Programmierstarthilfe SS 2008
Fakultät für Ingenieurwissenschaften und Informatik

4. Blatt
Für den 19. und 20.5.2008

Organisatorisches

Um auf die Mailingliste aufgenommen zu werden schicke einfach eine Mail an `guido.de-melo@uni-ulm.de`.

Die Webseiten zur Veranstaltung sind unter `http://www.uni-ulm.de/in/mi/lehre/ss2008/programmierstarthilfe.html` zu finden.

1. Methoden

1.1. Einführung in Methoden

Mit Methoden kann man Code an verschiedenen Stellen wiederverwenden. Dadurch spart man Schreibarbeit und bekommt ein strukturierteres Programm. Eine Methodendeklaration sieht zum Beispiel so aus:

```
1 public static int machWas(int parameter1, String parameter2){
2   // Code
3   return 5;
4 }
```

Diese Methode heißt `machWas` und hat zwei Parameter vom Typ `int`, bzw. `String`. Methodenname und Parameter zusammen nennt man Signatur.

Direkt vor dem Methodennamen steht der so genannte Rückgabewert (in diesem Fall `int`). Dieser wird nach einem Aufruf mit `return` an die aufrufende Stelle zurückgegeben. Ein Aufruf könnte so in der `main`-Methode stehen:

```
1 int ergebnis = machWas(15, "Hallo");
```

In diesem Fall wird `machWas` ausgeführt, und das Ergebnis der Methode der Variablen `ergebnis` zugewiesen.

Das Attribut `static` benötigt man, damit die Methode aus der `main`-Methode aufgerufen werden kann.

Der Modifizierer `public` kann auch weggelassen werden. Dieser ist später für die Objektorientierung wichtig.

1.2. Methode für Summierung

a) Schreibe eine Methode, die zwei natürliche Zahlen a und b übergeben bekommt und alle Zahlen von a bis b aufsummiert und die Summe zurückgibt.

b) Erweitere das Programm so, dass immer von der kleineren zur größeren Zahl gezählt wird, unabhängig davon, in welcher Reihenfolge sie als Parameter der Methode übergeben werden.

1.3. Rechnen mit Methoden

a) Schreibe jeweils eine Methode für Addition, Subtraktion und Multiplikation. Die Methoden sollen als Parameter zwei Zahlen entgegen nehmen, und das Ergebnis der Rechnung zurückgeben.

b) Teste deine Methoden mit einfachen eigenen Rechenbeispielen.

c) Berechne das Ergebnis des Rechenterms (5+3)(6-2) mithilfe deiner Methoden. Überlege dir zunächst, wie du schrittweise vorgehen musst.

1.4. *Zählen von Buchstaben in Texten

Es soll eine Methode geschrieben werden, die einen Text und einen Buchstaben als Parameter entgegennimmt und zurückgibt, wie oft der Buchstabe im Text vorhanden ist.
Wenn beispielweise der Text „das ist ein test“übergeben wird, und alle Vorkommen von 't' gezählt werden, so soll die Methode die Zahl 3 zurückgeben

a) Wie sieht die sogenannte Methodensignatur aus? Also welche Datentypen werden für die Paramter benötigt und was wird zurückgegeben?

b) Schreibe die Methode in Java und teste sie Anhand von verschiedenen Testfällen. Beachte zur Vereinfachung ausschließlich das kleine Alphabet.

c) Schreibe nun eine zweite Methode, die bei einem übergebenem Text eine Häufigkeitstabelle der Vorkommen der Buchstaben a - z ausgibt. Für die einzelnen Häufigkeiten der Buchstaben kannst du jeweils auf die vorherige Methode zurückgreifen.

1.5. Stringmatching

Informiere dich für diese Aufgabe auf `http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html` über die Methoden `indexOf()` und `lastIndexOf()` der Klasse String.

a) Schreibe eine Methode welche das erste Vorkommen eines Musters in einem Text findet und die Fundstelle zurückgibt. Wird das Muster nicht gefunden, wird -1 zurückgegeben.

Beispiel:
`Text="DiesisteineFurzkissenfabrik!",Muster="Furz",Ausgabe: 14`

b) Schreibe eine main()-Methode, um die geschriebene Methode zu testen.

c) Modifiziere deine Methode so, dass nicht das erste, sondern das letzte Vorkommen gefunden wird.

*d) Versuche, die Aufgaben zu lösen, ohne `indexOf()` bzw. `lastIndexOf()` zu benutzen, indem du diese Methoden selbst implementierst. Dabei darf aber die Methode `charAt()` benutzt werden.

1.6. **Nimm-Spiel

Schreibe ein kleines Spiel: Zunächst soll die Anzahl der verwendeten Stäbchen eingegeben werden, dann nehmen der Spieler und der Computer abwechselnd 1 bis 3 Stäbchen weg (wobei der Computer immer die optimale Strategie verfolgt, schreibe dazu eine Methode `rechnerNimmt()`). Verloren hat derjenige, der das letzte Stäbchen wegnimmt. Der Spieler kann anschließend entscheiden, ob er nochmal spielen will.

Überlege dir zunächst den Programmablauf und skizziere ihn durch Kommentare in deinem Quellcode.

1.7. *Quersummen

Eine Quersumme einer Zahl ist die Summe der einzelnen Ziffern.

a) Es soll eine Methode erstellt werden, die die Quersumme einer übergebenen Zahl berechnet und zurückgibt. Es ist allerdings etwas trickreich, die einzelnen Ziffern zu bestimmten. Deswegen hier ein kleiner Tipp: Nehme eine beliebige Zahl und teile sie auf einem Blatt Papier modulo und diviso 10, also eine ganzzahlige Division mit Rest. Welche zwei Zahlen hast du hiermit errechnet? Wie kannst du das nun für einen Algorithmus nutzen?

b) Von einer Quersumme kann man wiederum eine Quersumme bilden. Eine besondere Quersumme ist die einstellige Quersumme. Hier wird solange von einer Quersumme wieder Quersumme gebildet, bis diese einstellig ist (Beispiel: 93: 9+3 = 12: 1+2 =3). Schreibe nun eine Methode, die eine Zahl übergeben bekommt und daraus die einstellige Quersumme berechnest. Hierfür kannst du (mehrmals) auf deine vorherige Methode zurückgreifen, bis du das gewünschte Ergebnis gefunden hast.