

Prova Substitutiva de Construção de Compiladores. Turma Terça.  
Prof. José de Oliveira Guimarães. Primeiro Semestre de 2002.

1 (6.0) O comando loop tem o seguinte formato:

```
loop C1; C2; ... Cn times E;
```

Onde Ci é um comando e E um expressão. loop e times são palavras chaves.

Um analisador sintático incompleto para este comando é dado abaixo. Copie este analisador para a folha de respostas acrescentando o restante da análise sintática, o código para a criação do objeto da ASA para o loop e a análise semântica. Faça também a classe da ASA LoopStatement com o método genC para gerar código.

Você talvez usará a classe Type. Esta possui subclasses IntegerType, CharType e BooleanType. Type possui variáveis estáticas integerType, charType e booleanType. Cada uma delas aponta para o único objeto do programa de cada subclasse de Type.

Utilize o método addElement para adicionar um elemento no vetor v. O método statement() analisa uma instrução retornando um objeto de uma subclasse de Statement.

```
LoopStatement loopStatement() {
    nextToken(); // pega o "loop"
    Vector v = new Vector();
    while ( lexer.token != Symbol.times )
        ...
}
```

2. (2.0) Qual é a função da tabela de símbolos ? Para auxiliar a sua resposta, siga as seguintes dicas: explique a sua utilidade, diga quando um elemento é inserido na TS, quando elementos são retirados da TS e mostre um e apenas um exemplo de sua utilização,.

3.(2.0) Copie o analisador léxico abaixo na folha de respostas e o modifique para que ele reconheça comentários como os de C++, delimitados por /\* e \*/. Não se esqueça de sinalizar o erro "comentário aberto e não fechado". Lembre-se de que após reconhecer o comentário, nextToken() deve ser recursivamente chamado.

```
public void nextToken() {
    char ch;

    while ( (ch = input[tokenPos]) == ' ' || ch == '\\r' ||
            ch == '\\t' || ch == '\\n')
        tokenPos++;
    if ( ch == '\\0') token = new Symbol(Symbol.EOF);
    else // coloque o seu código aqui.
        if ( Character.isLetter( ch ) ) {
            // get an identifier or keyword
            ...
        }
        else {
            ...
        }
}
```