

Übungsblatt 4

Abgabe bis Donnerstag, 24.11.11, 12:00 Uhr

Hinweis:

Lösungen immer per E-Mail an den zuständigen Tutor schicken (Bei Programmieraufgaben den Java Quellcode und eventuell benötigte Datendateien).

Aufgabe 4.1

Erweitern Sie die Klasse `Triangle` von Übungsblatt 3. Sie finden die Klasse ab dem 18.11.2011 auf der Vorlesungshomepage.

Hinweis: Gehen Sie davon aus, dass zwei `double`-Variablen a und b gleich sind, wenn gilt: $|a - b| < 10^{-6}$.

1. Schreiben Sie eine Methode `public boolean isEquilateral()`, die genau dann `true` zurückgibt, wenn das Dreieck gleichseitig ist.
2. Schreiben Sie eine Methode `public boolean isIsosceles()`, die genau dann `true` zurückgibt, wenn das Dreieck gleichschenkelig ist.

Aufgabe 4.2

Die Kreiszahl π kann mit Hilfe der folgenden Rechenvorschrift (Leibniz-Reihe) näherungsweise berechnet werden:

$$\frac{\pi}{4} = \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} \cdots$$

Schreiben Sie ein Java-Programm, das den Benutzer zur Eingabe einer ganzen Zahl i auffordert und anschließend den Wert von π bis zum i -ten Term berechnet und ausgibt.

Aufgabe 4.3

Betrachten Sie folgenden Auszug der Klasse MyCalculator.

```
public class MyCalculator {
    public static int add(int a, int b){
        ...
        while(...){...}
        ...
    }

    public static int multiply(int a, int b) {
        ...
        while(...) {...}
        ...
    }

    public static int power(int a, int p) {
        ...
        for(...) {...}
        ...
    }
}
```

- Die Methode `add(int a, int b)` soll für zwei positive `int`-Variablen $a + b$ berechnen.
- Die Methode `multiply(int a, int b)` soll für zwei positive `int`-Variablen $a \cdot b$ berechnen.
- Die Methode `power(int a, int p)` soll für zwei positive `int`-Variablen a^p berechnen.

Fügen Sie dafür an den Stellen, die mit “...” markiert sind, Ihren Code ein. Benutzen Sie dafür nur Increment (++) und Decrement (--) - Operatoren. Verwenden Sie keine der Operatoren +, -, *, / oder Methoden der Klasse Math.

Hinweis: Verwenden Sie die Methode `add` in `multiply`, sowie die Methode `multiply` in `power`.

Aufgabe 4.4

Auf der letzten Seite des Übungsblattes finden Sie einige Konventionen für die Formatierung von Java-Code. Betrachten Sie folgendes Programm und korrigieren Sie die Stellen, die nicht mit den Konventionen übereinstimmen.

```
import java.net.*;
import java.io.*;

class webpageread {
    public static void main(String[] arg) throws Exception{
        URL u = new URL("http://www.informatik.uni-freiburg.de/");
        InputStream ins = u.openStream();
        InputStreamReader isr = new InputStreamReader(ins);
        BufferedReader WebPageBuffer = new BufferedReader(isr);

        int NoOfLines =10;

        if(NoOfLines==0){
            return;
        }

        if(NoOfLines == 1){
            System.out.println("Read " + NoOfLines + " line of " + u );
        }
        else{
            System.out.println("Read " + NoOfLines + " lines of " + u );
        }

        for(int i=1; i<=NoOfLines ; i++)
        {
            System.out.println("line #" +i+ " : \""+ WebPageBuffer.readLine() +"\"");
        }
    }
}
```

Codestyle - Konventionen

Ihre Programme sollten folgende Konventionen einhalten:

1. Variablen- und Methodennamen: $[a - z][a - zA - Z0 - 9_]^*$
(d.h. erstes Zeichen Kleinbuchstabe, folgende Zeichen beliebige Buchstaben oder Unterstriche). Die Bezeichnung der Variablen bzw. Methoden sollte möglichst klar ihre Bedeutung im Programm beschreiben.
2. Klassennamen: $[A - Z][a - zA - Z0 - 9_]^*$
(d.h. erstes Zeichen Großbuchstabe, folgende Zeichen beliebige Buchstaben oder Unterstriche).
3. Zeilenlänge maximal 80 Zeichen.
4. Leerzeichen nach “,”.
5. Leerzeichen um zweistellige Operatoren, wie z.B. “+”, “-”, “<” oder “=”.
6. If-Blöcke in der Form:

```
if (i < j) {  
    System.out.println("i < j");  
} else {  
    System.out.println("j <= i");  
}
```

mit Leerzeichen nach `if` und `else` sowie Leerzeichen vor geschweiften Klammern.

7. For-Schleifen in der Form:

```
for (int i = 0; i < 10; ++i) {  
    System.out.println("i");  
}
```

mit Leerzeichen nach `for` sowie Leerzeichen vor geschweiften Klammern.

8. While-Schleifen in der Form:

```
while (i < 10) {  
    System.out.println("i");  
    ++i;  
}
```

mit Leerzeichen nach `while` sowie Leerzeichen vor geschweiften Klammern.