Algorithmen und Programmierung III

WS 2002/2003

Übung 3

H. Schweppe

Ausgabe: 31.10.2002
Abgabe: 12.11.2002 , 16.00
30 Punkte.
Verspätete Abgabe führt zu Punktabzug.

Aufgabe 3.1 (6 P)
Erweitern Sie die in der Vorlesung angegebene Haskell-Funktion `checkBrack` um eckige und geschweifte Klammern. Damit ist ein korrekt geklammerter Ausdruck folgendermaßen definiert:
Eine beliebige Zeichenkette S , *in der keine Klammern vorkommen, ist ein korrekt geklammerter Ausdruck.*
Sind S1, S2 *und* S3 *korrekt geklammert Ausdrücke, dann auch* S1(S2)S3, S1{S2}S3 *und* S1[S2]S3 .

Aufgabe 3.2 (6 P)
Geben Sie die modellierende Spezifikation für die Prioritätsschlange an. Gehen Sie davon aus, dass es für den Datentyp fünf Funktionen gibt: `isEmpty ()`, `enqueueP ()`, `dequeueP ()`, `length ()`, `createP ()`.
In Prioritätsschlangen hat jedes Element eine numerische Priorität. `dequeueP ()` liefert das in der Schlange befindliche Element mit höchster Priorität.
Hinweis: Folgen sind als Modell gut geeignet. Sie können aber auch ein anderes Modell wählen.

Aufgabe 3.3 (6 P)

a) (3) Es soll ein Java-Interface für Polynome mit reellen Koeffizienten $p(x) = \sum a_i x^i$, $i=0,...,n$ spezifiziert werden.
Operationen sollen mindestens sein: Addition und Multiplikation von Polynomen, Grad eines Polynoms (= größter Exponent), Wert für ein Argument x =const, sowie ein Prädikat 'Nullpolynom'. (Ein Polynom ist das Nullpolynom, wenn alle Koeffizienten null sind. Als Modell bieten sich die aus der Mathematik bekannten Polynome an. Die Spezifikation kann halbformal oder umgangssprachlich sein, Hauptsache sie ist präzise.

b)(3) Welche Laufzeit hat die "naive" Auswertung eines Polynoms vom Grad n für x=const? Die naive Methode Methode wertet jeden Summanden einzeln aus (z.B. x^k *

a_k) und summiert sie auf. Welche Laufzeit hat die Auswertung mit dem Horner-Schema? Wenn Sie nicht mehr wissen was das ist: Schauen Sie einfach im Netz nach: http://google.com o.ä.

Aufgabe 3.4 (5 P)
Gesucht ist eine Klasse `class MySingleton` , für die sichergestellt ist, dass es nur ein Objekt dieser Klasse gibt. Ein zweiter Versuch eines Klienten, ein `MySingleton` -Objekt zu erzeugen, soll scheitern.
Hier eine Klientenklasse, mit der getestet werden kann:

```
class ClientSingleton  {

 private MySingleton aSingleton, another;

 public static void main (String[] args){
    ClientSingleton c = new ClientSingleton ();
    c.aSingleton      = MySingleton.newMySingleton ();
    if (c.aSingleton != null) System.out.println("got first
Object");
    c.another         = MySingleton.newMySingleton ();
    if (c.another !=null ) System.out.println("should be only
one");
        else System.out.println("Is null, only one Object
allowed");
 }
}
```

Dieses sogenannte Singleton-Muster, eines der einfachsten überhaupt, kann auf verschiedene Art implementiert werden. Achten Sie auf die Modifikatoren. (Was war das nochmal? -> Alp2-Skript!) Die Klasse `MySingleton` sollte `final` sein. Warum?
Bemerkung: diese Aufgabe zuletzt bearbeiten. Es handelt sich neben den Alp2-Anteilen um Zusatzstoff, den man sich aber leicht selbst erarbeiten kann.

Aufgabe 3.5 (5 P)

a) (2) Zeigen Sie: $\sum i^2 = O(n^3)$, i=1...n
b) (3) Wie sieht der Laufzeitstapel einer imperativen Sprache in folgender Situation aus: Eine Funktion (Methode) `f` ruft mit `p(23)` die Methode

```
int p(int arg){
  if (arg >1) return arg*p(arg-1);
     else return 1;}
```

vor bzw. nach dem ersten rekursiven Aufruf von p auf.

Aufgabe 3.6 (2 P)
a) (1) Angenommen es existiert eine exakte Schranke für die Anzahl der Operationen $T_A(n)$ des Algorithmus A bei Problemgröße n im mittleren Fall, d.h. $T_A(n) = \Theta(g(n))$ Liefert die exakte Schranke den genauen Wert für die Anzahl der Operationen $T_A(n)$ bei Problemgröße n? (Begründung)

b) (1) Geben Sie fünf verschiedene Anwendungen von Stapeln an.