Übungen zu Softwareentwicklung 1 **WS 2000/2001**
Übung 2

Name: _____ Tutor: _____

Matrikelnummer: _____ Punkte: _____

Gruppe: _____ **Abzugeben bis: 25.10.00, 12:00**

Übungsleiter: _____ Bearbeitungsdauer: _____

Aufgabe 1 (6 Punkte): Grammatik I

Betrachten Sie folgenden (formalen) Satz, in dem jedes Symbol genau ein Zeichen lang ist:

BACBCACBCBA

Aufgabe:
- Finden Sie mindestens drei verschiedene Grammatiken, die — ausgehend von einem Startsymbol S — diesen Satz erzeugen können. Zeichnen Sie für jede einen Syntaxbaum.
- Worin unterscheiden sich diese Grammatiken?
- Was ist die kürzeste, was die längste Erzeugung (in Anzahl der Schritte vom Startsymbol bis zum Satz)?
- Zeichnen Sie für zwei der Grammatiken je ein Syntaxdiagramm.

Aufgabe 2 (6 Punkte): Testen von Algorithmen

Vor dem Testen eines Algorithmus bzw. des realisierenden Programmes überlegen Sie sich zunächst, WAS Sie testen wollen (also welches Verhalten und welche (Sonder-)Fälle). Dann überlegen Sie sich, WIE Sie diesen Test durchführen, und zwar durch Festlegung geeigneter Eingabewerte. Ihre Überlegungen halten Sie in einem **Testplan** fest: WAS, WIE, erwartetes Resultat.

Bei der Festlegung des Testplanes sollen auch die "Grenzen" eines Algorithmus "abgesteckt" werden. Die Beschreibung einer Schnittstelle enthält neben der Beschreibung der Ein- und Ausgabeparameter insbesondere die Bedingungen und Einschränkungen, die der betreffende Algorithmus bzw. das realisierende Programm besitzen. Oft werden Sie feststellen, dass die Problembeschreibung konkretisiert werden muss.

Aufgabe: Für Ihren GGT-Algorithmus aus Übung 1 / Aufgabe 1:
- Beschreiben Sie dessen "Grenzen".
- Erstellen Sie einen Testplan.
- Führen Sie einen Schreibtischtest für zwei (nicht triviale) Testfälle durch.

Anmerkung: Legen Sie den GGT-Algorithmus Ihrer Ausarbeitung bei. Wenn Sie keine eigene GGT-Lösung (mehr) haben, verwenden Sie die Beispiellösung Ihrer Übungsgruppe.

Aufgabe 3 (6 Punkte): Grammatik II

- Suchen Sie sich ein strukturierbares Objekt Ihrer Wahl: z.B. Aktenkoffer, Geldbörse, ein Gebäude, Fahrrad, etc.
- Beschreiben Sie seine Struktur in Worten. Z.B.: Die Geldbörse hat drei Bereiche; ein Bereich ist für Festgeld, dort verwahre ich Münzen und Jetons. Münzen sind 1, 5, 10 oder 20 Schilling…

- Geben Sie eine Grammatik in EBNF an, die geeignete Symbolfolgen zur Beschreibung dieses strukturierten Inhaltes produziert. (Natürlich sollten auch Iterationen und Optionen vorkommen.)
- Vergleichen Sie die umgangssprachliche Formulierung mit der in EBNF.
- Zeichnen Sie für einen aktuellen Inhalt einen (nicht leeren) Syntaxbaum. Gab es ein Problem beim Zeichnen?

Anmerkung: Achten Sie auf gute Strukturierung der Syntaxregeln. (Schreiben Sie vor allem nicht alles in eine einzige Regel.)

Aufgabe 4 (6 Punkte): JDK-Benutzung

Um mit dem JDK vertraut zu werden, versuchen Sie, das folgende Programm `DoTheCoding` einzutippen, zu übersetzen und auszuführen. Gehen Sie dabei folgendermaßen vor:

- Starten Sie eine Java-Programmierumgebung Ihrer Wahl oder einen gewöhnlichen Texteditor und erstellen Sie ein neues Dokument.
- Geben Sie das Programm mit Hilfe des Editors ein. Bedenken Sie dabei, dass der Zweck dieser Übung ist, mit dem System vertraut zu werden. Probieren Sie daher beim Eintippen schon die verschiedenen Funktionen Ihres Editors aus.
- Speichern Sie das Programm als `DoTheCoding.java` auf Ihrer Festplatte ab.
- Versuchen Sie, das Programm zu übersetzen (`javac`) und anschließend zu starten (`java`). Falls der Übersetzer Fehler entdeckt, beheben Sie diese, übersetzen Sie das Programm erneut und starten Sie es anschließend. *Hinweis*: Sie können die Programmausführung durch Eingabe eines Sterns ("*") beenden.
- Testen Sie das Programm ausführlich und drucken Sie das Programmlisting sowie das Testprotokoll aus.
- Beschreiben Sie die Funktion des Programms möglichst genau. Versuchen Sie auch Sonderfälle zu testen. Wie verhält sich das Programm in “Extremsituationen”?

```
class DoTheCoding extends Basic {

  public static void main (String[] arg) {
    String theLine;
    theLine = readLine();
    while (IODone && (theLine.charAt(0) != '*') {
      writeLn(doSomething(theLine));
      theLine = readLine();
    }

  static String doSomething (String line) {
    char[] newLine;
    int i = 0, j = 0;
    newLine = line.toCharArray();
    while (i < newLine.length) {
      if ((newLine[i]!=' ')||(j==0)||(newLine[j-1]!=' ')) {
        newLine[j] = newLine[i];
        j++;
      }
      i++
    }
    return line.copyValueOf(newLine, 0, j);
  }
}
```