

Plan

2023-09-25

Повторение и практика

Theory

Theory for the week

Arrays

Given an array `int[] array = new array[10];`

How to find out the length of an array? - What element is at the beginning of the array? - What element is at the end of the array? - What does the current array element look like? - What does the next array element after the current one look like? - What does the previous one before the current array element look like? -

Filling an array with random data

- `double a = Math.random();` - generates a random number in the range from $[0, 1)$ - parentheses from mathematics $[\Rightarrow 0$ - maybe $) \Rightarrow 1$ - can't be
- `double x = (Math.random() * (b-a)) + a` - generates a random number in the interval from $[a, b)$ ($a < b$) a - maybe, b - can't be
- `int n = (int)(Math.random() * (b - a + 1) + a)` - generates a random **integer** number in the interval $[a, b]$ a - maybe, b - maybe

The for loop loops through all the elements of the array

```
for( i = 0, i < array length, i++) {  
    doSomething;  
}
```

Теория за неделю

Массивы

Задан массив `int[] array = new array[10];` - длина массива 10

Как узнать длину массива? - `array.length`, [написана скобках]

Какой элемент стоит в начале массива? - `array[0]`, это первый элемент массива

Какой элемент стоит в конце массива? - `array[lenght - 1]`, это последний элемент массива

Как выглядит текущий элемент массива? - `array[i]`

Как выглядит следующий за текущим элементом массива? - `array[i + 1]`

Как выглядит предыдущий перед текущим элементом массива? - `array[i - 1]`

Заполнение массива случайными данными

- `double a = Math.random();` - генерирует случайное число в интервале от `[0, 1)` - скобки из математики `[` => 0 - может быть, `)` => 1 - не может быть
- `double x = (Math.random() * (b-a)) + a` - генерирует случайное число в интервале от `[a, b)` (`a < b`) `a` - может быть, `b` - не может быть
- `int n = (int)(Math.random() * (b - a + 1) + a)` - генерирует случайное **целое** число в интервале `[a, b]` `a` - может быть, `b` - может быть

Цикл for обеспечивает "обегание" всех элементов массива

```
for( i = 0, i < длины массива, i++) {  
    doSomething;  
}
```

Сортировки массивов

- Сортировка выбором
- Пузырькова (bubble sort)

```
public class Main {  
    public static void bubbleSort(int[] arr) {  
        for (int i = 0; i < arr.length - 1; i++) {  
            for (int j = 0; j < arr.length - 1 - i; j++) { // правый конец  
                if (arr[j] > arr[j + 1]) { // если левый из соседей больше  
                    int tmp = arr[j];  
                    arr[j] = arr[j + 1];  
                    arr[j + 1] = tmp;  
                }  
            }  
        }  
    }  
}
```

Стандартные задачи и операции с массивами:

- поиск элемента массива;
- добавление нового элемента (в начало, в конец, в середину);
- редактирование(изменение) элемента массива;
- удаление элемента массива;
- поиск максимального/минимального значения;
- сортировка по возрастанию/убыванию;
- удаление дубликатов;
- сдвиг массива;
- удаление пустых (нулевых) элементов.

Задачи

Задача 1: Сравнение эффективности линейного и бинарного поиска

Описание

Сравнить эффективность линейного и бинарного поиска, определив количество итераций, которое потребуется для каждого метода.

Дано

- Отсортированный массив целых чисел.
- Целевое значение для поиска.

Вывод

- Количество итераций для линейного поиска.
- Количество итераций для бинарного поиска.

Примеры

Пример

Дано: Массив [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], Целевое значение 7

Вывод: Линейный поиск: 7 итераций, Бинарный поиск: 3 итерации

Ограничения

- Массив не пустой и содержит только уникальные значения.
- Массив уже отсортирован в возрастающем порядке.

Задача 2: Клонирование массива с использованием циклов for и while

Описание

Ваша задача создать копию исходного массива, используя циклы `for` и `while`.

Дано

- Массив целых чисел.

Вывод

- Новый массив, являющийся копией исходного массива.

Примеры

Пример 1

Дано: [1, 2, 3, 4, 5]

Вывод: [1, 2, 3, 4, 5]

Ограничения

- Не использовать библиотеки или встроенные функции для клонирования массива.

Дополнительные задачи

Задача 3: Генерация и вывод первых N чисел Фибоначчи

Описание

Создайте массив, который будет содержать первые N чисел последовательности Фибоначчи. После этого выведите этот массив на экран.

Дано

- Число N, определяющее количество чисел Фибоначчи для генерации.

Вывод

- Массив из первых N чисел Фибоначчи.

Примеры

Пример 1

Дано: N = 5

Вывод: [1, 1, 2, 3, 5]

[\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # ()

[\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # () [\:](#) # ()

Homework

► English

▼ На русском

Задача 1. Создать массив из 20-ти случайных целых чисел из интервала от -100 до 100. Выполните сортировку полученного массива. Попало ли число 0 в этот массив? (выяснить с помощью `binary search`) Если да, то на какое место (индекс) в массиве? Если нет, то на какой индекс его следует поставить?

Задача 2. Создать массив из 20-ти случайных целых чисел из интервала от -100 до 100. Реализуйте метод, который получает на вход исходный массив и возвращает массив, содержащий только положительные числа из исходного массива.

Задача 3.(*). Реализовать способ обмена значениями двух переменных целого типа, не используя третьей переменной. В идеале написать метод `swap(a, b)`.

Code

code/HWSolution14/src/Task1.java

```
import java.util.Arrays;

/**
 * @author Andrej Reutow
 * created on 22.09.2023
 */
public class Task1 {

    public static void main(String[] args) {
        System.out.println(Arrays.toString(createArrayOfIntegers1(10)));
        System.out.println(Arrays.toString(createArrayOfIntegers2(10)));
        System.out.println(Arrays.toString(createArrayOfIntegers3(10)));

        // Output: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
    }
}
```

```
public static int[] createArrayOfIntegers1(int n) {
    int[] array = new int[n];
    for (int i = 1; i <= n; i++) {
        array[i - 1] += i * 2;
    }

    return array;
}

public static int[] createArrayOfIntegers2(int n) {
    int[] array = new int[n];
    for (int i = 0; i < n; i++) {
        if (i % 2 == 0) {
            array[i] += i;
        }
    }

    return array;
}

public static int[] createArrayOfIntegers3(int n) {
    int[] array = new int[n];
    for (int i = 1; i < n; i++) {
        if (i % 2 != 0) {
            array[i] += i * 2;
        }
    }

    return array;
}
}
```

code/HWSolution14/src/Task2.java

```
import java.util.Arrays;

/**
 * @author Andrej Reutow
 * created on 24.09.2023
 */
public class Task2 {

    public static void main(String[] args) {
        int[] source = {10, 4, 2, 3, 1};
```

```
int elementN = 10;

int minElementIndex = findMinElementIndex(source);
int elementIndex = findElementIndex(source, elementN);

if (elementIndex >= 0) {
    source[elementIndex] = source[minElementIndex];
} else {
    System.out.println("Element " + elementN + " not found");
}

System.out.println(Arrays.toString(source));
}

private static int findElementIndex(int[] source, int element) {
    for (int i = 0; i < source.length; i++) {
        if (source[i] == element) {
            return i;
        }
    }

    return -1;
}

private static int findMinElementIndex(int[] source) {
    int minElementIndex = 0;
    int min = source[minElementIndex];
    for (int i = 0; i < source.length; i++) {
        if (source[i] < min) {
            min = source[i];
            minElementIndex = i;
        }
    }

    return minElementIndex;
}

}
```

code/HWSolution14/src/Task3.java

```
import java.util.Arrays;

/**
 * @author Andrej Reutow
 * created on 22.09.2023
 */
public class Task3 {
    public static void main(String[] args) {
        int[] mas = new int[10];

        for (int i = 0; i < mas.length; i++) {
            mas[i] = (int) (Math.random() * 90) + 10;
        }
        System.out.println(Arrays.toString(mas));

        Arrays.sort(mas);
        System.out.println(Arrays.toString(mas));

        boolean flag = true;
        for (int i = 1; i < mas.length; i++) {
            if (mas[i] <= mas[i - 1]) {
                flag = false;
                break;
            }
        }
        flag = isIncreasingSequence2(mas);

        if (flag) {
            System.out.println("Массив является строго возрастающей последо
        } else {
            System.out.println("Массив не является строго возрастающей посл
        }
    }

    public static boolean isIncreasingSequence(int[] arr) {
        for (int i = 0; i < arr.length - 1; i++) {
            if (arr[i] >= arr[i + 1]) {
                return false;
            }
        }
        return true;
    }
}
```



```
public static boolean isIncreasingSequence2(int[] arr) {
    for (int i = 0; i < arr.length - 1; i++) {
        if (arr[i] > arr[i + 1]) { // Изменили здесь
            return false;
        }
    }
    return true;
}
}
```

code/HWSolution14/src/Task4.java

```
/**
 * @author Andrej Reutow
 * created on 22.09.2023
 */
public class Task4 {
    public static void main(String[] args) {

        int[] arr = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
        int evenCount = 0;

        // Подсчет количества четных чисел в массиве
        for (int i = 0; i < arr.length; i++) {
            int num = arr[i];
            if (num % 2 == 0) {
                evenCount++;
            }
        }

        // Создание массива для четных чисел
        int[] evenArr = new int[evenCount];
        int index = 0;

        // Заполнение массива четными числами
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] % 2 == 0) {
                evenArr[i] = arr[i];
            }
        }

        // Сортировка второго массива
        for (int i = 0; i < evenArr.length; i++) {
```

```

        for (int j = 0; j < evenArr.length - i - 1; j++) {
            if (evenArr[j] > evenArr[j + 1]) {
                int temp = evenArr[j];
                evenArr[j] = evenArr[j + 1];
                evenArr[j + 1] = temp;
            }
        }
    }

    // Вывод обоих массивов
    System.out.print("Исходный массив: ");
    for (int num : arr) {
        System.out.print(num + " ");
    }
    System.out.println();

    System.out.print("Массив четных чисел: ");
    for (int num : evenArr) {
        System.out.print(num + " ");
    }
}
}

```

code/HWSolution14/src/Task5.java

```

/**
 * @author Andrej Reutow
 * created on 22.09.2023
 */
public class Task5 {

    public static void main(String[] args) {
        int[] arr1 = {1, -10, 70, 101};
        int[] arr2 = {1, 99};

        findIndexes(arr1, arr2);
    }

    public static void findIndexes(int[] arr1, int[] arr2) {
        for (int j = 0; j < arr2.length; j++) {
            int num = arr2[j];
            boolean found = false;
            for (int i = 0; i < arr1.length; i++) {
                if (arr1[i] == num) {

```

```
        System.out.println("Число " + num + ", найдено, индекс  
        found = true;  
        break;  
    }  
}  
if (!found) {  
    System.out.println("Число " + num + ", индекс не найден");  
}  
}  
}  
}
```

code/Lesson15/src/Task1.java

```
/**  
 * @author Andrej Reutow  
 * created on 25.09.2023  
 */  
  
//Задача 1: Сравнение эффективности линейного и бинарного поиска  
public class Task1 {  
    public static void main(String[] args) {  
        System.out.println("Hello world!");  
    }  
}
```

code/Lesson15/src/Task2.java

```
import java.util.Arrays;  
  
/**  
 * @author Andrej Reutow  
 * created on 25.09.2023  
 */  
public class Task2 {  
    //Задача 2: Клонирование массива с использованием циклов for и while  
    //Описание  
    //Ваша задача создать копию исходного массива, используя циклы for и while.  
    //  
    //Дано  
    //Массив целых чисел.  
    //Вывод  
    //Новый массив, являющийся копией исходного массива.  
    //Примеры
```

```
//Пример 1
//Дано: [1, 2, 3, 4, 5]
//Вывод: [1, 2, 3, 4, 5]

public static void main(String[] args) {
    int[] arrayToCopy = new int[10];

    fillArray(arrayToCopy, 88);

    System.out.println(Arrays.toString(arrayToCopy));

    //      int[] arrayCopy = copyArray(arrayToCopy);
    //
    //      printArray(arrayToCopy);
    //      printArray(arrayCopy);
    ////      System.out.println(Arrays.toString(arrayToCopy));
    ////      System.out.println(Arrays.toString(arrayCopy));
    //
    //      int[] arrayCopy2 = copyArrayTo(arrayToCopy, 3);
    //      printArray(arrayCopy2);

    int[] arrayCopy3 = copyArrayFromTo(arrayToCopy, 2, 5);
    printArray(arrayCopy3);
}

public static void fillArray(int[] arrayToFill, int from) {
    for (int i = 0; i < arrayToFill.length; i++) {
        arrayToFill[i] = from++;
    }

    //      если from = 88
    //      arrayToFill[0] = 88;
    //      arrayToFill[1] = 89;
    //      arrayToFill[2] = 90;
    //      arrayToFill[3] = 91;
}

public static int[] copyArray(int[] source) {
    int[] newArray = new int[source.length];

    for (int i = 0; i < source.length; i++) {
        newArray[i] = source[i];
    }
}
```

```

        return newArray;
    }

    public static int[] copyArrayTo(int[] source, int toLength) {
        int[] newArray = new int[toLength];

        for (int i = 0; i < newArray.length; i++) {
            newArray[i] = source[i];
        }

        return newArray;
    }

    public static int[] copyArrayFromTo(int[] source, int fromLength, int toLength) {
        // fromLength = 5
        // toLength = 10
        // size = 10 - 5
        int size = toLength - fromLength;
        int[] newArray = new int[size + 1];
        for (int i = 0, j = fromLength; i < newArray.length; i++, j++) {
            newArray[i] = source[j];
        }

        return newArray;
    }

    public static void printArray(int[] array) {
        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i] + "\t");
        }
        System.out.println();
    }
}

```

code/Lesson15/src/MyArrayUtils.java

```

/**
 * @author Andrej Reutow
 * created on 25.09.2023
 */

```

```

//поиск элемента массива;
//добавление нового элемента (в начало, в конец, в середину);

```

```
//редактирование(изменение) элемента массива;
//удаление элемента массива;
//поиск максимального/минимального значения;
//сортировка по возрастанию/убыванию;
//удаление дубликатов;
//сдвиг массива;
public class MyArrayUtils {

    public static void main(String[] args) {
        int[] testArray = {1, 10, 1111082, -5, 20, 1111082, 1, 10, 1, 1};

        printArray(testArray);

        sortAz(testArray);

        printArray(testArray);

        //      int findIndex = findIndexByElement(testArray, 1111082);
        //      System.out.println(findIndex);
        //
        //      if (findIndex >= 0) {
        //          System.out.println("Element is found");
        //      } else {
        //          System.out.println("Element not found");
        //      }
        //
        //      findIndex = findIndexByElementN(testArray, 1, 4);
        //      System.out.println("4 " + findIndex); // 9
        //
        //      findIndex = findIndexByElementN(testArray, 1, 2);
        //      System.out.println("2 " + findIndex); // 6
        //
        //      findIndex = findIndexByElementN(testArray, 1, 100);
        //      System.out.println("100 " + findIndex); // -1
        //
        //      findIndex = findIndexByElementN(testArray, 7777, 1);
        //      System.out.println("1 " + findIndex); // -1
        //
        //      System.out.println(Arrays.toString(testArray));
        //      removeElement(testArray, testArray.length - 1);
        //
        //      int maxElement = findMaxElement(testArray);
```

```
//      System.out.println(maxElement);
//
//      int minElement = findMinElement(testArray);
//      System.out.println(minElement);
}

public static void printArray(int[] arrayToPrint) {
//      for (int i = 0; i < arrayToPrint.length; i++) {
//          System.out.print(arrayToPrint[i] + " ");
//      }

    int index = 0;
    while (index < arrayToPrint.length) {
        System.out.print(arrayToPrint[index] + " ");
        index++;
    }

    System.out.println();
}

/**
 * @param sourceArr    исходный массив
 * @param searchValue значение для поиска в массиве
 * @return индекс найденного значения. Если значение не найдено возвращает -1
 */
public static int findIndexByElement(int[] sourceArr, int searchValue) {
    int elementIndex = -1;
    for (int i = 0; i < sourceArr.length; i++) {
        if (sourceArr[i] == searchValue) {
//            return i;
            elementIndex = i;
            break;
        }
    }

    return elementIndex;
}

/**
 * @param sourceArr    исходный массив
 * @param searchValue значение для поиска в массиве
 * @param n             количество вхождений искомого элемента в массив
 */
```

```
* @return индекс найденного значения. Если значение не найдено возвращает -1;
*/

public static int findIndexByElementN(int[] sourceArr, int searchValue,
    int counter = 0;

    for (int i = 0; i < sourceArr.length; i++) {
        if (sourceArr[i] == searchValue) {
            counter++;
            if (counter == n) {
                return i;
            }
        }
    }

    return -1;
}

/**
 * добавление нового элемента;
 *
 * @param sourceArr исходный массив
 * @param index индекс для вставки элемента
 * @param value значение для вставки по index
 */
public static void setElement(int[] sourceArr, int index, int value) {
    sourceArr[index] = value;
}

/**
 * @param sourceArr исходный массив
 * @param index индекс для изменения элемента
 * @param value значение
 */
//редактирование(изменение) элемента массива;
public static void editElement(int[] sourceArr, int index, int value) {
    setElement(sourceArr, index, value);
}

// удалить элемент по индексу - установить значение по умолчанию
public static void removeElement(int[] sourceArr, int index) {
    // sourceArr[index] = 0;
    setElement(sourceArr, index, 0);
}
```



```
// нужно вернуть максимальный элемент в массиве
public static int findMaxElement(int[] sourceArr) {
    int maxValue = sourceArr[0];

    for (int i = 0; i < sourceArr.length; i++) {
        if (sourceArr[i] > maxValue) {
            maxValue = sourceArr[i];
        }
    }
    // maxValue , elmt array,  sourceArr[i] > maxValue
    // 1, 1          1 > 1
    // 1, 10         10 > 1
    // 10, 1111082   1111082 > 10
    // 1111082, -5   -5 > 1111082
    // 1111082, 20   20 > 1111082
    // 1111082, 1111082 1111082 > 1111082
    // 1111082, 1     1 > 1111082

    return maxValue;
}

// нужно вернуть максимальный элемент в массиве
public static int findMinElement(int[] sourceArr) {
    int minElement = sourceArr[0];

    for (int i = 0; i < sourceArr.length; i++) {
        if (sourceArr[i] < minElement) {
            minElement = sourceArr[i];
        }
    }

    return minElement;
}

public static void sortAz(int[] sourceArr) {
    // {5,2,10}
    for (int i = 0; i < sourceArr.length; i++) {
        for (int j = 0; j < sourceArr.length - 1; j++) {
            if (sourceArr[j] > sourceArr[j + 1]) { // 5 > 2 {5,2,10}
                int temp = sourceArr[j];          // 5
                sourceArr[j] = sourceArr[j + 1];  // {2,2,10}
                sourceArr[j + 1] = temp;          // {2,5,10}
            }
        }
    }
}
```

```
        }  
    }  
}  
  
public static void sortZa(int[] sourceArr) {  
    //todo  
}  
  
//удалить из массива одинаковые элементы - установить значения по умолчанию  
public static void deleteDuplicate(int[] sourceArr) {  
    //todo  
}  
}
```