

Лабораторная работа №1

Лебедев Кирилл Дмитриевич 6204-010302D

1) Задание 1.

На рисунке 1 показан запуск компилятора `javac` без параметров. После ввода `javac` в консоль вышел список команд, синтаксис и их описание.

```
C:\Users\kiril>javac
Usage: javac <options> <source files>
where possible options include:
  @<filename>                Read options and filenames from file
  -Akey[=value]              Options to pass to annotation processors
  --add-modules <module>(,<module>)*
                             Root modules to resolve in addition to the initial modules,
                             or all modules on the module path if <module> is ALL-MODULE-PATH.
  --boot-class-path <path>, -bootclasspath <path>
                             Override location of bootstrap class files
  --class-path <path>, -classpath <path>, -cp <path>
                             Specify where to find user class files and annotation processors
  -d <directory>             Specify where to place generated class files
  -deprecation
                             Output source locations where deprecated APIs are used
  --enable-preview
                             Enable preview language features.
                             To be used in conjunction with either -source or --release.
  -encoding <encoding>       Specify character encoding used by source files
  -endorseddirs <dirs>       Override location of endorsed standards path
  -extdirs <dirs>            Override location of installed extensions
  -g                          Generate all debugging info
  -g:{lines,vars,source}     Generate only some debugging info
  -g:none                     Generate no debugging info
  -h <directory>             Specify where to place generated native header files
```

Рис.1

На рисунке 2 показан запуск программы `java` без параметров. После ввода `java` в консоль, вышел список команд, синтаксис и их описание.

```
C:\Users\kiril>java
Usage: java [java options...] <application> [application arguments...]

Where <application> is one of:
  <mainclass>                to execute the main method of a compiled main class
  -jar <jarfile>.jar          to execute the main class of a JAR archive
  -m <module>[/<mainclass>]  to execute the main class of a module
  <sourcefile>.java           to compile and execute a source-file program

Where key java options include:
  --class-path <class path>
    where <class path> is a list of directories and JAR archives to search for class files, separated by ";"
  --module-path <module path>
    where <module path> is a list of directories and JAR archives to search for modules, separated by ";"
  -version
    to print product version to the error stream and exit

For additional help on usage:      java --help
For an interactive Java environment: jshell
```

Рис.2

2) Задание 2.

Создал файл MyFirstClass.java (рисунок 3) и добавил следующие строчки:

```
class MyFirstClass {  
}
```

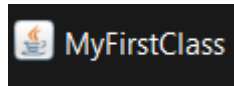


Рис.3

После этого я откомпилировал файл с помощью команды в консоли

```
javac MyFisrtClass.java
```

и получил файл MyFirstClass.class (рисунок 4).

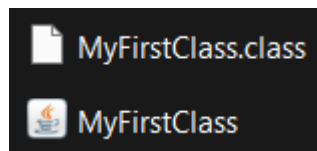


Рис. 4

Затем я запустил программу командой

```
java MyFirstClass
```

Согласно заданию, добавил код в тело класса:

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

После чего проделал всё тоже самое, откомпилировал и запустил.

Чтобы программа заработала, добавил static, тк на момент запуска программы ни одного объекта типа MyFirstClass не существует, перед void main(String[] s).

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

```
}
```

После изменений увидел следующий результат (рисунок 5).

```
C:\Users\kiril\Рабочий стол\sadfasdf\task2>javac MyFirstClass.java  
C:\Users\kiril\Рабочий стол\sadfasdf\task2>java MyFirstClass  
Hello World!!!
```

Рис. 5

3) Задание 3.

Согласно заданию, внес следующие изменения в метод main():

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

После чего запустил программу с добавлением аргументов к команде:

```
java MyFirstClass arg1 arg2 arg3 arg4 arg5
```

Результат(рисунок 6):

```
C:\Users\kiril\Рабочий стол\sadfasdf\task3>javac MyFirstClass.java  
C:\Users\kiril\Рабочий стол\sadfasdf\task3>java MyFirstClass arg1 arg2 arg3 arg4 arg5  
arg1  
arg2  
arg3  
arg4  
arg5
```

Рис. 6

4) Задание 4.

Согласно заданию, добавил второй класс в файл MyFirstClass.java:

```
class MySecondClass {  
    private int a;  
    private int b;
```

```

MySecondClass(int a, int b) {
    this.a = a;
    this.b = b;
}

public void setA(int val) {
    this.a = val;
}

public int getA() {
    return a;
}

public void setB(int val) {
    this.b = val;
}

public int getB() {
    return b;
}

public int multiAB() {
    return a*b;
}
}

```

И заменил код метода main() на следующие строчки:

```

MySecondClass o = new MySecondClass(10, 10);
int i, j;
for (i = 1; i <= 8; i++) {

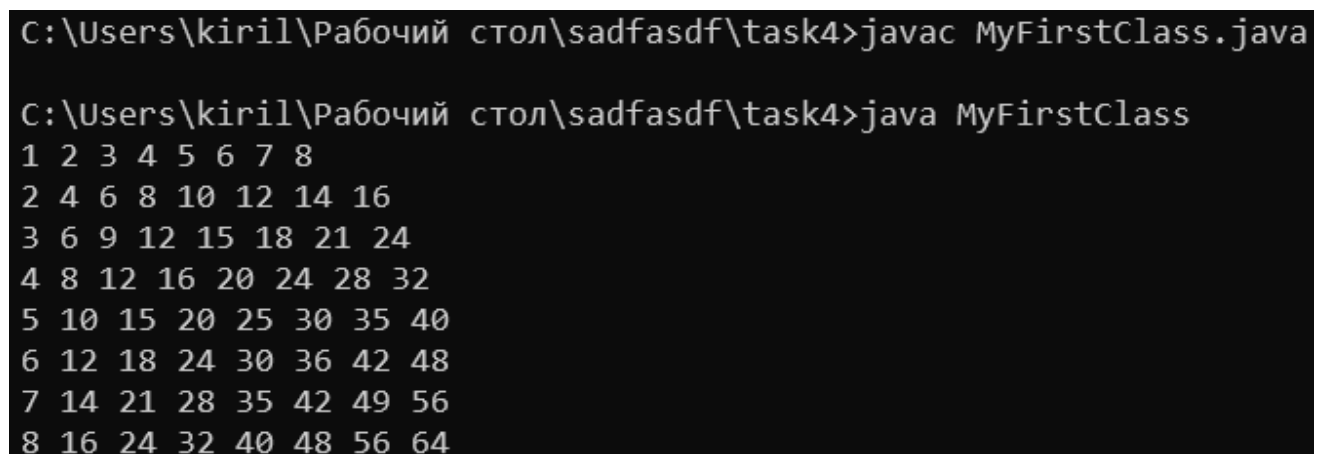
```

```

        for(j = 1; j <= 8; j++) {
            o.setA(i);
            o.setB(j);
            System.out.print(o.multiAB());
            System.out.print(" ");
        }
        System.out.println();
    }
}

```

После чего откомпилировал и получил результат(рисунок 7):



```

C:\Users\kiril\Рабочий стол\sadfasdf\task4>javac MyFirstClass.java

C:\Users\kiril\Рабочий стол\sadfasdf\task4>java MyFirstClass
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

```

Рис. 7

5) Задание 5.

Согласно заданию, вынес второй класс в отдельный файл и назвал его:

MyFirstPackage.java

После чего добавил в начало файла MyFirstPackage.java следующую строку:

```
package myfirstpackage;
```

а в файл MyFirstClass.java:

```
import myfirstpackage.*;
```

Появилось несколько ошибок, которые я исправил следующим образом:

1. Добавил модификатор доступа public к MySecondClass
2. Добавил модификатор доступа public к конструктору класса
3. Переименовал файл MyFirstPackage.java в MySecondClass.java

Результат(рисунок 8):

```
C:\Users\kiril\Рабочий стол\sadfasdf\task5>javac myfirstpackage\MySecondClass.java
C:\Users\kiril\Рабочий стол\sadfasdf\task5>javac MyFirstClass.java
C:\Users\kiril\Рабочий стол\sadfasdf\task5>java MyFirstClass
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
```

Рис. 8

6) Задание 6.

Запустил программу jar для создание архива. И получил формат записи для создания архива:

```
C:\Users\kiril\Рабочий стол\sadfasdf\task6>jar
```

```
Usage: jar [OPTION...] [ [--release VERSION] [-C dir] files] ...
```

```
Try `jar --help' for more information.
```

Затем скопировал файлы с расширением *.class в папку с заданием 6 и создал файл manifest.mf

В manifest.mf добавил следующие строки:

```
Manifest-Version: 1.0
```

```
Created-By: Lebedev
```

```
Main-Class: MyFirstClass
```

Обязательно с переносом строки в конце (после MyFirstClass).

После создал архив следующей командой:

```
jar cfm myfirst.jar manifest.mf *.class myfirstpackage\*.class
```

В результате чего появился архив myfirst.jar содержащий файлы с расширением .class и manifest.mf, который я перенес в папку myjar.

Результат открытия архива myfirst.jar(рисунок 9):

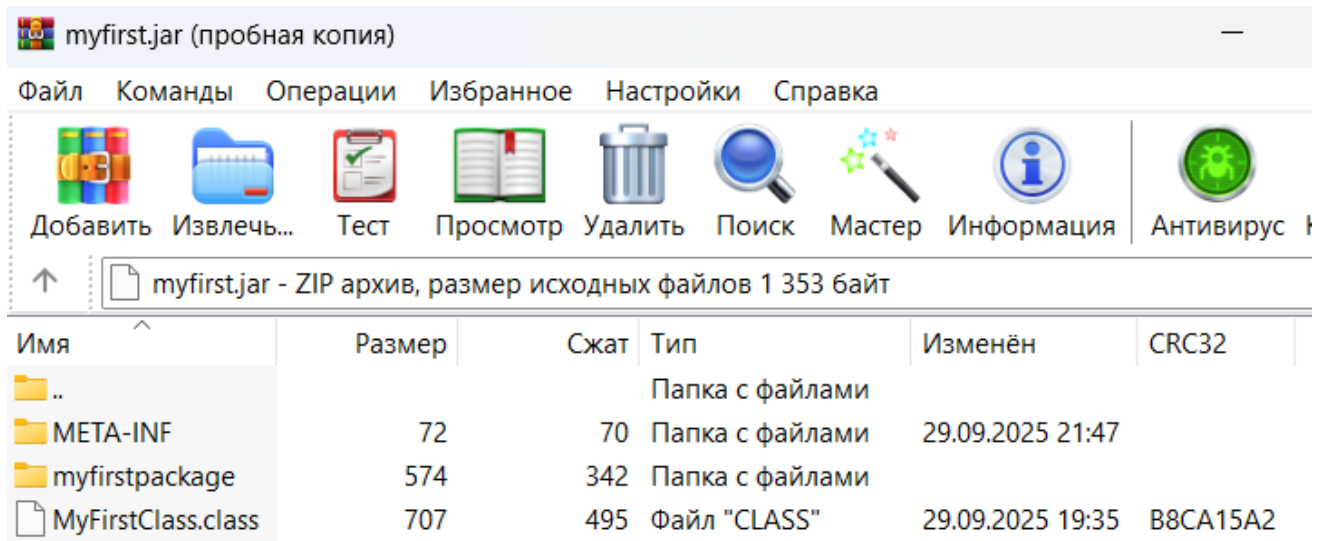


Рис. 9