

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский университет
имени академика С.П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Кафедра технической кибернетики

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\OOP_JAVA\Lab1SkibaVladimir6204_010302\task4> javac MyFirstProgram.java
PS D:\java\OOP_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. Класс `MySecondClass` перемещён в пакет `myfirstpackage`.
2. Добавлена строка импорта:

```
import myfirstpackage.*;
```

3. Компиляция:

```
javac MyFirstProgram.java
```

Результат:

```
PS D:\java\00P_JAVA\Lab1SkibaVladimir6204_010302\task5> javac MyFirstProgram.java
PS D:\java\00P_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

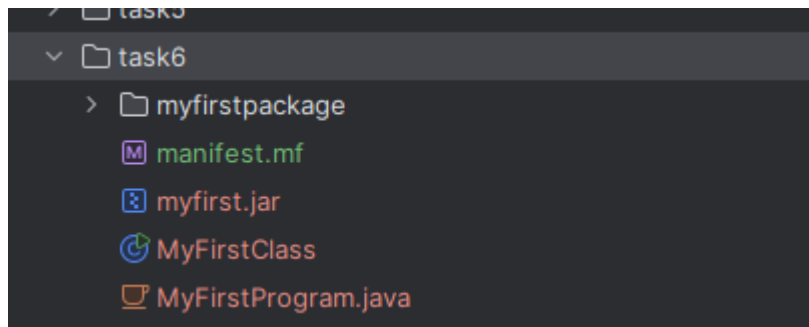
2. Создан JAR-архив:

```
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.

Точка входа в программу является методом `main()` с параметрами `String[] s`.

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

Откомпилируйте и запустите программу для выполнения. Логично предположить, что метод `main()` должен быть статическим, т.к. на момент запуска программы ни один тип объекта `MyFirstClass` не существует. Используйте статический метод, снова откомпилируйте и запустите программу.

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

аницы в отчете кроме титульной должны быть пронумерованы.

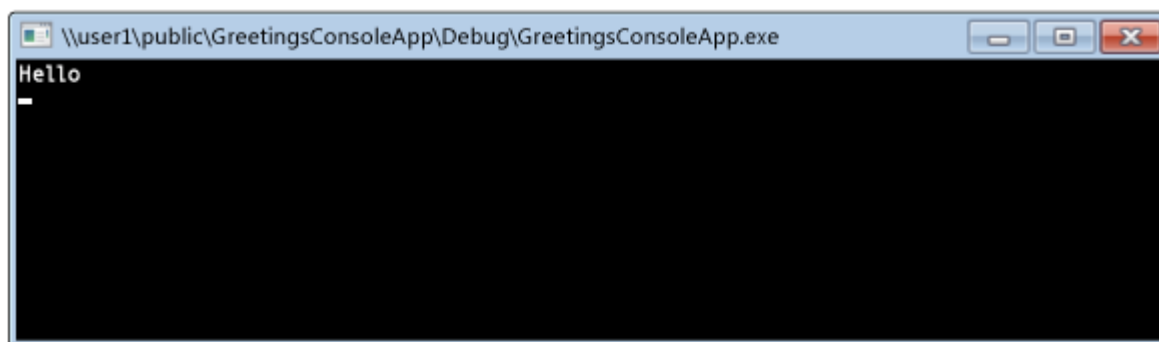


Рисунок 1 – Снимок экрана с результатами выполнения программы «Hello World!»

Формат страницы отчетов – А4, ориентация – книжная (портретная). Обычные поля с отступами сверху – 20 мм, снизу – 20 мм, слева – 30 мм, справа – 15 мм.

Основной текст отчета набирается шрифтом Times New Roman, размер шрифта 14 пт, форматирование по ширине. Междустрочный интервал полуторный, отступ первой строки 1 см. Заголовок отчета набирается шрифтом Times New Roman Bold, размер – 14, форматирование по центру, абзацный отступ – 0 мм.

Подпись к рисунку начинается со слова «Рисунок» и номера по порядку, затем следует тире и сама подпись, выполненная шрифтом Times New Roman, размер – 14 пт,

форматирование по центру, отступ – 0 мм. Наличие подрисовочных подписей является обязательным. Если имеется несколько рисунков, объединённых одной подписью, отдельные части обозначаются русскими буквами: а), б), в) и т.д.

Название таблицы начинается со слова «Таблица» и номера по порядку, затем следует тип и само название, выполненное шрифтом Times New Roman, размер – 14 пт, форматирование по центру, отступ – 0 мм. Наличие названий таблиц является обязательным.

После заголовков, подписей таблиц и рисунков точка не ставится.

Для ЛР №1 в основную часть необходимо поместить:

- 1) Снимок экрана с окном ошибок (должна наличествовать минимум одна ошибка). К рисунку прилагается объяснение причин ошибки, способа исправления.
- 2) Снимок экрана работы программы "Hello World". Описание основных сложностей при написании программы.
- 3) Снимки экрана работы программы, реализующей вариант задания на ЛР №1. Описание основного оператора для реализации требуемого алгоритма. Описание всех особенностей программы (проверки ввода-вывода, возможности повторного запуска и т.д.), если таковые имеются. Снимки экрана должны отражать полный функционал программы.

Тестовые примеры для снимков экрана уточняются у преподавателя.

Отчет проще всего оформить путем вставки в представленный шаблон своих результатов работы. Все рекомендации по оформлению, приведенные в данном шаблоне, в таком случае, конечно же, должны быть удалены или скорректированы.

Заключение

В данном разделе подводятся итоги выполнения ЛР, делаются выводы о полученных результатах и приобретенных навыках. Заключение должно содержать итоги и выводы по всем заданиям на ЛР, включая дополнительное.

Далее приведена формулировка, которая позволит написать заключение по проделанной ЛР. В "_____" вставляем свой текст, лишнее убираем.

Цель лабораторной работы – написать программу, которая _____, достигнута. В ходе выполнения лабораторной работы я изучил(а) _____, приобрел (ла) навыки по _____.

В заключении допускается указать наиболее интересные или сложные аспекты задания. Например: "В ходе выполнения работы наибольшие затруднения вызвали _____." Или: "Самой интересной частью работы оказалась _____."

Отчет должен содержать только описание выполненной ЛР. Все рекомендации по оформлению, приведенные в данном шаблоне должны быть удалены или скорректированы.

Литература

- [1] Безус, Е.А. Лабораторные работы по курсу «Методы программирования» (язык программирования C++): Методические указания [Текст] / Сост. Е.А. Безус, Е.С. Козлова, С.С. Стафеев - Самара, 2019. – 56 с.
- [2] Страуструп, Б. Язык программирования C++ / Б. Страуструп. – М:Бином, 2011 –1136 с.
- [3] Шилдт. Г. C++. Полное руководство / Г. Шилдт. – М: "ООО И.Д. Вильямс", 2019 – 800 с.
- [4] Страуструп, Б. Программирование: принципы и практика использования C++, испр. изд.: Пер. с англ. / Б. Страуструп. – М: "ООО И.Д. Вильямс", 2011 –1244 с.
- [5] Прата, С. Язык программирования C++. Лекции и упражнения , 6-е изд.: Пер. с англ. / С. Прата. – Киев: "Диалектика", 2018 –1248 с.
- [6] Библиографическое описание документа. Общие требования и правила составления [Электронный ресурс] / сост.: В.С. Крылова, С.М. Григорьевская, Е.Ю. Кичигина // Официальный интернет-сайт научной библиотеки Томского государственного университета. – Электрон. дан. – Томск, [2010]. – <http://www.lib.tsu.ru/win/produkcija/metodichka/metodich.html> (дата обращения: 10.09.2019).

Приложение

Сюда вставляем свой код. Чтобы было читабельнее, можно сделать его 10 кеглем (размер шрифта). В правильно оформленном коде комментарии составляют третью часть. Пример:

```
#include "stdafx.h"

#include <iostream>

#include "math.h"
```

```
#include <iomanip>
```

```
using namespace std;
```

```
long double ProverkaX(long double a);
```

```
long double ProverkaAlpha(long double a);
```

```
int main()
```

```
{
```

```
    bool p = true;
```

```
    while (p)
```

```
    {
```

```
        long double x = 0, alpha = 0, sum = 0, An, As, E = 1000;
```

```
        cout << "Enter x>";
```

```
        x = ProverkaX(x);
```

```
        cout << "Enter alpha>";
```

```
        alpha = ProverkaAlpha(alpha);
```

```
        if (alpha - (int)alpha == 0)
```

```
        {
```

```
            for (int n = 1; n <= alpha; n++)
```

```
            {
```

```
                An = (pow(-1.0, n + 1)*pow(x, 2 * n)) / (2 * n*(2 * n - 1));
```

```
                sum = sum + An;
```

```
                As = (pow(-1.0, n + 2)*pow(x, 2 * (n + 1))) / (2 * (n + 1)*(2 * (n + 1) - 1));
```

```
                E = abs(As / sum);
```

```
                cout << left << setw(5) << n << left << setw(15) << left << An << left << setw(15) <<  
sum << left << setw(15) << E << endl;
```

```

    }
}
else
{
    int n = 1;
    while (alpha <= E)
    {

        An = (pow(-1.0, n + 1)*pow(x, 2 * n)) / (2 * n*(2 * n - 1));

        sum = sum + An;

        As = (pow(-1.0, n + 2)*pow(x, 2 * (n + 1))) / (2 * (n + 1)*(2 * (n + 1) - 1));

        E = abs(As / sum);

        cout << left << setw(5) << n << left << setw(15) << left << An << left << setw(15) <<
sum << left << setw(15) << E << endl;

        n++;
    }
}

cout << "Repeat(y/n): ";

char k = ' ';

cin >> k;

if (k == 'y')
    p = true;

else if (k == 'Y')
    p = true;

else
    p = false;

}

```

```
return 0;
```

```
}
```