

Лабораторная работа по №1

Выполнила: Качкуркина Арина Валерьевна

6204-010302D

2025 г.

Задание 1.

Я запустила компилятор `javac` и ознакомилась с форматом задания. Так-же запустила программу `java` без параметров и ознакомилась с форматом задания параметров запуска виртуальной машины.

Задание 2.

Изначально создала файл через командную строку `MyFirstProgram.java` и поместила код одного пустого класса с именем `MyFirstClass` по заданию:

```
class MyFirstClass {  
  
    public static void main(String[] s) {  
  
        System.out.println("Hello world!!!");  
  
    }  
  
}
```

Скомпилировала командой: `javac MyFirstProgram.java`

Запустила программу: `java MyFirstClass`

Система выдала сообщение об ошибке, указывающее на отсутствие главного метода в классе. Был добавлен метод `main`:

```
class MyFirstClass {  
  
    void main(String[] s) {  
  
        System.out.println("Hello world!!!");  
  
    }  
  
}
```

Повторная компиляция и запуск снова привели к ошибке. На этот раз система сообщила, что метод `main` должен быть статическим. Я добавила `static` к методу и повторной компиляции программа работала корректно

Программа успешно вывела в консоль фразу *"Hello world!!!"*.

Задание 3.

Я изменила метод main для работы с аргументами командной строки:

```
class MyFirstClass {  
    public static void main(String[] s) {  
        for (int i = 0; i < s.length; i++)  
            System.out.println(s[i]);  
    }  
}
```

Запустила с тестовыми аргументами:

```
java MyFirstClass cat dog edgebug cat1 dog1
```

Аргументы вывелись в столбик, каждый с новой строки:

cat

dog

edgebug

cat1

dog1

Задание 4.

Добавила в файл второй класс MySecondClass, который работает с двумя числами. Класс включает:

-два приватных поля oneNum и twoNum;

- методы для изменения значений methodOne и methodTwo;
- метод actionOnNumbers, который возвращает произведение чисел.

Далее я изменила основной метод:

```
class MyFirstClass {  
    public static void main(String[] s) {  
        MySecondClass o = new MySecondClass(0, 0);  
        int i, j;  
        for (i = 1; i <= 8; i++) {  
            for(j = 1; j <= 8; j++) {  
                o.methodOne(i);  
                o.methodTwo(j);  
                System.out.print(o.actionOnNumbers());  
                System.out.print(" ");  
            }  
            System.out.println();  
        }  
    }  
}
```

После компиляции и запуска ,программа вывела таблицу умножения от 1 до 8, каждая строка содержит результаты умножения одного числа на числа от 1 до 8:

1 2 3 4 5 6 7 8

2 4 6 8 10 12 14 16

3 6 9 12 15 18 21 24

4 8 12 16 20 24 28 32
5 10 15 20 25 30 35 40
6 12 18 24 30 36 42 48
7 14 21 28 35 42 49 56
8 16 24 32 40 48 56 64

Задание 5.

В этом задании мне нужно было разнести классы по разным файлам и папкам. Я создала папку myfirstpackage и перенесла в неё класс MySecondClass в файл MySecondClass.java.

Сначала я скомпилировала новый файл:

```
javac myfirstpackage/MySecondClass.java
```

Потом попробовала скомпилировать основной файл, но получила ошибку. Программа не видела класс MySecondClass.

Я добавила строчку *import myfirstpackage.*;* в начало файла, но снова получила ошибку. Оказалось, нужно добавить *package myfirstpackage;* в сам файл MySecondClass.java.

После всех исправлений программа заработала правильно.

Итоговый текст работы программы:

1 2 3 4 5 6 7 8
2 4 6 8 10 12 14 16
3 6 9 12 15 18 21 24
4 8 12 16 20 24 28 32
5 10 15 20 25 30 35 40
6 12 18 24 30 36 42 48

7 14 21 28 35 42 49 56

8 16 24 32 40 48 56 64

Задание 6.

Я запустила и ознакомилась с jar, создала файл manifest.mf и поместила туда следующий код:

Manifest-Version: 1.0

Created-By: Качкуркина Арина

Main-Class: MyFirstClass

Сборка архива jar:

```
jar cfm myfirst.jar manifest.mf MyFirstClass.class  
myfirstpackage/MySecondClass.class
```

Переместила архив в другую папку и запустила:

```
java -jar myfirst.jar
```

Программа успешно выполнилась, показав ту же таблицу умножения:

1 2 3 4 5 6 7 8

2 4 6 8 10 12 14 16

3 6 9 12 15 18 21 24

4 8 12 16 20 24 28 32

5 10 15 20 25 30 35 40

6 12 18 24 30 36 42 48

7 14 21 28 35 42 49 56

8 16 24 32 40 48 56 64

