



Grundlagen der Informatik 1

Wintersemester 2009/2010

Prof. Dr. Max Mühlhäuser, Dr. Rößling
<http://proffs.tk.informatik.tu-darmstadt.de/teaching>

Übung 8 Version: 1.0

07. 12. 2009

1 Mini Quiz

1. ☐ Scheme ist näher an der Funktionsweise eines Rechners orientiert als Java.
2. ☐ '–' kann als unäre Operation verwendet werden.
3. ☐ $a \gg b$ verschiebt die Bitrepräsentation von a um b Stellen nach rechts.
4. ☐ $<$ hat eine niedrigere Piorität als $\&\&$.

2 Fragen

1. Was ist der Hauptunterschied zwischen Java und Scheme?
2. Erklären Sie, was Operator-Priorität ist und bringen Sie folgende Operatoren in die richtige Reihenfolge: $*$, $=$, $\&\&$, \leq , $++$, $-$.
3. Warum spielte in Scheme die Operator-Priorität keine Rolle?
4. Erklären Sie, was Operator-Assoziativität ist. Welche Assoziativität haben die oben in 2.2 genannten Operatoren?

3 Schleifen

Gegeben sei folgende while Schleife in Java:

```
1 int x = 2;  
2 int number = 10;  
3 int count = 1;  
4 while (Math.pow(x, count) <= number){  
5     count++;  
6 }
```

Hinweis: `Math.pow(a,b)` berechnet a^b .

1. Was macht die Funktion? Kommt sie Ihnen bekannt vor?
2. Ersetzen Sie die while-Schleife durch eine for-Schleife.
3. Ersetzen Sie die while-Schleife durch eine `do ... while`-Schleife.

4 Scheme in Java

4.1 RGB Werte

In Übung 3.3 haben Sie eine Scheme-Prozedur geschrieben, die aus einem gegebenen Farbwert in RGB den zugehörigen Farbwert (Hue-Wert) ermittelt. Schreiben Sie nun die entsprechende Methode `double getHue(int red, int green, int blue)` in Java. Zur Erinnerung, die Berechnung erfolgt wie folgt:

$$\begin{aligned} r &= red/255, g = green/255, b = blue/255 \\ M &= \max(r, g, b), m = \min(r, g, b) \\ H &= \begin{cases} 0 & \text{wenn } M = m \\ 60(g - b)/(M - m) & \text{wenn } r = M \\ 120 + 60(b - r)/(M - m) & \text{wenn } g = M \\ 240 + 60(r - g)/(M - m) & \text{wenn } b = M \end{cases} \end{aligned}$$

4.2 Skalarprodukt

In Übung 2.6.1 haben Sie das Skalarprodukt zweier Vektoren berechnet. Schreiben Sie diese Methode in Java als `int computeScalar(int[] a, int[] b)`. Vektoren sollen dabei als ein Array von Zahlen repräsentiert werden. Zur Erinnerung: Das Skalarprodukt zweier Vektoren ist die Summe der Produkte der Komponenten der Vektoren:

`scalarProduct(new int[]{3, 1, 0}, new int[]{1, 0, 2}) = 3*1 + 1*0 + 0*2.`

4.3 Selection Sort

In Übung 5.6 haben Sie Selection Sort implementiert. Zur Erinnerung: Beim Durchlaufen des Arrays der Größe n (in Scheme: der Liste) wurde im Durchlauf i ($= 0, \dots, n-1$) das Minimum der Elemente ab Position $i+1$ bestimmt und mit dem Element an Position i vertauscht. Daher verringert sich die Länge der zu sortierenden Arrayteilstücke mit jeder Iteration um eins, womit sichergestellt ist, dass die Prozedur terminiert.

Wenn Sie am Rechner arbeiten, schreiben Sie zur Kontrolle der Sortierung eine weitere Methode `void printArray(int[] array)`, die ein Array auf der Konsole ausgibt. Verwenden Sie dazu die Methoden aus ACM JTF und den in Foliensatz T11.57 vorgestellten `StringBuffer`.

Implementieren Sie nun Selection Sort in Java (`void selectionSort(int[] array)`). Verwenden Sie `printArray` wenn Sie am Rechner arbeiten, um sich die Schritte der Sortierung anzeigen zu lassen.

Hausübung

Die Vorlagen für die Bearbeitung werden im Gdl1-Portal bereitgestellt. Kommentieren Sie Ihren selbst erstellten Code. Die Hausübung muss bis zum Abgabedatum im Gdl1-Portal abgegeben werden. Der Fachbereich Informatik misst der Einhaltung der Grundregeln der wissenschaftlichen Ethik großen Wert bei. Zu diesen gehört auch die strikte Verfolgung von Plagiarismus. Mit der Abgabe Ihrer Hausübung bestätigen Sie, dass Sie bzw. Ihre Gruppe alleiniger Autor des gesamten Materials sind. Falls Ihnen die Verwendung von Fremdmaterial gestattet war, so müssen Sie dessen Quellen deutlich zitieren. Falls Sie die Hausübung in einer Lerngruppe bearbeitet haben, geben Sie dies bitte deutlich bei der Abgabe an. Alle anderen Mitglieder der Lerngruppe müssen als Abgabe einen Verweis auf die gemeinsame Bearbeitung einreichen, damit die Abgabe im Portal auch für sie bewertet werden kann.

Abgabedatum: Freitag, 18. 12. 2009, 16:00 Uhr

Denken Sie bitte daran, Ihren Code hinreichend gemäß der Vorgaben zu kommentieren (Scheme: Vertrag, Beschreibung und Beispiel sowie zwei Testfälle pro Funktion; Java: JavaDoc). Zerlegen Sie Ihren Code sinnvoll und versuchen Sie, wo es möglich ist, bestehende Funktionen wiederzuverwenden. Wählen Sie sinnvolle Namen für Hilfsfunktionen und Parameter.

5 Hello Java (0 Punkte)

Bevor Sie mit der eigentlichen Hausübung beginnen, sollten Sie sich zuerst mit den Sprachelementen von Java und Ihrer Entwicklungsumgebung vertraut machen. Wir empfehlen Ihnen *Eclipse* als Entwicklungsumgebung. Eclipse kann mit `eclipse` auf den Poolrechnern aufgerufen werden. Auf der Gdl 1 Veranstaltungsseite finden Sie unter *Java Materialien - bitte unbedingt lesen!* neben dem Downloadlink von *Java* und *Eclipse* auch einen Link zum 'Java-Tutorial von Sun'.

Wenn Sie Eclipse benutzen möchten, empfehlen wir Ihnen, das „Create a Hello World application“ Tutorial durchzuarbeiten, das Sie in Eclipse über `Help> Welcome> Tutorials` finden können.

Programme werden in Eclipse über `Run> Run As> Java Application` ausgeführt; eventuelle Ausgaben sehen Sie in der *Console*. Sie können zum Testen den folgenden Programmcode nutzen:

```

1  class HelloJava {
2      public static void main(String[] args) {
3          System.out.println(" Hello Java!");
4      }
5  }
```

6 Chaos' Code (4 Punkte)

Die Firma "Schematic" hat beschlossen, einen Teil ihrer Softwareentwicklung fortan in Java zu machen. Der Mitarbeiter Herr Chaos hat dabei wohl seinen Namen ein wenig zu ernst genommen und folgenden Code zustande gebracht (den Sie auch auf der Vorlesungsseite herunterladen können) - und von dem er stolz sagt, dass er "im Wesentlichen" schon funktioniert:

```

1  public class Confused {
2      int[] b; public void
3  procC(int i, int j){int c = b[i];b[i]=
4  b[j];b[j]=c;}
5
6      public void procB() {
7  boolean a=true;while(a){a = false;for
8      (int i=0; i < b.arraySize-1; i++)if
9      (b[i] > b[i+1]) {procC(i,i+1); a = true;}}
10
11     public String procA(){String d="";for (int i=0;
12 i<b.arraySize; i++)d= d+b[i]+" ";return d;}
13
14     public static void main(String[] args) {Confused c
15 = new Confused();c.b= new int[] {0,9,4,4,6,2,8,5,1,7,3};
16 c.procB();c.procA();System.out.println(c.procA());}}
```

Damit wir erraten können, was hier passiert, sollten zunächst einige Schritte zur Verschönerung und Bereinigung des Codes vorgenommen werden. Wir empfehlen hierzu Eclipse zu verwenden, damit Sie Eclipse genauer kennen lernen (die in Eclipse benötigten Operationen werden im Folgenden angegeben), aber Sie können das auch mit jedem anderen Tool machen.

1. Formatieren Sie den Code, so dass nur noch ein Befehl in einer Zeile steht. In Eclipse können Sie dazu `Source->Format` verwenden (0.5 Punkte).
2. Der Code lässt sich so wie er ist nicht kompilieren, da ein Attribut `arraySize` verwendet wird, welches nicht existiert. Der Entwickler wollte hier wohl die Länge des Arrays herausfinden. Wie heißt das Attribut richtig? Wenn Sie Eclipse verwenden, gehen Sie dazu mit dem Cursor hinter `b.` und drücken `Strg+Leertaste`. Eclipse zeigt Ihnen dann an, welche Attribute und Funktionen für Arrays zur Verfügung stehen. Wählen Sie das Passende aus (0.5 Punkte).
3. Als nächstes benennen wir einige Variablen um, da `a`, `b`, `c` und `d` nicht besonders aussagekräftige Namen sind. `a` soll `sorted` heißen, `b` `intArray`, `c` `temp` und `d` `result`. Dies kann

- mit Eclipse wie folgt gemacht werden: selektieren Sie die entsprechende Variable (Cursor auf den Namen setzen), wählen Sie Refactor->Rename, geben Sie den neuen Variablennamen ein und bestätigen Sie den neuen Namen mit Return (0.5 Punkte).
4. Geben Sie nun auch den Methoden (procA, procB, procC) und der Klasse Confused aussagekräftigere Namen. Gehen Sie dabei mit Eclipse wie bei der Umbenennung der Variablen vor (0.5 Punkte).
 5. Die Firmenpolitik untersagt eigentlich, dass Variablen (hier intArray) in der main-Methode direkt gesetzt werden dürfen. Schreiben Sie daher eine Methode getIntArray, die intArray zurückgibt und eine Methode setIntArray, die ein int[] entgegennimmt und in der Variablen intArray speichert. In Eclipse können Sie dazu die entsprechende Variable anwählen und über Source ->Generate Getters and Setters... die zu erzeugenden Methoden auswählen (0.5 Punkte).
 6. Ändern Sie die Methode, die bis eben procA hieß, so ab, dass sie statt einem String und dem "+"-Operator einen StringBuffer (siehe T11.57) verwendet, um das Integer-Array auszugeben. Passen Sie dann Ihr Programm so an, dass es ein *ACM JTF DialogProgram* ist und verwenden Sie *println*, um den im vorherigen Schritt erzeugten String auszugeben (0.5 Punkte).
 7. Fügen Sie zuletzt auch noch angemessene Kommentare im JavaDoc-Format hinzu (1 Punkt).

7 Gdl Kalender (6 Punkte)

In dieser Aufgabe sollen Sie einen stark vereinfachten Terminkalender implementieren. Ein Terminkalender besteht dabei aus mehreren öffentlichen oder privaten Einträgen. Die Einträge haben alle einen Zeitpunkt, einen Besitzer, eine Beschreibung und können optional als privat markiert werden. Eine Vorlage als Projekt für Eclipse können Sie aus dem Portal herunterladen.

1. Beginnen Sie damit, das Archiv des Projektes in Eclipse zu importieren. Legen Sie dazu ein neues Java Projekt namens "GDI1Calendar" an, wählen Sie aus dem Menu Datei -> Importieren und importieren Sie das ZIP-Archiv aus dem Forum in Ihr Projekt. Öffnen Sie die Eigenschaften des Projektes durch einen Rechtsklick auf den Namen und fügen Sie JUnit 4 als Bibliothek zum "Build Path" hinzu (1 Punkt).

Hinweis: Die JUnit4 Bibliotheken sind in Eclipse vordefiniert und einfach über "Add Library" in Java Build Path -> Libraries verfügbar.

2. Öffnen Sie die Klasse User.java und fügen Sie die privaten Attribute givenName und familyName ein. Schreiben Sie entsprechende Zugriffsmethoden und implementieren Sie den vorgegebenen Konstruktor. Wenn Sie alles richtig gemacht haben, sollte die bereitgestellte Methode testUserCreation() in CalendarTest.java frei von Fehlern und Warnungen sein (1 Punkt).

Hinweis: Wenn Sie ihren Parameter in Methodenaufrufen den gleichen Namen wie Attributen der Klasse geben, "verstecken" die Parameter die gleichnamigen Klassenattribute (-> lexical scoping). Die getter/setter Methoden, welche von Eclipse angelegt werden, umgehen dieses Problem, indem Klassenattributen bei der Zuweisung explizit ein this. vorangestellt wird. Damit ist es möglich, Parametern und den zugehörigen Klassenattributen den gleichen Namen zu geben.

3. Öffnen Sie nun die Klasse CalendarEntry.java und implementieren Sie auch hier alle ange-deuteten Methoden. Schreiben Sie ebenfalls die Zugriffsmethoden für die benötigten Attribute der Klasse.

Zusätzlich soll es ein boolean-Attribut namens `priv` geben, welches die Sichtbarkeit der Objekte der Klasse regelt. Die `toString()` Methode soll das Datum des Eintrages, den Besitzer und die Beschreibung in einer Zeile ausgeben. Beachten Sie hier den zweiten Hinweis (1 Punkt).

Hinweis: Die Dokumentation für `SimpleDateFormat` und `GregorianCalendar` finden Sie, zusammen mit allen anderen Klassen der JVM auf <http://java.sun.com/javase/6/docs/api/>. Für die Arbeit am Laptop oder heimischen PC ist es sehr sinnvoll, die Dokumentation unter <http://java.sun.com/javase/downloads/index.jsp> herunterzuladen und damit immer auf Ihrem Rechner verfügbar zu haben.

Hinweis: Die Darstellung eines Kalendareintrages soll die folgende Form haben:

Mon, 12 April 2010 - 18:30:00 by Paul Anderson | Meet Mary for cinema.

4. Implementieren Sie nun ebenso die Klasse `Calendar.java`. Limitieren Sie nicht die maximale Anzahl der `CalendarEntry` Objekte pro `Calendar` Objekt. Verwenden Sie in der Methode `listEntries` die oben implementierte Methode `toString()` aus der `CalendarEntry` Klasse. Prüfen Sie hier auch die Sichtbarkeit eines Kalendareintrages - private Einträge sind nur für den entsprechenden Besitzer ersichtlich (2 Punkte).

Hinweis: Mit `System.arraycopy(src, offset, dest, offset, length)` können Sie ein Array in ein anderes, evtl. größeres Array kopieren.

Hinweis: Wenn Sie alles richtig gemacht haben, sollten alle Testfälle aus `CalendarTest.java` erfolgreich durchlaufen.

5. Fügen Sie zuletzt auch noch angemessene Kommentare im JavaDoc-Format hinzu (1 Punkt).

Hinweis: Die getter und setter Methoden der Attribute brauchen Sie nicht zu kommentieren.