

## Übungsblatt 6

Ausgabe: 22.11.2000

**Hinweise:** Zur Lösung der Aufgaben dürfen nur Konzepte benutzt werden, die in der Vorlesung schon behandelt worden sind.

### Aufgabe 23 (Funktionen): 65 Punkte

Implementieren Sie in Java folgende Prozeduren/Funktionen (je 5 Punkte)! Achten Sie auf Randfälle und nicht korrekte Parameterübergaben! Überprüfen Sie aber zunächst für jede Funktion, ob sie überhaupt mit den (bisher bekannten) Konzepten in Java implementiert werden kann und wenn nicht, begründen sie, wieso nicht!

- (1) eine Funktion, die testet, ob eine als Parameter übergebene natürliche Zahl eine Fibonacci-Zahl ist oder nicht (Beispiel:  $f(8) == true$ ),
- (2) eine Prozedur, die die Werte zweier als Parameter übergebener `double`-Variablen vertauscht (Beispiel: `double a=2.0; double b=5.9; f(a, b);` Ergebnis: `a==5.9; b==2.0;`),
- (3) eine Funktion, die einen übergebenen `double`-Wert rundet und als `int`-Wert zurückliefert (Beispiel:  $f(-2.6) == 3$ ),
- (4) eine Funktion, die die nächst kleinere Primzahl einer als Parameter übergebenen natürlichen Zahl (größer als 2) liefert (Beispiel:  $f(11) == 7$ ),
- (5) eine Funktion, die als Parameter einen `int`-Wert übergeben bekommt und die überprüft, ob die Ziffer 7 in dem `int`-Wert vorkommt (Beispiel:  $f(-2578) == true$ ),
- (6) eine Funktion, die als ersten Parameter eine Funktion  $g: \text{char} \rightarrow \text{int}$  und als zweiten Parameter einen `char`-Wert *zeichen* übergeben bekommt und die als Ergebnis den Wert  $g(\text{zeichen})$  liefert (Beispiel: `public static char pos(char zeichen) {return zeichen - 'a' ;}` und  $f(\text{pos}, 'b') == 2$ ),
- (7) eine Funktion, die als Parameter einen `int`-Wert  $n$  übergeben bekommt und die als Ergebnis die Summe der Zahlen zwischen 0 und  $n$  zurückliefert; ist der Wert des übergebenen Parameters jedoch kleiner als 0, soll die Funktion den Wert `false` liefern (Beispiel:  $f(4) = 10$ ,  $f(-2) == false$ ),
- (8) eine Funktion, die als ersten Parameter einen `int`-Wert  $n$  und daraufhin  $n$  `float`-Parameter übergeben bekommt, deren Summe geliefert werden soll (Beispiel:  $f(3, 1.1f, 2.2f, 3.3f) == 6.6f$ ),
- (9) eine Funktion, die als ersten Parameter einen `float`-Wert  $x$  (zwischen 0 und 1000) und als zweiten Parameter einen positiven `int`-Wert  $n$  (zwischen 1 und 5) übergeben bekommt. Die Funktion soll den Wert  $x$  auf  $n$  Nachkommastellen runden (Beispiel:  $f(2.2576F, 3) == 2.258F$ ),
- (10) eine Funktion, die die Summe zweier Uhrzeiten als Ergebnis liefert; Uhrzeiten werden dabei als `float`-Werte realisiert, wobei die Vorkommastellen die Stunden und die

Nachkommastellen die Minuten darstellen (Beispiel:  $f(22.13f, 3.48f) == 2.01f$ )

- (11) eine Funktion, die die Summe und die Differenz zweier als Parameter übergebener `int`-Werte zurückliefert (Beispiel:  $f(4, 3) = (7, 1)$ )

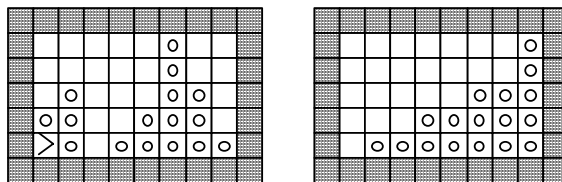
Schreiben Sie ein Programm, das einem Benutzer eine Auswahl zur Ausführung der implementierbaren Funktionen anbietet, anschließend jeweils passende Werte für die aktuellen Parameter einliest, die ausgewählte Funktion aufruft und ein Ergebnis auf dem Bildschirm ausgibt (10 Punkte). Achten Sie darauf, dass die Funktionen nur mit zulässigen Werten aufgerufen werden.

### Aufgabe 24 (Funktionen): 20 Punkte

Schauen Sie sich das Programm „Haus.java“ an, das Sie auf dem ARBI-Cluster unter `/home/is/fb10/dibo/java` bzw. auf der Homepage zum PK Java im WWW finden. Was leistet es? Überlegen Sie: Wo wäre der Einsatz von Prozeduren bzw. Funktionen sinnvoll? Schreiben Sie das Programm mit Hilfe von Prozeduren und Funktionen so um, dass es verständlicher und kürzer wird!

### Aufgabe 25 (Hamster-Modell): 15 Punkte

Der Hamster steht in der linken unteren Ecke (Blickrichtung Ost) eines durch Mauern abgeschlossenen ansonsten aber mauerlosen rechteckigen Raumes unbekannter Größe. Im Territorium befinden sich - wie in der linken Abbildung skizziert - Spalten („Körnertürme“) mit keinem, einem oder mehreren Körnern (auf jeder Kachel liegt jeweils maximal ein Korn). Der Hamster bekommt die Aufgabe, die Körnertürme so zu sortieren, dass die Körnertürme zum Schluss in aufsteigender Größe im Territorium angeordnet sind (siehe rechte Abbildung).



Entwickeln Sie ein Hamsterprogramm, das die Aufgabe löst!