

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего
образования

Санкт-Петербургский национальный исследовательский университет информационных
технологий механики и оптики

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторная работа № 1

По дисциплине «Технологии программирования»

**Выполнил студент группы
№М32091**

Бабурин Тимур

Проверил

Собенников Виктор Леонидович

***САНКТ-ПЕТЕРБУРГ
2021***

УПРАЖНЕНИЕ 1 КОМПИЛЯЦИЯ И ЗАПУСК ПЕРВОЙ JAVA 1.

Цель упражнения: Научиться работать с компилятором Java и JVM.

Описание упражнения: В этом упражнении Вы напишете, скомпилируете и запустите Вашу первую программу на Java. Обратите внимание на файл *.class, появившийся в текущем каталоге после компиляции. Помните, что программа начинает выполнение с метода `public static void main(String[] args)`.

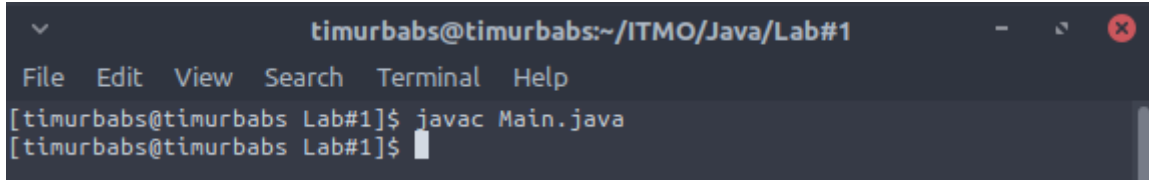
- 1) Текстовый редактор: Создайте новый текстовый файл и введите следующий

java-код:

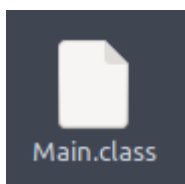
```
class Main{  
    public static void main(String[] args){  
        System.out.println("Starting project");  
    }  
}
```

- 2) Текстовый редактор: Сохраните файл `Main.java`.
- 3) Командная строка: Скомпилируйте файл `Main.java` командой `javac Main.java`

Результатом успешной компиляции станет появление файла `Main.class`.



```
timurbabs@timurbabs:~/ITMO/Java/Lab#1  
File Edit View Search Terminal Help  
[timurbabs@timurbabs Lab#1]$ javac Main.java  
[timurbabs@timurbabs Lab#1]$
```



- 4) Выполните файл `Main.class` командой `java Main`.

```
timurbabs@timurbabs:~/ITMO/Java/Lab#1
File Edit View Search Terminal Help
[timurbabs@timurbabs Lab#1]$ java Main
This is a byte: 120
This is a short: 129
This is a char: a
This is a int: 65999
This is a long: 429496729
This is a float: 0.33333334
This is a double: 0.3333333333333333
This is a boolean: true
Starting project
abcdefghijklmnopqrstuvwxyz
12
[timurbabs@timurbabs Lab#1]$
```

- 5) (Факультативно) Добавьте в исходный код комментарии типа javadoc.

Модифицируйте исходный код следующим образом: объявление класса должно выглядеть так:

```
public class Main{
...
}
```

```
1 /**
2  * Класс "Main"
3  * @author Baburin Timur
4  * @version 1.0
5  */
6 public class Main {
7     /**
8     * main() - основной метод
9     * @param args - массив типа String
10    */
11 }
```

- 6) В случае успешного выполнения задания 5, сгенерируйте html-документацию на Ваш проект командой `javadoc Main.java`. Изучите полученную документацию.

The screenshot shows a web browser displaying the Javadoc HTML documentation for the `Main` class. The browser's address bar shows the file path `/home/timurbabs/ITMO/Java/Lab%231/Main.html`. The page has a navigation bar with tabs for `PACKAGE`, `CLASS` (which is selected), `TREE`, `DEPRECATED`, `INDEX`, and `HELP`. Below the navigation bar, there are links for `SUMMARY: NESTED`, `FIELD`, `CONSTR`, `METHOD`, and `DETAIL: FIELD`, `CONSTR`, `METHC`. The main content area is titled **Class Main** and shows the inheritance hierarchy `java.lang.Object` and `Main`. Below this, the class declaration is shown: `public class Main` and `extends java.lang.Object`. At the bottom, there is a section titled "Класс "Main"".

Method Summary

All Methods

Static Methods

Concrete Methods

Modifier and Type	Method	Description
static void	<code>main(java.lang.String[] args)</code>	<code>main()</code> - основной метод

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Details

main

```
public static void main(java.lang.String[] args)
```

main() - основной метод

Parameters:

`args` -- массив типа String

Упражнение 2 Использование примитивных типов и операторов

Цель упражнения: Научиться использовать переменные и примитивные типы данных языка Java.

Описание упражнения: В этом упражнении вы воспользуетесь базовыми операциями для работы с переменными примитивных типов данных. Далее в курсе вы создадите собственные типы данных, определив классы и интерфейсы в проекте.

Продолжайте работу с файлом `Main.java` из предыдущего упражнения.

- 1) Создайте восемь локальных переменных в методе `main`, по одной каждого типа данных.

```
byte v_byte=120;
short v_short=129;
char v_char='a';
int v_int=65999;
long v_long=429496729;
float v_float=0.33333334f;
double v_double=0.3333333333333333;
boolean v_boolean=true;
```

7) Распечатайте значения каждой переменной.

Например: `System.out.println("This is a byte: "+v_byte);`

```
System.out.println("This is a byte: "+v_byte);
System.out.println("This is a short: "+v_short);
System.out.println("This is a char: "+v_char);
System.out.println("This is a int: "+v_int);
System.out.println("This is a long: "+v_long);
System.out.println("This is a float: "+v_float);
System.out.println("This is a double: "+v_double);
System.out.println("This is a boolean: "+v_boolean);
System.out.println("Starting project");
```

Вывод:

```
[timurbabs@timurbabs Lab#1]$ java Main
This is a byte: 120
This is a short: 129
This is a char: a
This is a int: 65999
This is a long: 429496729
This is a float: 0.33333334
This is a double: 0.3333333333333333
This is a boolean: true
Starting project
abcdefghijklmnopqrstuvwxyz
9
```

8) Добавьте в метод `main` следующие объявления переменных и проверьте их работоспособность. Исправьте ошибки, в случае, если они присутствуют:

- Значение переменной `char` всегда в одинарных кавычках;
- После значения переменной типа `long` всегда ставится `L`;
- После значения переменной типа `float` всегда ставится `f`;
- `double` может принимать только численные значения, а значения `true` `false` может принимать только `boolean`;

```
byte v_byte=120;
short v_short=129;
char v_char='a';
int v_int=65999;
long v_long=429496729L;
float v_float=0.33333334f;
double v_double=0.3333333333333333;
boolean v_boolean=true;
```

Упражнение 3. Использование циклов

Цель упражнения: Научиться использовать конструкции языка Java.

Описание упражнения: В этом упражнении вы используете циклы для выполнения повторяющихся однотипных действий.

- 1) Напишите программу, распечатавшую буквы латинского алфавита от 'a' до 'z':

'a' 'b' ... 'z' *Замечание:* Вспомните, что буквы латинского алфавита представляются в Java значениями типа `char`, хранящими `UNICODE`-коды соответствующих символов. Коды символов 'a'...'z' хранятся в таблице последовательно.

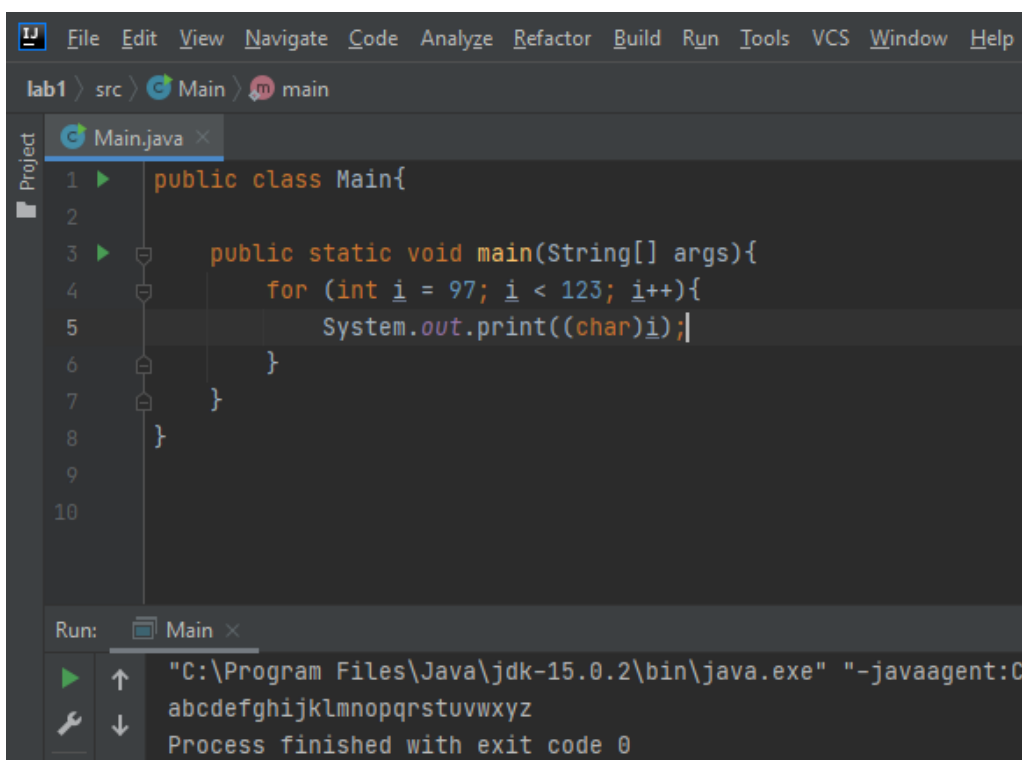
- 9) Напишите программу, изменяющую значение целочисленной переменной `i` от 0 до 1000000000 и проведите замер производительности.

- a) Объявите переменную `begin` типа `long` и инициализируйте ее следующим образом:

```
long begin = new java.util.Date().getTime();
```

Замечание: Данная строка помещает в переменную `begin` количество миллисекунд, прошедшее с 01.01.1970 по настоящее время.

- b) Объявите переменную `i` типа `int` и присвойте ей начальное значение 0: `int i = 0;`



The screenshot shows an IDE window with a Java file named `Main.java`. The code defines a `Main` class with a `main` method that uses a `for` loop to print the letters 'a' through 'z'. The IDE's output window at the bottom shows the command used to run the program and the resulting output, which is the alphabet. The process finished with exit code 0.

```
public class Main{  
    public static void main(String[] args){  
        for (int i = 97; i < 123; i++){  
            System.out.print((char)i);  
        }  
    }  
}
```

Run: Main ×
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-javaagent:C:\Program Files\Java\jdk-15.0.2\bin\javaagent.jar" -Djconsole.port=5044
abcdefghijklmnopqrstuvwxyz
Process finished with exit code 0

- c) Напишите цикл, увеличивающий на каждом витке `i` на 1 до значения 1000000000.
- d) Объявите переменную `end` типа `long` и инициализируйте ее следующим образом:
- ```
long end = new java.util.Date().getTime();
```
- e) Выведите на экран разницу `end-begin`:
- ```
System.out.println(end-begin);
```
- Разность `end-begin` представляет собой (с определенным допуском) время выполнения программы. Показанный здесь способ позволяет достаточно приближенно отслеживать производительность ваших программ.
- f) Измените тип переменной `i` с `int` на `long`. Каким образом изменился результат выполнения?

Время с "long": 9

```
[timurbabs@timurbabs Lab#1]$ java Main
This is a byte: 120
This is a short: 129
This is a char: a
This is a int: 65999
This is a long: 429496729
This is a float: 0.33333334
This is a double: 0.3333333333333333
This is a boolean: true
Starting project
abcdefghijklmnopqrstuvwxyz
9
```

Время с "int": 6

```
[timurbabs@timurbabs Lab#1]$ java Main
This is a byte: 120
This is a short: 129
This is a char: a
This is a int: 65999
This is a long: 429496729
This is a float: 0.33333334
This is a double: 0.3333333333333333
This is a boolean: true
Starting project
abcdefghijklmnopqrstuvwxyz
6
```

Исходный код:

```

30     long begin = new java.util.Date().getTime();
31
32     for(char i = 97; i <= 122; i++) {
33         System.out.print((char)i);
34     }
35     System.out.println("");
36
37     for(long i = 0; i <= 10000000; i++) {}
38
39     /*int i = 0;
40     for (; i<1000000000;) {}*/
41
42     long end = new java.util.Date().getTime();
43     System.out.println(end-begin);

```

Упражнение 5-1. Использование массивов

Цель упражнения: Научиться использовать переменные и примитивные типы данных языка **Java**.

Описание упражнения:

- 1) Объявите в программе массив целых чисел и присвойте его элементам произвольные значения. Например:
`int[] mas = {12,43,12,-65,778,123,32,76};`
- 10) Напишите алгоритм, находящий максимальное число в данном массиве.

```

45     int[] mas = {12,43,12,-65,778,123,32,76};
46     int max = mas[0];
47     for (int i = 0; i < mas.length; ++i) {
48         if (mas[i] > max) {
49             max = mas[i];
50         }
51     }

```

Упражнение 5-2 (Опционально).

Расширенное использование массивов

- 1) Создайте двумерный массив размером 3×3 целых чисел и заполните его случайными значениями от 1 до 10. Элемент [i, j] массива можно заполнить с помощью вызова метода `random()` класса `Math`:

```
matrix[i][j]=(int)Math.round(Math.random()*10);
```

Замечание: Метод `public static double random()` класса `Math` возвращает случайное значение типа `double` в диапазоне от 0 (включительно) до 1 (невключительно).

Метод `public static long round(double d)` округляет параметр `d` до целого.

Транспонируйте полученный массив (поменяйте местами его столбцы и строки).

Например, для массива с числами Транспонированная версия будет
выглядеть как

1 2 3

1 4 7

4 5 6

2 5 8

7 8 9

3 6 9

Исходный код:

```
53     int[][] matrix = new int[3][3];
54     for (int i = 0; i < matrix.length; i++) {
55         for (int j = 0; j < matrix[i].length; j++) {
56             matrix[i][j] = (int) Math.round(Math.random() * 10);
57         }
58     }
59     int t = 0;
60     for (int i = 0; i < matrix.length; i++) {
61         for (int j = i; j < matrix[i].length; j++) {
62             t = matrix[i][j];
63             matrix[i][j] = matrix[j][i];
64             matrix[j][i] = t;
65         }
66     }
67 }
```