

Instruções para o Trabalho I

Compiladores I

As seguintes instruções devem ser consideradas para realização do trabalho.

- Não tente plagiar o trabalho do(a) seu(sua) colega, você pode prejudicar você e seu(sua) colega;
- O trabalho poderá ser realizado em grupos de **até** 4 pessoas;
- A entrega do código do trabalho deve ser feita via AVA/Moodle até o dia (18/02) às 23:59.

1 MiniJava

MiniJava é um subconjunto da linguagem Java. Como é um subconjunto, todo programa MiniJava é um programa Java válido que pode ser executado pela JVM.

MiniJava restringe a linguagem Java para ter apenas inteiros, booleanos, vetores de inteiros e classes, removendo interfaces, números de ponto flutuante, classes abstratas, strings, vetores de outros tipos etc. Também não há sobrecarga de métodos, ou métodos estáticos, exceto pelo método main da classe principal do programa. O comando `System.out.println` de MiniJava só pode imprimir números.

Um exemplo simples de MiniJava:

```
class Factorial{
    public static void main(String[] a){
        System.out.println(new Fac().ComputeFac(10));
    }
}
class Fac {
    public int ComputeFac(int num){
        int num_aux;
        if (num < 1)
            num_aux = 1;
        else
            num_aux = num * (this.ComputeFac(num-1));
        return num_aux ;
    }
}
```

1.1 Especificação Léxica

- Espaços em branco:

[\n \t \r \f]

- Identificadores: uma letra $[a - zA - Z]^+$, seguido de zero ou mais letras $[0 - 9a - zA - Z]^*$, dígitos ou `_`

- Numerais: apenas números inteiros [0-9]

- Operadores:

=
<
==
!=
+
-
*
&&
!

- Pontuação:

(
)
[
]
{
}
;
.
,

- Palavras reservadas:

boolean, class, extends, public, static, void, main,
String, return, int, if, else, while, System.out.println,
length, true, false, this, new, null

- Você deverá desenvolver um programa que realize a análise léxica da linguagem MiniJava. A partir de um arquivo contendo um código (programa) em MiniJava, o analisador léxico deverá produzir um arquivo com os tokens identificados e seus valores correspondentes. Cada linha do arquivo de saída deve conter um token reconhecido. O analisador léxico deve ser capaz de lidar com os erros léxicos encontrados no programa. A mensagem de erro, “ERRO LÉXICO” juntamente com a sequência lexicamente errada. Você pode utilizar a ferramenta Lex/Flex para realizar a implementação do Analisador Léxico.