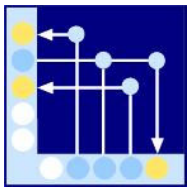
Hochschule Aalen
Fakultät Elektronik und Informatik
Studiengang Informatik



Programmieren 2

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

5. Übungsblatt (11. November 2014)

Aufgabe 5: Klassen

Implementieren Sie eine Java-Klasse `StringSet` zur Repräsentation von Mengen mit Elementen des Typs `String`, die die folgenden öffentlichen Konstruktoren und Methoden besitzt:

```
// Leere Menge mit Kapazität n >= 0 (d. h. Platz für n Elemente) erzeugen.
StringSet (int n)

// Leere Menge mit Kapazität n >= 0 erzeugen und anschließend
// Element s einfügen, falls dies möglich ist (vgl. insert).
StringSet (int n, String s)

// Kapazität der Menge (d. h. Wert des Konstruktorparameters n) liefern.
int capacity ()

// Kardinalität der Menge (d. h. tatsächliche Anzahl ihrer Elemente) liefern.
int card ()

// Menge in der Form "{ }", "{ a }", "{ a, b }" etc. ausgeben.
void print () {

// Enthält die Menge das Element s?
boolean contains (String s)

// Element s einfügen und true liefern, falls es noch nicht enthalten
// und nicht null ist und die Kapazität der Menge noch nicht erschöpft ist;
// andernfalls false liefern.
boolean insert (String s)

// Element s entfernen und true liefern, falls es enthalten ist;
// andernfalls false liefern.
boolean remove (Strings s)

// Schnittmenge der Mengen first und second als neue Menge mit
// geeigneter Kapazität liefern (first und second bleiben unverändert).
static StringSet intersection (StringSet first, StringSet second)
```

Speichern Sie die Elemente einer Menge mit Kapazität `n` und Kardinalität `m` in den ersten `m` Plätzen eines Arrays der Größe `n`!
Alle Objektvariablen der Klasse sowie eventuelle Hilfsmethoden (zur Vermeidung von Codeverdopplung!) sollen privat sein.
Außer der Methode `equals` der Klasse `String` sowie `System.out.print` und `System.out.println` sollen keinerlei Java-Bibliotheksmethoden verwendet werden!

Das folgende Programm kann zum interaktiven Testen der Klasse `StringSet` verwendet werden:

```
// Testprogramm für die Klasse StringSet.
class StringSetTest {
    public static void main (String [] args) {
        // Abbruch, wenn keine Konsole verfügbar ist (z. B. in Eclipse).
        if (System.console() == null) {
            System.out.println("console not available");
            return;
        }

        // Aktuelle Menge s.
        StringSet s = null;

        // Endlosschleife.
        while (true) {
            // Kommandozeile lesen und in Wörter zerlegen.
            // Abbruch bei Ende der Eingabe oder leerer Eingabezeile.
            String line = System.console().readLine("command: ");
            if (line == null || line.equals("")) return;
            String [] cmd = line.split(" ");

            // Fallunterscheidung anhand des ersten Zeichens des ersten Worts.
            switch (cmd[0].charAt(0)) {
            default: // new set
                int n = Integer.parseInt(cmd[0]);
                if (cmd.length == 1) s = new StringSet(n);
                else s = new StringSet(n, cmd[1]);
                break;
            case '+': // insert
                System.out.println(s.insert(cmd[1]));
                break;
            case '-': // remove
                System.out.println(s.remove(cmd[1]));
                break;
            case '?': // contains
                System.out.println(s.contains(cmd[1]));
                break;
            case '&': // intersection
                StringSet t = new StringSet(cmd.length - 1);
                for (int i = 1; i < cmd.length; i++) t.insert(cmd[i]);
                s = StringSet.intersection(s, t);
                break;
            }

            // Kardinalität, Kapazität und Inhalt der Menge s ausgeben.
            System.out.print(s.card() + " of " + s.capacity() + " element(s): ");
            s.print();
            System.out.println();
        }
    }
}
```