



Probeklausur

Einführung in die Informatik I

Hinweis: Diese Probeklausur ist eine kleine Aufgabensammlung, die etwa dem Schwierigkeitsgrad der Teilleistung TL 2 (Programmiertest) des Moduls Inftech I entspricht. Die Probeklausur ist umfangreicher als die Teilleistung TL 2. Die Aufgaben decken jedoch nicht alle behandelten Themenbereiche ab.

Aufgabe 1 (Bedingte Anweisungen).

1. Betrachten Sie die folgende Java-Methode

```
boolean f(boolean x, boolean y) {  
    if(x) {  
        return y;  
    } else if(y) {  
        return x;  
    } else {  
        return x;  
    }  
}
```

- (a) Stellen Sie eine Tabelle auf, die für alle möglichen Wertepaare (x, y) der Parameter den zugehörigen Rückgabewert der Methode $f()$ angibt.
- (b) Implementieren Sie eine möglichst einfache Methode $f2(\text{boolean } x, \text{boolean } y)$, die ohne `if`-Anweisung auskommt und für alle möglichen Wertekombinationen von x und y die gleichen Werte wie die Methode $f()$ zurückliefert.

2. Betrachten Sie das folgende Punktesystem zur Notenvergabe:

Punkte	Note
36-40	1
31-35	2
26-30	3
21-25	4
0-20	5

Implementieren Sie eine Methode `int getNote(int punkte)`, die als Wert die Note der übergebenen Punkte zurückliefert. Liefern Sie den Wert `-1` zurück, falls der Parameter `punkte` keine gültige Punktezahl darstellt.

Aufgabe 2 (Schleifen).

1. Was ist der Wert von `k` nach Ausführung der folgenden Anweisungen?

```
int k = 2;
for(int i = 2; i < 7; i += 2) {
    k *= 2;
}
```

2. Wandeln Sie die folgende `for`-Anweisung in eine `while`-Anweisung um.

```
for(int i = 533; i > -12; i -= 7) {
    System.out.println(i);
}
```

3. Implementieren Sie eine Methode `int[] countZeroes(int[][] tabelle)`, die als Wert die Anzahl der Nullen in jeder Spalte der übergebenen Tabelle zurückliefert. Dabei sei `tabelle[i][j]` der Wert in der i -ten Zeile und j -ten Spalte der Tabelle.

Aufgabe 3 (Klassen und Objekte).

1. Ergänzen Sie die folgende Java-Klasse wie in den Kommentaren beschrieben.

```
class C {  
  
    int a;  
  
    C(int a) {  
        // initialisiere Attribut mit uebergebenen Parameter  
    }  
  
    void f() {  
        a *= 2;  
        System.out.println("a = " + a);  
    }  
  
    public static void main(String[] args) {  
        // Erzeuge ein Objekt der Klasse C. Dabei soll der Wert des  
        // Attributs a mit 5 initialisiert werden.  
  
        // Rufe die Methode f() auf.  
    }  
}
```

2. Implementieren Sie eine Klasse **Fahrrad**. Die Klasse soll folgenden Anforderungen genügen:

- Der Zustand eines Objekts der Klasse **Fahrrad** wird durch seine Geschwindigkeit und die Anzahl der Gänge beschrieben. Beide Attribute sind vom Typ **int**. Ein Zugriff auf die Attribute außerhalb der Klasse ist nicht möglich.
- Die Klasse besitzt einen Standardkonstruktor, der die Geschwindigkeit mit 0 und die Anzahl der Gänge mit 1 initialisiert.
- die Klasse besitzt einen erweiterten Konstruktor mit zwei Parametern zur Initialisierung der Attribute.
- Die Klasse besitzt eine Methode zum Ändern der Geschwindigkeit. Der Betrag zur Geschwindigkeitsänderung wird als Parameter der Methode übergeben.
- Die Klasse besitzt eine Methode zum Setzen der Anzahl der Gänge. Die Anzahl wird dabei als Parameter der Methode übergeben.
- Auf die Klasse, die Konstruktoren und auf die Methoden kann von jeder anderen Klasse zugegriffen werden.

Aufgabe 4 (Vererbung).

Betrachten sie die Klassenhierarchie in Abbildung 1.

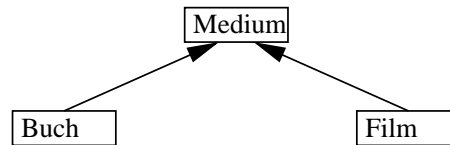


Abbildung 1: Klassenhierarchie

Bücher besitzen einen Titel und einen Autor. Filme werden durch ihren Titel und ihre Spieldauer beschrieben. Implementieren Sie die Klassenhierarchie von Abbildung 1. Gehen Sie dabei folgendermaßen vor:

1. Implementieren Sie eine Klasse `Medium`. Die Klasse `Medium` besitzt als Attribut einen Titel vom Typ `String`. Implementieren Sie einen erweiterten Konstruktor zur Initialisierung des Titels. Stellen Sie sicher, dass Unterklassen der Klasse `Medium` eine abstrakte Methode `void ausgeben()` implementieren. Die Methode `ausgeben()` gibt die Werte aller Attribute der jeweiligen Unterklasse auf der Konsole aus.
2. Implementieren Sie eine Unterklasse `Buch` der Klasse `Medium`. Die Klasse `Buch` besitzt als zusätzliches Attribut einen Autor. Implementieren Sie einen erweiterten Konstruktor zur Initialisierung der Attribute der Klasse `Buch`. Stellen Sie sicher, dass von der Klasse `Buch` Objekte erzeugt werden können.
3. Implementieren Sie eine Unterklasse `Film` der Klasse `Medium`. Die Klasse `Film` besitzt als zusätzliches Attribut die Spieldauer. Implementieren Sie einen erweiterten Konstruktor zur Initialisierung der Attribute der Klasse `Film`. Stellen Sie sicher, dass von der Klasse `Film` Objekte erzeugt werden können.

