

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 последовательностей целых чисел с порядком на основе разности максимального и минимального числа. |
| 2 | | | Класс полиномов с порядком на основе максимального значения полинома на интервале $[0; 1]$. |
| 3 | | | Класс, представляющий множество материальных точек, заданных координатами в трёхмерном пространстве и массой, с порядком на основе расстояния от центра масс до начала координат. |
| 4 | | | Класс вершин бинарного дерева поиска с порядком на основе высоты поддерева, растущего из вершины. (Потребуется дополнительный класс – бинарное дерево с операциями поиска и добавления словарной пары.) |
| 5 | | | Класс, представляющий вершину мультиграфа, рёбра которого помечены строками, с порядком на основе степени вершины. |
| 6 | | | Класс, представляющий список публикаций учёного, с порядком на основе индекса Хирша. |
| 7 | | | Класс, представляющий координаты слова в тексте с порядком на основе частоты встречаемости слова и операций, меняющей два слова местами. (Потребуется дополнительный класс, представляющий изменяемый текст, с операцией доступа к слову по его номеру. Тестирование работоспособности нужно произвести двумя способами: отсортировав массив слов стандартным способом и реализовав алгоритм быстрой сортировки слов в тексте.) |
| 8 | | | Класс, представляющий натуральное число, с порядком на основе расстояния на числовой прямой до ближайшего простого числа. |
| 9 | | | Класс состоящих из слов предложений с порядком на основе количества слов в предложении. |
| 10 | | | Класс шаров, определяемых их координатами в трёхмерном пространстве и радиусами, с порядком на основе площади сечения шара плоскостью XOY. |
| 11 | | | Класс полиномов с порядком на основе суммы коэффициентов производной. |
| 12 | | | Класс, представляющий арифметическую прогрессию с порядком на основе количества членов, попадающих в интервал $[0; 100]$. |
| 13 | | | Класс, представляющий шестизначный номер, который составлен из десятичных цифр, с порядком на основе счастливости номера. Счастливость определяется разностью сумм цифр левой и правой половины номера. |
| 14 | | | Круг, заданный координатами центра и радиусом, с порядком на основе количества точек с целочисленными координатами, попадающими внутрь круга. |
| 15 | | | Класс ферзей на шахматной доске, помнящих свою позицию, с порядком на основе количества ферзей, которых данный ферзь бьёт. (Потребуется дополнительный класс – шахматная доска.) |
| 16 | | | Класс квадратных целочисленных матриц размера n с порядком на основе |

- количества элементов, нарушающих симметричность матрицы относительно главной диагонали.
- 17 Класс отрезков прямых на плоскости с порядком на основе количества точек, принадлежащих отрезку и имеющих целочисленные координаты. Отрезок задаётся координатами своих концов, которые также целочисленные.
- 18 Класс, представляющий комбинацию карт в покере, с порядком, определяемым правилами игры.
- 19 Класс ломаных линий на плоскости с порядком на основе количества пересечений ломаной линии с осями координат.
- 20 Класс программ, написанных на языке C, с порядком на основе суммарной длины комментариев в программе.
- 21 Класс пар векторов в трёхмерном пространстве с порядком на основе длины их векторного произведения. (Потребуется дополнительный класс для представления вектора.)
- 22 Класс целых чисел с порядком на основе количества различных цифр в десятичном представлении.
- 23 Класс, представляющий клетку на доске для игры в крестики-нолики размера $m \times n$ (потребуется отдельный класс, представляющий доску). Клетка может либо быть пустой, либо содержать крестик, либо содержать нолик. Пусть порядок на множестве клеток задаётся на основе количества дружественных соседей клетки. Дружественной называется соседняя клетка, содержащая тот же символ, т.е. для крестиков дружественны крестики, для ноликов – нолики, для пустых клеток – пустые клетки.
- 24 Класс коней на шахматной доске с порядком на основе количества допустимых ходов. Конь может быть белым или чёрным и может атаковать коня другого цвета. (Потребуется дополнительный класс – шахматная доска, помнящая множество коней.)
- 25 Класс последовательностей char'ов с порядком на основе максимального значения префиксной функции.
- 26 Класс, представляющий резюме программиста, претендующего на некоторую вакансию. Как резюме, так и вакансия характеризуются перечнем языков программирования. Порядок для резюме, относящихся к одной и той же вакансии, определяется величиной пересечения перечня языков резюме с перечнем языков вакансии. Порядок для резюме, относящихся к разным вакансиям, определяется размером перечня языков вакансии.
- 27 Класс, представляющий библиографическое описание книги, с порядком на основе следующих компонентов описания (в порядке убывания значимости): год издания, лексикографически наименьшее имя среди имён авторов книги, название книги.
- 28 Класс, представляющий номер года, с порядком на основе количества дней, в которых 13-е число месяца приходится на пятницу.
- 29 Класс, представляющий ипотечный кредит, с порядком на основе размера ежемесячной выплаты. (Кредит определяется суммой, процентной ставкой и сроком, измеряемым в месяцах.)
- 30 Класс, представляющий множество точек в трёхмерном пространстве с

порядком на основе минимального объёма параллелепипеда, содержащего все точки (любая сторона параллелепипеда параллельна одной из осей координат).

- 31 Класс предложений, состоящих из разделённых пробелами слов, с порядком на основе количества слов, представляющих целые числа в десятичной записи.
- 32 Класс, представляющий идеальный размен некоторой суммы денег набором рублёвых купюр и монет, с порядком на основе общего количества купюр и монет. Идеальный размен содержит минимально возможное количество монет и купюр.
- 33 Класс треугольников, заданных координатами точек на плоскости, с порядком на основе площади треугольника. (Потребуется дополнительный класс, представляющий координаты точки.)