**Массивы**

Если в переменной мы хранили лишь одну переменную, то в массиве можно хранить множество значений, но значения должны быть одного типа.

Массивы бывают одномерными, двумерными, трёхмерными и так далее. Рассмотрим в начале одномерные массивы.

Размер массива записывают в квадратных скобках [ ],

Имя массива – arr

в массиве будет храниться 3 элемента

int [] arr = new int [3]; // создали массив с именем **arr** и в нем укажем **3** элемента

Если мы хотим элементам присвоить значение, то пишем:

arr [0] = 20; //присвоили значение первому элементу

arr [1] = 40; //присвоили значение второму элементу

arr [2] = 60; //присвоили значение третьему элементу



вывод на экран последнего элемента:

System.out.print( arr[2] );

Запишем немного короче, удалим переменные и пропишем значения в фигурных скобках:

int [] arr = new int [] {20,40,60} ;

System.out.print( arr[2] );

Мы опустили размер массива, поставили фигурные скобки после квадратных и перечислили все значения через запятую. Размер массива теперь определяется числом элементов в фигурных скобках.

Но нет предела совершенству, сократим еще немного нашу запись:



int [] arr = {20,40,60};

System.out.println( arr[2] );

После знака «=» остались только фигурные скобки с перечислением значений через запятую.

Чтобы вывести все элементы в массиве в консоль нужно воспользоваться циклом for:

for (int i=0; i<3; i++){

System.out.println( arr [ i ] );

}

**Задание 1**

Создайте массив из всех чётных чисел от 2 до 20 и выведите элементы массива на экран сначала в строку, отделяя один элемент от другого пробелом, а затем в столбик

**Задание 2**

Создайте массив из всех нечётных чисел от 1 до 99, выведите его на экран в строку, а затем этот же массив выведите на экран тоже в строку, но в обратном порядке

(1 3 5 … 95 97 99)

(99 97 95 … 5 3 1)