

KIV/FJP

Samostatná práce

Jiří Sedmihradský, A02323

[sedmihradskyj@centrum.cz](mailto:sedmihradskyj@centrum.cz)

Datum odevzdání : 8.ledna 2007

## Formulace úlohy:

Vytvořte kompilátor a interpret jazyka PL/0 v programovacím jazyce C# a prostředí .NET.

## Popis řešení:

Program byl realizován v prostředí .NET 2.0 a toto prostředí je vyžadováno pro činnost programu. Program je realizován jako konzolová aplikace, je-li spouštěn s parametrem je tento parametr brán jako soubor obsahující zdrojový kód určený k zpracování. Při spuštění bez parametrů je nutno jméno (cestu) souboru zadat. Proběhne-li překlad úspěšně je generován na obrazovku a do souboru přeložený kód a automaticky následuje interpretace generovaného kódu. Je-li v programu nalezena chyba je vypsán její kód a slovní popis. Při realizaci jsem vycházel z již existující verze kompilátoru a překladače jazyka PL/0 v jazyce Java[1.], z něhož jsem převzal i objektovou dekompozici.

## Popis programu:

Vzhledem k tomu že celý program je členěn do 19 tříd, budou zde popsány pouze ty nejdůležitější (přiložený obrázek obsahuje diagram všech tříd). U všech tříd jsem se snažil o okomentování podstatných metod a proměnných, proto zde uvedený popis berte pouze jako orientační.

Program začíná voláním metody *main* v třídě *Mainclass* (soubor *Program.cs*). Zde se získá jméno souboru s vstupními daty a spustí se kompilace prostřednictvím *PL0Compiler* a v případě úspěšného zkompilování je spuštěna interpretace *PL0Interpreter*.

*PL0Compiler* : Třída zajišťující kompilaci zdrojového textu programu. Obhospodařuje objekty provádějící lexikální analýzu, kompilaci bloku a správu chyb odhalených během překladu.

*LexicalA* : Třída realizující lexikální analýzu zpracovávaného souboru. Třída načítá ze vstupního souboru lexémy – tj. základní prvky jazyka a předává je dále. Obsahuje vnořenou třídu *SourceReader* která slouží k čtení zpracovávaného souboru.

*CodeGen* : Uchovává generované instrukce v tabulce. Umožňuje generované instrukce vypsát do souboru a na obrazovku.

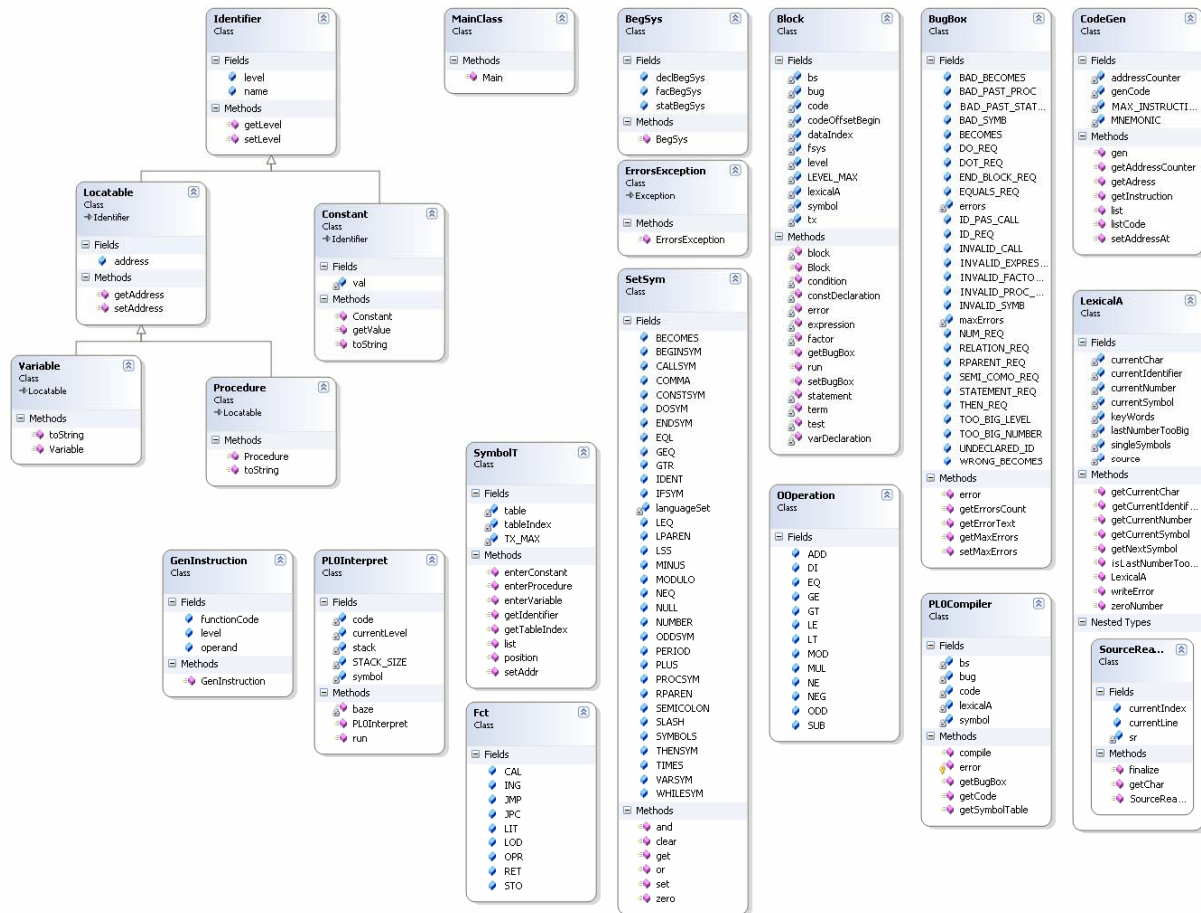
*BugBox* : Spravuje chyby generované při překladu.

*Block* : Provádí samotnou kompilaci zdrojového textu – tj. syntaktickou analýzu, využívá výše popsaných objektů. Kompilace začíná u nejvyššího bloku a postupuje rekurzivně přes bloky vnořené. Přijímá jednotlivé symboly od lexikálního analyzátoru a zpracovává je. Zpracované symboly jsou ukládány do tabulky symbolů představované třídou *SymbolT*. Je-li identifikována syntaktická chyba je zpracována s využitím objektu třídy *BugBox*.

*PL0Interpreter* : Realizuje interpretaci přeloženého zdrojového textu. Při interpretaci jsou jednotlivé instrukce čteny z tabulky vygenerovaných instrukcí (třída *CodeGen*) kterou vytvoří kompilátor.

*SymbolT*: Spravuje tabulku obsahující symboly nalezené při překladu – tj. konstanty, proměnné a procedury.

## Diagram tříd:



## Zdroje:

- [1]: Michal Beran, Překladač a interpret jazyka pl/0 v jazyce Java.
- [2]: Ostatní implementace překladačů jazyka PL/0, viz. stránky předmětu KIV/FJP.
- [3]: Materiály k předmětu KIV/FJP.