

## Übungen zu Einführung in die Informatik

**Hinweis** Die Aufgaben der Probeklausur entsprechen vom Schwierigkeitsgrad her den Klausuraufgaben. Sie sollten in der Lage sein, sie in 80 Minuten zu bearbeiten. Beachten Sie bitte, dass die Probeklausur nicht als Hinweis auf Klausurstoff zu verstehen ist!

**Besprechung** 21./22. Jan. 2008

### Aufgabe PK-1

### Lernfragen

- a) Ein Java-Programm besteht aus Klassen und Objekten. (Ja/Nein)
- b) In der Informatik wird die  $O$ -Notation verwendet, um die Komplexität eines Algorithmus zu charakterisieren. Seien  $f : \mathbb{N} \rightarrow \mathbb{N}$  und  $s : \mathbb{N} \rightarrow \mathbb{N}$  zwei Funktionen,  $f \in O(s)$  bedeutet, dass es  $m \in \mathbb{N}$  gibt, so dass gilt: Für alle  $n \in \mathbb{N}, n \geq m$  gibt es  $k \in \mathbb{N}$ , so dass  $f(n) \leq k \cdot s(n)$  gilt. (Ja/Nein)
- c) Es gilt  $O(n) = O(2n)$  (Ja/Nein)
- d) Der Wert des Java-Ausdrucks `o1 == o2` ist immer `true`, wenn `o1.equals(o2) && o2.equals(o1)` wahr ist. (Ja/Nein)
- e) Enthält folgendes Java-Programm Syntaxfehler? Wenn ja, welche?

```
int i = 0;
Object o1 = "123" + i;
if (true == false) i /= 1;
```

### Aufgabe PK-2

### Vererbung

Welche Ausgabe erzeugt folgendes Programm?

```
public class A {
    public int f() {
        return 1;
    }
    public int g() {
        return h();
    }
    public int h() {
        return 20;
    }
    public int h(Object o) {
        return 5;
    }
    public int h(String s) {
        return 15;
    }
}

public class B extends A {
    public int f() {
        return 2;
    }
    public int g() {
        return super.g();
    }
    public int h() {
        return 30;
    }
    public int h(Object o) {
        return 25;
    }
}

public class Main1 {
    public static void main(String[] args) {
        A a = new A();
        System.out.println("a.f() = " + a.f());
        System.out.println("a.g() = " + a.g());
        a = new B();
        System.out.println("a.f() = " + a.f());
        System.out.println("a.g() = " + a.g());
        System.out.println("a.h(x) = " + a.h("x"));
    }
}
```

### Aufgabe PK-3

### S-Expressions

In der Programmiersprache Scheme wird die Syntax von Programmen durch so genannte S-Expressions gegeben. In dieser Aufgabe soll eine vereinfachte EBNF-Grammatik für S-Expressions entwickelt werden.

- a) Ein Symbol ist eine nichtleere Folge von Buchstaben, Ziffern und Sonderzeichen, die mit einem Buchstaben beginnt. Zum Beispiel sind `abc` und `x%y` Symbole, dagegen sind `%1a` und `123` keine Symbole. Geben Sie eine EBNF-Grammatik von Symbolen an. Sie können dabei die Regeln

$$\begin{aligned} \text{Buchstabe} &= \text{"a"} \mid \dots \mid \text{"z"} \\ \text{Ziffer} &= \text{"0"} \mid \text{"1"} \mid \dots \mid \text{"9"} \\ \text{Sonderzeichen} &= \text{"+"} \mid \text{"-"} \mid \text{"\%"} \mid \text{"*"} \end{aligned}$$

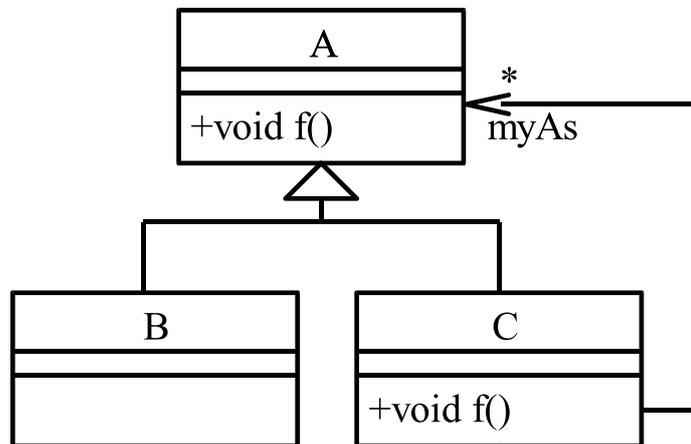
als gegeben betrachten.

- b) Eine S-Expression ist entweder ein Symbol oder eine von runden Klammern ( und ) eingeschlossene Folge von S-Expressions. Zum Beispiel sind ( ), (print this) und (x (y (z) a)) S-Expressions, dagegen sind (a)(b) und )( keine S-Expressions. Geben Sie eine EBNF-Grammatik für S-Expressions an.
- c) Geben Sie eine Ableitung von (x (y (z) a)) aus Ihrer Grammatik an.

### Aufgabe PK-4

### UML

Gegeben sei folgendes UML-Klassendiagramm.



Geben Sie eine Implementierung der Klassen B und C gemäß dem Klassendiagramm in Java an. Implementieren Sie für die Klasse C Getter- und Setter-Methoden für alle Attribute, sowie einen Konstruktor, in dem alle Attribute initialisiert werden. Für B sind keine Getter- und Setter-Methoden und keine Konstruktoren anzugeben.

Implementieren Sie die Methode f für die Klasse C. Dabei soll die Methode f für alle mit this assoziierten Objekte aus myAs aufgerufen werden.

```
class A {
    public void f() {
        System.out.println("A.f()");
    }
}
```

### Aufgabe PK-5

### Laufängen-Kodierung

Die Laufängen-Kodierung ist ein Datenkomprimierungsverfahren, das häufig auf Bilder angewandt wird. „Sie ist besonders gut geeignet, Wiederholungen oder Sequenzen von gleichen Werten verkürzt darzustellen. Liegt eine Wiederholung vor, wird die Anzahl der Wiederholungen sowie der wiederholte Wert gespeichert“ [Wikipedia]. Beispielsweise kann das int-Array {0,0,0,0,0,0,1,1,1,1,3,3,3,3,2} mit {6,0,4,1,4,3,1,2} und das Array {0,0,0,0,1,1,1,1,1,1,1,2,2,2} mit {4,0,7,1,3,2} kodiert werden.

Implementieren Sie in der Klasse RunLength eine Methode `public static int[] coding(int[] a)`, die das Array a nach obigem Beispiel kodiert und zurückgibt. Das Array a soll nicht verändert werden.