
/**
 * Главный метод программы
 * Выводит приветствие пользователю
 * @param args Аргументы командной строки
 */
public static void main(String [] args) {
    System.out.println("Hello, world");
}
```

Типы данных

Java является строго типизированным языком. Это означает, что любая переменная и любое выражение имеют известный тип еще на момент компиляции. Такое строгое правило позволяет выявлять многие ошибки уже во время компиляции. Компилятор, найдя ошибку,

указывает точную строку и причину ее возникновения, а динамические "баги" необходимо сначала выявить тестированием, а затем найти место в коде, которое их породило.

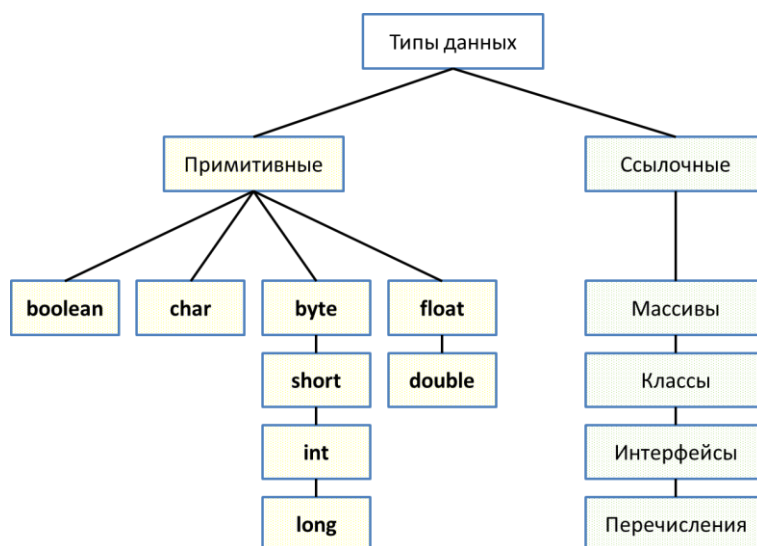


Рисунок 5. Типы данных в Java.

Тип	Описание	По умолчанию
byte	8-битное знаковое целое число. Значения от -128 до 127 (включительно).	0
short	16-битное знаковое целое число. Значения от -32768 до 32767 (включительно).	0
int	32-битное знаковое целое число. Значения от -2147483648 до 2147483647 (включительно)	0
long	64-битное знаковое целое число. Значения от -9223372036854775808 до 9223372036854775807 (включительно)	0L
float	32-битное число с плавающей точкой, соответствующее IEEE 754.	0.0f
double	64-битное число с плавающей точкой двойной точности, соответствующее IEEE 754.	0.0
boolean	Логический тип данных с двумя возможными значениями: true и false	false
char	16-битный символ Unicode	'\u0000'

Значение ссылочной переменной может быть null или ссылаться на другой объект.

Массив – это контейнерный объект, содержащий фиксированное количество значений одного типа. Длина массива устанавливается при его создании. После создания длину массива изменить нельзя. Элемент массива – это одно из значений в массиве, к которому можно обратиться, используя его позицию внутри массива - индекс элемента. Нумерация элементов начинается с 0.

- Неинициализированная ссылка на массив

```
int[] array;
```

- Создание массива

```
int[] array = new int[10];
```

- Доступ к элементам

```
array[0] = 100;
```

- Инициализация при создании

```
int[] array = {100, 200, 300, 400, 500, 600};
```

- Длина массива

```
array.length;
```

- Массивы могут быть многомерными

```
String[][] names = {  
    {"Mr. ", "Mrs. ", "Ms. "},  
    {"Smith", "Jones"},  
};
```

- Копирование массивов – это стандартная операция

```
public static void arraycopy(Object src, int srcIndex, Object dest, int  
destIndex, int length)
```

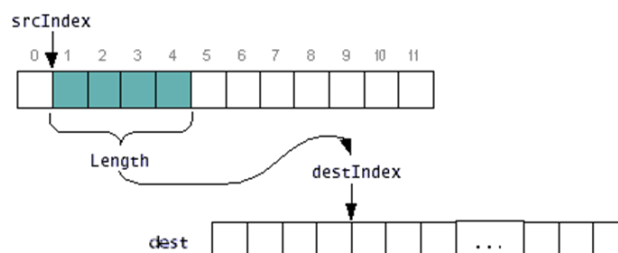


Рисунок 6. Копирование массива.

Строки

`java.lang.String` – класс, описывающий строку. Объекты класса `String` являются неизменяемыми.

Объявление и создание строки

```
String s1 = "Привет, мир";  
char[] arr = {'П', 'р', 'и', 'в', 'е', 'т'};  
s2 = new String(arr);
```

Определение длины строки

```
int len = s1.length();
```

Конкатенация строк

```
String s3 = "Привет";  
String s4 = "мир";  
String s5 = s3.concat(" ").concat(s4);  
String s6 = s3+" "+s4;
```

Сравнение строк

```
if (s5.equals(s6)) {}
```

String substring(int ind1, int ind2) String substring(int beginIndex)	Возвращает подстроку между заданными номерами символов / начиная с заданного номера символа
String trim()	Возвращает строку без лидирующих и завершающих пробелов
String format(String f, [a, b,...])	Возвращает форматированную строку
String toLowerCase() String toUpperCase()	Возвращает строку в нижнем/верхнем регистре
int indexOf(int ch) int lastIndexOf(int ch) int indexOf(String str) int lastIndexOf(String str)	Возвращает индекс первого/последнего вхождения в строку символа ch / подстроки str
boolean contains(CharSequence s)	Проверяет, содержит ли строка заданную последовательность символов
String replace(char oldC, char newC) String replace(CharSequence oldC, CharSequence newC)	Возвращает строку, в которой произведена замена символа/последовательности символов oldC на newC
boolean equalsIgnoreCase(String s) int compareToIgnoreCase(String str)	Сравнивает строки без учета регистра

Класс Math

Класс java.lang.Math содержит только статические методы для физических и технических расчетов, а также константы E и PI.

Все методы класса вызываются без создания экземпляра класса (создать экземпляр класса Math невозможно). В классе определено большое количество методов для математических вычислений, а также ряд других полезных методов, таких как floor(), ceil(), rint(), round(), max(), min(), которые выполняют задачи по округлению, поиску экстремальных значений, нахождению ближайшего целого и т.д.

double abs(double d) float abs(float f) int abs(int i) long abs(long lng)	Взятие модуля числа
double ceil(double d)	Округление в большую сторону
double floor(double d)	Округление в меньшую сторону
double rint(double d) long round(double d) int round(float f)	Округление до ближайшего целого

double min(double a, double b) float min(float a, float b) int min(int a, int b) long min(long a, long b)	Определение минимального значения
double max(double a, double b) float max(float a, float b) int max(int a, int b) long max(long a, long b)	Определение максимального значения
double exp(double d)	Функция экспоненты
double log(double d)	Натуральный логарифм
double pow(double a, double b)	Возведение числа a в степень b
double sqrt(double d)	Квадратный корень
double sin(double d)	Синус
double cos(double d)	Косинус
double tan(double d)	Тангенс
double asin(double d)	Арксинус
double acos(double d)	Арккосинус
double atan(double d)	Арктангенс
double toDegrees(double d)	Преобразование радиан в градусы
double toRadians(double d)	Преобразование градусов в радианы
double random()	Генерация случайной величины в диапазоне [0;1)

Переменные

Имя переменной – последовательность из латинских букв и цифр, начинающаяся с буквы. Могут также встречаться подчеркивания, знак \$, буквы других алфавитов. Имя переменной чувствительно к регистру и не должно совпадать с ключевыми словами.

Рекомендации по именованию переменных:

- если значение будет меняться:
 - имя начинается с маленькой буквы;
 - если имя состоит из нескольких слов, второе и последующие слова начинаются с большой буквы, например myVar.
- если значение не будет меняться (именованные константы):
 - имя записывается в верхнем регистре;
 - слова отделяются друг от друга символом "_".

Операции

Арифметические операции

Пример	Название	Результат
$a + b$	Сложение/конкатенация	Сумма a и b . Также складывает строки
$a - b$	Вычитание	Разность a и b .
$a * b$	Умножение	Произведение a и b .
a / b	Деление	Частное от деления a на b .
$a \% b$	Остаток от деления	Целочисленный остаток от деления a на b .

Операции присвоения

Пример	Название	Результат
$a += b$	Сложение/конкатенация	Сумма a и b . Также складывает строки
$a -= b$	Вычитание	Разность a и b .
$a *= b$	Умножение	Произведение a и b .
$a /= b$	Деление	Частное от деления a на b .
$a \% = b$	Остаток от деления	Остаток от деления числа a на b .

Операции сравнения

Пример	Название	Результат
$a == b$	равно	TRUE , если a равно b .
$a != b$	не равно	TRUE , если a не равно b .
$a < b$	меньше	TRUE , если a строго меньше b .
$a > b$	больше	TRUE , если a строго больше b .
$a <= b$	меньше или равно	TRUE , если a меньше или равно b .
$a >= b$	больше или равно	TRUE , если a больше или равно b .

Операции инкремента/декремента

Пример	Название	Эффект
$++a$	Pre-increment	Увеличивает a на 1, затем возвращает a .
$a++$	Post-increment	Возвращает a , затем увеличивает a на 1.
$--a$	Pre-decrement	Уменьшает a на 1, затем возвращает a .
$a--$	Post-decrement	Возвращает a , затем уменьшает a на 1.

Логические операции

Пример	Имя	Результат
$! a$	Not	TRUE , если a не TRUE .
$a \&\& b$	And	TRUE , если и a , и b TRUE .
$a \parallel b$	Or	TRUE , если a или b TRUE .

Тернарная операция

Выражение вычисляется в оператор1, если условие равно **TRUE**, и в оператор2, если условие равно **FALSE**.

`var = условие ? оператор1 : оператор2;`

Условный оператор

```
if (условие) {  
    операторы  
} else {  
    операторы  
}
```

Циклы

С предусловием

```
while (условие) {  
    операторы  
}
```

С постусловием

```
do {  
    операторы  
} while (условие);
```

С параметром

```
for (инициализация; условие; приращение) { операторы }
```

Итерационный

```
for(переменная : массив_или_объект) { операторы }
```

Оператор break

Заканчивает выполнение текущей структуры for, while, do..while или switch.

Оператор continue

Используется в циклических структурах для пропуска оставшейся части текущей итерации цикла и продолжения выполнения с начала следующей итерации (прохода) цикла.

Оператор switch

Напоминает выполнение серии операторов IF над одним выражением. Во многих случаях вам нужно будет сравнить одну переменную (или выражение) с различными значениями и выполнить разные участки кода в зависимости от того, каково значение

```
switch (выражение) {  
    case значение1:  
        statement  
        break;  
    case значение2:  
        statement  
        break;  
    case значение3:  
        statement
```

```
        break;
    }
```

Оператор return

Если вызывается из функции, оператор **return** немедленно заканчивает выполнение текущей функции и возвращает свой аргумент как значение вызова функции.

Простой консольный ввод-вывод

`System.out` - объект, связанный со стандартным потоком вывода (обычно консолью).

`System.err` - объект, связанный со стандартным потоком вывода ошибок (обычно той же консолью).

`System.in` - объект, связанный со стандартным потоком ввода (обычно клавиатурой).

`Scanner` - удобная обертка над потоком ввода, позволяющая интерпретировать вводимые данные, как целые числа, вещественные числа, строки и т.п.

После использования объект `Scanner` надо не забыть закрыть.

```
public static final void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int i = scanner.nextInt();
    String text = scanner.next();
    System.out.println("Integer: " + i);
    System.out.println("String: " + text);
    System.err.println("This is error stream");
    scanner.close();
}
```

Пакеты

Программа на Java представляет собой набор пакетов (packages). Каждый пакет может включать вложенные пакеты, а так же может содержать классы и интерфейсы. Каждый пакет имеет свое пространство имен, что позволяет создавать одноименные классы в различных пакетах. Имена бывают простыми (simple), состоящими из одного идентификатора, и составными (qualified), состоящими из последовательности идентификаторов, разделенных точкой. Составное имя любого элемента пакета составляется из составного имени этого пакета и простого имени элемента. Простейшим способом организации пакетов и типов является обычная файловая структура. Таким образом разделяется логическая структура проекта. Например, исходный код класса `space.sunsystem.Moon` хранится в файле `space\sunsystem\Moon.java`. Запуск программы на JAVA стоит производить из директории, в которой содержатся пакеты. Было бы ошибкой запускать Java прямо из папки `space\sunsystem` и пытаться обращаться к классу `Moon`, несмотря на то, что файл-описание лежит именно в

ней. Необходимо подняться на два уровня директорий выше, чтобы Java, построив путь из имени пакета, смогла обнаружить нужный файл.

При использовании классов перед именем класса через точку надо добавлять полное имя пакета, к которому относится данный класс. При создании пакета всегда следует руководствоваться простым правилом: называть его именем простым, но отражающим смысл, логику поведения и функциональность объединенных в нем классов.

В реальных проектах пакеты часто именуются следующим образом:

- обратный интернет-адрес производителя программного обеспечения, а именно для bstu.ru получится ru.bstu;
- далее следует имя проекта, например: it31;
- затем располагаются пакеты, определяющие собственно приложение.

Вновь созданные пакеты именуются при помощи команды package.

```
package ru.bstu.it31.petrov.lab1;
```

Подключение пакета осуществляется командой import.

```
import ru.bstu.it31.petrov.lab2;
```

Подробная документация находится по адресу – <http://docs.oracle.com/javase/8/docs/api/>

Пример простой программы

```
import java.util.Scanner;
public class HelloWorld {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int i = scanner.nextInt();
        String text = scanner.next();
        System.out.println("Hello, World!!!");
        System.out.println("Integer: " + i);
        System.out.println("String: " + text);
        System.err.println("This is error stream");
        scanner.close();
    }
}
```

Задание к лабораторной работе:

1. Скачать и установить JDK
(<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>)
2. Изучить файловый ввод и вывод в Java.
3. Разработать программы для решения заданий по вариантам.
4. Для решения каждого задания необходимо создать отдельный метод с осмысленным названием. Методы могут быть как в одном классе, так и в нескольких.
5. Все параметры, которые есть в задании, должны считываться как с консоли, так и из файла.

Варианты заданий

1. Условные выражения

№	Задание
1	Даны действительные числа a, b, c . Удвоить эти числа, если $a > b > c$, и заменить их абсолютными значениями, если это не так
2	Даны три положительных числа a, b, c . Проверить, будут ли они сторонами треугольника. Если да, то вычислить площадь этого треугольника.
3	Заданы размеры A, B прямоугольного отверстия и размеры x, y, z кирпича. Определить, пройдет ли кирпич через отверстие.
4	Дано число x . Напечатать в порядке возрастания числа: $\sin x, \cos x, \ln x$. Если при каком-либо x некоторые из выражений не имеют смысла, вывести сообщение об этом и сравнивать значения только тех, которые имеют смысл
5	Даны три числа, определить является ли сумма каких-либо двух из них положительным числом.
6	Определить взаимное расположение точки с координатами (x_0, y_0) и окружности радиуса R с центром в точке (x_1, y_1)
7	Если сумма трех различных чисел X, Y, Z меньше единицы, то наименьшее из этих трех чисел заменить полусуммой двух других; в противном случае заменить меньшее из X, Y полусуммой двух оставшихся значений.
8	Даны три положительных числа. Определить, можно ли построить треугольник со сторонами, длины которых равны этим числам. Если возможно, то найти его площадь.
9	На оси OXY расположены три точки a, b, c . Определить, какая из точек b или c расположена ближе к a .
10	Дан круг радиуса R . Определить, поместится ли правильный треугольник со стороной a в этом круге.

2. Оператор выбора

№	Задание
1	Написать программу, которая по номеру дня недели (целому числу от 1 до 7) выдает количество пар в вашей группе.
2	Составить программу, которая по заданному году и номеру месяца m определяет количество дней в этом месяце.
3	Составить программу, которая по данному числу (1—12) выводит название соответствующего ему месяца.
4	Написать программу, которая по данному номеру времени года (1 — зима, 2 — весна, 3 — лето, 4 — осень) выдавала соответствующие этому времени года месяцы, количество дней в каждом из месяцев.
5	Написать программу, которая бы по заданному номеру единицы измерения (1 — дециметр, 2 — километр, 3 — метр, 4 — миллиметр, 5 — сантиметр) и длине отрезка L выдавала бы соответствующее значение длины отрезка в метрах.
6	Написать программу, которая по введенному числу от 1 до 12 (номеру месяца) выдает все приходящиеся на этот месяц праздничные дни (например, если введено число 1, то должно получиться 1 января — Новый год, 7 января — Рождество)
7	Даны два действительных положительных числа x, y . Арифметические действия над числами пронумерованы (1 — сложение, 2 — вычитание, 3 — умножение, 4 — деление). Составить программу, которая по введенному номеру выполняет то или иное действие над числами.
8	Вычислить номер дня в невисокосном году по заданным числу и месяцу.

9	Для каждой введенной цифры (0 — 9) вывести соответствующее ей название на английском языке (0 — <i>zero</i> , 1 — <i>one</i> , 2 — <i>two</i> ,...).
10	Составить программу, позволяющую получить словесное описание школьных отметок (1 — «плохо», 2 — «неудовлетворительно», 3-«удовлетворительно», 4— «хорошо», 5 — «отлично»).

3. Циклы - данное задание необходимо выполнить 2-мя способами, с помощью цикла с параметрами (for) и с помощью цикла с условием (while)

1	Составить программу для проверки утверждения: «Результатами вычислений по формуле $x^2 + x + 17$ при $0 < x < 15$ являются простые числа». Все результаты вывести на экран.
2	Начав тренировки, спортсмен в первый день пробежал n км. Каждый день он увеличивал дневную норму на $m\%$ нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за k дней?
3	Составить программу, которая печатает таблицу умножения и сложения натуральных чисел в десятичной системе счисления. В качестве параметра принять размер таблицы n т. е. при $n=2$ должна быть таблица следующего вида: $1 \times 1 = 1$ $2 \times 1 = 2$ $1 \times 2 = 2$ $2 \times 2 = 4$
4	Напечатать в возрастающем порядке все трехзначные числа не превышающие n , в десятичной записи которых нет одинаковых цифр (операции деления и нахождения остатка от деления не использовать).
5	Составить программу, которая запрашивает пароль (например, четырехзначное число) до тех пор, пока он не будет правильно введен или пока количество попыток не превысит n
6	Составить программу, которая среди всех трехзначных положительных чисел находит наибольшее значение отношения числа к сумме его цифр.
7	Найти сумму всех n -значных чисел, кратных k ($1 < n < 4$).
8	Составить программу для вычисления значений функции $F(x)$ на отрезке $[a, b]$ с шагом n . Результат представить в виде таблицы, первый столбец которой — значения аргумента, второй — соответствующие значения функции $F(x) = x \sin x$
9	Составить программу для вычисления значений функции $F(x)$ на отрезке $[a, b]$ с шагом n . Результат представить в виде таблицы, первый столбец которой — значения аргумента, второй — соответствующие значения функции $F(x) = \sin x + \lg x$
10	Составить программу для вычисления значений функции $F(x)$ на отрезке $[a, b]$ с шагом n . Результат представить в виде таблицы, первый столбец которой — значения аргумента, второй — соответствующие значения функции $F(x) = \sin x + 0,5 \cos x$

4. Массивы

1	В массив $A[N]$ занесены натуральные числа. Найти сумму тех элементов, которые кратны данному K .
2	Дана последовательность целых чисел a_1, a_2, \dots, a_n . Выяснить, какое число встречается раньше — положительное или отрицательное.
3	Дан массив действительных чисел, размерность которого N . Подсчитать, сколько в нем отрицательных, положительных и нулевых

	элементов.
4	В заданном одномерном массиве поменять местами соседние элементы, стоящие на четных местах, с элементами, стоящими на нечетных местах. Пример: Пусть дан массив $A=\{1,2,3,4,5,6\}$, на выходе должен быть массив $B=\{2,1,4,3,6,5\}$
5	Задана последовательность из N вещественных чисел. Определить, сколько среди них чисел меньших K , равных K и больших K
6	Задан массив чисел. Определить, сколько раз меняется знак в данной последовательности чисел, запомнить номера позиций, в которых происходит смена знака. Пример: Дан массив $A=\{1,2,-3,-4,-5,6,-7,8,9,10\}$ Ответ: Знак меняется 4 раза. Номера позиций в которых происходит смена знака $\{3,6,7,8\}$
7	Дан массив действительных чисел. Найти его первый максимальный элемент и заменить его нулем.
8	В массиве целых чисел с количеством элементов n найти наиболее часто встречающееся число. Если таких чисел несколько, то определить наименьшее из них.
9	Задан массив с количеством элементов N . Сформировать два массива: в первый включить элементы исходного массива с четными номерами, а во второй — с нечетными.
10	Дана последовательность n различных целых чисел. Найти сумму ее членов, расположенных между максимальным и минимальным значениями (в сумму включить и оба этих числа).

Каждый отчет должен содержать:

1. Заголовок лабораторной работы (название и цель работы).
2. Фамилия, инициалы и группа студента.
3. Задание к лабораторной работе.
4. Краткие теоретические сведения.
5. Описание алгоритмов, функций, примененных решений.
6. Результаты выполнения программ.
7. Исходный код программ.
8. Выводы о проделанной работе.