

Zuname _____ Vorname _____ Matr.-Nr. _____

Übungsgruppe Punkte _____ korr. _____

- | | | |
|---------------------------------------|---------------------------------------|--|
| <input type="checkbox"/> 1 (Kotzmann) | Do 10 ¹⁵ -11 ⁴⁵ | |
| <input type="checkbox"/> 2 (Kotzmann) | Do 13 ⁴⁵ -15 ¹⁵ | Letzter Abgabetermin |
| <input type="checkbox"/> 3 (Wimmer) | Do 10 ¹⁵ -11 ⁴⁵ | Mittwoch, 12.10.2005, 20 ¹⁵ Uhr |

Grammatiken

1. Grundbegriffe

(2+3+3+6 Punkte)

Die Grammatik der Sprache *MicroJava* finden Sie im VO-Skriptum im Kapitel 5.2 auf Seite 8.

- Geben Sie den Satz mit der minimalen Anzahl von Terminalsymbolen an, den man mit dieser Grammatik erzeugen kann und der mindestens das Terminalsymbol "*break*" enthält. Aus wie vielen Terminalsymbolen besteht er?
- Betrachten Sie die Nonterminalsymbole *Factor*, *MethodDecl* und *Statement*. Geben Sie für jedes dieser Nonterminalsymbole an, ob es links-, zentral- oder rechtsrekursiv ist, und ob es direkt oder indirekt rekursiv ist.
- Zeichnen Sie den Syntaxbaum für folgenden Satz:

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

 Gibt es mehrere Syntaxbäume für diesen Satz?
- Bestimmen Sie alle terminalen Anfänge und Nachfolger für die Regeln *Expr*, *Statement*, *Term* und *Relop*.

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 müssen mit "*0x*" beginnen und bestehen aus Ziffern (Terminalklasse *z*) oder Buchstaben von "*A*" bis "*F*" (Terminalklasse *b*).

Beispiele für gültige Zahlen: 0, 123, 45.678, 0x8, 0x6B53

Beispiele für ungültige Zahlen: 1234, 567.8, 8B, 0x12.345, 0x6G53

3. Beseitigung von Linksrekursionen

(5 Punkte)

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

```
ArrayAccess = Array "[" Index "]" .  
Index       = { number "," } number .  
Array       = ident | ArrayAccess .
```

Beseitigen Sie alle Linksrekursionen und geben Sie die transformierte Grammatik in EBNF an. *ident* und *number* sind Terminalklassen, die einen Namen (Buchstabe gefolgt von Ziffern und Buchstaben) bzw. eine Zahl (bestehend aus Ziffern) definieren.