UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
FACOM - FACULDADE DE COMPUTAÇÃO

COMPILADORES I – 2023/1

PROFA. BIANCA DE ALMEIDA DANTAS

Trabalho Prático
3ª Etapa – Análise Semântica

1 DESCRIÇÃO

A terceira parte do trabalho prático de nossa disciplina consiste na implementação de algumas funcionalidades para verificação semântica da linguagem de programação alvo de nossos estudos, a MiniJava modificada. As funcionalidades que devem ser implementadas deverão ser realizadas concomitantemente à análise sintática (com o auxílio da tabela de símbolos), sendo, assim, um processo guiado pela sintaxe.

As seguintes atividades devem ser realizadas:

- Verificação de declaração de classes: ao declarar uma classe, ela deve ser inserida na tabela de símbolos do escopo corrente;
- Verificação de hierarquia de classes: ao acessar um membro de uma classe, deve-se verificar se a classe o possui em sua definição;
- Verificação de declaração de variáveis e métodos: ao acessar uma variável ou um método, deve-se verificar se sua declaração existe e pode ser acessada a partir do escopo corrente;
- Verificação de tipos em comandos de atribuição e chamadas de métodos: quando uma atribuição for realizada deve-se verificar se a expressão à direita é compatível com o *lvalue*. Ao invocar um método, deve-se verificar a compatibilidade dos tipos de seus parâmetros reais com os parâmetros formais, bem como a compatibilidade do tipo de retorno com o *lvalue*, caso a invocação esteja em um comando de atribuição.

Observe que para a realização das atividades descritas, é necessário que a tabela de símbolos seja manipulada de maneira coerente, lidando com o gerenciamento dos ponteiros para a tabela de símbolos global e para a tabela de símbolos corrente. Isso envolve saber o momento correto para a criação de novas tabelas para novos escopos e, eventualmente, para abandonar um escopo e voltar para o escopo que o englobe.

Erros apenas semânticos não devem parar a execução do processo de compilação. O seu compilador pode continuar executando normalmente apenas apresentando a mensagem adequada.

Os códigos fontes do programa desenvolvido serão compilados usando o compilador g++.

2 EXECUÇÃO E ENTRADA

O seu programa deve ser capaz de realizar a compilação de um arquivo de texto com a extensão **.mj**, cujo nome será fornecido na linha de comando do terminal logo após o nome do executável de seu compilador. Por exemplo, se seu executável possuir o nome **mjc** e o arquivo de entrada for **teste1.mj**, a seguinte instrução será digitada no terminal:

```
./mjc teste1.mj
```

3 SAÍDA

O compilador deve emitir mensagens de erros, caso encontre algum, informando claramente o erro e a linha de ocorrência. Caso não sejam encontrados erros, o compilador deve imprimir que a compilação foi encerrada com sucesso. Todas as mensagens devem ser mostradas no terminal. Sugere-se usar como inspiração mensagens geradas por compiladores reais (como o próprio g++).

4 AVALIAÇÃO

O seu programa será compilado usando o comando:

```
g++ -Wall *.cpp -o mjc
```

Caso a compilação gere erros e o executável não seja gerado, o trabalho receberá nota zero.

O programa será executado com n arquivos fontes, podendo conter erros ou não, e a nota atribuída será proporcional ao número de testes cuja execução de seu compilador conseguir detectar os erros (ou a falta deles) corretamente. Testes em que a execução não gerar o resultado esperado serão zerados.

O programa deve receber a entrada e gerar a saída **exatamente** como especificado nas descrições das etapas, caso isso não ocorra, a nota será penalizada.

O programa entregue deverá conter, pelo menos, as análises léxica e sintática para ser considerado. Sendo assim, a análise semântica não é obrigatória, mas sem ela o trabalho valerá no máximo $8{,}0$**.**

5 ESPECIFICAÇÕES

- O trabalho prático poderá ser realizado em grupos de, no máximo, 3 alunos **sem exceções**.
- A linguagem C++ deverá ser utilizada na implementação do trabalho.
- A entrega de todas as etapas deve ser realizada até o dia: **23/06/2023**.

6 AVALIAÇÃO

O trabalho prático será avaliado utilizando os seguintes critérios:

- A etapa de análise semântica valerá 20% da nota total.
- O funcionamento correto do analisador sintático em conjunto com o léxico valerá 80% da nota total.
- Trabalhos que não compilarem receberão nota **zero**.
- Um conjunto contendo $numTestes$ programas de teste será executado, com possíveis erros, e para cada teste seu programa deve emitir mensagens de erro (se houver) e, ao final, informar o término do processo de compilação.
- A cada teste para o qual o seu compilador executar corretamente, o grupo receberá pontuação equivalente $\frac{1}{numTestes} * nota\ total$. Executar corretamente significa terminar o percurso do

programa de entrada e emitir mensagens para todos os erros do programa, se houver.

- Execuções parcialmente corretas receberão uma porcentagem da nota total.
- Se o seu programa travar na execução de um teste, a pontuação para tal teste será zerada.