

## PROJETO – PARTE 2

CCMP0006 - COMPILADORES

PROF. GUSTAVO CARVALHO

**Enviar o código/lista de regras por e-mail & Perguntas individuais**

### OBJETIVOS

- |   |   |
|---|---|
| 1 | Implementar um analisador semântico utilizando o padrão <i>Visitor</i> que verifica se a árvore atende às restrições da linguagem. A saída do analisador semântico é a AST anotada (com ligações uso x declaração, tipos inferidos e outras informações julgadas pertinentes). Para tanto, as classes da AST devem ser modificadas. |
| 2 | Cada equipe deve verificar as 9 restrições contextuais abaixo (ou variações das mesmas). Além destas 9 restrições, a equipe deve definir pelo menos mais 1 restrição contextual e verificá-la.  |
| 3 | Escrever pelo menos <b>20 programas</b> que ilustrem as restrições contextuais definidas.   |

### RESTRIÇÕES CONTEXTUAIS

- |    |   |
|----|---|
| 1  | Todos os identificadores precisam ser declarados antes de serem utilizados.   |
| 2  | Não pode haver mais de um identificador (global   local) com o mesmo <i>spelling</i> .  |
| 3  | Ao chamar uma função os tipos dos argumentos devem ser iguais ao dos parâmetros.  |
| 4  | Toda função com retorno diferente de void, precisa ter um ou mais return deste tipo.  |
| 5  | Ao retornar um valor, a função em questão deve ter como retorno o mesmo tipo.   |
| 6  | Break e continue podem ser utilizados somente dentro do escopo de um while.   |
| 7  | Todos os operadores devem ser aplicados a operandos do mesmo tipo.<br>Operadores +, -, * e / devem ser aplicados a operandos int<br>Operadores +, -, * e / retornam o tipo dos seus operandos.<br>Operadores == e != devem ser aplicados a operandos int, ou boolean.<br>Operadores >, <, >=, <= devem ser aplicados a operandos int<br>Operadores ==, !=, >, <, >=, <= retornam valor de tipo boolean. |
| 8  | Em A = B, o tipo de B precisa ser igual ao tipo de A e A é uma variável.  |
| 9  | Todo código deve ter uma função principal (ponto de entrada).   |
| 10 | ? Definida pela equipe!   |