# Практическое задание №3 Java

1. Задать целочисленный массив, состоящий из элементов 0 и 1. Например: [ 1, 1, 0, 0, 1, 0, 1, 1, 0, 0 ]. С помощью цикла и условия заменить 0 на 1, 1 на 0;

2. Задать пустой целочисленный массив длиной 100. С помощью цикла заполнить его значениями 1 2 3 4 5 6 7 8 … 100;

3. Задать массив [ 1, 5, 3, 2, 11, 4, 5, 2, 4, 8, 9, 1 ] пройти по нему циклом, и числа меньшие 6 умножить на 2;

4. Создать квадратный двумерный целочисленный массив (количество строк и столбцов одинаковое), и с помощью цикла(-ов) заполнить его диагональные элементы единицами (можно только одну из диагоналей, если обе сложно). Определить элементы одной из диагоналей можно по следующему принципу: индексы таких элементов равны, то есть [0][0], [1][1], [2][2], …, [n][n];

5. Написать метод, принимающий на вход два аргумента: len и initialValue, и возвращающий одномерный массив типа int длиной len, каждая ячейка которого равна initialValue;

6. \* Задать одномерный массив и найти в нем минимальный и максимальный элементы ;

7. \*\* Написать метод, в который передается не пустой одномерный целочисленный массив, метод должен вернуть true, если в массиве есть место, в котором сумма левой и правой части массива равны.

\*\*Примеры:  
checkBalance([2, 2, 2, 1, 2, 2, ||| 10, 1]) → true, т.е. 2 + 2 + 2 + 1 + 2 + 2 = 10 + 1  
checkBalance([1, 1, 1, ||| 2, 1]) → true, т.е. 1 + 1 + 1 = 2 + 1

граница показана символами |||, эти символы в массив не входят и не имеют никакого отношения к ИЛИ.

8. \*\*\* Написать метод, которому на вход подается одномерный массив и число n (может быть положительным, или отрицательным), при этом метод должен сместить все элементы массива на n позиций. Элементы смещаются циклично. Для усложнения задачи нельзя пользоваться вспомогательными массивами. Примеры: [ 1, 2, 3 ] при n = 1 (на один вправо) -> [ 3, 1, 2 ]; [ 3, 5, 6, 1] при n = -2 (на два влево) -> [ 6, 1, 3, 5 ]. При каком n в какую сторону сдвиг можете выбирать сами.