

Лабораторная работа №1

По дисциплине «Объектно-ориентированное программирование»

Тема «Ознакомление со структурой исходного кода Java, системой пакетов и организацией компиляции»

Студент: Родомакин М.И

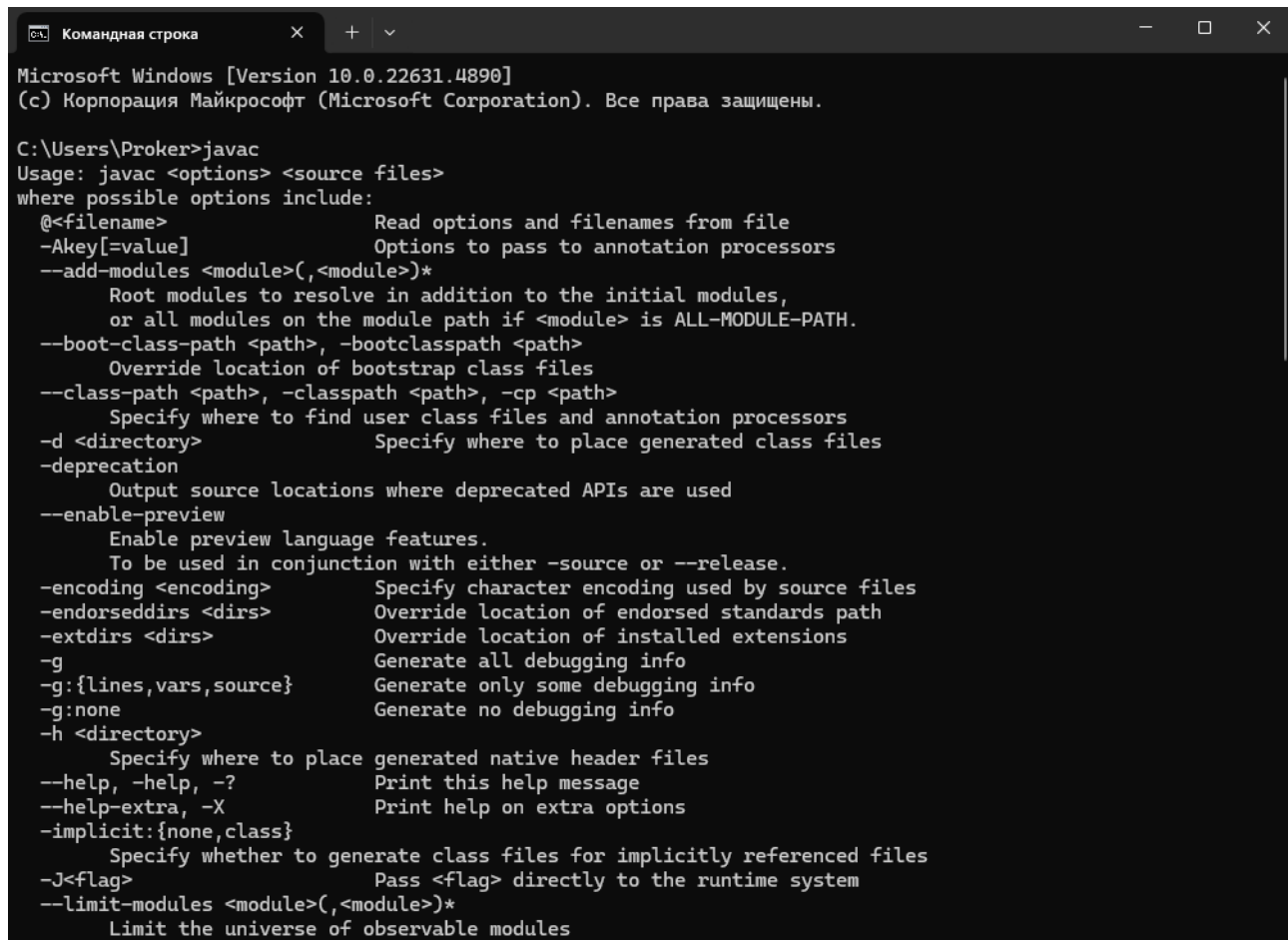
Группа: 6203-010302D

Содержание

Задание 1	3
Задание 2	4
Задание 3	5
Задание 4	6
Задание 5	7
Задание 6	8

Задание 1

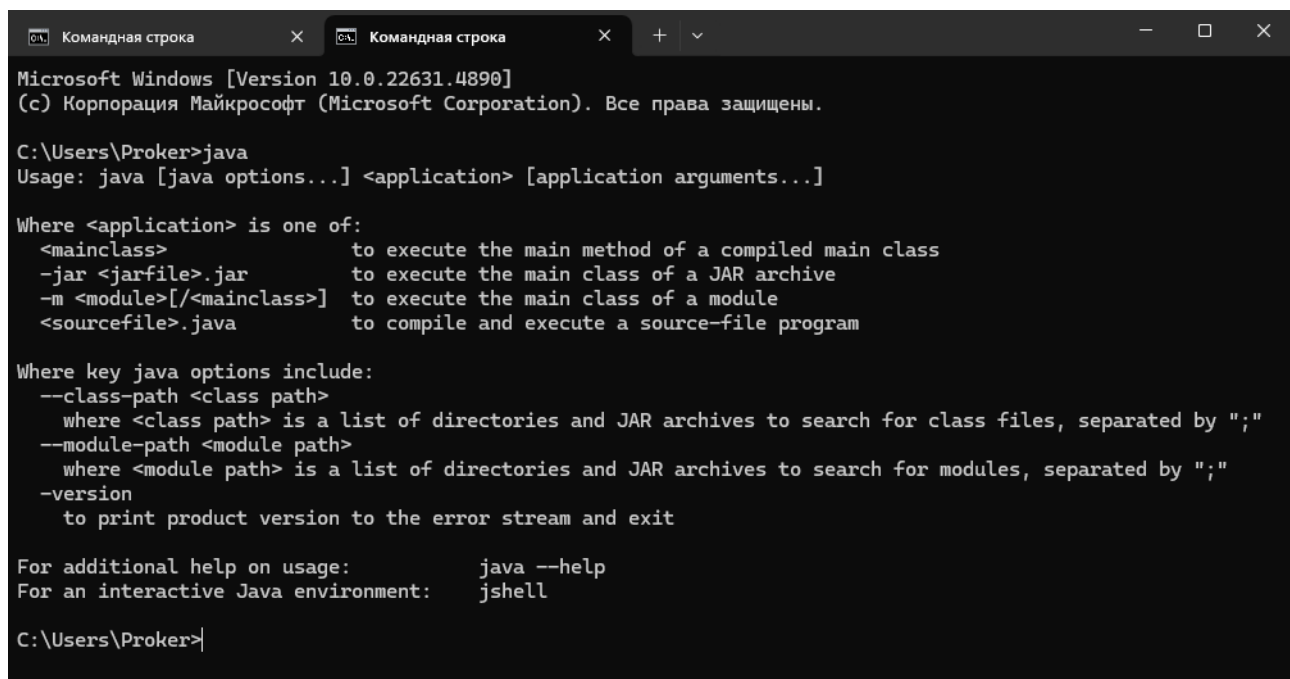
Были запущены компилятор `javac` и программа `java` без параметров для ознакомления с форматом задания параметров.



```
Командная строка
Microsoft Windows [Version 10.0.22631.4890]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Proker>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
```

Рисунок 1 - Текст при вводе `javac` в консоль



```
Командная строка
Microsoft Windows [Version 10.0.22631.4890]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Proker>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\Proker>
```

Рисунок 2 - Текст при вводе java в консоль

Задание 2

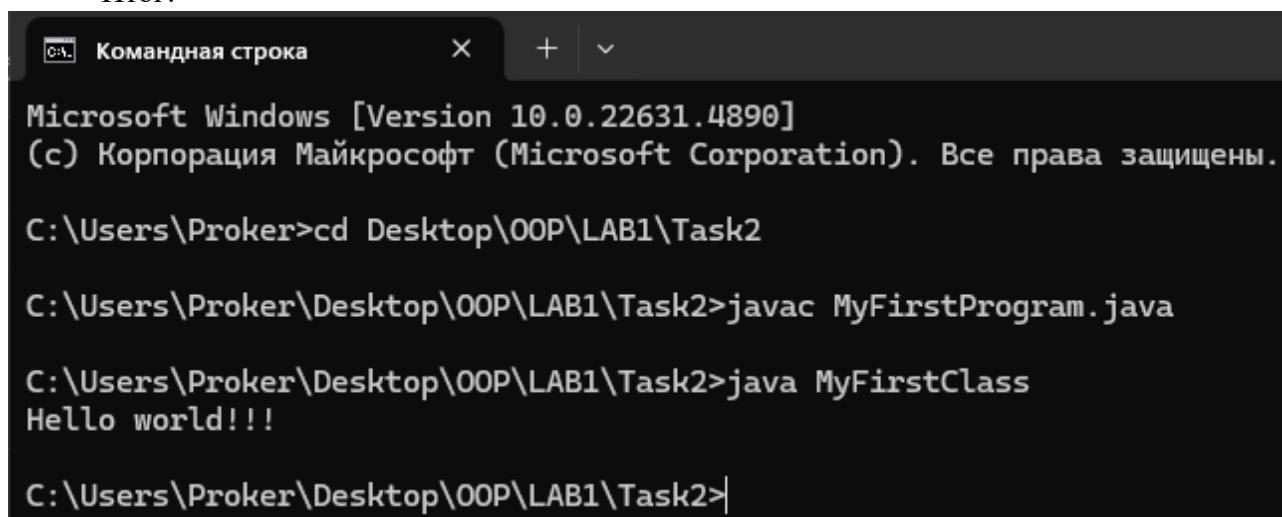
Создаем файл MyFirstProgram.java с пустым классом MyFirstClass и компилируем его с помощью javac, с помощью консольной команды `javac MyFirstProgram.java` и получаем файл MyFirstClass.class

Для успешного запуска через `java MyFirstClass`, потребовалось добавить метод `main()` с параметрами `String[] s`. Программа будет выводить в консоль послание на иностранных языках.

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

Рисунок 3 – Файл MyFirstClass.java

Итог:

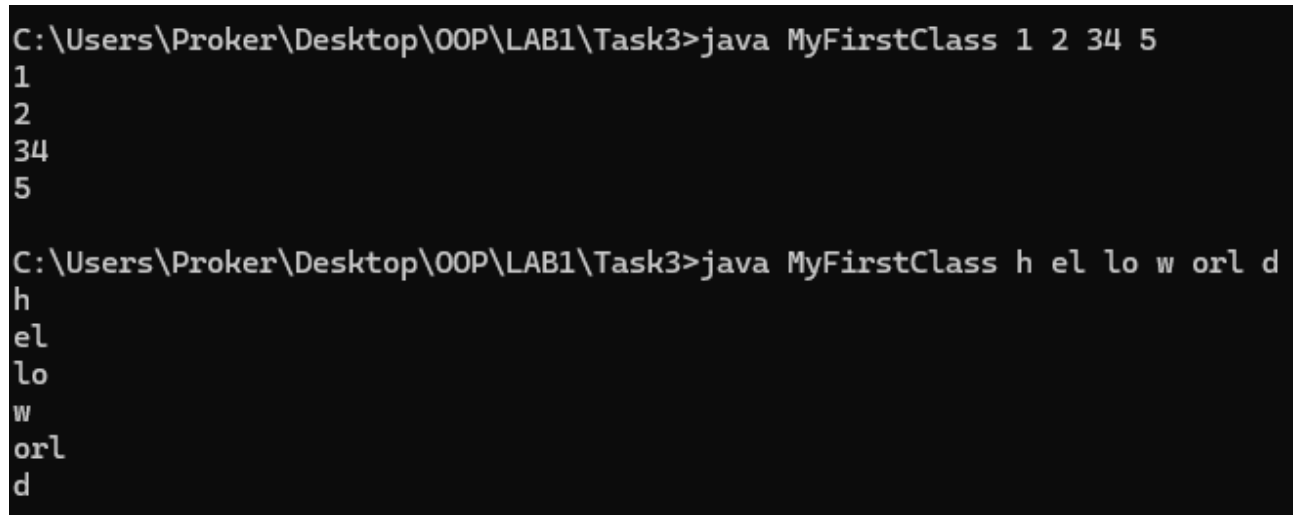


```
Командная строка  
Microsoft Windows [Version 10.0.22631.4890]  
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.  
  
C:\Users\Proker>cd Desktop\OOP\LAB1\Task2  
  
C:\Users\Proker\Desktop\OOP\LAB1\Task2>javac MyFirstProgram.java  
  
C:\Users\Proker\Desktop\OOP\LAB1\Task2>java MyFirstClass  
Hello world!!!  
  
C:\Users\Proker\Desktop\OOP\LAB1\Task2>
```

Рисунок 4 - Работа с консолью при заднии №2

Задание 3

Добавляем в main цикл, выводящий элементы массива строк, которые мы вводим при запуске программы. Полученные элементы выводятся каждый на новой строке.



```
C:\Users\Proker\Desktop\OOP\LAB1\Task3>java MyFirstClass 1 2 34 5
1
2
34
5

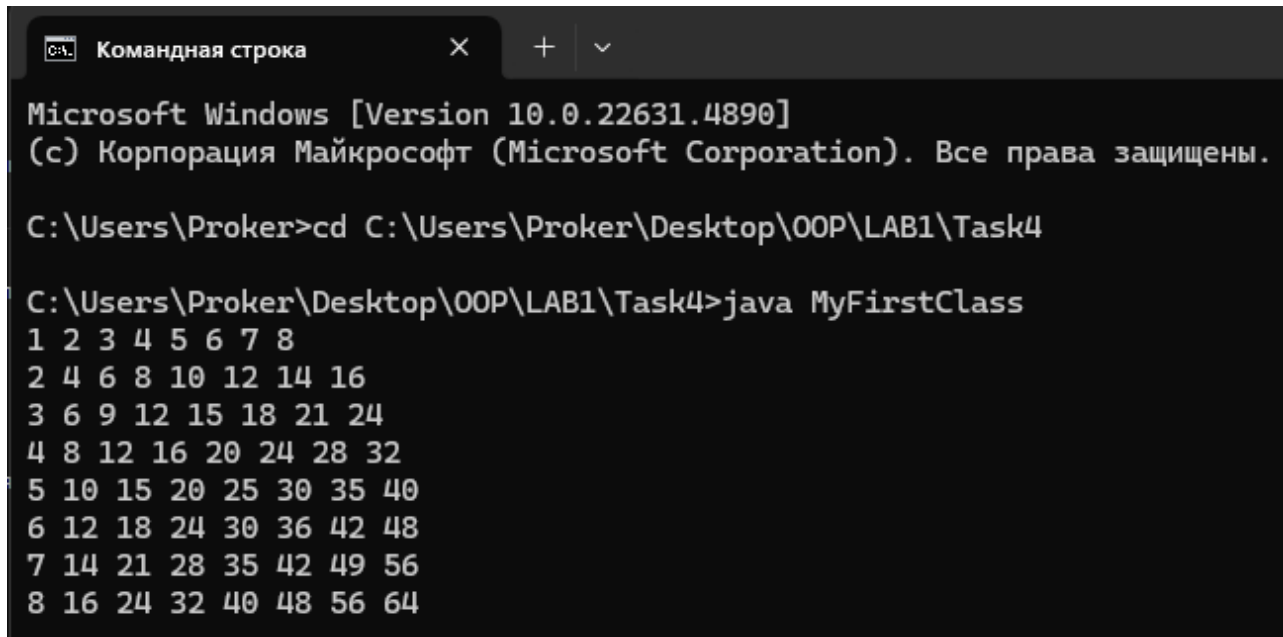
C:\Users\Proker\Desktop\OOP\LAB1\Task3>java MyFirstClass h el lo w orl d
h
el
lo
w
orl
d
```

Рисунок 5 - Работа с консолью при заднии №3

Задание 4

Создаем дополнительный класс `MySecondClass` в `MyFirstProgram.java`. Указываю ему два приватных поля типа `int val1` и `val2`, конструктор объектов класса, геттеры и сеттеры для двух значений, а также метод перемножения `val1` и `val2`.

В классе `MyFirstClass` мы создаем объект класса `MySecondClass`, после чего с помощью вложенного цикла двум переменным `i` и `j` присваиваем значения и перемножаем, после чего результаты выводим в консоль через пробел для `j` [1:8], и через строку для `i` [1:8]. Получается таблица умножения 8 на 8.



```
Microsoft Windows [Version 10.0.22631.4890]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Proker>cd C:\Users\Proker\Desktop\OOP\LAB1\Task4

C:\Users\Proker\Desktop\OOP\LAB1\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
```

Рисунок 6 - Работа с консолью при заднии №4

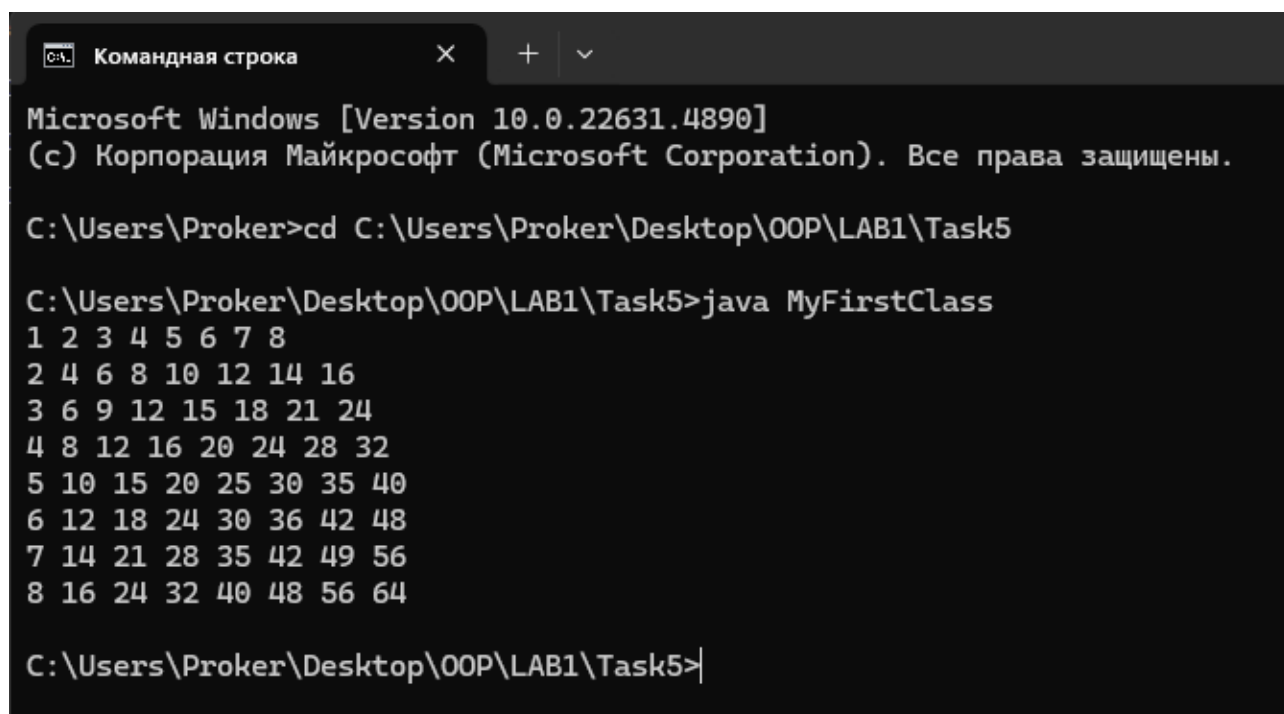
Задание 5

В соответствии с требованиями задания все скомпелированные файлы классов были удалены, а исходный код класса `MySecondClass.java` выведен в отдельный файл `MyFirstPackage.java` в поддиректории `myfirstpackage`.

При попытке компиляции `MyFirstProgram.java` в виде, указанном в задании, компиляция не удалась из-за отсутствия доступа к классу `MySecondClass`.

Для решения этой проблемы потребовалось добавить строку `import myfirstpackage.*` в `MyFirstProgram.java`, а `MyFirstPackage.java` был переименован в `MySecondClass.java`

После этого компиляция и запуск программы прошел успешно.



```
Командная строка
Microsoft Windows [Version 10.0.22631.4890]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Proker>cd C:\Users\Proker\Desktop\OOP\LAB1\Task5

C:\Users\Proker\Desktop\OOP\LAB1\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

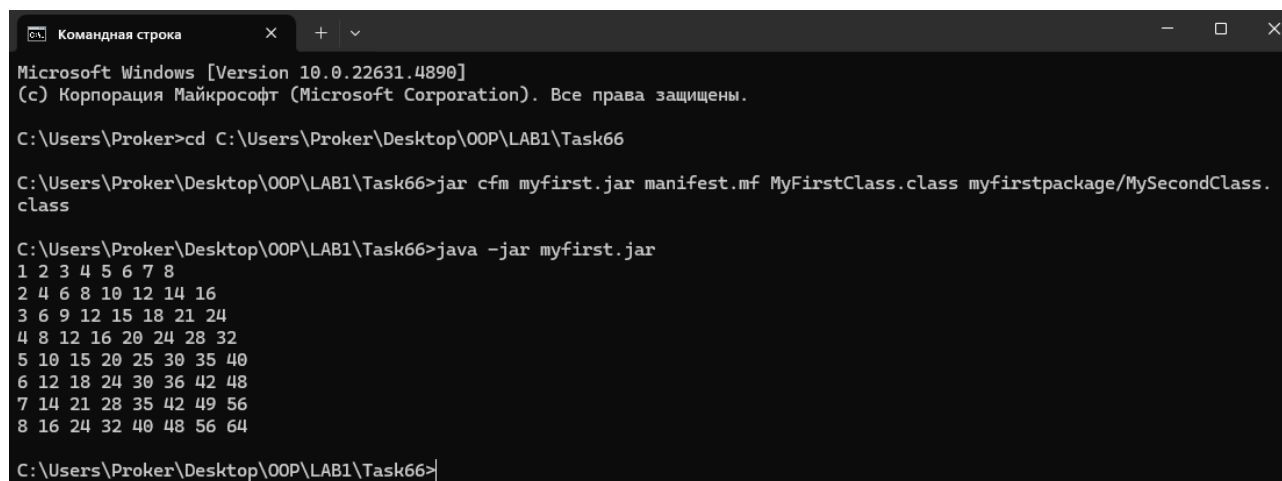
C:\Users\Proker\Desktop\OOP\LAB1\Task5>
```

Рисунок 7 - Работа с консолью при заднии №5

Задание 6

Была запущена утилита `jar` для ознакомления с форматом задания ключей для формирования архивов. В рабочую директорию была перенесена структура каталогов с файлами `.class` из предыдущего задания. Также был создан файл `manifest.mf` с указанием версии, автора и главного класса.

С использованием этого манифеста был создан архив `myfirst.jar`, который был перемещен в отдельную директорию `MyJar` и успешно запущен.



```
Командная строка
Microsoft Windows [Version 10.0.22631.4890]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Proker>cd C:\Users\Proker\Desktop\OOP\LAB1\Task66

C:\Users\Proker\Desktop\OOP\LAB1\Task66>jar cfm myfirst.jar manifest.mf MyFirstClass.class myfirstpackage/MySecondClass.class

C:\Users\Proker\Desktop\OOP\LAB1\Task66>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

C:\Users\Proker\Desktop\OOP\LAB1\Task66>
```

Рисунок 8 - Работа с консолью при заднии №6