

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

ОТЧЕТ ПО
ЛАБОРАТОРНОЙ РАБОТЕ № 1

**«Ознакомление со структурой Java и
работой пакетов в ней»**

по курсу
Объектно-ориентированное программирование

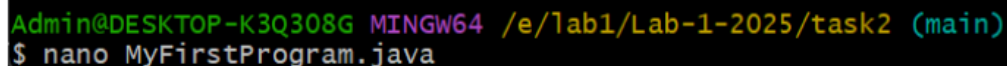
Выполнил: Петрухин Роман,
студент группы 6203-010302D

Оглавление

<u>Задание №2.....</u>	<u>3-4</u>
<u>Задание №3.....</u>	<u>4</u>
<u>Задание №4.....</u>	<u>4-5</u>
<u>Задание №5.....</u>	<u>5-6</u>
<u>Задание №6.....</u>	<u>6-7</u>

Задание №2

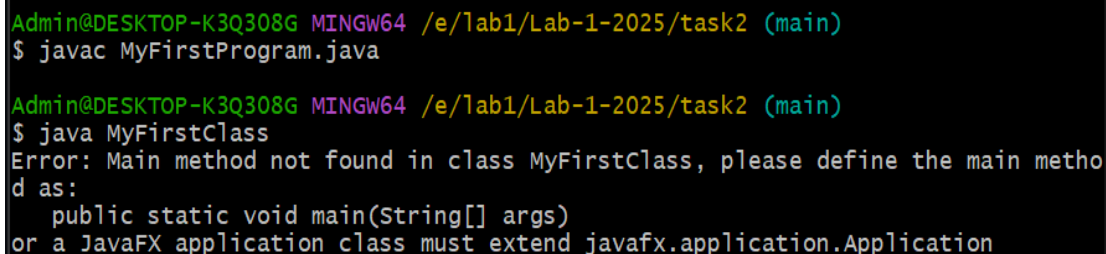
Для того чтобы выполнить данное задание я создал отдельную директорию в своём проекте «Lab-1-2025» под названием «task2», используя команду «mkdir task1», изначально перейдя в нужную мне папку. В ней я создал файл «MyFirstProgram.java» и написал в нём, требуемый заданием код (см. Изображение 1).



```
Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task2 (main)
$ nano MyFirstProgram.java
```

Изображение 1

Далее я скомпилировал файл, используя команду «javac MyFirstProgram.java» и получил ошибку, которая гласила, что в коде отсутствует точка входа - функция main. (см. Изображение 2).



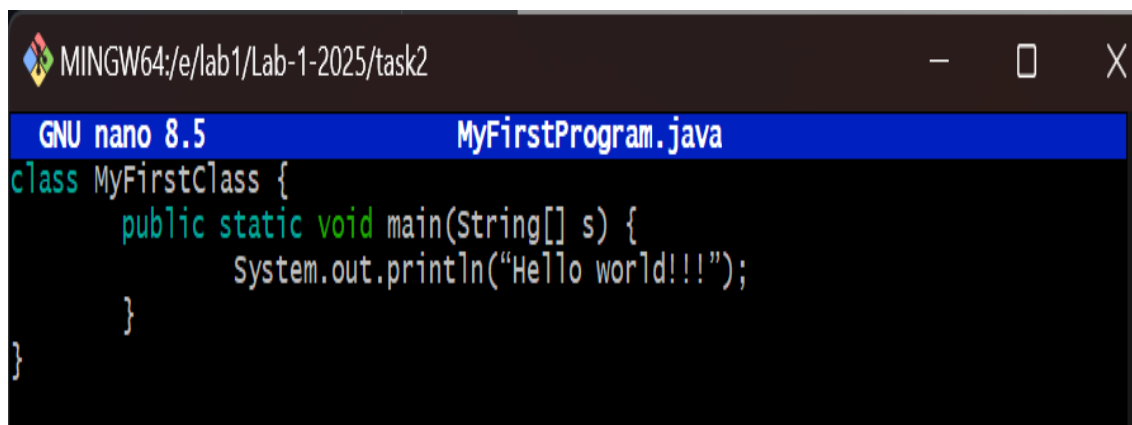
```
Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task2 (main)
$ javac MyFirstProgram.java

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task2 (main)
$ java MyFirstClass
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
```

Изображение 2

После этого я изменил код, добавив метод main(). Скомпилировав его я вновь получил ошибку, которая гласила, что функция main() должна выглядеть так: «public static void main(String[] s) ...».

Я исправил все указанные ошибки (см. Изображение 3) и получил нужный вывод: «Hello world!!!» (см. Изображение 4).



```
MINGW64:/e/lab1/Lab-1-2025/task2
GNU nano 8.5 MyFirstProgram.java
class MyFirstClass {
    public static void main(String[] s) {
        System.out.println("Hello world!!!");
    }
}
```

Изображение 3

```
Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task2 (main)
$ javac MyFirstProgram.java

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task2 (main)
$ java MyFirstClass
Hello world!!!
```

Изображение 4

Задание №3

После выполнения задания №2 я перешел к заданию №3. В соответствии с заданием я изменил уже написанный код, скомпилировал его и запустил с аргументами (см. Изображение 5).

```
Admin@DESKTOP-K3Q308G MINGW64 ~
$ cd E:

Admin@DESKTOP-K3Q308G MINGW64 /e
$ cd lab1

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1
$ cd Lab-1-2025

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025 (main)
$ cd task3

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task3 (main)
$ javac MyFirstProgram.java

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task3 (main)
$ java MyFirstClass arg1 arg2 arg3 arg4 arg5
arg1
arg2
arg3
arg4
arg5
```

Изображение 5

Задание №4

После выполнения задания №3 я перешел к заданию №4. В соответствии с заданием я написал второй класс «MySecondClass» с соответствующим функционалом, в качестве функции для двух чисел я выбрал сложение и назвал её «add». В качестве методов позволяющих извлекать и менять значения я использовал стандартные геттеры-сеттеры. После реализации всей структуры я откомпилировал и запустил программу и получил весьма интересный набор чисел (см. Изображение 6).

```

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025 (main)
$ cd task4

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task4 (main)
$ javac MyFirstProgram.java

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task4 (main)
$ java MyFirstClass
2 3 4 5 6 7 8 9
3 4 5 6 7 8 9 10
4 5 6 7 8 9 10 11
5 6 7 8 9 10 11 12
6 7 8 9 10 11 12 13
7 8 9 10 11 12 13 14
8 9 10 11 12 13 14 15
9 10 11 12 13 14 15 16

```

Изображение 6

Задание №5

Для выполнения задание под номер 5 я, как и было сказано, удалил все байт-коды классов и перенес всё, что осталось в соответствующую директорию под названием «task5», которую я сделал ранее уже известной мне командой «mkdir». Далее разбил единый файл на два и перенес файл с классом «MySecondClass» в директорию «myfirstpackage» (см. Изображение 7)

```

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025 (main)
$ mkdir task5

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025 (main)
$ cd task5

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task5 (main)
$ nano MyFirstProgram.java

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task5 (main)
$ nano MyFirstPackage.java

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task5 (main)
$ nano MyFirstProgram.java

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task5 (main)
$ mkdir myfirstpackage

```

Изображение 7

После в соответствии с условием я откомпилировал полученную программу и получил ошибки (см. Изображение 8)

```

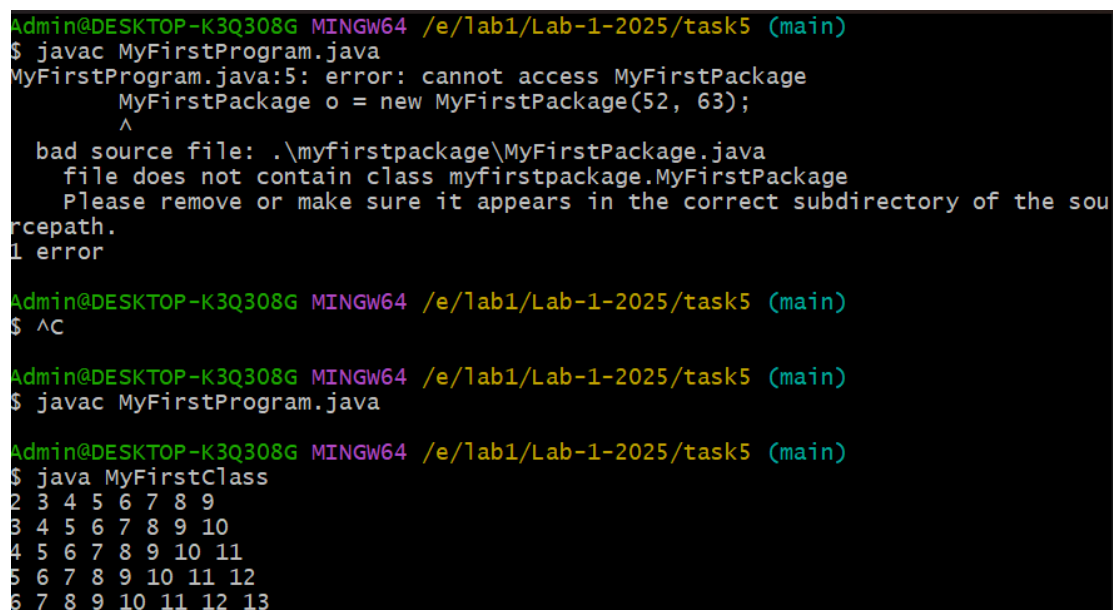
Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task5 (main)
$ javac MyFirstProgram.java
MyFirstProgram.java:5: error: cannot find symbol
    MySecondClass o = new MySecondClass(52, 63);
    ^
  symbol:   class MySecondClass
  location: class MyFirstClass
MyFirstProgram.java:5: error: cannot find symbol
    MySecondClass o = new MySecondClass(52, 63);
                                ^
  symbol:   class MySecondClass
  location: class MyFirstClass
2 errors

```

Изображение 8

Для решения данной проблемы я добавил перед «class MySecondClass» индентификатор «public» для того чтобы этот класс стал виден не только внутри директории, но и вне её. Но чтобы это работало мне пришлось переименовать класс «MySecondClass» в класс «MyFirstPackage». После этого я вновь скомпилировал программу и получил вот такую ошибку (см. Изображение 9)

Для решения данной проблемы в файле «MyFirstPackage.java» я добавил надпись «package myfirstpackage», которая означает, что этот класс теперь можно импортировать (см. Изображение 9).



```
Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task5 (main)
$ javac MyFirstProgram.java
MyFirstProgram.java:5: error: cannot access MyFirstPackage
    MyFirstPackage o = new MyFirstPackage(52, 63);
    ^
bad source file: .\myfirstpackage\MyFirstPackage.java
file does not contain class myfirstpackage.MyFirstPackage
Please remove or make sure it appears in the correct subdirectory of the sourcepath.
1 error

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task5 (main)
$ ^C

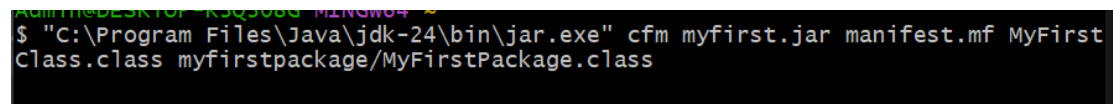
Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task5 (main)
$ javac MyFirstProgram.java

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task5 (main)
$ java MyFirstClass
2 3 4 5 6 7 8 9
3 4 5 6 7 8 9 10
4 5 6 7 8 9 10 11
5 6 7 8 9 10 11 12
6 7 8 9 10 11 12 13
```

Изображение 9

Задание №6

Для выполнения данного задания я создал соответствующую номерную директорию «task6». Перенёс в неё вручную все файлы формата .class, сохранив первоначальную структуру. Далее я создал файл «manifest.mf», используя стандартный блокнот. В него я поместил те данные, которые указаны в примере, указав лишь в качестве имени своё собственное «Петрухин Роман». После этого я используя специализированную команду создал jar-архив (см. Изображение 10)



```
Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task6 (main)
$ "C:\Program Files\Java\jdk-24\bin\jar.exe" cfm myfirst.jar manifest.mf MyFirstClass.class myfirstpackage/MyFirstPackage.class
```

Изображение 10

Далее я перенес его в отдельную директорию под название «MyJar» и запустил специальной командой, получив тот же результат, что и ранее (см. Изображение 11).

```
Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025 (main)
$ cd task6

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task6 (main)
$ cd MyJar

Admin@DESKTOP-K3Q308G MINGW64 /e/lab1/Lab-1-2025/task6/MyJar (main)
$ java -jar myfirst.jar
2 3 4 5 6 7 8 9
3 4 5 6 7 8 9 10
4 5 6 7 8 9 10 11
5 6 7 8 9 10 11 12
6 7 8 9 10 11 12 13
7 8 9 10 11 12 13 14
8 9 10 11 12 13 14 15
9 10 11 12 13 14 15 16
```

Изображение 11