Übungen zu Übersetzerbau WS 2001/2002
Übung 5

Zuname _____ Matr. Nr. _____

Übungsgruppe
- ❑ 1 (Wöß) Do 10^{15}-11^{45}
- ❑ 2 (Wöß) Do 12^{45}-14^{15}
- ❑ 3 (Rammerstorfer) Do 14^{30}-16^{15}

Punkte _____ korr. _____

Letzter Abgabetermin
Donnerstag, 6.12.2001, 8^{15} Uhr

Symbolliste (15 Punkte)

Erweitern Sie Ihren Parser um eine Symbolliste *Tab*. Legen Sie dafür für alle folgenden Klassen neue Dateien im Package *MJ.SymTab* an (Angabe ohne Access Modifier).

```
class Obj {
  static final int Con, Var, Type, Meth, Fld, Prog;

  int kind;                  // Con, Var, Typ, Fld, Meth, Prog
  String name;
  Struct type;
  Obj next;                  // next local object in this scope
  int adr;                   // Con: value ; Meth, Var, Fld: offset
  int level;                 // Var: declaration level ; Meth: # of parameters
  Obj locals;                // Meth: reference to list of local variables
}

class Struct {
  static final int None, Int, Char, Arr, Class;

  int kind;                  // None, Int , Char, Class, Arr
  Struct elemType;           // Arr: type of array elements
  int n;                     // Class: # of fields
  Obj fields;                // Class: reference to list of fields
}

class Scope {
  Scope outer;               // reference to enclosing scope
  Obj locals;                // symbol table of this scope
  int nVars;                 // # of  variables in this scope
}

class Tab {
  static final Struct noType, intType, charType, nullType;  // predefined types
  static Obj noObj, chrObj, ordObj, lenObj;    // predefined objects

  static Scope topScope;    // current scope
  static int level;         // nesting level of current scope

  static void init ();
  static void openScope ();
  static void closeScope ();
  static Obj insert (int kind, String name, Struct type);
  static Obj find (String name);
  static Obj findField (String name, Struct type);
}
```

Die Methode *init* initialisiert die Symbolliste und trägt alle vordeklarierten Namen (Funktionen, Typen und Konstanten) von MicroJava ein.

Die Methoden *openScope* und *closeScope* legen einen neuen *topScope* an bzw. entfernen den aktuellen *topScope*.

Die Methode *insert* erzeugt ein *Obj*, trägt seine Attribute ein und fügt es im aktuellen Gültigkeitsbereich in die Symbolliste ein. Wenn dort bereits ein Eintrag mit dem gleichen Namen vorhanden ist, soll ein semantischer Fehler ausgegeben werden.

Die Methoden *find* und *findField* werden für die Erzeugung der Symbolliste noch nicht benötigt. Sie dienen dazu, später auf die Symbollisteneinträge zugreifen zu können.

find sucht nach einem Namen beginnend im aktuellen bis zum äußersten Gültigkeitsbereich.

findField sucht nach einem Namen in einer inneren Klasse, deren *Struct* in der Schnittstelle mitgegeben wird.

Test (5 Punkte)

Testen Sie Ihren neuen Parser ausführlich. Erweitern Sie dafür Ihre Symbolliste um Methoden, die ein Objekt (*Obj*), einen Typ (*Struct*), sowie die ganze Symbolliste am Bildschirm ausgeben. Achten Sie darauf, dass Sie alle Attribute ausgeben, z.B. auch die Adresse einer Variablen.

Geben Sie am Ende der Syntaxanalyse mit Hilfe obiger Methoden die gesamte Symbolinformation (inklusive aller vordeklarierten Objekte) für das übersetzte Programm aus.