Prüfungsklausur zu den Vorlesungen
„Digitale Informationsverarbeitung“ und „Programmieren und Programmiersprachen“
Dr. Gräbe, Sommersemester 2002

Aufgabe D 1: Programmanalyse (14 Punkte)

Gegeben ist das folgende C-Programmfragment.

```
extern int f(int);
extern int g(int);
int test(int n) {
  int i,s=0;
  for(i=0;i<f(n);i++) s+=g(i);
  return s;
}
```

a. Welche Bezeichner werden in diesem Programmfragment deklariert und was bezeichnen sie jeweils? Welche Bezeichner werden bei der Deklaration auch definiert? (6 Pkt.)

b. Es sei bekannt, dass `f(n)` von der Größenordnung $O(n)$ und Berechnungsdauer $O(n^2)$ ist, während `g(n)` von der Größenordnung $O(n^2)$ und Berechnungsdauer $O(n)$ ist. Bestimmen Sie daraus Größenordnung und Berechnungsdauer von `test(n)`. (8 Pkt.)

Aufgabe D 2: Zahldarstellungen (10 Punkte)

a. Bestimmen Sie die Bit-Darstellung der Zahl $z = -47$ als `signed char`. (2 Pkt.)

b. Bestimmen Sie die Bit-Darstellung der Zahl $z = -47$ als `float` im 32-Bit-IEEE-Format. (3 Pkt.)

c. Bestimmen Sie die Positionsdarstellung der Zahl $u = [0.\overline{27}]_8$ im Hexadezimalsystem (also zur Basis 16). (5 Pkt.)

Aufgabe D 3: Höhere Programmiersprachen (16 Punkte)

a. Nennen Sie die vier Schritte, die in einem Compiler zur Analyse und Übertragung des Quelltexts eines Programms in die Zielsprache durchlaufen werden. Charakterisieren Sie jeden dieser Schritte mit einem Satz. (8 Pkt.)

b. Geben Sie einen regulären Ausdruck an, der die ganzen Zahlen (mit und ohne Vorzeichen) in dezimaler Notation beschreibt. (3 Pkt.)

c. Erläutern Sie den Unterschied zwischen Wertparametern und Referenzparametern.

   Welches dieser Konzepte unterstützt C, wie kann man das andere modellieren und welche Besonderheiten sind dabei zu beachten? (5 Pkt.)

Fragenteil DF **(10 Punkte)**

DF-1. Geben Sie zu folgenden Zahlen jeweils ihre Positionsdarstellung im Dezimalsystem an. Markieren Sie syntaktisch nicht korrekte Angaben durch ein Kreuz. (6 Pkt.)

$[0.C]_{16}$	$[110]_3$	$[1.01]_2$	$[2.B]_8$	$[10]_7$	$[10]_{12}$

DF-2. Kreuzen Sie an, zu welchen Wachstumsklassen die beiden angegebenen Funktionen für $n \to \infty$ gehören. (4 Pkt.)

	$O(1)$	$O(n)$	$O(n^2)$
$\frac{n^3+1}{(n+1)^3-(n-1)^3}$			
$\sum_{i=1}^{n} \frac{2\,i+1}{i+2}$			

Zusatzblatt zu Aufgabe P 1

```
class A {
    int x = 12, dx = 3;
    A() { dx = 1; }
    int f() { x-=dx; return x; }
}

class B extends A {
    int x = 7;
    B(int n) { x=x+n; }
    int f() { x+=dx; return x; }
    int f(int n) { return x-n; }
}

class ABC {
    static void testA(A a) {
        System.out.println(a.x);
        System.out.println(a.dx);
        System.out.println(a.f());
    }

    static void testB(B b) {
        System.out.println(b.x);
        System.out.println(b.f());
        System.out.println(b.f(5));
    }

    static public void main(String[] argv) {
        A x=new A(); B y=new B(2);
        testA(x); testA(y); testB(y);
    }
}
```

Aufgabe P 1: Überladen und Überschreiben (14 Punkte)

Auf dem Zusatzblatt zu dieser Aufgabe ist ein Java-Programm gegeben, in dem drei Klassen zusammenwirken.

a. Markieren Sie im Quelltext je eine Stelle, an der Überschreiben, Überladen bzw. Überdecken verwendet wird. (3 Pkt.)

b. Tragen Sie die 9 Werte, die vom Hauptprogramm nacheinander ausgegeben werden, in die folgende Tabelle (von links nach rechts) ein. (5 Pkt.)

testA(x)			testA(y)			testB(y)		

c. Erläutern Sie, wie die Felder von y beim Aufruf des Konstruktors `y=new B(2)` belegt werden und wie sich diese Belegung während des Aufrufs von `a.f()` in `testA(y)` ändert. (6 Pkt.)

Aufgabe P 2: Generische Klassen (16 Punkte)

Gegeben ist das folgende Fragment einer Implementierung der generischen Klasse `ListFunctions`, die auf die beiden Klassen `E` (wie Element) und `L` (wie Liste) zugreift.

```
public class ListFunctions {

    static boolean isMember(L f, E o) {
        if(L.isEmpty(f)) return false;
        if(L.first(f).eq(o)) return true;
        return isMember(L.rest(f),o);
    }
}
```

a. Welche Methoden müssen von den Klassen `E` und `L` bereitgestellt werden? Geben Sie die genauen Methodendeklarationen mit ihren Modifikatoren in Java-Syntax an. (6 Pkt.)

b. Begründen Sie, welche der Klassen `E` und `L` als `interface` vereinbart werden können. (4 Pkt.)

c. Begründen Sie, ob die Klasse `ListFunctions` als `abstract` deklariert werden kann. (2 Pkt.)

d. Wie ist der Quelltext der Methode `isMember` zu ändern, wenn `L` die Methoden `first`, `rest` und `isEmpty` als Instanzmethoden exportiert? (4 Pkt.)

Aufgabe P 3: Abstraktionskonzepte (12 Punkte)

a. Nennen Sie zwei Abstraktionskonzepte, die von Java unterstützt werden. Beschreiben Sie jedes von ihnen in einem Satz. (4 Pkt.)

b. Nennen Sie zwei Abstraktionskonzepte, die von Java nicht unterstützt werden. Beschreiben Sie jedes von ihnen in einem Satz. (4 Pkt.)

c. Java ist eine statisch getypte Sprache mit dynamischer Bindung. Erläutern Sie diese Aussage. (4 Pkt.)

Fragenteil PF **(8 Punkte)**

PF-1. Ein (abstraktes) Objekt wird gegeben durch seine *Zustand* und seine *Funktionalität*. Wie wird das durch Java-Objekte umgesetzt? (2 Pkt.)

PF-2. Geben Sie zwei Eigenschaften an, in denen sich Templates und generischen Klassen unterscheiden, indem Sie in der folgenden Tabelle die unterschiedliche Ausprägung dieser Eigenschaften gegenüberstellen. (6 Pkt.)

generische Klassen	**Templates**