

## **Trabalho Prático III – Análise Semântica – 17 pontos**

### **Descrição do trabalho**

Nesta etapa, você deverá implementar um analisador semântico para a linguagem *Pyscal*, cuja descrição inicial encontra-se no enunciado do Trabalho Prático I (porém, você deve seguir a versão correta da gramática que você alterou no trabalho prático anterior).

Deve-se usar uma implementação de tabela de símbolos e árvore sintática abstrata com nodos correspondentes às conversões implícitas de tipos.

Itens a verificar (12.0 pontos, sendo 1.2 pontos por item):

1. Declaração duplicada no mesmo escopo;
2. Declaração somente dos tipos permitidos;
3. Utilização de identificadores antes da declaração;
4. Verificação do tipos das expressões entre double e integer resultando em double;
5. Verificação do tipo resultante das expressões dos comandos if, while e das operações lógicas e relacionais, que deve ser bool;
6. Verificação se o tipo do índice em um vetor é inteiro (ConstInt ou ID do tipo integer);
7. Em uma atribuição a expressão que recebe o valor deve ser um identificador (variável) ou uma posição de uma variável vetor (por exemplo, v[2]) compatível com o valor atribuído;
8. Número e tipos dos parâmetros das chamadas de funções com a declaração dessas funções;
9. Verificação se a função retorna expressão compatível com o tipo de retorno declarado na função;
10. Verificação da não atribuição de funções, cujo tipo de retorno seja void.

Além da verificação de tipos, seu compilador deverá gerar código assembly para a JVM (4.0 pontos para geração e 1.0 ponto para execução) utilizando o software Jasmin. No SOL, durante a semana e a próxima, serão disponibilizados alguns exemplos desse Assembly JVM. Embora a maioria dos comandos necessários para a tradução sejam contemplados em sala, existem outros comandos que não serão apresentados, mas que podem ser obtidos na especificação completa da linguagem em <http://jasmin.sourceforge.net/instructions.html> (inglês) e na apostila colocada no SOL (português).

### **O que entregar?**

Você deverá entregar nesta etapa:

1. Programa com todos os arquivos fonte (será apresentado em data previamente marcada) ;
2. Testes realizados com programas corretos e errados com relação a verificação de tipos (no mínimo, 3 certos e 3 errados), contendo todas as verificações especificadas.

Para avaliar a correção, o programa deverá exibir a Árvore Sintática Abstrata em pré-ordem com o tipo anotado em cada nodo (tipo vazio, erro ou o tipo macro correspondente).