

ЛАБОРАТОРНАЯ РАБОТА № 1

Номер группы 6204-01.03.02

Студент Скиба В.А

Самара 2025

Задание 1

Запустите компилятор `javac` без параметров и ознакомьтесь с форматом задания параметров компилятора. Запустите программу Java без параметров и ознакомьтесь с форматом параметров запуска устройства Java (JVM).

Ход выполнения работы

```
PS D:\java\00P_JAVA\Lab1SkibaVladimir6204_010302> 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
  --help, -help, -?          Print this help message
  --help-extra, -X           Print help on extra options
  -implicit:{none,class}
                             Specify whether to generate class files for implicitly referenced files
  -J<flag>                   Pass <flag> directly to the runtime system
  --limit-modules <module>(<module>)*
                             Limit the universe of observable modules
  --module <module>(<module>)*, -m <module>(<module>)*
                             Compile only the specified module(s), check timestamps
  --module-path <path>, -p <path>
                             Specify where to find application modules
  --module-source-path <module-source-path>
                             Specify where to find input source files for multiple modules
  --module-version <version>
                             Specify version of modules that are being compiled
  -nowarn                     Generate only mandatory warnings
  -parameters
                             Generate metadata for reflection on method parameters
  -proc:{none,only,full}
                             Control whether annotation processing and/or compilation is done.
  -processor <class1>[,<class2>,<class3>...]
                             Names of the annotation processors to run;
                             bypasses default discovery process
  --processor-module-path <path>
                             Specify a module path where to find annotation processors
  --processor-path <path>, -processorpath <path>
                             Specify where to find annotation processors
  -profile <profile>
                             Check that API used is available in the specified profile.
                             This option is deprecated and may be removed in a future release.
  --release <release>
                             Compile for the specified Java SE release.
                             Compile for the specified Java SE release.
                             Supported releases:
                                8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25
  -s <directory>             Specify where to place generated source files
  --source <release>, -source <release>
                             Provide source compatibility with the specified Java SE release.
                             Supported releases:
                                8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25
  --source-path <path>, -sourcepath <path>
                             Specify where to find input source files
  --system <jdk>|none         Override location of system modules
  --target <release>, -target <release>
                             Generate class files suitable for the specified Java SE release.
                             Supported releases:
                                8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25
  --upgrade-module-path <path>
                             Override location of upgradeable modules
  -verbose                    Output messages about what the compiler is doing
  --version, -version         Version information
  -Werror                     Terminate compilation if warnings occur
```

```
PS D:\java\OOP_JAVA\Lab1SkibaVladimir6204_010302> 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
PS D:\java\OOP_JAVA\Lab1SkibaVladimir6204_010302> █
```

Задание 2

Цель: Создание, компиляция и запуск простейшего Java-класса.

Ход работы:

1. Создан файл `MyFirstProgram.java` с содержимым:

```
class MyFirstClass {

    public static void main(String[] args) {

        System.out.println("Hello world!!!");

    }

}
```

2. Выполнена компиляция:

```
javac MyFirstProgram.java
```

3. Запуск программы:

```
java MyFirstClass
```

Результат:

```
PS D:\java\OOP_JAVA\Lab1SkibaVladimir6204_010302\task2> javac MyFirstProgram.java
PS D:\java\OOP_JAVA\Lab1SkibaVladimir6204_010302\task2> java MyFirstClass
Hello world!!!
```

Задание 3

Цель: Передача аргументов командной строки и их вывод.

Ход работы:

1. Изменён метод `main()`:

```
for (int i = 0; i < s.length; i++) {
    System.out.println(s[i]);
}
```

2. Компиляция и запуск с аргументами:

```
javac MyFirstProgram.java
```

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

Результат:

```
PS D:\java\OOP_JAVA\Lab1SkibaVladimir6204_010302\task3> javac MyFirstProgram.java
PS D:\java\OOP_JAVA\Lab1SkibaVladimir6204_010302\task3> java MyFirstClass arg1 arg2 arg3 arg4 arg5
arg1
arg2
arg3
arg4
arg5
```

Задание 4

Цель: Создание второго класса с методами и полями, вывод таблицы значений.

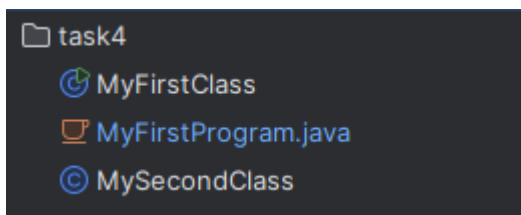
Ход работы:

1. Добавлен класс `MySecondClass` с конструктором, геттерами, сеттерами и методом умножения.
2. Изменён метод `main()` для вывода таблицы умножения.
3. Компиляция и запуск:

```
javac MyFirstProgram.java
```

```
java MyFirstClass
```

Результат:



```
PS D:\java\00P_JAVA\Lab1SkibaVladimir6204_010302\task4> javac MyFirstProgram.java
PS D:\java\00P_JAVA\Lab1SkibaVladimir6204_010302\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
```

Задание 5

Цель: Работа с пакетами.

Ход работы:

1. Перемещение в пакет:

- Класс `MySecondClass` был перемещен в пакет `myfirstpackage`
- В начало файла `MyFirstPackage.java` добавлена строка:
- `package myfirstpackage;`

2. Создание структуры каталогов:

- Создана поддиректория `myfirstpackage`
- Файл `MyFirstPackage.java` помещен в эту поддиректорию

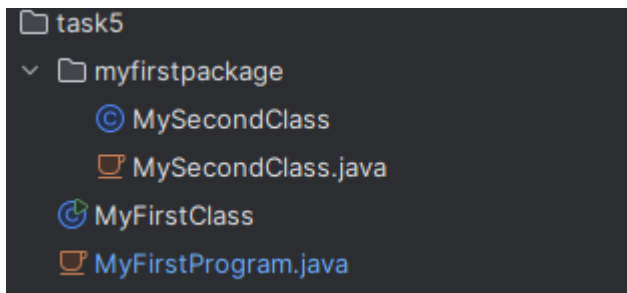
3. Изменения в основном классе:

- В файл `MyFirstProgram.java` добавлен импорт пакета:
- `import myfirstpackage.*;`
- Все обращения к `MySecondClass` теперь используют полное квалифицированное имя или импорт

4. Компиляция:

- Сначала компилируется пакет:
- `javac myfirstpackage/MyFirstPackage.java`
- Затем основной класс:
- `javac MyFirstProgram.java`

Результат:



```
PS D:\java\OOP_JAVA\Lab1SkibaVladimir6204_010302\task5> javac MyFirstProgram.java
PS D:\java\OOP_JAVA\Lab1SkibaVladimir6204_010302\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
```

Задание 6

Цель: Создание JAR-архива с манифестом.

Ход работы:

1. Создан файл `manifest.mf`:

Manifest-Version: 1.0

Created-By: <Скиба Владимир>

Main-Class: MyFirstClass

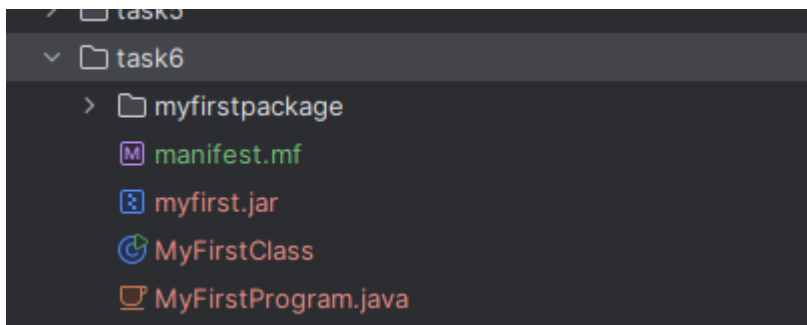
2. Создан JAR-архив:

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

3. Запуск архива:

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

Результат:



```
PS D:\java\OOP_JAVA\Lab1SkibaVladimir6204_010302\task6> jar cfm myfirst.jar manifest.mf MyFirstClass.class myfirstpackage/MySecondClass.class
PS D:\java\OOP_JAVA\Lab1SkibaVladimir6204_010302\task6> java -jar myfirst.jar
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
PS D:\java\OOP_JAVA\Lab1SkibaVladimir6204_010302\task6> 
```

Вывод

В ходе лабораторной работы были изучены основы компиляции и запуска Java-программ, работа с аргументами командной строки, создание классов, использование пакетов и упаковка проекта в JAR-архив. Все задания выполнены успешно.