

Zuname _____ Matr. Nr. _____

Übungsgruppe: Punkte _____ korr. _____

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

Letzter Abgabetermin:
Donnerstag, 13.01.2005, 8¹⁵ Uhr

Codeerzeugung – Teil 2 (24 Punkte)

Vervollständigen Sie nun Ihren Compiler, indem Sie auch die fehlenden Teile der Codeerzeugung gemäß der in den Unterlagen ausgegebenen Spezifikation der *MicroJava*-VM hinzufügen.

Um nun Sprünge innerhalb des Codes realisieren zu können, benötigen wir zunächst sogenannte *Labels*, also Sprungmarken, mit denen die Ziele der Sprünge greifbar gemacht werden. Führen Sie dazu die Klasse `Label` im Package `ssw.mj.codegen` ein:

```
class Label {
    boolean defined;           // is destination of jump already defined?
    int adr;                   // jump destination address (defined) or
                              // head of unresolved forward jumps (!defined)
    void put();                // generates code for a jump to this label
    void here();               // defines this label to be at the current pc position
    void setTo (Label dest);   // defines this label to be at the position of dest
}
```

Außerdem muss die Klasse *Item* wie folgt erweitert werden:

```
class Item {
    static final int // item kinds
        Con = 0,    // constant
        Local = 1, // local variable
        Static = 2, // global variable
        Stack = 3, // expression on stack
        Fld = 4,   // field of inner class
        Elem = 5,  // array element
        Meth = 6,  // method call
        Cond = 7;  // condition

    int kind;           // Con, Local, Static, Stack, Fld, Elem, Meth, Cond
    Struct type;        // item type
    int val;            // Con:Wert
    int adr;           // Local,Static,Fld,Meth:Adresse; Cond: Operator
    Obj obj;           // Meth: method object from symboltable
    Label tLabel, fLabel; // Cond: True Jumps, False Jumps
}
```

Die Angabedatei `UB-UE6-Angabe.zip` enthält bereits die vollständige Klasse `Label` sowie den Quellcode der virtuellen Maschine (Klasse `Run`).

Testen Sie Ihren Generator ausführlich. Lassen Sie Ihre Testprogramme auch tatsächlich auf der MJ-VM (`java ssw.mj.Run <obj-Datei>`) laufen.

Auf der Übungsseite finden Sie ein Testprogramm (`TestProgram.mj`), das Ihr Compiler korrekt übersetzen muss. Läuft das Programm korrekt ab, so wird die Zeichenfolge „1234“ ausgegeben