

COMPILADORES

Nomes: Diego Jornada¹

Isadora Kurtz²

Marina Barros³

¹ 112040787

² 13104445

³ 11104974

Descrição do Trabalho

O trabalho tem como objetivo, a implementação dos analisadores léxico e sintático para a linguagem Minijava. Para isso, foi utilizado a gramática BNF for MiniJava que foi disponibilizada. Também foi implementado um sistema de tratamento de erros para entradas inválidas.

Léxico

Para o léxico foram utilizadas

-Palavras Reservadas

```
"$TRACE_ON" { yyparser.setDebug(true); }
"$TRACE_OFF" { yyparser.setDebug(false); }

"public"      { return Parser.PUBLIC; }
"static"      { return Parser.STATIC; }
"void"        { return Parser.VOID; }
"class"       { return Parser.CLASS; }
"main"        { return Parser.MAIN; }
"String"      { return Parser.STRING; }
"extends"     { return Parser.EXTENDS; }
"return"      { return Parser.RETURN; }
"int"         { return Parser.INT; }
"boolean"     { return Parser.BOOLEAN; }
"if"          { return Parser.IF; }
"else"        { return Parser.ELSE; }
"while"       { return Parser.WHILE; }
"length"      { return Parser.LENGTH; }
"System.out.println" { return Parser.PRINT; }
"true"        { return Parser.TOP; }
"false"       { return Parser.BOTTOM; }
"this"        { return Parser.THIS; }
"new"         { return Parser.NEW; }
"&&"         { return Parser.AND; }
```

¹ 112040787

² 13104445

³ 11104974

-Tokens

```
0 | [1-9][0-9]*      { return Parser.INTEGER; }
{ID}                  { return Parser.IDENT; }
{WHITE_SPACE}+       { }
```



```
"(" |
")" |
"{" |
"}" |
"[" |
"]" |
"," |
"." |
";" |
"|" |
"=" |
"<" |
"_" |
"*" |
"+" { return (int) yycharat(0); }
```

Sintático

Programas Testes

Para utilizar o programa é necessário usar um Make File e os programas rodam de maneira iterativa ou recebendo um arquivo como entrada.

```
class Factorial{
    public static void main(String[] a){
        System.out.println(new Fac().ComputeFac(10, 4));
    }
}
class Fac {

    public int ComputeFac(int num, int y){
        int x;
        int[] z;
        return 10;
    }

}
```

¹ 112040787

² 13104445

³ 11104974

