

Probeklausur

erstellt von J. Bremer, B. Rapp

17.01.2002

Nachname:

Vorname:

Die Klausur besteht aus 4 Aufgaben. Verschaffen Sie sich einen kurzen Überblick über die Aufgaben. Beginnen Sie mit der Aufgabe, die Ihnen am ehesten ein Erfolgserlebnis zu bringen scheint.

- Sie können 70 Punkte erreichen.
- Kennzeichnen Sie jedes(!) Blatt mit Ihrem Namen.
- Bei Verständnisfragen heben Sie bitte den Arm; das "Aufsichtspersonal" ;o) bemüht sich dann um Klärung.
- Schreiben Sie leserlich! Nicht lesbare oder unklare Teile werden mit 0 Punkten bewertet.

Viel Erfolg!

Aufgabe 1 (Vererbung)

```
class A {
    int x1=2;
    int x2;
    public A(int i) {
        this.x2=i;
    }
    public void print() {
        switch(x2) {
            case 1: System.out.println("a"); return;
            case 3: System.out.println("b");
            case 2: System.out.println("c");
        }
    }
}

public class B extends A {
    int x1=1;
    int x2=2;
    public B(int x1) {
        super(1);
        ((A)this).x2=x1;
    }
    public void print() {
        System.out.println(super.x1);
    }
    public static void main(String[] args) {
        A a=new A(3);
        B b=new B(3);
        A c=new A(1);
        c.print();
        a.print();
        ((A)b).print();
        System.out.println(((A)b).x1);
    }
}
```

Analysieren Sie obige Klassen. Geben Sie an, welche Ausgaben von dem Programm in welcher Reihenfolge erzeugt werden. (15 Punkte)

Aufgabe 2 (Schleifen)

```
public class Schleife {
    public static void main(String[] args) {
        int i=1;
        while(i<=10) {
            i++;
            if(i<4) {
                System.out.println(i);
                continue;
            }
            System.out.println(i);
            i++;
        }
    }
}
```

Wandeln Sie die while-Schleife aus dem obigen Programm in eine äquivalente for-Schleife um. Äquivalent bedeutet dabei, daß die gleichen Ausgaben erzeugt werden. (10 Punkte)

Aufgabe 3 (Klassen)

Implementieren Sie eine Klasse **Vektor**, welche den Umgang mit Vektoren

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

aus \mathbf{R}^2 ermöglicht. Die Klasse **Vektor** soll folgende Methoden besitzen:

1. einen Konstruktor, der ein **Vektor**-Objekt mit zwei übergebenen float-Werten initialisiert. (3 Punkte)
2. einen Copy-Konstruktor, der ein **Vektor**-Objekt mit einem anderen **Vektor**-Objekt initialisiert. (3 Punkte)
3. eine Methode *equals*, die die entsprechende von der Klasse **Objekt** geerbte Methode überschreibt. (3 Punkte)
4. eine Methode *toString*, die die entsprechende von der Klasse **Objekt** geerbte Methode überschreibt. (3 Punkte)
5. eine Klassenmethode, welches das Standardskalarprodukt

$$\mathbf{R}^2 \rightarrow \mathbf{R}$$
$$\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \right) \mapsto \sum_{i=1}^2 x_i y_i$$

zurückliefert. (5 Punkte)

6. Schreiben Sie ein kleines Testprogramm, welches die Methode aus Teilaufgabe 5 testet. (3 Punkte)

Aufgabe 4 (Monte-Carlo-Integration)

In dieser Aufgabe sollen Sie ein Programm entwickeln welches einen Näherungswert für die Kreiszahl π bestimmt. Stellen Sie sich vor, Donald D. schießt mit einem Gewehr auf eine quadratische Fläche. Wir berücksichtigen nur Schüsse, die diese Fläche treffen. Donald ist ein so schlechter Schütze, daß die Punkte des Auftreffens völlig zufällig sind und daher die Treffer im Mittel gleichförmig auf dem Quadrat verteilt sind. Die Zahl dieser Treffer sei N . Nun zeichnen wir eine Flächenumrandung - in diesem Fall einen Kreis - in dieses Quadrat und zählen separat die Treffer, die im Inneren des Kreises liegen. Die Zahl dieser Treffer sei M . Wenn man N proportional zur Fläche des Quadrates setzt, dann muß offenbar M proportional zur Kreisfläche sein. Die Wahrscheinlichkeit p , daß ein Treffer im Kreisinnern liegt, ist gleich dem Verhältnis von Kreisfläche zu Quadratfläche.

Tipp: Arbeiten Sie mit einem Viertel des Einheitskreises (d.h. Radius=1) und einem Quadrat der Kantenlänge 1. Dann konvergiert das Verhältnis M/N gegen das Flächenverhältnis $\pi/4$.

Für die Lösung dieser Aufgabe sei Ihnen die Klasse `Random` gegeben. Diese Klasse enthält u.a. die Methode `public float nextFloat()`, welche bei jedem Aufruf eine neue Pseudozufallszahl aus dem Intervall $[0,1]$ liefert.

Schreiben Sie ein Programm, welches π näherungsweise für $N=1000$ berechnet. (25 Punkte)