

Zuname _____ Vorname _____ Matr. Nr. _____

Übungsgruppe Punkte _____ korr. _____

- 1 (Wöß) Do 10¹⁵-11⁴⁵
- 2 (Wöß) Do 12⁰⁰-13³⁰
- 3 (Rammerstorfer) Do 17¹⁵-18⁴⁵

Letzter Abgabetermin
Donnerstag, 16.10.2003, 8¹⁵ Uhr

Grammatiken

1. Grundbegriffe (2+3+3+6 Punkte)

Die Grammatik der Sprache *MicroJava* finden Sie im VO-Skriptum auf Seite 9, Kapitel 5.2.

- a) Geben Sie den Satz mit der minimalen Anzahl von Terminalsymbolen an, den man mit dieser Grammatik erzeugen kann und der mindestens einen Stern ('*', s. NTS `MulOp`) enthält.
- b) Betrachten Sie die Nonterminalsymbole *Expr*, *BlockStat* und *Statement*. Geben Sie für jedes dieser Nonterminalsymbole an, ob es links-, zentral- oder rechtsrekursiv und direkt oder indirekt rekursiv ist.
- c) Zeichnen Sie den Syntaxbaum für folgenden Satz:

```
class Test class Circle { int r; } { Circle create() { return new Circle; } }
```

Gibt es mehrere Syntaxbäume für diesen Satz?

- d) Bestimmen Sie alle terminalen Anfänge und Nachfolger für die Regeln *Type*, *Term*, *CondTerm* und *Designator*.

2. Konstruktion einer Grammatik (5 Punkte)

Geben Sie eine Grammatik (in EBNF) für die Bezeichner einer fiktiven Programmiersprache gemäß den folgenden Bedingungen an:

Die Bezeichner dürfen aus Buchstaben (Terminalklasse *b*), Ziffern (Terminalklasse *z*) und dem Unterstreichungszeichen ("_") bestehen.

Ein Bezeichner muss mit einem Buchstaben beginnen und darf anschließend beliebig viele Buchstaben und Unterstreichungszeichen enthalten, wobei Unterstreichungszeichen immer nur zu zweit (nicht mehr und nicht weniger) in Folge auftreten dürfen. Auch Ziffernfolgen sind erlaubt, allerdings müssen sie von Unterstreichungszeichen umgeben sein oder sie stehen am Schluss des Bezeichners.

Bsp.

gültige Bezeichner: ungültige Bezeichner

<i>b</i>	<i>bz</i>
<i>b__</i>	<i>b_zzzz</i>
<i>b_z</i>	<i>__zzz</i>
<i>b_z__b</i>	<i>zzz__</i>
<i>bb__b__zzz__z__b</i>	<i>bz__b</i>

3. Beseitigung von Linksrekursionen (5 Punkte)

Gegeben sei folgender Auszug aus einer fiktiven Grammatik, der Methodenaufrufe beschreibt:

```
MethodCall = Designator "." ident "(" [ ParamList ] ")" .  
ParamList  = { Designator "," } Designator .  
Designator = ident | MethodCall .
```

Beseitigen Sie alle Linkrekursivitäten und geben Sie die transformierte Grammatik in EBNF an. *ident* ist eine Terminalklasse und definiert einen Namen (Buchstabe gefolgt von Ziffern und Buchstaben).