Hochschule Aalen
Fakultät Elektronik und Informatik
Studiengang Informatik

Programmieren 2

Vorlesung im Wintersemester 2014/2015
Prof. Dr. habil. Christian Heinlein

7. Übungsblatt (28. November 2014)

Aufgabe 7: Überschreiben von Standardmethoden

Transformieren Sie die Klasse `StringSet` (Musterlösung von Aufgabe 5) in eine Klasse `ObjectSet`, die anstelle von Strings beliebige Objekte als Elemente enthalten kann!

Fügen Sie dann die folgenden drei Standardmethoden hinzu (vgl. Skript, § 5.10):

- `boolean equals (Object other)`
  Liefert genau dann `true`, wenn das Objekt `other` ebenfalls ein `ObjectSet` ist und die gleichen Elemente wie die aktuelle Menge enthält. (Zum Vergleich der Elemente soll wiederum `equals` verwendet werden.)

- `int hashCode ()`
  Liefert als Ergebnis die Summe der Hashwerte aller Elemente der Menge. (Die Hashwerte der einzelnen Elemente sollen wiederum mittels `hashCode` bestimmt werden. Für eine leere Menge erhält man die Summe 0.)

- `String toString ()`
  Liefert die gleiche Zeichenfolge, die die Methode `print` ausgibt. (Zur Umwandlung der einzelnen Elemente in Zeichenfolgen soll wiederum `toString` verwendet werden.)

Vermeiden Sie Code-Verdopplungen!

Verwenden Sie zum Testen u. a. folgendes Programm und erklären Sie, wie bei einem Aufruf der Art

```
java ObjectSetTest 1 2 3 4 5 6 7 8
```

die einzelnen Ausgabezeilen genau zustandekommen!

```
// Punkt im zweidimensionalen Raum.
class Point {
  // Koordinaten des Punkts.
  public final double x, y;

  // Punkt mit Koordinaten x und y konstruieren.
  public Point (double x, double y) { this.x = x; this.y = y; }

  // Zeichenketten-Darstellung des aktuellen Punkts liefern.
  public String toString () { return "(" + x + ", " + y + ")"; }
```

```
  // Vergleich des aktuellen Punkts mit einem anderen Objekt other.
  public boolean equals (Object other) {
    if (!(other instanceof Point)) return false;
    Point that = (Point)other;
    return this.x == that.x && this.y == that.y;
  }

  // Hashwert für den aktuellen Punkt liefern.
  public int hashCode () { return (int)(x + y); }
}

// Testprogramm für die Klasse ObjectSet.
class ObjectSetTest {
    public static void main (String [] args) {
        int n = args.length;

        // Je zwei Kommandozeilenargumente definieren einen Punkt p.
        // Jeder solche Punkt wird in die Menge s1 eingefügt.
        ObjectSet s1 = new ObjectSet(n);
        for (int i = 0; i < n; i += 2) {
            double x = Double.parseDouble(args[i]);
            double y = Double.parseDouble(args[i+1]);
            Point p = new Point(x, y);
            s1.insert(p);
        }
        System.out.println(s1 + " " + s1.hashCode());

        // In die Menge s2 werden die gleichen Punkte eingefügt
        // wie in s1, aber in umgekehrter Reihenfolge.
        ObjectSet s2 = new ObjectSet(n/2);
        for (int i = n - 2; i >= 0; i -= 2) {
            double x = Double.parseDouble(args[i]);
            double y = Double.parseDouble(args[i+1]);
            Point p = new Point(x, y);
            s2.insert(p);
        }
        System.out.println(s2 + " " + s2.hashCode());

        // Vergleich von s1 und s2.
        System.out.println(s1.equals(s2));

        // Da s1 und s2 selbst wieder Objekte sind, können sie in eine
        // dritte Menge s3 eingefügt werden.
        ObjectSet s3 = new ObjectSet(2);
        s3.insert(s1);
        s3.insert(s2);
        System.out.println(s3);
    }
}
```