Informatik II (ITET)
SS 2007
Prof. Dr. F. Mattern, Dr. K. Römer
Silvia Santini (santinis@inf.ethz.ch)

ETH Zürich
Institut für Pervasive Computing
Verteilte Systeme
http://www.inf.ethz.ch/vs

Übungsserie Nr. 1

Ausgabe: 28. März 2007
Abgabe: 4./5. April 2007

Bitte geben Sie das bearbeitete Übungsblatt immer eine Woche später **in der Übungsstunde** ab. Schicken Sie die **Programme** zusätzlich auch per **E-Mail** an Ihren Tutor. In den E-Mails sollten Sie als Betreff unter anderem „`I2 Blatt x Team y`" eintragen (das erleichtert das Sortieren der Mails). Ihre Team Nummer wird Ihnen in der Übungsstunde von Ihrem Tutor zugewiesen.

1. Aufgabe: (4 Punkte) Altägyptische Multiplikation

In der Vorlesung wurde die Korrektheit des altägyptischen Multiplikationsschemas für alle $a, b \in \mathbb{N}^+$ mittels Induktion über b bewiesen (Skript S. 25).

(1a) *(1 Punkt)* Könnte man die Korrektheit (in entsprechend einfacher Weise) auch durch Induktion über a zeigen? Geben Sie eine Begründung für Ihre Antwort.

(1b) *(1 Punkt)* Kann man auch die Terminierung des Algorithmus (für alle erlaubten Eingabewerte) beweisen? Begründen Sie auch hier Ihre Antwort.

(1c) *(2 Punkte)* Wie ist der Beweis zu ändern, wenn das Programm auf Seite 17 des Skripts so modifiziert wird, dass es statt

```
if (b == 1) return a;
```

heisst:

```
if (b == 0) return 0;
```

2. Aufgabe: (6 Punkte) Aufwandsabschätzung

In der Vorlesung werden folgende Methoden eines funktionalen Programms diskutiert:

```
static boolean gerade( int x ) {
    if( x == 0 ) return true;
    return !gerade( x - 1 );
}

static int verdopple( int x ) {
    if( x == 0 ) return 0;
    return 2 + verdopple( x-1 );
}

static int halbiere( int x ) {
    if( x == 0 ) return 0;
    if( x == 1 ) return 0;
    return halbiere( x - 2 ) + 1;
}

 static int f(int a, int b) {
     if (b == 0) return 0;
     if (gerade(b)) return f(verdopple(a),halbiere(b));
     else return a + f(verdopple(a), halbiere(b));
 }
```

(2a) *(1 Punkt)* Wie oft rufen sich die Methoden *gerade*, *verdopple*, *halbiere* bei Eingabe eines Wertes x jeweils selbst rekursiv auf?

(2b) *(2 Punkte)* In f werden alle drei Methoden aufgerufen. Wieviele Aufrufe (in Abhängigkeit von a, b) der drei Methoden gibt es insgesamt bei **einem** Aufruf von f? Ignorieren Sie dabei zunächst, dass f sich selbst rekursiv aufruft (d.h., betrachten Sie nur eine einzige Instanz der Methode).

(2c) *(3 Punkte)* Nun sollen Sie berücksichtigen, dass in $f(a, b)$ wiederum $f(2a, b/2)$ aufgerufen wird u.s.f., bis $b = 1$ erreicht wird. Ermitteln Sie die Gesamtzahl aller Methodenaufrufe (als Funktion von a und b). Um dies mit einfachen Formeln ausdrücken zu können, sollten Sie kleinere Fehler (z.B. Auf- oder Abrunden, Vernachlässigung von Nachkommastellen bei der Logarithmierung etc.) bewusst in Kauf nehmen, und nur eine (natürlich möglichst gute) Abschätzung liefern.

3. Aufgabe: (4 Punkte) Rekursion und Iteration

Beachten Sie die folgende rekursive Definition der Funktionen x^n und $n!$:

$$x^n = \begin{cases} 1 & , n = 0 \\ x \cdot x^{n-1} & , n > 0 \end{cases}$$

$$n! = \begin{cases} 1 & , n = 0 \\ n \cdot (n-1)! & , n > 0 \end{cases}$$

(3a) *(2 Punkte)* Schreiben Sie die Java Methoden:

```
int power_rec(int x, int n)
```

und

```
int factorial_rec(int n)
```

die die zwei oben gegebenen Funktionen **rekursiv** implementieren.

(3b) *(2 Punkte)* Können die zwei Funktionen x^n und $n!$ auch mittels eine `while`-Schleife, d.h. **iterativ**, implementiert werden? Wenn ja, schreiben Sie die entsprechenden Java Methoden:

```
int power_it(int x, int n)
```

und

```
int factorial_it(int n)
```

4. Aufgabe: (6 Punkte) Javadoc

In der Praxis kommt es gelegentlich vor, dass man auf alte, undokumentierte und unkommentierte Programmteile stösst, die man verstehen muss, um beispielsweise Änderungen daran vornehmen zu können. Die tiefer liegende Idee aus einem Programm herauszuarbeiten, ist oft schwierig, wenn das Programm nicht gut kommentiert ist.

(4a) *(2 Punkte)* Probieren Sie es selbst aus. Das Programm `ClassX.java` finden Sie zum Herunterladen auf der Vorlesungswebseite. Erkennen Sie die dem Programm `ClassX.java` zugrunde liegende Idee durch blosses Draufschauen und Nachdenken und geben Sie sie an.

(4b) *(4 Punkte)* Um eine einfache Wiederverwendung von Java-Code zu erlauben, hat Sun Microsystems das Programm *Javadoc* entwickelt, das Bestandteil des Java 2 Software Development Kits (J2SDK) ist. *Javadoc* ist ein Software-Dokumentationswerkzeug, das aus Java-Quelltexten automatisch HTML-Dokumentationsdateien erstellt. Dazu müssen spezielle Kommentare im Quelltext eingefügt werden.
Fügen Sie jetzt in das oben angegebene Programm geeignete Kommentare ein (Sie können das Programm `ClassX.java` von der Webseite herunterladen).
Wenden Sie das Programm Javadoc auf das so kommentierte Programm an. Führen Sie dazu folgenden Befehl aus:

```
$ javadoc ClassX.java
```

oder benutzen Sie ggf. die eingebauten Befehle Ihres Java-Editors. Die Ausführung des obigen Befehls erzeugt eine `html`-Datei, die mit einem Internet-Browser geöffnet werden kann. Schicken Sie die erzeugte Datei (`ClassX.html`) und die modifizierte Version der Datei `ClassX.java` an Ihren Tutor!
Wie genau Sie Kommentare für *Javadoc* erstellen können (Sie müssen dabei einen gewissen Dokumentationsstil verwenden), wird auf der Sun-Webseite:

```
http://java.sun.com/j2se/javadoc/writingdoccomments/index.html
```

beschrieben. Dort erhalten Sie auch weitere Hinweise zur Benutzung von *Javadoc*.

5. Aufgabe: (5 Punkte) Halteproblem

Von einem Algorithmus weiss man gerne, ob er terminiert, also zu Ende kommt (man sagt, er hält) oder ob er ewig weiterrechnet. Dies sieht man ihm im Allgemeinen nicht an. Betrachten Sie den folgenden Algorithmus:

1. Starte mit einer beliebigen ganzen Zahl $n \in \mathbb{N}$
2. Falls n und $n + 2$ Primzahlen sind: fertig
3. Sonst: ersetze n durch $n + 1$ und gehe zu Schritt 2

(5a) *(3 Punkte)* Führen Sie eine (manuelle!) Berechnung des Algorithmus für die Anfangswerte $n \in \{2, 7, 98\}$ durch.

(5b) *(2 Punkte)* Hält der angegebene Algorithmus für beliebige Anfangswerte $n \in \mathbb{N}$? Begründen Sie Ihre Antwort.

Summe: 25 Punkte