

Массивы. Array

Массивы в Java — это структура данных, которая хранит упорядоченные коллекции фиксированного размера элементов определенного типа.

Упражнения

1. Нумерация элементов массива

Выведите на экран первый и последний элементы массивов.

```
// Exercise 1  
  
int[] ar1 = {10, 20, 30};  
int[] ar2 = new int[3];
```

2. foreach

Используйте циклы

```
for (int element : a) System.out.print(element+ " ") ;  
  
for (int i = 0; i <a.length ; i++)  
    System.out.print(a[i]+ " ");
```

и выведите на экран элементы всех объявленных и инициализированных массивов.
Выведите на экран длину массива `a.length`

```
// Exercise 2  
  
// объявление массивов  
int[] a;  
double[] b;  
  
// инициализация массивов  
a=new int[10];  
int n = 5;  
b=new double[n];  
char c[]={ 'p', 'q', 'r' };  
  
String d[]={ "ab", "cd" };
```

1

3. Arrays.toString(a)

Подключите библиотеку

```
import java.util.Arrays;
```

и выведите на экран элементы массива `a` с помощью команды

```
System.out.println(Arrays.toString(a));
```

4. Двумерный массив

Опишите двумерный массив

```
double[][] ar = new double[2][3];  
заполните его, используя вложенные циклы  
  
for (int i = 0; i < 2 ; i++) {  
    for (int j = 0; j < 3; j++) {  
        ar[i][j] = Math.pow((j + 1) , (i + 1));  
    }  
}
```

5. Arrays.deepToString(ar)

Выведите на экран элементы массива **ar** с помощью команды

```
System.out.println(Arrays.deepToString(ar));
```

6. По строкам и по столбцам

Инициализируйте двумерный массив

```
double[][] ar2 = {{1,10, 100},{2,20, 200},{3,30,300}};
```

Выведите на экран элементы массива **ar2** сначала по строкам, а потом по столбцам, используя вложенные циклы

```
for(int i = 0; i<ar2.length; i++){  
    for(int j = 0; j<ar2[i].length; j++){  
        System.out.print(ar2[i][j] + " ");  
    }  
    System.out.println();  
}
```

2

7. Неровные массивы

Инициализируйте двумерный неровный массив

```
double[][] ar7 = {{1},{10,20},{100,200,300}};
```

Выведите на экран элементы массива **ar7**

```
System.out.println(Arrays.deepToString(ar7));
```

8. Инициализация неровного массива

Опишите двумерный неровный массив

```
int[][] ar8 = new int[5][];// двумерный массив с 5 строками
```

Выведите на экран элементы массива **ar8**

```
System.out.println(Arrays.deepToString(ar8));
```

Выделите память для каждой строки массива **ar8**

```
ar8[0] = new int [1];  
ar8[1] = new int [2];
```

```
ar8[2] = new int [3];
ar8[3] = new int [4];
ar8[4] = new int [5];
```

Выведите на экран элементы массива **ар8**

```
System.out.println(Arrays.deepToString(ar8));
```

9. Заполнение массива **Arrays.fill(массив, значение)** ;

Заполните массив **ар8**

```
Arrays.fill(ar8[0], 1);
Arrays.fill(ar8[1], 2);
Arrays.fill(ar8[2], 3);
Arrays.fill(ar8[3], 4);
Arrays.fill(ar8[4], 5);
```

Выведите на экран элементы массива **ар8**

10. Многомерные массивы

Инициализируйте массив **ар10**

```
int[][][] ar10 = {{{1, 2, 3}, {4, 5, 6}}, {{10, 20, 30}, {40, 50, 60}}, {{100, 200, 300}, {400, 500, 600}}, {{1000, 2000, 3000}, {4000, 5000, 6000}}};
```

3

Выведите на экран размерность массива **ар10**

```
System.out.println(ar9.length);
System.out.println(ar9[0].length);
System.out.println(ar9[0][0].length);
```

11. проверка на идентичность **Arrays.equals(a1,a2)**

Инициализируйте массивы

```
int[] d1={1,2};
int[] d2={1,2};
```

Проверьте идентичность массивов

```
System.out.println(Arrays.equals(d1,d2));
```

12. проверка на идентичность **Arrays.deepEquals(a1,a2)**

Инициализируйте массивы

```
int[][] d3={{1,2},{3,4}};
int[][] d4=d3;
```

Проверьте идентичность массивов `System.out.println(Arrays.deepEquals(d3,d4));`

13. преобразование строкового массива в список `Arrays.asList(a)`

```
String aa[] = new String[]{"abc","klm","xyz","pqr"};  
List list1 = Arrays.asList(aa);  
System.out.println(Arrays.asList(aa));
```

14. копирование диапазона массива `Arrays.copyOfRange(a, from: , to:)`

```
int[] z = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};  
int[] y = Arrays.copyOfRange(z, 3, 7);  
System.out.println(Arrays.toString(y));
```

15. копирование массива `Arrays.copyOf(a, n)` ;

```
int[] A = { 1, 2, 3, 4, 5};  
int[] B = Arrays.copyOf(A, A.length);  
System.out.println(Arrays.toString(B));  
  
A = Arrays.copyOf(A, 2*A.length);  
System.out.println(Arrays.toString(A));
```

16. сортировка массива `Arrays.sort(a)`

```
int [] C={1,5,2,3,4};  
Arrays.sort(C);  
System.out.println(Arrays.toString(C));
```

17. бинарный поиск в массиве `Arrays.binarySearch(a, key)`

Чтобы осуществить бинарный поиск нужно сначала выполнить операцию сортировки

```
byte byteArr[] = {10,20,15,22,35};  
char charArr[] = {'g','p','q','c','i'};  
int intArr[] = {10,20,15,22,35};  
double doubleArr[] = {10.2,15.1,2.2,3.5};  
float floatArr[] = {10.2f,15.1f,2.2f,3.5f};
```

```

short shortArr[] = {10, 20, 15, 22, 35};

Arrays.sort(byteArr);
Arrays.sort(charArr);
Arrays.sort(intArr);
Arrays.sort(doubleArr);
Arrays.sort(floatArr);
Arrays.sort(shortArr);

byte byteKey = 35;
char charKey = 'g';
int intKey = 22;
double doubleKey = 1.5;
float floatKey = 35;
short shortKey = 5;

System.out.println(byteKey + " found at index = "
    + Arrays.binarySearch(byteArr, byteKey));
System.out.println(charKey + " found at index = "
    + Arrays.binarySearch(charArr, charKey));
System.out.println(intKey + " found at index = "
    + Arrays.binarySearch(intArr, intKey));
System.out.println(doubleKey + " found at index = "
    + Arrays.binarySearch(doubleArr, doubleKey));
System.out.println(floatKey + " found at index = "
    + Arrays.binarySearch(floatArr, floatKey));
System.out.println(shortKey + " found at index = "
    + Arrays.binarySearch(shortArr, shortKey));

```

5

18. java.util.ArrayList. Добавление элементов add

ArrayList – автоматически расширяемый массив.

Подключите библиотеку

```
import java.util.ArrayList;
```

Опишите списочный массив

```
ArrayList <String> list = new ArrayList <String>();
```

Добавьте элементы массива

```
list.add("a");
list.add("b");
list.add("c");
```

Выведите размер списочного массива

```
System.out.println ("Размер массива равен " + Integer.valueOf (list.size()) + " элементам");
```

19. java.util.ArrayList. set и get

Чтобы вернуть элемент массива по индексу, используйте `get`.
Чтобы установить новое значение элемента, используйте `set`.

```
ArrayList <String> list = new ArrayList <String>();  
list.add("a");  
list.add("b");  
  
System.out.println ("Первый элемент массива = "+ list.get(0) );  
  
list.set(0,"A");  
System.out.println ("Первый элемент массива = "+ list.get(0) );
```

20. java.util.ArrayList. Удаление элементов remove

Можно удалить элемент, указав его индекс. Можно удалить элемент, указав сам элемент.

```
ArrayList <String> list = new ArrayList <String>();  
list.add("a");  
list.add("b");  
list.add("c");  
  
list.remove(1); // удаляем по индексу  
  
list.remove("c"); // удаляем по объекту  
  
System.out.println(list);
```

21. java.util.ArrayList. Очистка всего массива clear

```
ArrayList <String> list = new ArrayList <String>();  
list.add("a");  
list.add("b");  
list.add("c");  
list.clear();  
  
System.out.println(list);  
System.out.println(list.size());
```

6

22. java.util.ArrayList. Определение позиции indexOf

```
list.add("a");  
list.add("Pink");  
list.add("c");  
int index = list.indexOf("Pink");  
  
System.out.println(index);
```

23. java.util.ArrayList. Проверка на наличие элемента. contains

```
System.out.println (list.contains("Pink") );
```

24. java.util.ArrayList. Преобразование в массив toArray

```

ArrayList<String> days = new ArrayList<>();
days.add("Monday");
days.add("Tuesday");
days.add("Wednesday");
days.add("Thursday");
days.add("Friday");
days.add("Saturday");
days.add("Sunday");

String[] daysArray = days.toArray(new String[days.size()]);

System.out.println("Массив: " + Arrays.toString(daysArray));

```

Задания

Создайте методы, которые выполняют следующие действия:

1. Создать массив четных чисел в диапазоне от n до m.
2. Вывод на экран элементов массива, записанных в строку, или в столбец, в зависимости от параметра.
3. Подсчет суммы четных и нечетных элементов для целочисленного массива.
4. Инвертирование массива.
5. Создание массива из n случайных целых чисел из отрезка [a,b].
6. Замена каждого элемента с нечётным индексом на ноль.
7. Подсчет среднего арифметического для целочисленного массива
8. Определение - является ли массив строго возрастающей последовательностью.
9. Создание массива n первых чисел Фибоначчи.
10. Определение максимального элемента массива и его индекса.
11. Определение наиболее часто встречающегося элемента массива.
12. Пользователь должен указать с клавиатуры чётное положительное число, а программа должна создать массив указанного размера из случайных целых чисел из [-5;5] и вывести его на экран в строку. После этого программа должна определить и сообщить пользователю о том, сумма модулей какой половины массива больше: левой или правой, либо сообщить, что эти суммы модулей равны. Если пользователь введёт неподходящее число, то программа должна требовать повторного ввода до тех пор, пока не будет указано корректное значение.
13. Программа должна создать массив из 12 случайных целых чисел из отрезка [-10;10] таким образом, чтобы отрицательных и положительных элементов там было поровну и не было нулей. При этом порядок следования элементов должен быть случаен (т. е. не подходит вариант, когда в массиве постоянно выпадает сначала 6 положительных, а потом 6 отрицательных чисел или же когда элементы постоянно чередуются через один и пр.). Вывести полученный массив на экран.

14. Пользователь вводит с клавиатуры натуральное число большее 3, которое сохраняется в переменную n . Если пользователь ввёл не подходящее число, то программа должна просить пользователя повторить ввод. Создать массив из n случайных целых чисел из отрезка $[0;n]$ и вывести его на экран. Создать второй массив только из чётных элементов первого массива, если они там есть, и вывести его на экран.
15. Создать двумерный массив из 8 строк по 5 столбцов в каждой из случайных целых чисел из отрезка $[10;99]$. Вывести массив на экран
16. Создать двумерный массив из 5 строк по 8 столбцов в каждой из случайных целых чисел из отрезка $[-99;99]$. Вывести массив на экран. После на отдельной строке вывести на экран значение максимального элемента этого массива (его индекс не имеет значения).
17. Создать двумерный массив из 7 строк по 4 столбца в каждой из случайных целых чисел из отрезка $[-5;5]$. Вывести массив на экран. Определить и вывести на экран индекс строки с наибольшим по модулю произведением элементов. Если таких строк несколько, то вывести индекс первой встретившейся из них.
18. Создать двумерный массив из 6 строк по 7 столбцов в каждой из случайных целых чисел из отрезка $[0;9]$. Вывести массив на экран. Преобразовать массив таким образом, чтобы на первом месте в каждой строке стоял её наибольший элемент. При этом изменять состав массива нельзя, а можно только переставлять элементы в рамках одной строки. Порядок остальных элементов строки не важен (т.е. можно совершить только одну перестановку, а можно отсортировать по убыванию каждую строку). Вывести преобразованный массив на экран.
19. Для проверки остаточных знаний учеников после летних каникул, учитель младших классов решил начинать каждый урок с того, чтобы задавать каждому ученику пример из таблицы умножения, но в классе 15 человек, а примеры среди них не должны повторяться. В помощь учителю напишите программу, которая будет выводить на экран 15 случайных примеров из таблицы умножения (от $2*2$ до $9*9$, потому что задания по умножению на 1 и на 10 — слишком просты). При этом среди 15 примеров не должно быть повторяющихся (примеры $2*3$ и $3*2$ и им подобные пары считать повторяющимися).