**Задачи на тему: Основные управляющие конструкции Java и массивы**

Для оценки времени выполнения того или иного участка Java кода разработчики часто используют метод System.nanoTime()

long startTime = System.nanoTime();

// ... the code being measured ...

long estimatedTime = System.nanoTime() - startTime;

Более подробно об этом методе можно почитать здесь: <http://docs.oracle.com/javase/6/docs/api/java/lang/System.html#nanoTime%28%29>

Для заполнения массивов случайными числами рекомендуется использовать классы Math (метод random()) или класс Random

**1.Вычисление n! факториала**

Необходимо реализовать алгоритм вычисления факториала двумя способами:

1. С помощью цикла
2. С помощью рекурсии

Сравнить время выполнения a и b используя System.nanoTime()

**2.Задачи на использование вложенных циклов**

a. Реализовать программу которая выводит в консоли прямоугольник из символов #

# # # # #

# # # # #

# # # # #

# # # # #

# # # # #

Пример кода:

***int size = 5;***

***for (int row = 1; row <= size; ++row) {***

***for (int col = 1; ......; ......) {***

***......***

***}***

***......***

***}***

b. Реализовать программу, которая рисует следующие фигуры (на выбор 2-3, но можно и все при желании):

# # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # #

(a) (b) (c) (d)

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

# # # # # # # # # #

# # # # # # # # # #

# # # # # # # #

# # # # # # # # # #

# # # # # # # # # #

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

(e) (f) (g) (h) (i)

**3.Задачи на одномерные массивы**

a. Создать массив нечётных чисел от 1 до 99 и вывести их в порядке возрастания и убывания

b. Создать массив из 20 случайных чисел лежащих в отрезке [0;10]. Вывести массив на экран.

Подсчитать и вывести на экран количество чётных и нечётных элементов в массиве.

c. Создайте массив из 10 случайных целых чисел на отрезке [1;100]. Выведите массив на экран в строку. Замените каждый элемент с нечётным индексом на ноль. Выведите новый результат на экран.

d. Создайте массив из 15 случайных целых чисел из отрезка [-50;50]. Найдите максимальный и минимальный элемент массива. Выведите их значения на экран и также индекс последнего вхождения элемента в массив.

e. Создайте два массива из 10 случайных целых чисел в отрезке [0;10]. выведите массивы. Вычислите среднее арифметическое элементов каждого массива и сообщите, для какого из массивов это значение оказалось больше (либо сообщите, что они равны).

f. Создайте массив из 20 случайных целых чисел в отрезке [-1;1] Определите какой элемент встречается в массиве чаще всего и выведите его на экран. Также если несколько элементов встречаются одинаковое количество раз, необходимо их вывести на экран.

**4. Задачи на многомерные массивы**

a. Создать двумерный массив размером 8 на 8. Заполнить его случайными числами в диапазоне [1;99] Вычислить и вывести на экран сумму и произведение элементов главной и побочной диагонали

b. Создать двумерный массив из 8 строк и 5 столбцов. Заполнить его случайными числами в отрезке [-99;99]. Вывести массив на экран. Вывести на экран значение и индексы максимального (по всем строкам и столбцам) элемента в массиве.

c. Cоздать двумерный массив из 8 строк и 5 столбцов из случайных целых чисел в отрезке [-10;10]. Вывести массив на экран. Определить и вывести на экран индекс строки с наибольшим по модулю произведением элементов.

d. Создать двумерный массив из 10 строк и 7 столбцов из целых чисел в отрезке [0;100]. Вывести массив на экран. Отсортировать каждую строку в порядке убывания. Вывести преобразованный массив на экран.

**5 Управляющие конструкции else-if и switch-case**

Реализовать произвольный Enum, состоящий из 5 элементов. Используя полученный Enum реализовать управляющие конструкции else-if и switch-case. Далее в каждую ветку конструкции добавляем простой консольный вывод. Необходимо сравнить время работы данных конструкций с помощью System.nanoTime();

Аналогичные измерения необходимо проделать для массива, состоящего из 50 элементов, заполненного целыми числами в диапазоне [0;10] Т.е. для каждого из возможного значений сделать свою отдельную ветку. Также необходимо подсчитать сколько раз каждый элемент попал в ту или иную ветку.