

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: Сучков Борис Антонович

Группа: 6201-120303D

Самара 2025

Задание на лабораторную работу

В процессе написания тестовых заданий ознакомиться со структурой исходного кода для Java, изучить особенности областей видимости и использования пакетов.

Каждое следующее задание надо выполнять в новом каталоге (например, Task1, Task2 и так далее).

Эта лабораторная работа выполняется без применения среды разработки.

Задание 1 (0,2 балла)

Запустите компилятор `javac` без параметров и ознакомьтесь с форматом задания параметров компилятора.

Запустите программу `java` без параметров и ознакомьтесь с форматом задания параметров запуска виртуальной машины Java (JVM).

Задание 2 (0,5 балла)

Создайте файл `MyFirstProgram.java`, содержащий исходный код одного пустого класса с именем `MyFirstClass`:

```
--- MyFirstProgram.java -----  
--  
  
class MyFirstClass {  
  
}  
  
-----  
--
```

Откомпилируйте его с помощью компилятора `javac`. Для этого необходимо выполнить команду “`javac MyFirstProgram.java`”. Запустите полученный файл класса на выполнение с помощью команды “`java <Имя класса>`”.

Обратите внимание на то, что на вход компилятора необходимо подавать имя файла с расширением, а на вход JVM – без расширения.

Также обратите внимание на то, что в операционных системах, в которых имена файлов чувствительны к регистру, имена файлов для компиляции и запуска следует указывать с учетом регистра. При этом расширение файла с исходным кодом должно быть “`.java`” (маленькими буквами).

Добавьте в класс метод `main()`:

```
--- MyFirstProgram.java -----  
--  
  
class MyFirstClass {
```

```

void main(String[] s) {

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

}

}

```

```

-----
--

```

Точкой входа программы является метод `main()` с параметрами `String[] s`.

Для вывода текстовой информации на экран в данном случае используется функция `println()` класса `PrintStream`. Поле `out` этого типа является статическим полем класса `System`, поэтому может использоваться без создания объекта типа `System`.

Откомпилируйте и запустите программу на выполнение.

Логично предположить, что метод `main()` должен быть статическим, т.к. на момент запуска программы ни одного объекта типа `MyFirstClass` не существует. Сделайте метод статическим, снова откомпилируйте и запустите программу.

Внесите в текст программы необходимые для ее запуска изменения, откомпилируйте и запустите программу.

Задание 3 (0,8 баллов)

Замените текст метода `main()` на следующий:

```

--- MyFirstClass.main -----
--

for (int i = 0; i < s.length; i++)

System.out.println(s[i]);

-----
--

```

Откомпилируйте и запустите программу, добавив в командную строку ряд аргументов. Например, следующим образом: `java MyFirstClass arg1 arg2 arg3 arg4 arg5`.

Задание 4 (1,5 балла)

В том же файле `MyFirstProgram.java` после описания класса `MyFirstClass` добавьте описание второго класса `MySecondClass`, реализующего следующую функциональность:

- имеет два приватных поля типа `int`;
- методы для получения и модификации их значений;

- конструктор, создающий объект и инициализирующий значения полей;
- метод с возвращаемым типом `int`, реализующий над этими числами действие на ваш выбор (умножение, вычитание и т.д.).

Код метода `MyFirstClass.main()` при этом следует заменить на следующий (в угловых скобках указаны элементы, имена которых следует подставить при написании программы):

```
--- MyFirstClass.main -----
--

<Создание и инициализация объекта "o" типа MySecondClass>;

int i, j;

for (i = 1; i <= 8; i++) {

for(j = 1; j <= 8; j++) {

o.<Метод установки значения первого числового поля>(i);

o.<Метод установки значения второго числового поля>(j);

System.out.print(o.<Метод, реализующий действие над числами>());

System.out.print(" ");

}

System.out.println();

}

-----
--
```

Откомпилируйте и запустите программу.

Задание 5 (1 балл)

Удалите все откомпилированные байт-коды классов. Вынесите код класса `MySecondClass` без изменений в отдельный файл с именем `MyFirstPackage.java`, и поместите его в поддиректорию `myfirstpackage`, откомпилируйте. Попробуйте откомпилировать файл `MyFirstProgram.java`.

Добавьте в начало исходного кода в файле `MyFirstProgram.java` следующий код:

```
--- MyFirstProgram.java -----
--

import myfirstpackage.*;
```

--
Снова попробуйте откомпилировать `MyFirstProgram.java`. Далее, следуя сообщениям компилятора и изменяя исходный код программы, добейтесь ее работоспособности.

Задание 6 (1 балл)

Запустите программу `jar`, предназначенную для создания архивов, и ознакомьтесь с форматом задания ключей для формирования архивов.

Скопируйте в рабочую папку, сохранив структуру каталогов, только файлы с расширением `class`, полученные в результате выполнения задания 5.

Создайте файл `manifest.mf`, содержащий следующий код:

--- `manifest.mf` -----
--

Manifest-Version: 1.0

Created-By: <Ваши фамилии>

Main-Class: `MyFirstClass`

--
Обратите внимание на то, что после имени класса надо обязательно поставить символ новой строки.

Создайте архив `myfirst.jar`, включив в него полученные ранее файлы классов и указав созданный вами манифест-файл. Переместите полученный файл в другую директорию (например, поддиректорию `MyJar`) и запустите его на выполнение.

Задание 1

Я запустил компилятор `javac` без параметров и ознакомился с форматом задания параметров компилятора.

```

C:\Users\Borus>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
                             8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
  -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
  --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
  --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

C:\Users\Borus>

```

Я запустил программу java без параметров и ознакомился с форматом задания параметров запуска виртуальной машины Java (JVM).

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Borus>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

C:\Users\Borus>
```

Задание 2

1. Создаю каталог Task2: `mkdir Task2`
2. Перехожу в этот каталог: `cd Task2`
3. Там создаю файл в блокноте: `notepad MyFirstProgram.java`
4. В файле создаю класс `MyFirstClass`:

```
class MyFirstClass {
}
```
5. Компилирую файл: `javac MyFirstProgram.java`
6. Запускаю `java MyFirstClass`. Получаю 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`."
7. Добавляю в класс метод `main()`:

```
class MyFirstClass {
    void main(String[] s) {
        System.out.println("Hello world!!!");
    }
}
```
8. Снова компилирую `javac MyFirstProgram.java`.
9. Запускаю `java MyFirstClass`. Опять получаю 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`"
10. Исправляю ошибку (делаю метод `main()` статическим):

```

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

```

11. Снова компилирую `javac MyFirstProgram.java`
12. Снова запускаю `java MyFirstClass` Получаю: `Hello world!!!`

Задание 3

1. Создаю каталог Task3: `mkdir Task3`
2. Создаю в этом каталоге файл: `notepad MyFirstProgram.java`
3. Пишу там код:

```

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

```
4. Компилирую файл: `javac MyFirstProgram.java`
5. Запускаю с аргументами: `java MyFirstClass arg1 arg2 arg3 ar4 arg5`
6. Получаю:

```

arg1
arg2
arg3
arg4
arg5

```

Задание 4

1. Создаю каталог Task4: `mkdir Task4`
2. Создаю в этом каталоге файл: `notepad MyFirstProgram.java`
3. Пишу там код:


```

MyFirstProgram.java
Файл  Изменить  Просмотр

class MyFirstClass {
    public static void main(String[] s) {
        MySecondClass o = new MySecondClass(0, 0);
        for (int i = 1; i <= 8; i++) {
            for(int j = 1; j <= 8; j++) {
                o.setNum1(i);
                o.setNum2(j);
                System.out.print(o.multiply() + "\t");
            }
            System.out.println();
        }
    }
}

class MySecondClass {
    private int num1;
    private int num2;

    // конструктор
    public MySecondClass(int n1, int n2) {
        this.num1 = n1;
        this.num2 = n2;
    }

    public void setNum1(int num1) {this.num1 = num1;}
    public void setNum2(int num2) {this.num2 = num2;}

    public int multiply() {
        return this.num1 * this.num2;
    }
}

```

4. Компилирую: `javac MyFirstProgram.java`
5. Запускаю: `java MyFirstClass` Получаю:

```

C:\Users\Borus\Desktop\вуз\Программирование\ООП\lr1\Task4>java MyFirstClass
1      2      3      4      5      6      7      8
2      4      6      8      10     12     14     1
6
3      6      9      12     15     18     21     2
4
4      8      12     16     20     24     28     3
2
5      10     15     20     25     30     35     4
0
6      12     18     24     30     36     42     4
8
7      14     21     28     35     42     49     5
6
8      16     24     32     40     48     56     6
4

C:\Users\Borus\Desktop\вуз\Программирование\ООП\lr1\Task4>

```

Задание 5

1. Создаю каталог Task5: `mkdir Task5`
2. Создаю в этом каталоге каталог myfirstpackage: `mkdir myfirstpackage`

3. Создаю в myfirstpackage файл MySecondClass.java: notepad myfirstpackage\MySecondClass.java и пишу туда код:

```
MySecondClass.java
Файл  Изменить  Просмотр

package myfirstpackage;

public class MySecondClass {
    private int num1;
    private int num2;
    public MySecondClass(int n1, int n2) { this.num1 = n1; this.num2 = n2; }
    public void setNum1(int num1) { this.num1 = num1; }
    public void setNum2(int num2) { this.num2 = num2; }
    public int multiply() { return this.num1 * this.num2; }
}
```

4. Создаю в каталоге Task5 файл MyFirstProgram.java: notepad MyFirstProgram.java и пишу туда код:

```
MyFirstProgram.java
Файл  Изменить  Просмотр

import myfirstpackage.*;

class MyFirstClass {
    public static void main(String[] s) {
        MySecondClass o = new MySecondClass(0, 0);
        for (int i = 1; i <= 8; i++) {
            for (int j = 1; j <= 8; j++) {
                o.setNum1(i);
                o.setNum2(j);
                System.out.print(o.multiply() + "\t");
            }
            System.out.println();
        }
    }
}
```

5. Компилирую: javac myfirstpackage/MySecondClass.java
javac MyFirstProgram.java
6. Запускаю: java MyFirstClass Получаю:

```
C:\Users\Borus\Desktop\вуз\Программирование\ООП\lr1\Task5>java MyFirstClass
1      2      3      4      5      6      7      8
2      4      6      8      10     12     14     1
6
3      6      9      12     15     18     21     2
4
4      8      12     16     20     24     28     3
2
5      10     15     20     25     30     35     4
0
6      12     18     24     30     36     42     4
8
7      14     21     28     35     42     49     5
6
8      16     24     32     40     48     56     6
4

C:\Users\Borus\Desktop\вуз\Программирование\ООП\lr1\Task5>
```

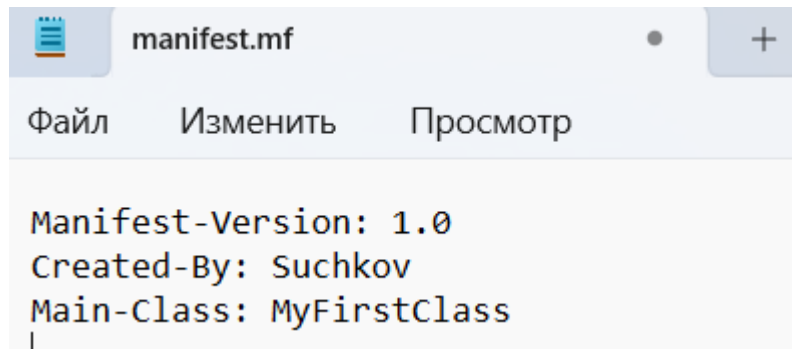
Задание 6

1. Я ознакомился с форматом задания ключей для формирования архивов:

```
C:\Users\Borus\Desktop\вуз\Программирование\ООП\lr1\Task6>jar
Usage: jar [OPTION...] [ [--release VERSION] [-C dir] files] ...
Try `jar --help' for more information.

C:\Users\Borus\Desktop\вуз\Программирование\ООП\lr1\Task6>
```

2. Создаю каталог Task6: `mkdir Task6`
3. Копирую туда файлы с расширением `.class` (в том числе и каталог `myfirstpackage` с файлом с расширением `.class` в нём)
4. Создаю файл манифеста в каталоге Task6: `notepad manifest.mf` и пишу туда код:



5. Создаю архив `myfirst.jar`: `jar cvfm myfirst.jar manifest.mf MyFirstClass.class myfirstpackage/MySecondClass.class`
6. Создаю каталог `MyJar`: `mkdir MyJar` и перемещаю в него архив `myfirst.jar`:
`move Task6\myfirst.jar MyJar`
7. Запускаю архив:

```

C:\Users\Borus\Desktop\вуз\Программирование\ООП\lr1>move Task6\myfirst.jar MyJar
Перемещено файлов:      1.

C:\Users\Borus\Desktop\вуз\Программирование\ООП\lr1>cd MyJar

C:\Users\Borus\Desktop\вуз\Программирование\ООП\lr1\MyJar>java -jar myfirst.jar
1      2      3      4      5      6      7      8
2      4      6      8      10     12     14     1
6
3      6      9      12     15     18     21     2
4
4      8      12     16     20     24     28     3
2
5      10     15     20     25     30     35     4
0
6      12     18     24     30     36     42     4
8
7      14     21     28     35     42     49     5
6
8      16     24     32     40     48     56     6
4

C:\Users\Borus\Desktop\вуз\Программирование\ООП\lr1\MyJar>

```