10. Übung zur Vorlesung „Programmiersprache Java“, SS 2001

Prof. Dr. Rudolf Berrendorf, FH Bonn-Rhein-Sieg

Aufgabe 1)

Auf der WWW-Seite zur Vorlesung finden Sie eine Datei alarm.jar, die den Quellcode eines Java-Programms enthält.
Was wird bei der Ausführung des Programms ausgegeben?
Beschreiben Sie detailliert, welche Exceptions auftreten und wo diese abgefangen werden!

Aufgabe 2)

Komplexe Zahlen lassen sich statt in kartesischen Koordinaten (a, b) auch in Polarkoordinaten (r, phi) mit $r \geq 0$ und phi in [- pi, pi] darstellen. Für die Umrechnung von kartesischen Koordinaten in Polarkoordinaten gilt

$r = sqrt(a^2 + b^2)$
$phi = arctan(b/a)$ sofern $a > 0$,
$phi = pi - arctan(b/|a|)$ sofern $a < 0$ und $b \geq 0$,
$phi = - pi + arctan(b/a)$ sofern $a < 0$ und $b \leq 0$,
$phi = pi/2$, für $a = 0$, $b \geq 0$ und
$phi = -pi/2$ für $a = 0$, $b < 0$

Für die Umrechnung von Polarkoordinaten in kartesische Koordinaten gilt

$a = r * cos (phi)$
$b = r * sin (phi)$

Erweitern Sie die Klasse Komplex der letzten Übung um Methoden zur Umrechnung von Polarkoordinaten in kartesische Koordinaten und umgekehrt.

Aufgabe 3)

Lassen Sie in einer Schleife 10000000-mal die Rechnung $y = x^2$ ausführen (z.B. mit double x = 2.0). Benutzen Sie einmal die Methode pow (double a, double b) der Klasse StrictMath, zum anderen die Formel y = x*x.
Messen Sie in beiden Fällen, wie lange das Programm für die Schleife benötigt.
Führen Sie die Messung ein drittes Mal durch, wobei Sie nun int - Variablen statt double - Variablen benutzen.

Aufgabe 4)

Der Abstrakte Datentyp Binärbaum wurde im 1.Semester eingeführt als

$binbaum = <\{M, M^\Delta, B\}, Ops_{binbaum}>$

mit den Operationen $Ops_{binbaum}$:

$emptytree : \varnothing \rightarrow M^\Delta$	mit emptytree() = ε
$maketree: M^\Delta \times M \times M^\Delta \rightarrow M^\Delta$	mit maketree(l,a,r) = <l,a,r>
$root: (M^\Delta - \{\varepsilon\}) \rightarrow M$	mit root(<l,a,r>) = a
$lefttree: (M^\Delta - \{\varepsilon\}) \rightarrow M^\Delta$	mit lefttree(<l,a,r>) = l
$righttree: (M^\Delta - \{\varepsilon\}) \rightarrow M^\Delta$	mit righttree(<l,a,r>) = r
$isemptytree: M^\Delta \rightarrow B$	mit isemptytree(t) = wahr, falls t=ε und falsch sonst
$isequal: M^\Delta \times M^\Delta \rightarrow B$	mit isequal(t1,t2) = wahr, wenn t1 und t2 komponentenweise gleich sind

In binBaum.jar (zu finden auf der WWW-Seite zur Vorlesung) ist ein konkreter Datentyp als Java-Klasse Binaerbaum zu finden (inkl. einer Fehlerklasse).

a) Erläutern Sie detailliert, wie der Abstrakte Datentyp in den konkreten Datentyp umgesetzt wurde. Beachten Sie dabei die Rekursion in den Daten- und Operationsdefinitionen.
b) Ergänzen Sie die Java-Klasse Binaerbaum so, dass sie das Klonen von Objekten unterstützt. Beachten Sie dabei, dass ein Binärbaum in der vorliegenden Implementierung ein tiefes Objekt ist. Testen Sie das Klonen in einem Beispielprogramm. Gehen Sie dabei so vor, dass Sie zuerst nach dem Klonen eines vorhandenen Baums die beiden dann existierenden Bäume über die Methode isequal auf Gleichheit testen. Aufgrund des Klonens müsste dieser Vergleich true ergeben. Modifizieren Sie anschließend den Ursprungsbaum über die Methode setDatum (im Java-Programm zu finden) und testen die beiden Bäume wiederum auf Gleichheit. Diesmal müsste der Test auf Gleichheit false ergeben.
c) Ergänzen Sie die Klasse um eine Methode, die die Knoteninhalte (datum) des Baums in inorder- (bzw. postorder- bzw. preorder-) Reihenfolge ausgibt.
d) Implementieren Sie die Methode contains: $M^\Delta \times M \rightarrow B$ mit

	= falsch,	wenn t = <>
contains (t, a)	= true ,	wenn a = root(t)
	= contains (leftree(t), a) oder contains (righttree(t), a)	sonst.