

```

        for (int i : c)
            System.out.print(i + " ");
        System.out.println();
    }
}
package chapt10;
import java.util.Comparator;

public class MyComparator<T> implements Comparator<Integer>
{
    public int compare(Integer n, Integer m) {
        return m.intValue() - n.intValue();
    }
}

```

В результате будет выведено:

```

4 3 5 1 2
5 4 3 2 1
1 2 3 4 5
3 4 5 1 2
min: 5
max: 1
71

```

Класс Arrays

В пакете **java.util** находится класс **Arrays**, который содержит методы манипулирования содержимым массива, а именно для поиска, заполнения, сравнения, преобразования в коллекцию и прочие:

int binarySearch(параметры) – перегруженный метод организации бинарного поиска значения в массивах примитивных и объектных типов. Возвращает позицию первого совпадения;

void fill(параметры) – перегруженный метод для заполнения массивов значениями различных типов и примитивами;

void sort(параметры) – перегруженный метод сортировки массива или его части с использованием интерфейса **Comparator** и без него;

static <T> T[] copyOf(T[] original, int newLength) – заполняет массив определенной длины, отбрасывая элементы или заполняя **null** при необходимости;

static <T> T[] copyOfRange(T[] original, int from, int to) – копирует заданную область массива в новый массив;

<T> List<T> asList(T... a) – метод, копирующий элементы массива в объект типа **List<T>**.

В качестве простого примера применения указанных методов можно привести следующий код.

```

/* пример # 21 : методы класса Arrays : ArraysEqualDemo.java */
package chapt10;
import java.util.*;

```

```

public class ArraysEqualDemo {
    public static void main(String[] args) {
        char m1[] = new char[3];
        char m2[] = { 'a', 'b', 'c' }, i;
        Arrays.fill(m1, 'a');
        System.out.print("массив m1:");
        for (i = 0; i < 3; i++)
            System.out.print(" " + m1[i]);
        m1[1] = 'b';
        m1[2] = 'c';
        //m1[2]='x'; //приведет к другому результату

        if (Arrays.equals(m1, m2))
            System.out.print("\nm1 и m2 эквивалентны");
        else
            System.out.print("\nm1 и m2 не эквивалентны");

        m1[0] = 'z';
        Arrays.sort(m1);
        System.out.print("\nmассив m1:");
        for (i = 0; i < 3; i++)
            System.out.print(" " + m1[i]);
        System.out.print(
            "\n значение 'с' находится в позиции-"
            + Arrays.binarySearch(m1, 'c'));
        Integer arr[] = {35, 71, 92};
        //вывод массива объектов в строку
        System.out.println(Arrays.deepToString(arr));
        //вычисление хэш-кода исходя из значений элементов
        System.out.println(Arrays.deepHashCode(arr));
        Integer arr2[] = {35, 71, 92};
        //сравнение массивов по содержимому
        System.out.println(Arrays.deepEquals(arr, arr2));
        char m3[] = new char[5];
        //копирование массива
        m3 = Arrays.copyOf(m1, 5);
        System.out.print("массив m3:");
        for (i = 0; i < 5; i++)
            System.out.print(" " + m3[i]);
    }
}

```

В результате компиляции и запуска будет выведено:

```

массив m1: a a a
m1 и m2 эквивалентны
массив m1: b c z
значение 'с' находится в позиции 1
[35, 71, 92]
65719
true
массив m3: b c z □ □

```