Implementação de Interpretador para Linguagem *Lang*

Diego Paiva - 201565516C
Thaynara Ferreira - 201565254C

Outubro, 2020

Resumo

O presente documento tem como objetivo mostrar como é feita a execução do interpretador desenvolvido para a linguagem fictícia *Lang*, referente ao trabalho prático da disciplina *DCC045 - Teoria dos Compiladores*.

1 Introdução

Um interpretador é um software que executa instruções contidas em um arquivo codificado em uma determinada linguagem de programação. As Seções 2 e 3 apresentam algumas decisões de projeto referentes à implementação do interpretador para a linguagem *Lang*, enquanto que as Seções 4 e 5 apresentam, respectivamente, instruções para a execução do interpretador e um exemplo de código interpretável.

2 Construção da AST

Antes de iniciar o processo de interpretação, é realizada a construção da Árvore de Sintaxe Abstrata (AST). Optou-se por realizar a conversão a Árvore de Sintaxe Concreta (CST) gerada pelo ANTLR [1] para uma AST, afim de tornar este procedimento mais maleável. Para tal, utilizamos o padrão de projeto **Visitor**, no qual um objeto *visitor* percorre a CST e, para cada nó, retorna a representação do mesmo em um objeto abstrato avaliável.

3 Estrutura do interpretador

Para a implementação do interpretador em si, optou-se também por utilizar o padrão de projeto **Visitor**, no qual um objeto *visitor* percorre a AST e, para cada nó, retorna o resultado da interpretação do objeto associado.

4 *Building*

Para realizar o *build* da aplicação, recomenda-se a utilização da ferramenta de gerenciamento **Maven** [2]. Com o Maven instalado, navegue até o diretório do projeto que contém o arquivo `pom.xml`. Após isso, rode o comando:

```
mvn clean compile assembly:single
```

Este comando irá compilar o código, incluir as dependências necessárias e empacotar em um formato distribuível `jar` dentro da pasta `target`. Todo código gerado automaticamente pelo ANTLR (`src/main/antlr4/lang/compiler/Lang.g4`) é incluido no diretório `target/generated-sources`. Para executar o programa principal, rode o seguinte comando:

```
java -jar target/lang-compiler-1.0-jar-with-dependencies.jar -i <file>
```

Onde `-i` é a opção para intepretar o arquivo passado como parâmetro.

5 Exemplo de Execução

Considere o seguinte código em *Lang*:

```
main() {
  divmod(3, 2)<x, y>;
  print x; -- 1.5
  print y; -- 1.0

  n = 0;

  -- Calculando fat de 0 a 10
  iterate(10) {
    print fat(n)[0];
    n = n + 1;
  }
}

divmod(q :: Int, r :: Int) : Int {
  return q / r, q % r;
}

fat(x :: Int) : Int {
  if (x < 1) {
    return 1;
  }
  else {
    return x * fat(x - 1)[0];
  }
}
```

Após execução conforme mostrado na Seção 4, o programa será interpretado e a saída será apresnetada como segue:

```
1.5
1.0
1
1.0
2.0
6.0
24.0
120.0
720.0
5040.0
40320.0
362880.0
```

Referências

[1] "Antlr." `https://www.antlr.org/`. Acesso em: 25/10/2020.

[2] "Apache maven project." `http://maven.apache.org/`. Acesso em: 04/10/2020.