Лабораторная работа №3
«Полиморфизм на основе интерфейсов в языке Java»

Скоробогатов С.Ю.

7 апреля 2016 г.

1 Цель работы

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

2 Исходные данные

Стандартная библиотека языка Java содержит «псевдокласс»[1] 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 стортировки, 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 *типовый 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.

[1] Класс 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:

```
1  public class FirstLetterString implements Comparable<FirstLetterString> {
2          private String s;
3
4          public FirstLetterString(String s) { this.s = s; }
5
6          public String toString() { return s; }
7
8          public int compareTo(FirstLetterString obj) {
9                  if (s.length() == 0 && obj.s.length() == 0) return 0;
10                 else if (s.length() == 0) return -1;
11                 else if (obj.s.length() == 0) return 1;
12                 else return s.charAt(0) - obj.s.charAt(0);
13         }
14 }
```

Продемонстрируем сортировку массива объектов класса FirstLetterString:

```
1  import java.util.Arrays;
2
3  public class Test {
4          public static void main(String[] args) {
5                  FirstLetterString[] a = new FirstLetterString[] {
6                          new FirstLetterString("gamma"),
7                          new FirstLetterString("beta"),
8                          new FirstLetterString("alpha")
9                  };
10                 Arrays.sort(a);
11                 for (FirstLetterString s : a) System.out.println(s);
12         }
13 }
```

3 Задание

Во время выполнения лабораторной работы требуется разработать на языке Java один из классов, перечисленных в таблицах 1 и 2. 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.

Таблица 1: Варианты классов

1	Класс нормализованных дробей с естественным порядком на множестве рациональных чисел.
2	Класс последовательностей целых чисел с лексикографическим порядком.
3	Класс последовательностей **char**'ов с порядком на основе количества букв 'a'.
4	Класс полиномов с порядком на основе суммы коэффициентов производной.
5	Класс состоящих из слов предложений с порядком на основе количества слов в предложении.
6	Класс знаковых целых чисел с порядком на основе суммы цифр десятичного представления.
7	Класс пар целых чисел с порядком на основе наибольшего общего делителя пары.
8	Класс последовательностей **char**'ов с порядком на основе близости первой латинской гласной буквы к началу последовательности.
9	Класс последовательностей целых чисел с порядком на основе количества пиков в последовательности.
10	Класс отрезков прямых на плоскости с порядком на основе длины отрезка.
11	Класс состоящих из слов предложений с порядком на основе средней длины слова в предложении.
12	Класс знаковых целых чисел с порядком на основе количества единичных бит в двоичном представлении.
13	Класс пар целых чисел с порядком на основе наименьшего общего кратного чисел пары.
14	Класс матриц с порядком на основе ранга матрицы.
15	Класс последовательностей целых чисел с порядком на основе максимальной суммы подпоследовательности (алгоритм Кадана).
16	Класс треугольников с порядком на основе площади треугольника.
17	Класс состоящих из слов предложений с порядком на основе максимальной длины слова в предложении.
18	Класс целых чисел с порядком на основе количества простых делителей.
19	Класс квадратных трёхчленов с порядком на основе суммы корней соответствующего квадратного уравнения.
20	Класс последовательностей целых чисел с порядком на основе количества обменов, которые нужно выполнить, чтобы отсортировать последовательность пузырьком.
21	Класс последовательностей целых чисел с порядком на основе разности максимального и минимального числа.
22	Класс знаковых целых чисел с порядком на основе количества младших нулевых бит в двоичном представлении числа.
23	Класс состоящих из слов предложений с порядком на основе близости слова минимальной длины к началу предложения.
24	Класс целых чисел с порядком на основе количества различных цифр в десятичном представлении.
25	Класс последовательностей **char**'ов с порядком на основе максимального значения префиксной функции.

Таблица 2: Варианты классов

26	Класс последовательностей целых чисел с порядком на основе количества различных чисел в последовательности.
27	Класс квадратных уравнений с порядком на основе количества действительных корней уравнения.
28	Класс многоугольников с порядком на основе максимальной длины стороны многоугольника.
29	Класс точек в трёхмерном пространстве с порядком на основе близости точки к началу координат.
30	Класс пар векторов в трёхмерном пространстве с порядком на основе длины их векторного произведения.
31	Класс четырёхугольников на плоскости с порядком на основе суммы длин диагоналей.
32	Класс пар окружностей с порядком на основе расстояния между точками пересечения окружностей (при совпадении окружностей считать расстояние нулевым, при непересечении – бесконечным).
33	Класс стеков целых чисел с порядком на основе максимального значения на стеке.
34	Класс последовательностей булевских значений с порядком на основе длины самой длинной подпоследовательности, состоящей из одинаковых значений.
35	Класс четырёхугольников на плоскости с порядком на основе площади четерёхугольника.
36	Класс векторов прозвольной размерности с порядком на основе длины вектора.
37	Класс пар комплексных чисел с порядком на основе произведения чисел пары.
38.	Класс предложений, состоящих из слов, разделённых пробелами и запятыми, с порядком на основе максимального количества слов, между которыми нет запятой.
39.	Класс последовательностей целых чисел с порядком на основе максимального количества одинаковых подряд идущих чисел.
40.	Класс предложений, состоящих из разделённых пробелами слов, с порядком на основе количества слов, представляющих целые числа в десятичной записи.
41.	Класс программ, написанных на языке C, с порядком на основе суммарной длины комментариев в программе.
42.	Класс последовательностей целых чисел с порядком на основе количества простых чисел в составе последовательности.
43	
44	
45	
46	
47	
48	
49	
50	