

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

1 (6.0) O comando for do compilador 10 tem o seguinte formato:

```
for variable = expression to expression do statement
```

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 for e a análise semântica. Faça também a classe da ASA ForStatement com o método genC para gerar código. Assuma que apenas inteiros podem ser variáveis de comando for.

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. A tabela de símbolos é referenciada pela variável st e possui métodos put(Object name, Object value) e Object get(Object name). O primeiro insere um símbolo e o segundo faz a busca por símbolo.

```
ForStatement forStatement() {
    nextToken();
    if ( lexer.token != Symbol.IDENT ) error();
    String name = lexer.getValue(); // nome do identificador
    nextToken();

    nextToken();
    Expr e2 = expr();
    if ( lexer.token != Symbol.DO ) error();
    nextToken();
    Statement s = statement();
}
```

2. (2.0) Faça um trecho de código em Java que monta a ASA para a seguinte declaração:

```
struct image {
    int backgroundColor;
    double dx, dy;
};
```

O objeto produzido será inserido na tabela de símbolos e referenciado pela ASA.

Desenhe o objeto criado usando bolinhas para representar objetos e setas para representar ponteiros (como feito inúmeras vezes em sala de aula). Não é preciso fazer as classes da ASA que você utilizar, apenas as utilize.

3. (2.0) Complemente o seguinte analisador léxico para que ele elimine comentários que começam com // e terminam no final da linha.

```
void nextToken() {
    while ( in[p] == ' ' ||
           in[p] == '\n' )
        p++;
    if(Character.isDigit(in[p]))
```

```
        reconheceDigito();  
switch ( in[p] ) {  
    '+' : ...  
}  
}
```

Copie o código acima na sua resposta.