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
```

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.

   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;`

   c) Напишите цикл, увеличивающий на каждом витке `i` на `1` до значения `100000000`.

   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?

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.

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

```
1 2 3
4 5 6
7 8 9
```

Транспонированная версия будет выглядеть как

```
1 4 7
2 5 8
3 6 9
```