Univerzita Pardubice
Fakulta Elektrotechniky a Informatiky

Interpreter jazyka Wren

Klečanský Pavel

Semestrální práce ITEJA
2021

OBSAH

ÚVOD

Cílem semestrální práce je vytvořit zjednodušení interpreter námi vybraného jazyka. V této semestrální práci byla provedena zjednodušená implementace jazyka Wren pomocí programovacího jazyka C#.

1 WREN – POPIS JAZYKA

Zjednodušení jazyk Wren podporuje globální i lokální proměnné. Proměnné jsou netypované a podporují dva datové typy, a to řetězec a číslice. Dále má jazyk podporu cyklů a podmínek. Wren také podporuje aritmetické výrazy. Dále je v jazyce také možné volat knihovní funkce které jsou rozdělené na dvě kategorie, a to System pomocí které je možné psát do standardního výstupu a Turtle které umožnuje používat želví grafiku.

2 IMPLEMENTACE JAZYKA

Interpreter jazyka je napsán v programovacím jazyce C# a postaven nad .Net frameworkem 4.7.2. Grafická část interpreteru je vytvořená pomocí Winforms.

2.1 Gramatika

Gramatiky jazyka je vytvořená za pomocí EBNF.

```
program = block$;

block = {statement};

statement = ifStatement | whileStatement | callStatement | var |
reassignment;

var = "var" identifier "=" (expression|string);

reassignment = identifier "=" (expression|string);

ifStatement = "if" "(" expression ")"  "{" block "}" ["else" "{" block
"}"];

whileStatement = "while" "(" expression ")"  "{" block "}";

callStatement = ("System." | "Turtle.") identifier "(" [paramList] ")";

paramList =  param {"," param};

param = (expression|string);

expression = equality;

equality = comparison {("!=" | "==") comparison};

comparison = term {( ">=" | ">" | "<=" | "<") term};

term = factor {("+" | "-") factor};

factor = unary {("*"|"/"|"%") unary};

unary = ["-" | "+"] unary | primary;

primary = identifier | number | "(" expression ")";

identifier = /[A-Za-z\_]+/;

number = /[+-]?([0-9]+([.][0-9]*)?|[.][0-9]+)/;

string = /"[^"]*"/;
```

2.2 Tvorba interpreteru

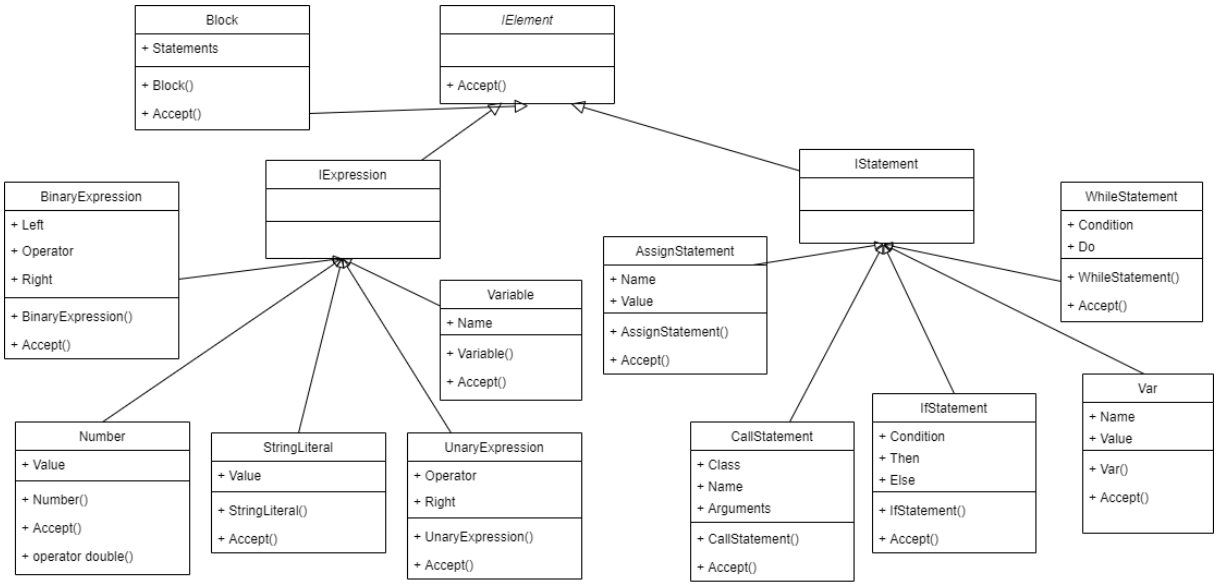
Interpreter je rozdělen na tři části a to lexer, parser, interpreter.

2.2.1 Lexer

Lexer vezme zdrojový kód jazyka Wren a vrátí ho ve formě pole tokenů. Implementaci lexeru v semestrální práci je možné najít ve složce Lexer pod názvem Lexer.cs. Ve složce Lexer najdete také třídu Token, pomocí které se vytvoří jednotlivé tokeny ve vráceném poli. Třída Token obsahuje dvě složky, a to její hodnotu a typ tokenu.

2.2.2 Parser

Parser zpracovává pole tokenů vrácených z lexeru a vytváří nový objekt Block který v sobě obsahuje list všech výrazů. Implementaci parseru je možné najít ve složce Parser pod názvem Parser.cs. V složce Parser najdete rozhraní IElement které implementují všechny elementy parseru. Dále jsou také ve složce rozhraní IExpression, které implementují všechny výrazy, a rozhraní IStatement, které implementují všechny příkazy. Na obrázku můžete vidět stručný UML diagram tříd AST stromu.



2.2.3 Interpreter

Interpreter zpracuje objekt Block který vrátil a parser a vykoná ho. Interpreter v semestrální práci je vytvořen pomocí návrhového vzoru Visitor. Díky tomu je přímo v interpreteru provedena i sémantická analýza. Implementaci interpreteru je možné najít ve složce Intepreter pod názvem Interpreter.cs. Dále je ve složce i podsložka NativeLibrary která obsahuje implementaci knihovních funkcí.

3 POUŽITÝ INTERPRETERU

Zdrojový kód můžeme spouštět dvěma různými způsoby, a to buďto pomocí příkazového řádku anebo pomocí DrWren.

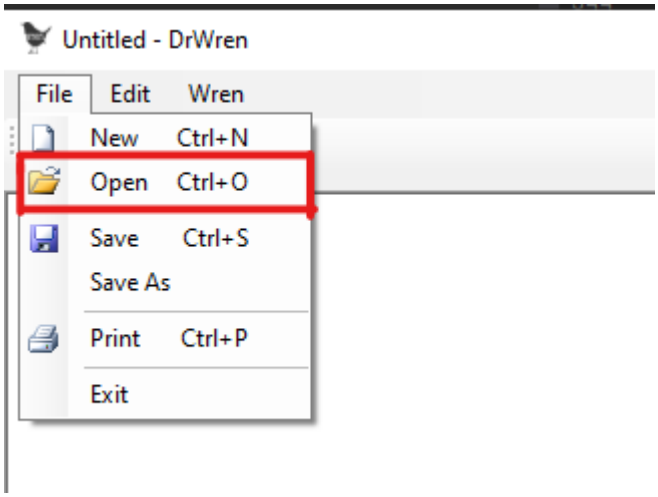
3.1 Příkazový řádek

Při spouštění interpreteru jazyka máte dvě možnosti jednou je, že si celý interpreter můžete sestavit ze zdrojového kódu anebo můžete použit již vytvoření spustitelní program. Druhý způsob spuštění nyní předvedu. Pro použití interpreteru otevřete příkazový řádek ve složce bin, kde se nachází spustitelní program pod názvem wren.exe. Nyní stačí jen napsat do příkazového řádku wren.exe a cestu ke zdrojovému kódu a tím spustíte interpreter.

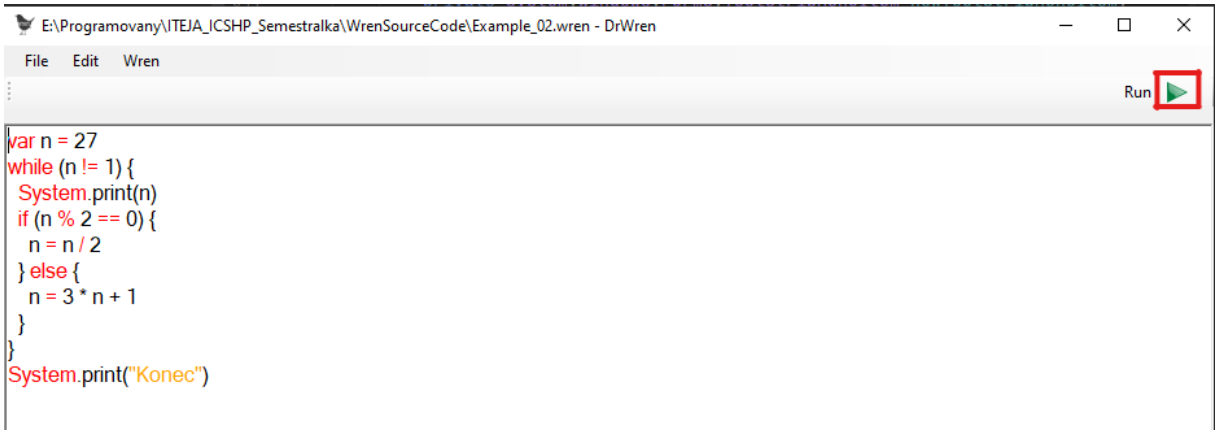
```
E:\Programovany\ITEJA_ICSHP_Semestralka\bin>wren.exe ../WrenSourceCode/Example_03.wren
FizzBuzz Game
1
2
Fizz
4
Buzz
Fizz
7
8
```

3.2 DrWren

DrWren je velmi jednoduché IDE pro spouštění zdrojových kódu jazyka Wren. DrWren se nachází také ve složce bin a podsložce DrWren ve které je spustitelný soubor s názvem DrWren.exe. Po spuštění aplikace můžeme načíst zdrojový kód pomocí položky v menu otevřít.



Po otevření zdrojového kódu ho můžeme spustit pomocí tlačítka Run.



3.3 Příklady zdrojových kódů

Příklady zdrojových kódů, pro jednoduché použití v programech, se nacházejí ve složce WrenSourceCode.

3.3.1 Želví grafika

```
var x = 0
while(x <= 360){
    Turtle.forward(x)
    Turtle.left(59)
    x = x + 1
}
Turtle.done()
```

3.3.2 Hailstone sequence

```
var n = 27
while (n != 1) {
  System.print(n)
  if (n % 2 == 0) {
    n = n / 2
  } else {
    n = 3 * n + 1
  }
}
System.print("Konec")
```

3.3.3 Fizz Buzz

```
System.print("FizzBuzz Game")
var i = 1
while (i <= 100) {
    if (i % 3 == 0) {
     if (i % 5 == 0) {
            System.print("FizzBuzz")
     }else{
         System.print("Fizz")
     }
    } else {
     if (i % 5 == 0) {
            System.print("Buzz")
     }else{
         System.print(i)
     }
    }
    i = i + 1
}
```