Theorie und Implementation objektorientierter Programmiersprachen
Theoretische Informatik, LMU München
Andreas Abel

Übungsblatt 10 (Sonderpunkte)

Abgabe **(freiwillig)** bis: Montag, **15. Juli 2002, 10.15 Uhr**

Aufgabe 1: [OOP in funktionaler Sprache]

Betrachten Sie das folgende Java-Programm (gekürzte Version von "Einzelschritt-Semantik in Java" `arith-2.java` aus Vorlesung 3).

```
class Exp {
    public String toString () { return "abstract Exp"; }
    public boolean isValue () { return false; }
    public int value () { return 0; }
    public Exp step() { return this; }
    public Exp steps() {
        if (isValue()) return this;
        else return step().steps();
    }
}

class CInt extends Exp {
    private final int i;
    CInt (final int i) { this.i = i; }

    public String toString() { return Integer.toString(i); }
    public boolean isValue () { return true; }
    public int value() { return i; }
}

class Plus extends Exp {
    private final Exp t1, t2;
    Plus (final Exp t1, final Exp t2) { this.t1 = t1; this.t2 = t2; }

    public String toString() {
        return "(" + t1.toString() + " + " + t2.toString() + ")";
    }
    public Exp step() {
        if (t1.isValue()) {
            if (t2.isValue())
                return new CInt (t1.value() + t2.value());
            else return new Plus(t1, t2.step());
        } else return new Plus(t1.step(), t2);
   }
}
```

1. Von welchen der 5 in der Vorlesung OO-Konzepte wird hier Gebrauch gemacht?
2. Setzen Sie die drei Klassen in ein funktionales Programm im OO-Stil um, ganz nach dem Vorbild der Vorlesung. Dabei dürfen Sie annehmen, dass es die String-Operationen `Integer.toString` und "`+`" auch in der funktionalen Sprache gibt.

(8 Punkte)

Aufgabe 2: [Simulation von Objektidentität]

Ein weiteres wichtiges Konzept der OO-Programmierung ist *Objektidentität*. Jedes Objekt hat einen einzigartigen Bezeichner (OID) anhand dessen festgestellt werden kann, ob zwei Objekte gleich sind. Beschreiben Sie eine allgemeine Methode, wie dieses Feature in unserer funktionalen Sprache mit Referenzen realisiert werden kann. Definieren Sie eine Mutter-Klasse `objectClass`, von der alle anderen Klassen abgeleitet werden müssen, die eine OID haben sollen.

```
Object = { id: Int }

objectClass : Unit -> Object
objectClass = ...
```

(4 Punkte)

Viel Spaß!