

Übungen zu Informatik B

Sommersemester 1970

Musterlösungen zum Übungsblatt 1

Aufgabe 1.1 (Thema der Aufgabe, 12 Punkte)

Hier den Aufgabentext einfügen. Bei Bezug auf Programmcode den `Teletype`-Schriftstil einsetzen (z. B. Methoden- oder Attributnamen). Einzelne oder mehrere Wörter können z. B. unterstrichen, **fett** oder *kursiv* gesetzt werden. Für einfache Formeln die entsprechende L^AT_EX-Notation verwenden (z. B. $2 \leq l \leq k$). Für Teilaufgaben ist die `enumerate`-Umgebung zu verwenden:

- a) Für Teilaufgaben möglichst die jeweils erreichbaren Punkte angeben. (4 Punkte)
- b) Noch eine Teilaufgabe. (4 Punkte)
- c) Und hier die letzte Teilaufgabe. (4 Punkte)

Musterlösung:

- a) Lösung für Teilaufgabe einfügen.
- b) Lösung für Teilaufgabe einfügen.
- c) Lösung für Teilaufgabe einfügen.

In der Musterlösung keine Punktezahlen eintragen, um Inkonsistenzen zum ausgegebenen Übungsblatt vorzubeugen.

Einzelne Formeln oder nicht zusammenhängende Teile werden mit der `displaymath`-Umgebung gesetzt:

$$VD[PM] = a \cdot (KLOC^b)$$

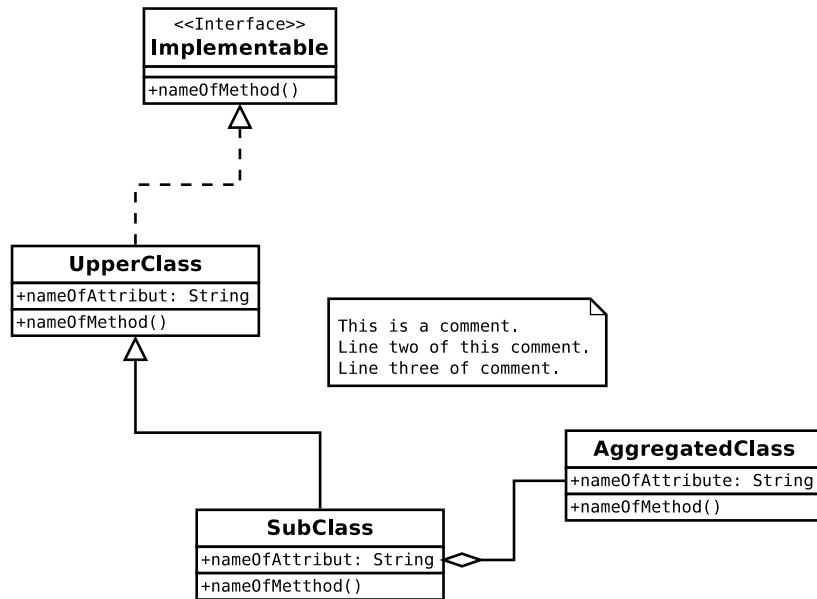
$$a = 1000/600 = 1 \frac{2}{3}$$

$$1 \frac{2}{3} \cdot 10^{1,19} = 25,81 \text{ PM}$$

Mehrere zusammengehörige Formeln können mit der `eqnarray*`-Umgebung gesetzt werden:

$$\begin{array}{rcl} 5 + 4 + 8 \cdot 5 + 2 \cdot 3 & = & 55 \\ 55 \cdot 0,01 + 0,65 & = & 1,2 \end{array}$$

Vorgehen zur Erstellung einer Abbildung im PDF-Format: Eine mit *Dia* erzeugte Abbildung kann in der Anwendung als .eps-Datei exportiert werden, welche wiederum mittels `epstopdf` in eine PDF-Datei konvertiert werden kann (geschieht automatisch wenn möglich, siehe Paket „epstopdf“). Eine Abbildung kann folgendermaßen eingebunden werden:



Beispiel für ein UML-Klassendiagramm

Bitte zur Erstellung von Abbildungen möglichst Werkzeuge verwenden, die im CIP-Pool verfügbar sind, also z. B. *Dia*, *xfig*, *OpenOffice.org*, *ArgoUML* etc.

Für Erstellung der HTML-Fassung mit $\text{T}_{\text{E}}\text{X}4\text{ht}$ ist zu beachten, dass $\text{T}_{\text{E}}\text{X}4\text{ht}$ zuerst per `latex`-Befehl die DVI-Ausgabe erzeugt und anschließend hieraus per `dvipng` die Abbildungen extrahiert. Bei Einsatz des `dvipng`-Konverters kann es allerdings vorkommen, dass dieser ggf. einige Pixel zuviel vom Rand der Abbildungen abschneidet. Um sicherzustellen, dass die Abbildungen vollständig sind, sollten diese in der Ursprungsfassung daher mit einem unsichtbaren (oder weißen) Rahmen versehen werden.

Aufgabe 1.2 (Thema der Aufgabe, 15 Punkte)

Hier wieder die allgemeine Aufgabenbeschreibung einfügen.

Um alle Aufgaben möglichst auf zwei Seiten (Vorder- und Rückseite) zu bekommen, können Aufzählungen z. B. mit `multicols` platzsparend mehrspaltig gesetzt werden:

- | | |
|---|--|
| a) Eine Teilaufgabe. (3 Punkte) | d) Noch eine Teilaufgabe. (3 Punkte) |
| b) Wieder Teilaufgabe. (3 Punkte) | e) Und hier die letzte Teilaufgabe. (3 Punkte) |
| c) Eine weitere Teilaufgabe. (3 Punkte) | |

Alternativ, um eine noch kompaktere Darstellung zu erreichen, kann z. B. auf die `enumerate`-Umgebung verzichtet und die Aufzählung manuell angegeben werden:

- | | |
|---|--|
| a) Eine Teilaufgabe. (3 Punkte) | d) Noch eine Teilaufgabe. (3 Punkte) |
| b) Wieder Teilaufgabe. (3 Punkte) | e) Und hier die letzte Teilaufgabe. (3 Punkte) |
| c) Eine weitere Teilaufgabe. (3 Punkte) | |

Musterlösung:

Als weitere Alternative kann die Umgebung `parcolumns` für mehrspaltigen Satz verwendet werden:

- a) Eine Antwort auf eine Teilaufgabe.
- b) Wieder Antwort auf eine Teilaufgabe.
- c) Eine weitere Antwort auf eine Teilaufgabe.
- d) Noch eine Antwort auf eine Teilaufgabe.
- e) Und hier die letzte Antwort auf eine Teilaufgabe.

Aufzählungen innerhalb der `parcolumns`-Umgebung sind allerdings etwas aufwendiger, weil der Aufzählungszähler für jede Spalte manuell gesetzt werden muss. Zudem ist die `parcolumns`-Umgebung für die HTML-Fassung ungeeignet, so dass hierfür als Alternative (für die entsprechende HTML/PDF-Weiche siehe L^AT_EX-Quelltext) eine „normale“ Aufzählung redundant angegeben werden muss (einspaltig).

Aufgabe 1.3 (Thema der Aufgabe, 0 Punkte)

Umlaute bitte ausschließlich in der folgenden Notation verwenden:

ä → "a
 ö → "o
 ü → "u
 ß → "s

Dies erleichtert den Dokumentenaustausch zwischen unterschiedlichen Betriebssystemen und Textkodierungen erheblich.

Beispiel für die Anzeige von mit dem Übungsblatt ausgegebenem Quellcode. *Anmerkungen:* Es wird ein klickbarer WWW-Link erzeugt. Der Linktext dagegen verweist auf das Dateisystem im CIP-Pool. Für den WWW-Link und den Linktext werden der aktuelle Jahrgang und die aktuelle Blattnummer automatisch ermittelt und eingefügt (vgl. Makro `myfile` in `blatt_masterfiles/blatt_latex_macros.tex`). Die Links auf den Übungsblättern werden somit **automatisch aktualisiert**. Neben Quellcode kann natürlich auf beliebige andere Dateien verwiesen werden:

```
/home/binf/aufgaben/blatt01/code/MatrixScalarMultiplication.java
```

Beispiel für die Anzeige von benötigtem Quellcode aus der Vorlesung. *Anmerkungen:* Für einige Aufgaben werden Verweise auf Dateien aus der Vorlesung benötigt. Wird das folgende Makro verwendet, gilt dasselbe wie oben, d. h. die Links werden im jeweiligen Jahrgang **automatisch aktualisiert** (vgl. Makro `myfileLecture` in `blatt_masterfiles/blatt_latex_macros.tex`):

```
/home/binf/code/xyz_src/xyz.java
```

Musterlösung:

Programmlistings werden mit Hilfe des Pakets `listings` eingebunden:

```
SqrtHeron.java
```

```

1
2 /**
3  * Berechnung einer Naeherung der Quadratwurzel einer Zahl mit einer Folge nach
4  * dem Heronverfahren.
5  *
6  * @author Wolfgang Runte
7  * @version 07.12.2010
8  */
9 public class SqrtHeron {
10
11     /**
12      * Berechnung einer Naeherung der Quadratwurzel einer Zahl mit einer Folge
13      * nach dem Heronverfahren.
14      *
15      * @param zahl eine Zahl, von der die Quadratwurzel berechnet werden soll
16      * @return Quadratwurzel zur uebergebenen Zahl, -1 fuer Eingaben < 0
17      */
18     public static double sqrtHeron(double zahl) {
19         // Schranke, vordefiniert fuer einstellige Zahl
20         double epsilon = 1e-15;
21         // max. Anzahl Folgenglieder
22         final int MAXIMUM = 100000;
23         // aktuelles Folgenglied
24         double x = 0;
25         // Folgengliednummer
26         int zaehler = 1;
27
28         // liefere -1 fuer Eingaben < 0
29         if (zahl < 0) {
30             x = -1;
31         }
32
33         // zahl <= 0 laesst sich nicht berechnen
34         if (zahl > 0) {
35             // aktuelles Folgenglied mit uebergebenen Zahl initialisieren
36             x = zahl;
37
38             // Kopie der uebergebenen Zahl erstellen
39             double kopie = zahl;
40
41             // Berechnung der Schranke
42             do {
43                 // solange 'kopie' noch mehr als eine Stelle vor dem Komma hat
44                 // ziehe eine Stelle ab...
45                 kopie = kopie / 10;
46                 // ... und multipliziere Schranke mit 10
47                 epsilon = epsilon * 10;
48             } while (kopie > 1);
49
50             // Berechnung der Quadratwurzel
51             do {
52                 // naechstes Folgenglied berechnen
53                 x = (x + zahl / x) / 2.0;
54                 zaehler++;
55             } while (zaehler < MAXIMUM && Math.abs(x * x - zahl) >= epsilon);
56         }
57         return x;
58     }
59 }

```

Aufgabe 1.4 (Thema der Aufgabe, 0 Punkte)

Musterlösung:

Aufgabe 1.5 (Offener Frageteil, 10 Punkte)

Beantworten Sie Ihrer Tutorin bzw. Ihrem Tutor Fragen zur Veranstaltung „Informatik B“.