

Atividade: Conceitos de Linguagem de Programação.

Aluno: Tiago Carvalho

#### Capítulo 1:

1 – A capacidade de compreensão de um código em geral é melhorada, facilitando a escrita de um código mais legível.

2- Melhora a capacidade de expressar ideias em forma de código, capacidade para escolher uma linguagem de programação sabendo da capacidade dela e onde aplicá-la.

7- Dificulta a tomada de decisão pois é possível fazer algo com mais de uma funcionalidade, dificultando a legibilidade para outros programadores que não iniciaram junto com o projeto.

8 – Prejudica a legibilidade pois o programador pode se confundir onde o operador está sendo usado.

13 – Se dá quando ele está dentro do planejamento no qual ele foi feito e realiza o que tem que ser feito de forma clara na escrita do código.

16 – É a capacidade do programa em interceptar erros em tempo de execução e tomar medidas para corrigi-los sem parar a execução.

#### Capítulo 5:

6a – Sub1: refere ao X da própria função sub1.

Sub2: refere ao X da função sub1.

Sub3: refere ao X da função maior.

6b - Sub1: refere ao X da própria função sub1.

Sub2: refere ao X da função sub1.

Sub3: refere ao X da função sub2.

7 – No escopo estático X vale 5, no outro vale 10.

8 - Sub1: A, X, Y, Z.

Sub2: A, B, X, Y, Z.

Sub3: A, W, X, Y, Z.

9 - Sub1: A, X, Y, Z.

Sub2: A, W, X, Y, Z.

Sub3: A, B, W, X, Y, Z.

10 – Ponta1: definição 1, 2 ; A, B, C, D.

Ponta2; definição 1, 2, 3; A, B, C, D.

Ponta3: definição 1, 2; A, B, C, D.

Ponta4: definição 1; A, B, C.

11<sup>a</sup> – main: A, B, C; func1: B, C, D; func2: C, D; func3: A, B, C, D, F.

11b – main: A, B, C; func1: B, C, D; func3: A, B, C, D, E, F.

11c -main: A, B, C; func2: C, D, E; func3: D, E, F; func1: A, B, C, D, E, F.

11d -main: A, B, C; func3: D, E, F; func1: A, B, C, D, E, F.

11e -main: A, B, C; func1: B, C, D; func3: D, E, F; func2: A, B, C, D, E, F.

11f -main: A, B, C; func3: D, E, F; func2: C, D, E; func1: A, B, C, D, E, F.

### Capítulo 3:

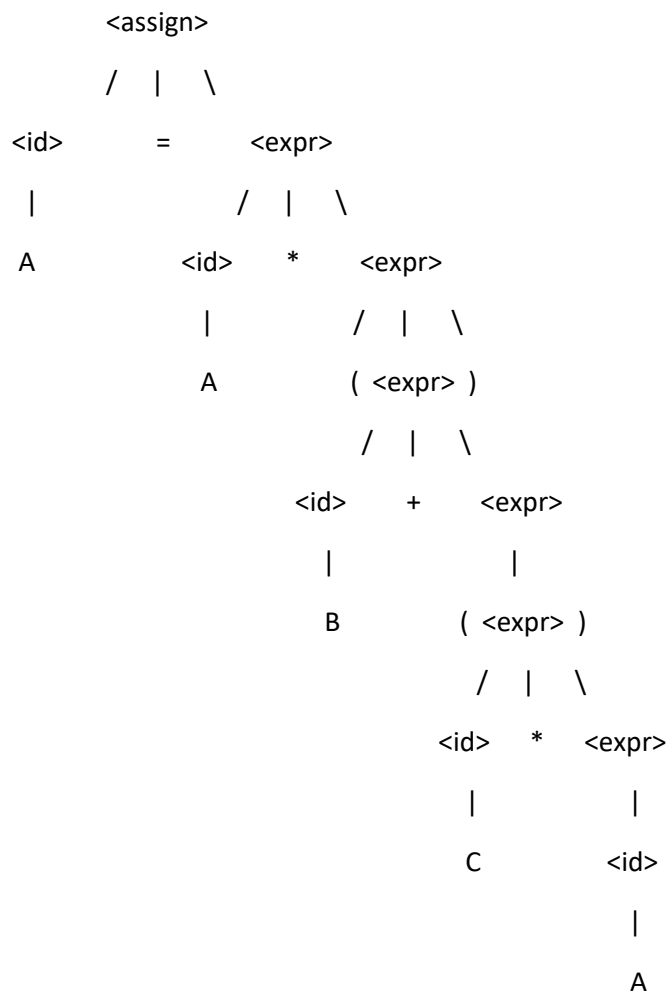
```
2g - <cmd_switch -> switch(<expr>)> {  
    {case <term>:}  
        <expr>;  
    break;  
    [default]:  
        <expr>;  
}
```

2d – Pode ser vista como uma lista de variáveis qualquer uma podendo ser de qualquer tipo. Mas todas estão no mesmo espaço da memória .

```
3 -    <assing> -> <id> = <expr>  
        <id> -> <expr> * <term>  
            | <term>  
        <term> -> <term> +< <factor>  
            | <factor>  
        <factor> -> (expr)  
            | <id>
```

6<sup>a</sup> <assing> -> <id> = <expr>

$A = \langle \text{expr} \rangle$   
 $A = \langle \text{id} \rangle * \langle \text{expr} \rangle$   
 $A = A * \langle \text{expr} \rangle$   
 $A = A * (\langle \text{expr} \rangle)$   
 $A = A * (\langle \text{id} \rangle + \langle \text{expr} \rangle)$   
 $A = A * (B + \langle \text{expr} \rangle)$   
 $A = A * (B + (\langle \text{expr} \rangle))$   
 $A = A * (B + (\langle \text{expr} \rangle))$   
 $A = A * (B + (\langle \text{id} \rangle * \langle \text{expr} \rangle))$   
 $A = A * (B + ((C + \langle \text{expr} \rangle)))$   
 $A = A * (B + (C + \langle \text{id} \rangle))$   
 $A = A * (B + (C * A))$



10 – A sentença inicia com um ou mais A seguido por um ou mais B seguido por um ou mais C.

11- A sentença inicia com um ou mais B seguido por dois ou mais terminada em B.

12- A sentença inicia com um A ou mais seguido de um C e um ou mais D, ou, inicia com um ou mais C ou terminando em um B.