

ЛАБОРАТОРНАЯ РАБОТА № 1

«Знакомство с языком программирования Java»

Цель: получить представление о написании программ на языке программирования Java с использованием командной строки.

Учебные вопросы:

1. Установка необходимого ПО;
 - а) Установка JDK (Java Development Kit – Комплект разработчика приложений на Java);
2. Компиляция и запуск java программ из командной строки;
3. Задания для самостоятельной работы;
4. Описание результата выполнения лабораторной работы;

1. УСТАНОВКА НЕОБХОДИМОГО ПО

Для того, чтобы была возможность писать программы на языке программирования java из командной строки, а также с помощью интегрированной среды разработки необходимо скачать Java Development Kit (Комплект разработчика приложений на Java) с сайта <https://www.oracle.com/index.html>.

После того, как вы скачаете файл необходимо его установить, после установки для того, чтобы существовала возможность вызова из командной строки компилятора **javac** (**Javac** — компилятор языка java) необходимо добавить путь до JDK в переменные среды Path.

2. КОМПИЛЯЦИЯ И ЗАПУСК JAVA ПРОГРАММ ИЗ КОМАНДНОЙ СТРОКИ

2.1 ВЫВОД НА КОНСОЛЬ

Для компиляции программы в командной строке необходимо вызвать командную строку от имени администратора. Как показано на рисунке 1.

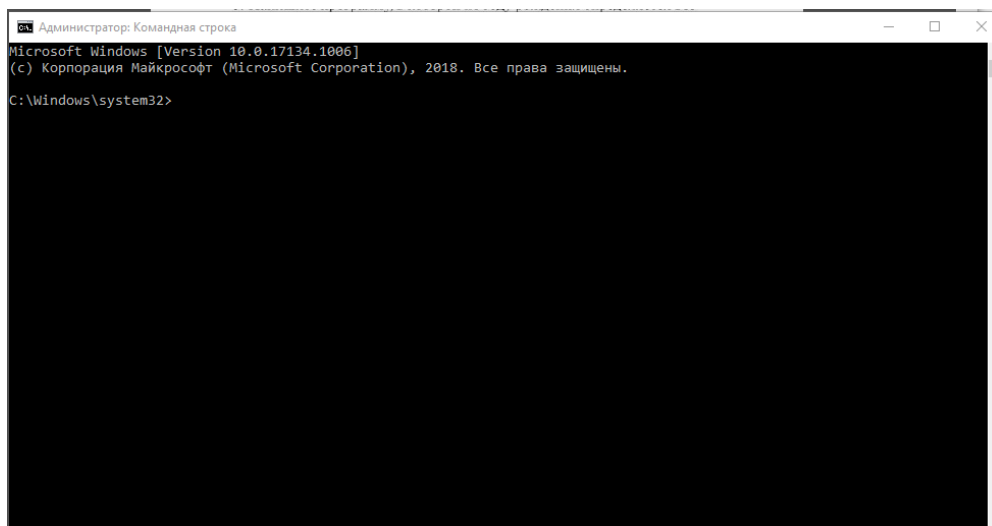


Рисунок 1. Командная строка

Затем необходимо создать в папку, в которой будут храниться текстовые файлы с текстом программы. Путь к файлам не должен содержать каталогов с наименованием на кириллице. Например, создайте папку в корне диска C. Назовите её «*java_ex*», *ex* – сокращенное от *examples*. Введите в командную строку команду *cd C:\java_ex*. Как показано на рисунке 2.

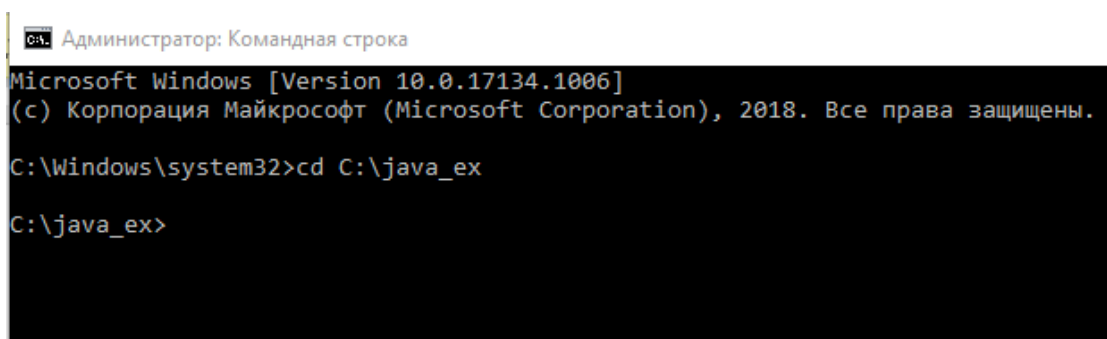


Рисунок 2. Результат выполнения команды *cd C:\java_ex*

После выполнения команды, необходимо создать текстовый файл «Блокнот», расширение ***.txt**, с наименованием `example1`, измените расширение файла с `txt` на `java`, скопируйте в него код, приведенный на листинге 1.

Листинг 1. Код программы «example1»

```
class example1
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

Затем вернитесь к командной строке введите в нее следующую команду **javac example1.java**, где **javac** это обращение к компилятору, а `example1.java` это имя созданного вами текстового файла. Обратите внимание, в каталоге **C:\java_ex** появился скомпилированный файл «example1.class». Теперь чтобы запустить выполнение программы необходимо в командной строке ввести команду **java example1**.

Результат показан на рисунке 3.

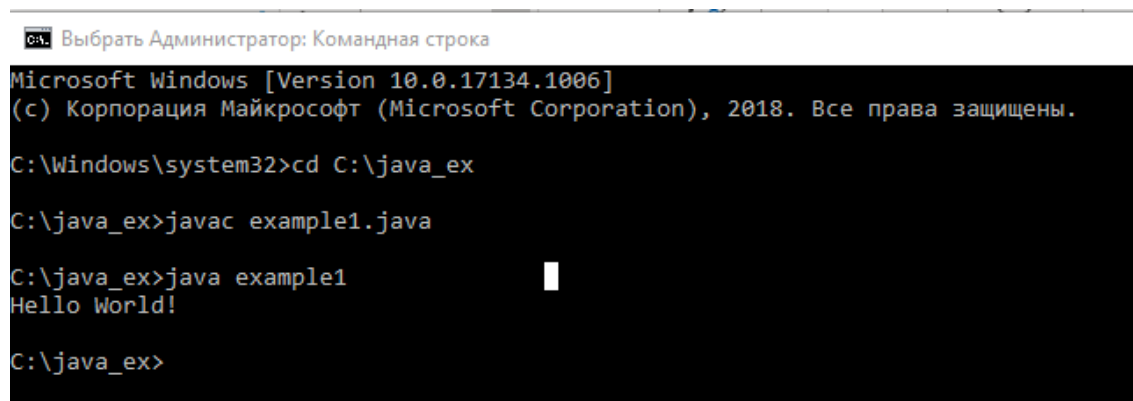


Рисунок 3. Результат выполнения программы example1

На листинге 2 представлена программа в которой указанное число умножается на 2.

Листинг 2. Код программы «example2»

```
class example2{
    public static void main(String args[])
```

```

{
int num; // в этой строке кода объявляется
// переменная с именем num
num = 100; // в этой строке кода переменной num
// присваивается значение 100
System.out.println("num: " + num);
num = num * 2;
System.out.print("Znachenie num * 2 равно ");
System.out.println(num);
}
}

```

На листинге 3 показан пример программы с использованием условного оператора if.

Листинг 3. Код программы «example3»

```

class example3
{
public static void main(String args[])
{
int S, D;
S = 10;
D = 20;
if(S < D) System.out.println("S < D");
S = S * 2;
if(S == D) System.out.println("S = D");
S = S * D;
if(S > D) System.out.println("S > D");

}
}

```

2.2 ВВОД С КОНСОЛИ

Для получения ввода с консоли в классе **System** определен объект **In**. Однако непосредственно через объект **System.In** не очень удобно работать, поэтому, как правило, используют класс **Scanner**, который, в свою очередь использует **System.in**. Пример программы, осуществляющей ввод чисел показан на листинге 4.

Листинг 4. Код программы «example4»

```
import java.util.Scanner;

public class example4{

    public static void main(String[] args) {

        Scanner InCMD = new Scanner(System.in);
        System.out.print("Input a number:");
        int num = InCMD.nextInt();

        System.out.printf("Your number: %d \n", num);
        InCMD.close();
    }
}
```

Так как класс **Scanner** находится в пакете **java.util**, то мы вначале его импортируем с помощью инструкции **import java.util.Scanner**.

Для создания самого объекта **Scanner** в его конструктор передается объект **System.in**. После этого мы можем получать вводимые значения. Например, в данном случае вначале выводим приглашение к вводу и затем получаем вводимое число в переменную **num**.

Чтобы получить введенное число, используется метод **InCMD.nextInt()**, который возвращает введенное с клавиатуры целочисленное значение. На рисунке показан пример работы программы.

```
C:\java_ex>java example4
Input a number:1
Your number: 1
```

Рисунок 4. Пример работы программы «example4»

Класс Scanner имеет еще ряд методов, которые позволяют получить введенные пользователем значения:

next(): считывает введенную строку до первого пробела

nextLine(): считывает всю введенную строку

nextInt(): считывает введенное число int

nextDouble(): считывает введенное число double

nextBoolean(): считывает значение boolean

nextByte(): считывает введенное число byte

nextFloat(): считывает введенное число float

nextShort(): считывает введенное число short

То есть для ввода значений каждого примитивного типа в классе Scanner определен свой метод.

Например, создадим программу для ввода информации о человеке показанной на листинге 5.

Листинг 5. Код программы «example5»

```
import java.util.Scanner;

public class example5{

    public static void main(String[] args) {

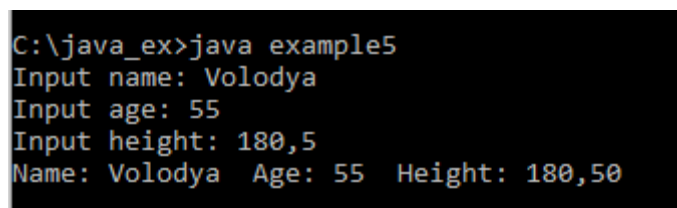
        Scanner in = new Scanner(System.in);
        System.out.print("Input name: ");
```

```

        String name = in.nextLine();
        System.out.print("Input age: ");
        int age = in.nextInt();
        System.out.print("Input height: ");
        float height = in.nextFloat();
        System.out.printf("Name: %s   Age: %d   Height:
%.2f \n", name, age, height);
        in.close();
    }
}

```

Здесь последовательно вводятся данные типов **string**, **int**, **float** и потом все введенные данные вместе выводятся на консоль. На рисунке показан пример работы программы:



```

C:\java_ex>java example5
Input name: Volodya
Input age: 55
Input height: 180,5
Name: Volodya   Age: 55   Height: 180,50

```

Рисунок 5. Пример работы программы

Обратите внимание для ввода значения типа **float** (то же самое относится к типу **double**) применяется число "180,5", где разделителем является запятая, а не "180.5", где разделителем является точка. В данном случае все зависит от текущей языковой локализации системы. В моем случае русскоязычная локализация, соответственно вводить необходимо числа, где разделителем является запятая. То же самое касается многих других локализаций, например, немецкой, французской и т.д., где применяется запятая.

Далее рассмотрим пример программы для расчета гипотенузы, показанной на листинге 6.

Листинг 6. Код программы «example6»

```

class example6

```

```

{
    static double a =10.0, b=4.0, c;
    public static double hyp(){
        return c = Math.sqrt(a*a + b*b);
    }
    public static void main(String[] args) {
        System.out.println("katet a=" + a);
        System.out.println("katet b=" + b);
        System.out.println("hypotenuse c=" + hyp());
    }
}

```

В листинге 6 использованы методы класса `Math`, предназначенного для выполнения различных математических операций. Описание класса вы можете найти по ссылке <https://metanit.com/java/tutorial/12.1.php>.

На листинге 7 представлена программа расчета радиуса круга.

Листинг 7. Код программы «example7»

```

import java.util.Scanner;

public class example7{

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        System.out.print("Radius kruga: ");
        int radius = in.nextInt();
        long area = Math.round(Math.PI * Math.pow(radius, 2));
        System.out.printf("S kruga s R %d = %d \n", radius,
area);
    }
}

```


}

3. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Напишите программу, в которой Пользователь вводит сначала фамилию, затем имя, затем отчество. После ввода программа выводит сообщение «Hallo <ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО>».
2. Напишите программу, в которой Пользователь вводит имя и возраст. Программа отображает сообщение об имени и возрасте пользователя.
3. Напишите программу, в которой Пользователь последовательно вводит название текущего дня недели, название месяца и дату (номер дня в месяце). Программа выводит сообщение о сегодняшней дате (день недели, дата, месяц).
4. Напишите программу, в которой пользователю предлагается ввести название месяца и количество дней в этом месяце. Программа выводит сообщение о том, что соответствующий месяц содержит указанное количество дней.
5. Напишите программу, в которой по году рождения определяется возраст пользователя.
6. Напишите программу, в которой Пользователь вводит имя и год рождения, в программе отображается сообщение содержащее имя пользователя и его возраст.
7. Напишите программу, в которой по возрасту определяется год рождения.
8. Напишите программу для вычисления суммы двух чисел. Оба числа вводятся пользователем. Для вычисления суммы используйте оператор +.
9. Напишите программу, в которой пользователь вводит число, а программой отображается последовательность из четырех чисел: число, на единицу меньше введенного, введенное число и число, на единицу больше введенного. Четвертое число должно быть квадратом суммы первых трех чисел.
10. Напишите программу, в которой Пользователь вводит два числа, а программой вычисляется и отображается сумма и разность этих чисел.
11. Добавьте в пример расчета гипотенузы (см. раздел 2) метод, вычисляющий a^b . Используйте для этого функции расчета натурального логарифма и экспоненты ($y = \exp(b * \log(a))$);
12. Используйте новый метод для расчета гипотенузы. Сделайте вариант метода `hyp()` с параметрами.
13. Сделайте вариант вычисления a^b с помощью встроенного метода (искать в Math).

4. ОПИСАНИЕ РЕЗУЛЬТАТА ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

В отчете по лабораторной работе должны быть представлены 20 листингов программ, 7 программ из 2 части лабораторной работы и 13 выполненных самостоятельно. Также необходимо в один архив вложить 20 файлов с расширением .class. Наименование файлов должно состоять из слова «example+_+номер студента в списке группы+_+ номер задания», например пятое задание студента с порядковым номером 7 будет называться «example07_05», а двадцатое задание этого студента будет «example07_20».

Лабораторная работа принимается при наличии отчета и двадцати, верно, выполненных программ, и наличии архива с двадцатью текстовыми файлами программ с расширением class. У каждого студента будут проверены выборочно программы (3-5 штук).

Структура отчета по лабораторной работе:

1. Титульный лист;
2. Цель работы;
3. Описание задачи;
4. Ход выполнения (содержит код программы);
5. Вывод;

Оформление:

- а) шрифт Times New Roman;
- б) размер шрифта 12 или 14;
- в) межстрочный интервал 1,5.

Отчет выполняется индивидуально и направляется по адресу электронной почты proverkalab@yandex.ru. В теле письма необходимо указать ФИО студента и номер группы.