

ЛАБОРАТОРНАЯ РАБОТА №1

**Знакомство с компилятором, запуском виртуальной машины Java
и работой с пакетами.**

**по курсу
Объектно-ориентированное программирование**

Группа 6204-010302D Студент:

С.О.Куропаткин.

Преподаватель: Борисов Дмитрий Сергеевич.

Задание 1

Знакомство с компилятором java и Jvm.

Ход выполнения работы

В ходе первого задания было необходимо запустить `javac` и программу `java` без параметров.

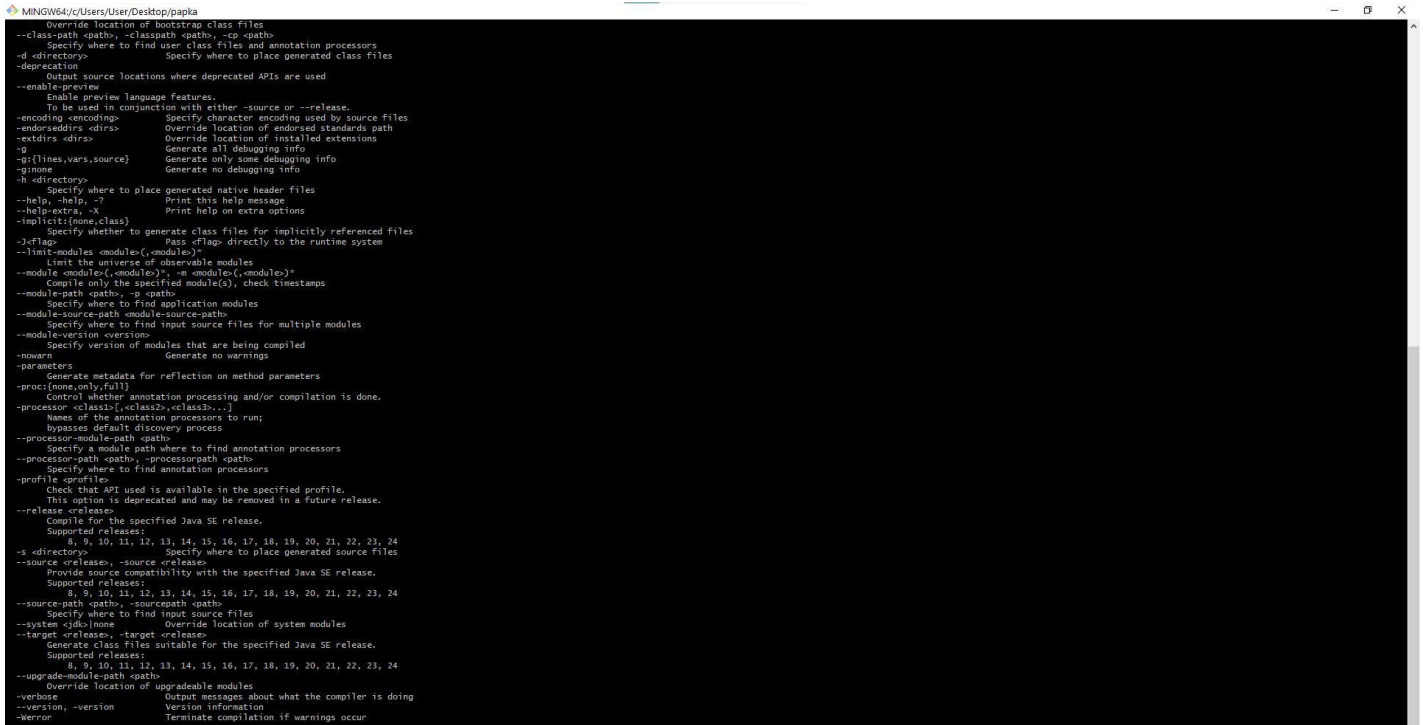


Рисунок 1 - Снимок экрана с результатами.

Задание 2

Создание класса

Во втором задании требовалось создать исходный файл `MyFirstProgram.java`, содержащий один пустой класс `MyFirstClass`, затем добавить в него метод `main` и добиться успешного запуска.

Ход выполнения работы

1. Создание файла с пустым классом. Компилятор создает файл но при запуске выходит ошибка компиляции.
2. Добавляем метод `main` без модификатора `static`. Компилятор принимает код, но JVM ожидает `public static void main(String[] s)`
3. Добавляем модификаторы и получаем результат.

```

User@DESKTOP-H7S1300 MINGW64 ~/Desktop/papka/Task2
$ nano MyFirstProgram.java

User@DESKTOP-H7S1300 MINGW64 ~/Desktop/papka/Task2
$ javac MyFirstProgram.java

User@DESKTOP-H7S1300 MINGW64 ~/Desktop/papka/Task2
$ 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

User@DESKTOP-H7S1300 MINGW64 ~/Desktop/papka/Task2

```

```

User@DESKTOP-H7S1300 MINGW64 ~/Desktop/papka/Task2
$ javac MyFirstProgram.java

User@DESKTOP-H7S1300 MINGW64 ~/Desktop/papka/Task2
$ java MyFirstClass
Hello World!!!

```

.Рисунок 2 - Снимок экрана с результатами

Задание 3 Замена метода main

В третьем задании необходимо заменить тело метода main.

Ход выполнения работы 1.

```

User@DESKTOP-H7S1300 MINGW64 ~/Desktop/papka/Task2
$ mkdir Task3

User@DESKTOP-H7S1300 MINGW64 ~/Desktop/papka/Task2
$ cd Task3

User@DESKTOP-H7S1300 MINGW64 ~/Desktop/papka/Task2/Task3
$ nano MyFirstProgram.java

User@DESKTOP-H7S1300 MINGW64 ~/Desktop/papka/Task2/Task3
$ javac MyFirstProgram.java

User@DESKTOP-H7S1300 MINGW64 ~/Desktop/papka/Task2/Task3
$ java MyFirstClass arg1 arg2 arg3 arg4 arg5
arg1
arg2
arg3
arg4
arg5

```

Изменяем тело метода main и выводим результат.

Рисунок 3

Задание 4 Работа с двумя классами

В этом задании необходимо добавить второй класс и реализовать в нём простую логику, а также использовать его в main.

Ход выполнения работы

1. Во втором классе объявлены два приватных поля.
2. Методы для получения и модификации значение
3. Метод, реализующий некоторые действия над числами

```
$ mkdir Task4

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3
$ cd Task4

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4
$ nano MyFirstProgram.java

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4
$ javac MyFirstProgram.java

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4
$ java MyFirstClass
0 -1 -2 -3 -4 -5 -6 -7
1 0 -1 -2 -3 -4 -5 -6
2 1 0 -1 -2 -3 -4 -5
3 2 1 0 -1 -2 -3 -4
4 3 2 1 0 -1 -2 -3
5 4 3 2 1 0 -1 -2
6 5 4 3 2 1 0 -1
7 6 5 4 3 2 1 0

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4
$ |
```

Рисунок 4

Задание 5

Работа с пакетами

В этом задании необходимо сделать пакет, в котором будет содержаться класс MySecondClass

Ход выполнения работы

1. Создали подкаталог **myfirstpackage** и файл **MySecondClass.java**
2. Удалили все скомпилированные файлы.
3. В соответствии с заданием мы вынесли описание класса **MySecondClass** в отдельный файл.
4. Для этого была создана поддиректория **myfirstpackage**, которая будет представлять пакет.
5. В файл **myfirstpackage/MySecondClass.java** был перенесён исходный код класса
6. Так как теперь **MySecondClass** находится в отдельном пакете, чтобы использовать его, в начало файла MyFirstProgram.java было добавлено:

Import myfirstpackage *

```

$ nano MyFirstProgram.java

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4/Task5
$ javac myfirstpackage/MyFirstProgram.java
error: file not found: myfirstpackage\MyFirstProgram.java
Usage: javac <options> <source files>
use --help for a list of possible options

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4/Task5
$ javac myfirstpackage/MyFirstPackage.java
myfirstpackage\MyFirstPackage.java:2: error: class MySecondClass is public, shou
ld be declared in a file named MySecondClass.java
public class MySecondClass{
    ^
1 error

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4/Task5
$ nano myfirstpackage/MyFirstPackage.java

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4/Task5
$ nano MyFirstProgram.java

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4/Task5
$ javac myfirstpackage/MyFirstPackage.java
myfirstpackage\MyFirstPackage.java:2: error: class MySecondClass is public, shou
ld be declared in a file named MySecondClass.java
public class MySecondClass{
    ^
1 error

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4/Task5
$ mv myfirstpackage/MyFirstPackage.java myfirstpackage/MySecondClass

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4/Task5
$ javac myfirstpackage/MySecondClass.java
error: file not found: myfirstpackage\MySecondClass.java
Usage: javac <options> <source files>
use --help for a list of possible options

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4/Task5
$ mv myfirstpackage/MySecondClass myfirstpackage/MySecondClass.java

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4/Task5
$ javac myfirstpackage/MySecondClass.java

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4/Task5
$ javac MyFirstProgram.java
MyFirstProgram.java:1: error: ';' expected
import myfirstpackage.*
    ^
1 error

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4/Task5
$ nano MyFirstProgram.java

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4/Task5
$ javac MyFirstProgram.java

User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Task2/Task3/Task4/Task5
$ java MyFirstClass
0 -1 -2 -3 -4 -5 -6 -7
1 0 -1 -2 -3 -4 -5 -6
2 1 0 -1 -2 -3 -4 -5
3 2 1 0 -1 -2 -3 -4
4 3 2 1 0 -1 -2 -3
5 4 3 2 1 0 -1 -2
6 5 4 3 2 1 0 -1
7 6 5 4 3 2 1 0

```

Рисунок 5

Задание 6

Создание MyFirst.jar

В этом задании необходимо добавить исполняемый JAR-архив.

Ход выполнения работы

1. Создали каталог *classes* и скопировали в него скомпилированные файлы из каталога *out* задания 5, сохраняя структуру каталогов.
2. В рабочей папке Task6, где находятся *MyFirstProgram.class* и каталог *myfirstpackage* с классом *MySecondClass.class*, была выполнена команда:

jar cfm myfirst.jar manifest.mf MyFirstProgram.class myfirstpackage/

3. Создали файл *manifest.mf* со следующим содержимым (последняя строка должна заканчиваться переводом строки):

Manifest-Version: 1.0

Created-By: <Фамилия>

Main-Class: MyFirstClass.



```
User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Lab-1-2025/MyJar (main)
$ java -jar myfirst.jar
0 -1 -2 -3 -4 -5 -6 -7
1 0 -1 -2 -3 -4 -5 -6
2 1 0 -1 -2 -3 -4 -5
3 2 1 0 -1 -2 -3 -4
4 3 2 1 0 -1 -2 -3
5 4 3 2 1 0 -1 -2
6 5 4 3 2 1 0 -1
7 6 5 4 3 2 1 0
User@DESKTOP-H751300 MINGW64 ~/Desktop/papka/Lab-1-2025/MyJar (main)
```

Рисунок 6

Заключение

- Я научился Запускать компилятор Java (*javac*) и виртуальную машину (*java*) из консоли и использовать их ключи для задания путей и каталогов.
- Приобрел навыки Создавать классы и различать имена файла и класса при компиляции и запуске.
- Обработать аргументы командной строки в методе *main*.
- Создавать классы с полями, конструкторами, методами доступа и выполняющими действия.
- Соединять код в пакеты: выносить классы в отдельные каталоги, импортировать их.
- Формировать исполняемые JAR-архивы с собственным манифестом и запускать их.