

Zuname _____ Vorname _____ Matr. Nr. _____

Übungsgruppe

- 1 (Kotzmann) Do 10¹⁵-1145
- 2 (Wimmer) Do 12⁰⁰-1330
- 3 (Wöß) Do 10¹⁵-1145

Punkte _____ korr. _____

Letzter Abgabetermin
Donnerstag, 14.10.2004, 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 das Terminalsymbol "if" enthält.
- b) Betrachten Sie die Nonterminalsymbole *Expr*, *Block* 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:

```
program Test int x, yi { int area() { return x * yi } }
```

Gibt es mehrere Syntaxbäume für diesen Satz?

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

2. Konstruktion einer Grammatik (5 Punkte)

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

Eine Zahl ist entweder eine Dezimalzahl oder eine Hexadezimalzahl.

Dezimalzahlen bestehen aus Ziffern (Terminalklasse *z*). Wenn eine Dezimalzahl aus mehr als drei Ziffern besteht, muss nach jeder dritten Ziffer (gerechnet von rechts) ein Punkt (".") stehen (Tausender-Punkte).

Hexadezimalzahlen bestehen aus Ziffern (Terminalklasse *z*) oder Buchstaben von "A" bis "F" (Terminalklasse *b*). Sie müssen mit einer Ziffer beginnen und mit einem "H" enden.

Bsp.

gültige Zahlen:	ungültige Zahlen
0	1234
123	567.89
45.678	8B
8H	12.345H
6B53H	B83H

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).