

Zuname _____ Vorname _____ Matr.-Nr. _____

Übungsgruppe

Punkte _____ korr. _____

- 1 (Löberbauer) Do 10¹⁵-11⁴⁵
- 2 (Löberbauer) Do 13⁴⁵-15¹⁵
- 3 (Wimmer) Do 10¹⁵-11⁴⁵

Letzter Abgabetermin

Mittwoch, 20.12.2006, 18⁰⁰ Uhr

Codeerzeugung – Teil 1

(24 Punkte)

Erweitern Sie Ihren Compiler um die Codeerzeugung gemäß der in den Unterlagen ausgegebenen Spezifikation der *MicroJava*-VM (siehe VO-Skriptum Kapitel 6, Seiten 13-17). Die dafür nötigen Klassen *Code* und *Item* befinden sich im Package *ssw.mj.codegen*.

Die Klasse *Item* ist bereits vollständig implementiert. In der Klasse *Code* sind Methoden zur Ausgabe häufig verwendeter Code-Muster (Laden von Werten, Zuweisungen, ...) zu implementieren. Die Klasse *Label* wird erst für die nächste Übung benötigt. Implementieren Sie auch in der Klasse *Struct* noch die Methoden, um die in *MicroJava* benötigten Arten der Typ-Kompatibilität zu prüfen (siehe VO-Skriptum, Kapitel 5.3, Seite 9).

Zusätzlich befindet sich im Package auch ein Decoder (*Decoder.java*), der Objektcode der *MicroJava*-VM in textueller Form ausgibt. Sie können ihn als Hilfsmittel verwenden, um Ihren erzeugten Objektcode zu überprüfen.

In diesem ersten Teil der Codeerzeugung sollen Sie *nur* die Teile implementieren, die im VO-Skriptum bis einschließlich Folie 6.54 (Zuweisungen) beschrieben sind, d.h. Sie brauchen noch *keinen Code für Sprünge und Methoden bzw. Methodenaufrufe* erzeugen. Daher sollen Sie bei dieser Übung in den folgenden Parsermethoden die für die korrekte Codeerzeugung notwendigen Änderungen noch nicht implementieren:

- *Program, MethodDecl, Block*.
- *ActPars*: nur *Code.load()* für jeden Parameter aufrufen.
- *Condition, CondTerm*
- *CondFact*: Nur *Code.load()* für beide Operanden aufrufen.
- Bei *Factor* die Alternative für Methodenaufrufe: nur die Art des Items auf *Stack* setzen.
- *Relop*

Außerdem sollen Sie auch die Parsermethoden für Anweisungen, die Sprünge oder Methodenaufrufe benötigen, noch unverändert lassen. Das betrifft die folgenden Alternativen in der Methode *Statement*:

- Bei *Designator (case ident)* die Alternative für Methodenaufrufe (*case lpar*)
- *if*-Anweisung (*case if_*)
- *while*-Anweisung (*case while_*)
- *break*-Anweisung (*case break_*)
- *return*-Anweisung (*case return_*): Nur *Code.load()* für den Rückgabewert aufrufen.

Vergessen Sie nicht, alle Kontextbedingungen, die Sie nun prüfen können, auch tatsächlich zu prüfen und entsprechende Fehlermeldungen auszugeben.

Abgabe

Die Abgabe der Übungen 2 – 6 muss auf Papier und elektronisch erfolgen. Geben Sie folgende Dateien ab:

- Ausgedruckt auf Papier: *Parser.java*, *Code.java*, *Tab.java*, *Struct.java*.
- Elektronisch als ZIP-Datei: **Alle** Quellcode-Dateien, die zum **Ausführen** des Compilers benötigt werden (Packages *ssw.mj*, *ssw.mj.codegen* und *ssw.mj.symtab*), also auch alle Klassen der Abgabe. Auf die Datei *messages.properties* nicht vergessen. Die Verzeichnis-Struktur muss in der ZIP-Datei erhalten bleiben.
- **Nicht abzugeben:** JUnit-Testfälle, *.class*-Dateien, Projekt-Dateien von IDEs.