

Лабораторная работа №3

Полиморфизм на основе интерфейсов в языке Java

Цель работы

Приобретение навыков реализации интерфейсов для обеспечения возможности полиморфной обработки объектов класса.

Исходные данные

Стандартная библиотека языка Java содержит псевдокласс `Arrays`, предоставляющий набор статических методов для манипуляции массивами различных типов. В частности, в классе `Arrays` имеется метод `sort`, осуществляющий сортировку массива объектов:

```
public static void sort (Object [ ] a)
```

В языке Java массивы объектов ковариантны. Это значит, что если класс `S` является подклассом класса `T`, то массив объектов класса `S` является подтипом массива объектов класса `T`. Например, из того, что класс `String` является подклассом класса `Object`, следует, что тип `String[]` является подтипом по отношению к типу `Object[]`. Тем самым, мы имеем право передавать методу `sort` массивы любых объектов.

В методе `sort` реализован вариант алгоритма быстрой сортировки, осуществляющий сравнение объектов путём вызова метода `compareTo`, объявленного в интерфейсе `Comparable<T>` стандартной библиотеки языка Java и выполняющего сравнение текущего объекта `this` с объектом `obj`, переданным этому методу в качестве параметра:

```
int compareTo (T obj)
```

При этом `compareTo` возвращает отрицательное число, если `this` меньше `obj`, положительно число, если `this` больше `obj`, и 0, если они равны.

Интерфейс `Comparable<T>` имеет так называемый типовой параметр `T`, то есть является обобщённым интерфейсом. Его можно параметризовать любым классом, подставив имя класса вместо параметра `T`. Тем самым, обобщённый интерфейс фактически представляет собой множество интерфейсов, которые различаются значением типового параметра: `Comparable<Object>`, `Comparable<Integer>`, `Comparable<String>` и т.п.

¹Класс `Arrays`, как и класс `Math`, не предназначен для создания объектов, а является по сути хранилищем статических методов.

Обратите внимание на то, что если интерфейс Comparable<T> параметризован некоторым классом SomeClass, то формальный параметр obj метода compareTo будет иметь тип SomeClass:

```
int compareTo (SomeClass obj)
```

Для того чтобы массив объектов некоторого класса SomeClass можно было отсортировать с помощью метода sort класса Arrays, этот класс должен реализовывать интерфейс Comparable<SomeClass>. Например, объявим класс FirstLetterString, объекты которого упорядочены по первой букве содержащейся в них строки:

Продemonстрируем сортировку массива объектов класса FirstLetterString:

FirstLetterString.java

```
public class FirstLetterString implements Comparable<FirstLetterString > {
    private String s;
    public FirstLetterString(String s) { this.s = s; }
    public char charAt(int i) { return s.charAt(i); }
    public int length() { return s.length(); }
    public String toString() { return s; }

    public int compareTo(FirstLetterString obj) {
        if (s.length()==0 && obj.s.length()==0) return 0;
        else if (s.length () == 0) return -1;
        else if (obj.s.length() == 0) return 1;
        else return s.charAt(0) - obj.s.charAt(0);
    }
}
```

Test.java

```
import java.util.Arrays ;

public class Test{
    public static void main (String[] args) {
        FirstLetterString[] a = new FirstLetterString[] {
            new FirstLetterString("gamma"),
            new FirstLetterString("beta"),
            new FirstLetterString("alpha"),
        };
        Arrays.sort(a);
        for (FirstLetterString s : a) System.out.println(s);
    }
}
```

Задание

Во время выполнения лабораторной работы требуется разработать на языке Java один из классов, перечисленных в таблице. В классе должен быть реализован интерфейс Comparable<T> и переопределён метод toString. В методе main вспомогательного класса Test нужно продемонстрировать работоспособность разработанного класса путём сортировки массива его экземпляров.

Варианты

п/н	Студент	Группа	Условие задачи
1			Класс квадратных трёхчленов с порядком на основе суммы корней соответствующего квадратного уравнения (вспомнить теорему Виета).
2			Класс стеков целых чисел с порядком на основе максимального значения на стеке.
3			Класс целочисленных матриц размера $m \times n$ с порядком на основе ранга матрицы.
4			Класс, представляющий военнослужащего, с порядком на основе (в порядке убывания значимости): звание, фамилия, имя, отчество, год рождения.
5			Класс целочисленных матриц размера $m \times n$ с порядком на основе величины седловой точки. (Седловая точка – элемент матрицы, одновременно наименьший в своей строки и наибольший в своём столбце). Если матрица не имеет седловой точки, считать величину седловой точки равной максимальному целому числу.
6			Класс, представляющий множество интервалов вида $[a; b]$, где a и b – вещественные числа, с порядком на основе размера покрываемой интервалами области числовой оси. Интервалы могут полностью или частично перекрываться.
7			Класс вещественных векторов произвольной размерности с порядком на основе длины вектора.
8			Класс целых чисел с порядком на основе количества простых делителей.
9			Класс, представляющий двойной стек целых чисел, с порядком на основе разности суммы элементов «левого» и «правого» стеков.
10			Класс, представляющий натуральное число, с порядком на основе расстояния на числовой прямой до ближайшего числа Фибоначчи.
11			Класс последовательностей char'ов с порядком на основе близости первой латинской гласной буквы к началу последовательности.
12			Класс, представляющий клетку в игре Конвея Жизнь, с порядком на основе количества заполненных соседних клеток. (Потребуется дополнительный класс – замкнутая вселенная размером $n \times n$, в которой верхняя граница соединена с нижней, а левая граница – с правой.)
13			Класс, представляющий множество вещественных векторов в n -мерном пространстве с порядком на основе длины суммы векторов множества.
14			Класс, представляющий множество арифметических прогрессий, с порядком на основе количества чисел из интервала $(0; 100)$, принадлежащих прогрессиям множества.
15			Класс пользователей социальной сети с порядком на основе количества френдов. (Потребуется дополнительный класс – социальная сеть, предоставляющий возможности добавления и удаления связей между пользователями.)
16			Треугольник, заданный координатами вершин на плоскости, с порядком на основе количества точек с целочисленными координатами,

- попадающими внутрь треугольника.
- 17 Класс состоящих из слов предложений с порядком на основе средней длины слова в предложении.
- 18 Класс многоугольников с порядком на основе максимальной длины стороны многоугольника.
- 19 Класс полиномов с порядком на основе количества целочисленных корней, принадлежащих интервалу $[0; 10]$.
- 20 Класс очередей целых чисел с порядком на основе суммы элементов очереди.
- 21 Класс, представляющий абитуриента, с порядком на основе суммы баллов по трём ЕГЭ: информатика, математика и русский язык. (Нужно учитывать, что ЕГЭ различаются по приоритетам, позволяющим упорядочивать абитуриентов, имеющих одинаковые суммы баллов.)
- 22 Класс булевских матриц размера $m \times n$ с порядком на основе суммарного количества строк и столбцов, все элементы которых равны между собой.
- 23 Класс вещественных векторов в трёхмерном пространстве с порядком на основе величины угла, который вектор образует с плоскостью XOY .
- 24 Класс, представляющий полином, с порядком на основе величины числа, которое надо прибавить к полиному, чтобы он без остатка делился на $(x-1)$.
- 25 Класс бинарных отношений на множестве целых чисел от 0 до n с порядком на основе количества пар чисел, принадлежащих отношению и нарушающих условие симметричности отношения.
- 26 Класс дробей, числитель и знаменатель которых взаимно просты, с естественным порядком на множестве рациональных чисел.
- 27 Класс пар окружностей с порядком на основе расстояния между точками пересечения окружностей. При совпадении окружностей считать расстояние нулевым, при непересечении – бесконечным. (Потребуется дополнительный класс – окружность.)
- 28 Класс последовательностей целых чисел с порядком на основе количества пиков в последовательности.
- 29 Класс, представляющий множество строк, с порядком на основе количества общих латинских букв в строках, принадлежащих множеству (например, если буквы a и b принадлежат всем строкам, а любая другая буква отсутствует хотя бы в одной строке, то количество общих латинских букв – 2).
- 30 Класс состоящих из слов предложений с порядком на основе максимальной длины слова в предложении.
- 31 Класс состоящих из слов предложений с порядком на основе максимальной длины слова в предложении.
- 32 Треугольник, в вершинах которого располагаются материальные точки. Каждая материальная точка задаётся координатами на плоскости и массой. Порядок на множестве таких треугольников должен быть основан на близости центра масс треугольника к началу координат.
- 33 Класс предложений, состоящих из слов, разделённых пробелами и запятыми, с порядком на основе максимального количества слов, между

которыми нет запятой.

34 Класс квадратных уравнений с порядком на основе количества действительных корней уравнения.