



**Lista de Exercícios – 05: Tipos de Dados, Classes de Objetos e Operadores**

1. Com relação às classes de objetos, marque a(s) alternativa(s) correta(s).

- (X) Variável é um valor imposto que pode ser alterado no decorrer do código.  
(X) Constante pode ser empregada em regiões de código sequencial.  
(X) Sinais são objetos que podem ter o seu valor alterado.  
( ) Constantes não podem ser declaradas em declaração da entidade, arquitetura da entidade, pacote, processo e subprogramas.  
(X) Sinais podem ser declarados na declaração da entidade e na arquitetura da entidade.  
( ) Variável não pode ser declarada e empregada em regiões de código sequencial.  
(X) Sinais são empregados em regiões de código concorrente e sequencial.  
(X) Constante é um objeto com um valor estático.

2. Diante das declarações abaixo, marque V (verdadeiro) ou F (falso). No caso da alternativa ser falsa, justifique.

<b>Justificativa abaixo:</b>	
(F) CONSTANT Pi : REAL <= 3.14;	CONSTANT Pi : REAL := 3.14;
(V) CONSTANT atraso : TIME := 50 ns;	
(F) VARIABLE var <= INTEGER;	VARIABLE var : INTEGER;
(F) VARIABLE aux := BIT : '0';	VARIABLE aux : BIT := '0';
(V) SIGNAL tempo : TIME := 50 ns;	
(F) SIGNAL clk : BIT <= '0';	SIGNAL clk : BIT := '0';
(V) CONSTANT Pi : REAL := 3.14;	
(F) CONSTANT atraso <= TIME := 50 ns;	CONSTANT atraso : TIME := 50 ns;
(F) VARIABLE var = INTEGER;	VARIABLE var : INTEGER;
(V) VARIABLE aux : BIT := '0';	
(F) SIGNAL tempo := TIME := 50 ns;	SIGNAL tempo : TIME := 50 ns;
(F) SIGNAL clk <= BIT := '0';	SIGNAL clk : BIT := '0';
(F) CONSTANT Pi := REAL : 3.14;	CONSTANT Pi : REAL := 3.14;
(F) CONSTANT atraso : TIME = 50 ns;	CONSTANT atraso : TIME := 50 ns;
(V) VARIABLE var : INTEGER;	
(F) VARIABLE aux = BIT = '0';	VARIABLE aux : BIT := '0';
(F) SIGNAL tempo : TIME <= 50 ns;	SIGNAL tempo : TIME := 50 ns;
(V) SIGNAL clk : BIT := '0';	

3. Com base no código em VHDL abaixo, verifique se há ocorrência de erros. Em caso afirmativo, proponha uma correção.

```
ENTITY signal_ent IS
    PORT (a, b : IN BIT;
          s : OUT BIT);
END signal_ent;

ARCHITECTURE signal_arc OF signal_ent IS
    BEGIN
        SIGNAL tempo : TIME := 50 ns;
        SIGNAL clk : BIT := '0';
        s <= a XOR b;
        clk <= NOT clk AFTER tempo;
    END signal_arc;
```

Proposta de correção:

```
ENTITY signal_ent IS
    PORT (a, b : IN BIT;
          s : OUT BIT);
END signal_ent;

ARCHITECTURE signal_arc OF signal_ent IS
    SIGNAL tempo : TIME := 50 ns;
    SIGNAL clk : BIT := '0';
```



```
BEGIN  
    s <= a XOR b;  
    clk <= NOT clk AFTER tempo;  
END signal_arc;
```

4. Com base no código em VHDL abaixo, verifique se há ocorrência de erros. Em caso afirmativo, proponha uma correção.

```
ENTITY exemplo IS  
    PORT (d0, d1, d2 : IN BIT;  
          s1, s2 : OUT BIT);  
END exemplo;  
  
ARCHITECTURE teste OF exemplo IS  
BEGIN  
    processo1 : PROCESS(d0, d1, d2)  
        VARIABLE var1 : STD_LOGIC;  
        BEGIN  
            var1 := d0 AND d1;  
            s1 <= var1 OR d2;  
        END processo1;  
    processo2 : PROCESS(d0, d1, d2)  
        SIGNAL signal1 : BIT;  
        BEGIN  
            signal1 <= d0 AND d1;  
            s2 <= signal1 OR d2;  
        END processo2;  
END teste;
```

Proposta de correção:

```
ENTITY exemplo IS  
    PORT (d0, d1, d2 : IN BIT;  
          s1, s2 : OUT BIT);  
END exemplo;  
  
ARCHITECTURE teste OF exemplo IS  
    SIGNAL signal1 : BIT;  
BEGIN  
    processo1 : PROCESS(d0, d1, d2)  
        VARIABLE var1 : STD_LOGIC;  
        BEGIN  
            var1 := d0 AND d1;  
            s1 <= var1 OR d2;  
        END processo1;  
    processo2 : PROCESS(d0, d1, d2)  
        BEGIN  
            signal1 <= d0 AND d1;  
            s2 <= signal1 OR d2;  
        END processo2;  
END teste;
```



5. Considere os valores inteiros  $a = 7$ ,  $b = -3$  e  $c = 3$ . Mostre os resultados para as sentenças com operadores aritméticos em VHDL, a seguir:
- a)  $a/c = 2$ ;
  - b)  $a/(a+b) = 1$ ;
  - c)  $(a*c)/c = 7$ ;
  - d)  $(a/c)*c = 6$ ;
  - e)  $(a+b)**c = 64$ ;
  - f)  $ABS(a) + ABS(b) = 10$ ;
  - g)  $a \text{ REM } b = 1$ ;
  - h)  $a \text{ MOD } b = -2$ ;

6. Considere  $a \leq "11001"$ . Mostre os resultados para as sentenças com operadores de deslocamento em VHDL, a seguir:

Resultado:

- |                                 |            |
|---------------------------------|------------|
| a) $x \leq a \text{ SLL } 2$ ;  | -- "00100" |
| b) $y \leq a \text{ SLA } 2$ ;  | .- "00111" |
| c) $w \leq a \text{ SLL } -3$ ; | .- "00011" |
| d) $z \leq a \text{ SRL } 2$ ;  | -- "00110" |
| e) $r \leq a \text{ SLA } -3$ ; | .- "11111" |
| f) $s \leq a \text{ SRL } -3$ ; | .- "01000" |
| g) $t \leq a \text{ ROL } 2$ ;  | -- "00111" |
| h) $u \leq a \text{ ROR } -3$ ; | .- "01110" |
| i) $v \leq a \text{ ROL } -3$ ; | .- "00111" |
| j) $k \leq a \text{ ROR } 2$ ;  | -- "01110" |
| k) $y \leq a \text{ SRA } 2$ ;  | .- "11110" |
| l) $w \leq a \text{ SRA } -3$ ; | .- "01111" |
7. Apresente o código de uma entidade de projeto que descreva as quatro expressões lógicas a seguir. Nessas expressões, considere o operador lógico de negação com maior precedência e o operador OR com menor precedência. A declaração da entidade deve ter quatro portas de entrada, **a**, **b**, **c** e **d**, e quatro portas de saída, **s1**, **s2**, **s3** e **s4**, todas do tipo bit.

$$s1 = a + \overline{b}$$

$$s2 = a + \overline{b}.c$$

$$s3 = (a + \overline{b}).(c + d)$$

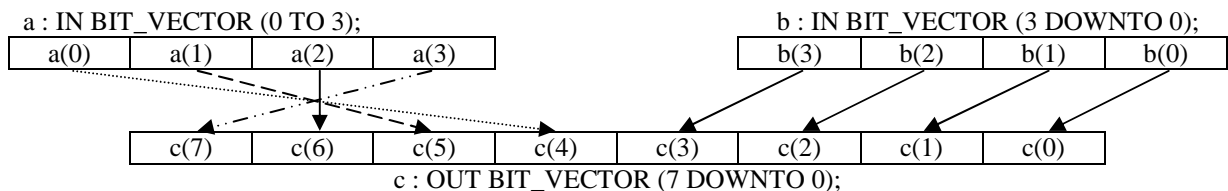
$$s4 = (a + \overline{b}).\overline{(c + a.d)}$$



```
ENTITY std_ya IS
  PORT (a, b, c, d : IN  BIT;
        s1         : BUFFER BIT;
        s2, s3, s4 : OUT  BIT);
END std_ya;

ARCHITECTURE teste OF std_ya IS
BEGIN
  s1 <= a OR NOT b;
  s2 <= a OR (NOT b AND c) OR d;
  s3 <= s1 AND (c OR d);
  s4 <= s1 AND NOT(c OR (a AND d));
END teste;
```

8. Apresente a descrição de uma entidade de projeto (declaração da entidade mais arquitetura da entidade = código em VHDL) com duas portas de entrada e uma porta de saída, todas do tipo bit\_vector. A figura a seguir ilustra o problema. Ambas as entradas possuem quatro bits, e o valor presente nessas entradas deve ser transferido para a saída, conforme ilustrado na figura.



```
ENTITY teste IS
  PORT (a : IN BIT_VECTOR (0 TO 3);
        b : IN BIT_VECTOR (3 DOWNT0 0);
        c : OUT BIT_VECTOR (7 DOWNT0 0));
END teste;
```

```
ARCHITECTURE vetor OF teste IS
BEGIN
  c(4) <= a(0);
  c(5) <= a(1);
  c(6) <= a(2);
  c(7) <= a(3);
  c(3 DOWNT0 0) <= b(3 DOWNT0 0);
  c <= c(7) & c(6) & c(5) & c(4) & c(3 DOWNT0 0);
END vetor;
```

9. Considerando a descrição ou código em VHDL a seguir, determine qual o valor de cada porta de saída. Observe que todos os comandos são concorrentes, portanto, a ordem nas linhas do código não importa.

```
ENTITY std_yc IS
  PORT (a, b : OUT BIT_VECTOR (2 DOWNT0 0);
        c, d : OUT BIT_VECTOR (0 TO 2));
END std_yc;

ARCHITECTURE teste OF std_yc IS
  CONSTANT x : BIT_VECTOR(0 TO 7) := B"1101_1001";
  SIGNAL y : BIT_VECTOR(3 DOWNT0 0);
BEGIN
  a <= x(1 TO 3);
  b <= y(3 DOWNT0 1);
  c <= x(5 TO 7);
  d <= y(2 DOWNT0 0);
  y <= x(2 TO 5);
END teste;
```



Resposta:

y= "0110"  
a="101"  
b="011"  
c="001"  
d="110"

10. Considerando a descrição ou código em VHDL a seguir, identifique as linhas que contêm erros no código e proponha uma solução.

```
ENTITY errad_1 IS
  PORT (a, b, c, d : IN BIT;
        S      : OUT BIT_VECTOR (5 DOWNT0 0));
END errad_1;

ARCHITECTURE teste OF errad_1 IS
BEGIN
  s(0) <= a AND b OR c AND d;
  s(1) <= a NOR b NOR c;
  s(2) <= a AND b OR c;
  s(3) <= NOT (a AND b) NAND c;
  s(4) <= a XOR b XOR c;
END teste;
```

Proposta de correção:

```
ENTITY corret_1 IS
  PORT (a, b, c, d : IN BIT;
        s      : OUT BIT_VECTOR (5 DOWNT0 0));
END corret_1;

ARCHITECTURE teste OF corret_1 IS
BEGIN
  s(0) <= (a AND b) OR (c AND d); -- opcao 1
  --s(0) <= a AND (b OR c) AND d; -- opcao 2

  s(1) <= (a NOR b) NOR c;          -- opcao 1
  --s(1) <= a NOR (b NOR c);       -- opcao 2

  s(2) <= (a AND b) OR c;          -- opcao 1
  --s(2) <= a AND (b OR c);       -- opcao 2

  s(3) <= NOT (a AND b) NAND c;    -- correta
  s(4) <= a XOR b XOR c;           -- correta
END teste;
```



11. Na descrição ou código em VHDL a seguir, para teste de operações lógicas, a declaração da entidade contém duas entradas e três saídas do tipo `bit_vector`. Considerando que `a_bit` tem o valor “010” e `b_bit` tem o valor “111”, determine qual o valor de cada porta de saída.

```
ENTITY std_c IS
  PORT( a_bit      : IN BIT_VECTOR(2 DOWNTO 0);
        b_bit      : IN BIT_VECTOR(2 DOWNTO 0);
        not_bit, and_bit, or_bit : OUT BIT_VECTOR(2 DOWNTO 0));
END std_c;
```

```
ARCHITECTURE exemplo OF std_c IS
BEGIN
  not_bit <= NOT a_bit;
  and_bit <= a_bit AND b_bit;
  or_bit  <= a_bit OR b_bit;
END exemplo;
```

Resposta:

```
not_bit  = “101”
and_bit  = “010”
or_bit   = “111”
```