МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Факультет информационных технологий и программирования

**Лабораторная работа № 1**

Выполнил студент группы № M3307  
Бойцов Виталий

САНКТ-ПЕТЕРБУРГ  
2019

# 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.
4. Выполните файл Main.class командой java Main.
5. (Факультативно) Добавьте в исходный код комментарии типа javadoc. Модифицируйте исходный код следующим образом: объявление класса должно выглядеть так:  
   public class Main{  
   ….  
   }
6. В случае успешного выполнения задания 5, сгенерируйте html-документацию на Ваш проект командой javadoc Main.java. Изучите полученную документацию.

# 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.
2. Распечатайте значения каждой переменной.   
   Например: System.out.println(“This is a byte: ”+v\_byte);
3. Добавьте в метод main следующие объявления переменных и проверьте их работоспособность. Исправьте ошибки, в случае, если они присутствуют:   
   v\_byte=120  
   v\_short=129  
   v\_char=a  
   v\_int=65999  
   v\_long=4294967296  
   v\_float=0.33333334  
   v\_double=0.3333333333333333  
   v\_double=true

|  |
| --- |
|  |
| public class Main { |
|  | public static void main(String[] args) { |
|  | System.out.println("Starting project"); |
|  | int i = 65999; //width = 32 |
|  | float f = 0.33333334f; //width = 32 |
|  | byte b = 120; //width = 8 |
|  | short s = 129; //width = 16 |
|  | char c = 'a'; //width = 8 |
|  | long l = 120; //width = 64 |
|  | double d = 0.3333333333333333; //width = 64 |
|  | boolean bool = true; |
|  | System.out.println("This is int " + i); |
|  | System.out.println("This is float " + f); |
|  | System.out.println("This is double " + d); |
|  | System.out.println("This is long " + l); |
|  | System.out.println("This is char " + c); |
|  | System.out.println("This is short " + s); |
|  | System.out.println("This is byte " + b); |
|  | System.out.println("This is boolean " + bool); |
|  |  |
|  |  |
|  |  |

# 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, распечатывающую 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.
2. Напишите программу, изменяющую значение целочисленной переменной i от 0 до 100000000 и проведите замер производительности.
   1. 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.
   2. Объявите переменную i типа int и присвойте ей начальное значение 0: int i = 0;
   3. Напишите цикл, увеличивающий на каждом витке i на 1 до значения 100000000.
   4. Объявите переменную end типа long и инициализируйте ее следующим образом:  
       long end = new java.util.Date().getTime();
   5. 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.
   6. Измените тип переменной i c int на long. Каким образом изменился результат выполнения?

|  |
| --- |
| //Упражнение 3.1 |
|  | for (char ch = 'a'; ch <= 'z'; ch++) { |
|  | System.out.println(ch); |
|  | } |
|  | System.out.println(); |
|  | //Упражнение 3.2 |
|  | long begin = new java.util.Date().getTime(); |
|  | long i3 = 0; |
|  | while (i3 <= 100000000) { |
|  | i3++; |
|  | } |
|  | long end = new java.util.Date().getTime(); |
|  | System.out.println("Perfomance " + (end - begin)); |
|  |  |

Время выполнения станет дольше.

# 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};
2. Напишите алгоритм, находящий максимальное число в данном массиве.

int[] mas = {12, 43, 12, -65, 778, 123, 32, 76};

System.out.print(max(mas));