

Zuname _____ Vorname _____ Matr. Nr. _____

Übungsgruppe

- 1 (Wöß) Do 10¹⁵-11⁴⁵
- 2 (Wöß) Do 12⁰⁰-13³⁰
- 3 (Rammerstorfer) Do 13⁴⁵-15¹⁵

Punkte _____ korr. _____

Letzter Abgabetermin
Donnerstag, 10.10.2002, 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 eine Expression (*Expr*) enthält.
- b) Betrachten Sie die Nonterminalsymbole *ActPars*, *Factor* 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 Circle int r; { int resize(int nr) { if (nr > 0) r = nr; return r; } }
```

Gibt es mehrere Syntaxbäume für diesen Satz?

- d) Bestimmen Sie alle terminalen Anfänge und Nachfolger für die Regeln *FormPars*, *Expr*, *CondFact* und *Designator*.

2. Konstruktion einer Grammatik (4 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 oder einem Unterstreichungszeichen beginnen und darf anschließend beliebig viele Buchstaben und Unterstreichungszeichen enthalten, wobei direkt hinter einem Unterstreichungszeichen kein weiteres Unterstreichungszeichen stehen darf. Auch Ziffernfolgen sind erlaubt, allerdings müssen sie von Unterstreichungszeichen umgeben sein.

Bsp.

gültige Bezeichner:

b
b_
_b_z_b
bb_b_zzz_z_b

ungültige Bezeichner

bz
b_zzzz
zzz
bz_b

3. Beseitigung von Linksrekursionen (6 Punkte)

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

```
MethodCall = Designator "." Ident "(" ParamList ")" .  
ParamList = ParamList "," Designator | Designator .  
Designator = Ident | MethodCall .  
Ident      = letter | Ident ( digit | letter ) .
```

Beseitigen Sie alle Linkrekursivitäten und geben Sie die transformierte Grammatik in EBNF an. *letter* und *digit* sind Terminalklassen und definieren die Mengen der Buchstaben und der Ziffern.