

Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное  
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет  
имени академика С.П. Королева»

Институт информатики и кибернетики

Кафедра технической кибернетики

Отчет по лабораторной работе №1

Дисциплина: «ООП»

Тема «Пакеты»

Выполнил: Гонеев Роман Олегович

Группа: 6102-120303D

Самара, 2025

## 1 Задание

1. `javac -help` #Посмотрел параметры компилятора
2. `java -help` #Смотрю параметры виртуальной машины

## 2 Задание

Создаю Java-программу в папке репозитория, в программе создаю сначала пустой класс, компилирую, получаю ошибку, что нет точки входа, изменяю код, добавляя метод `main`, компилирую.

Порядок команд в терминали и их результат:

1. `cd Git_Rep` #Перехожу в папку репозитория
2. `mkdir Task2` #Создаю папку для второго задания
3. `cd Task2`
4. `nano MyFirstProgram.java` #Создаю файл программы  
`class MyFirstClass {` #Создаю пустой класс `MyFirstClass`  
`}`
5. `javac MyFirstProgram.java` #Компилирую программу
6. `ls` #Проверяю созданы ли файлы `MyFirstProgram.java` и `MyFirstClass.class`
7. `java MyFirstClass` #Запускаю программу

Выдает ошибку:

Error: Main method not found in class MyFirstClass, please define the main method as:

```
public static void main(String[] args)
```

or a JavaFX application class must extend `javafx.application.Application`

из-за того, что нет точки входа - `main`

8. `nano MyFirstProgram.java` #Открываю файл программы  
Добавляю пустой метод `main`  
`class MyFirstClass {`  
`public static void main(String[] args) {`  
`}`  
`}`
9. `javac MyFirstProgram.java` #Компилирую программу
10. `java MyFirstClass` #Запускаю программу

Ничего не вывелось, значит скомпилированно без ошибок

### 3 Задание

1. `cp ~/Git_Rep/Task2/MyFirstProgram.java ~/Git_Rep/Task3/` #Скопировал программу из папки второго задания в папку третьего

2. `nano MyFirstProgram.java`

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

Добавил `String[] args` массив строк; `args.length`- длина массива; `args[i]`- j

`javac MyFirstProgram.java` #Компилирую программу

`java MyFirstClass` #Запускаю программу без аргументов

`java MyFirstClass arg1 arg2 arg3 arg4 arg5` #Запускаю программу с аргументами (arg1, arg2, arg3, arg4, arg5)

Вывод:

arg1

arg2

arg3

arg4

arg5

### 4 Задание

1. `cp ~/Git_Rep/Task3/MyFirstProgram.java ~/Git_Rep/Task4/` #Скопировал файл программы из папки 3 задания

2. `nano MyFirstProgram.java` #Открыл файл

3. Заменяю метод `MyFirstMain.main()` на новый

4. `nano MyFirstProgram.java` #Исправил ошибки

5. `javac MyFirstProgram.java` #Скомпилировал программу

6. `java MyFirstClass` #Запускаю программу

Вывод в терминале:

```
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
```

Теперь в Task4 3 файла — 2 файла для 2 классов:

MyFirstClass.class MyFirstProgram.java MySecondClass.class

## 5 Задание

1. `cp ~/Git_Rep/Task4/MyFirstProgram.java ~/Git_Rep/Task5/` #Скопировал файл программы из папки 4 задания
2. `~/Git_Rep/Task5` #Перехожу в папку с 5 заданием
3. `mkdir myfirstpackage` #Создаю файл пакета
4. `nano MyFirstProgram.java` #Открываю файл, удаляю код класса MySecondClass и добавляю `import myfirstpackage.*;`
5. `nano myfirstpackage/MyFirstPackage.java` #Создал и открыл файл в пакете
6. Вставил в него код класса MySecondClass с модификатором `public`
7. `javac -d . MyFirstClass.java myfirstpackage/MySecondClass.java` #компилирую оба класса
8. `java MyFirstClass` #Запускаю программу

Вывод:

```
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

## 6 Задание

1. `~/Git_Rep/Task6` #Перехожу в папку 6 задания
2. `cp -r ~/Git_Rep/Task5/myfirstpackage/ ~/Git_Rep/Task6/` #Копирую файлы класса
3. `cp -r ~/Git_Rep/Task5/MyFirstClass.class ~/Git_Rep/Task6/` #Копирую папку myfirstpackage с MySecondClass.class
4. `nano manifest.mf` #Создаю файл manifest.mf и копирую в него код
5. `jar cfm myfirst.jar manifest.mf MyFirstClass.class myfirstpackage/*.class` #Создаю jar-архив
6. `mkdir MyJar` #Создаю папку MyJar
7. `mv myfirst.jar MyJar/` #Перемещаю архив в папку
8. `cd MyJar/` #Перехожу в архив
9. `java -jar myfirst.jar` #Запускаю архив

Вывод:

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