FTL066 – Programação em Tempo Real
Terceira Lista de Exercícios

1) Ada termina cada construtor com **end** <nome do construtor>; C não usa um marcador final. Quais são as vantagens e desvantagens dos projetos destas linguagens?

2) Java e C são *case-sensitive*; Ada não é. Quais são os argumentos a favor e contra do *case-sensitive*?

3) Uma linguagem deve sempre solicitar que sejam dados valores iniciais para as variáveis? Cite pelo menos um problema relacionado a não inicialização de variáveis usadas no programa.

4) O uso do comando **exit** em Ada leva a programas legíveis e confiáveis?

5) Escreva programas para a solução dos seguintes problemas usando a linguagem Ada95:

a) Lido um string, escreva-o na ordem inversa (palíndromo).
b) Lidos dois vetores A, de tamanho m e B, de tamanho n, calcule o vetor C, soma de A e B.
c) Lido um string, conte quantas vogais existem.
d) Lido um string, escrever quantas e quais são as vogais.
e) O produto interno entre dois vetores A e B é o escalar obtido por $\Sigma\, a_i x b_i$, onde n é o tamanho dos dois vetores.
f) Lidos dois strings A e B, verifique se o string B está contido em A.
g) Lido um array numérico de n elementos, ache o maior.
h) Lido um conjunto de n elementos, ordená-lo crescentemente.
i) Leia uma matriz A e calcule a média aritmética de seus elementos.
j) Lidas duas matrizes A, de dimensões mxn e B, de dimensões pxq, calcule a matriz C, soma de A e B.
k) Lida uma matriz, gere sua matriz transposta. **Dica:** Matriz transposta, em matemática, é o resultado da troca de linhas por colunas em uma determinada matriz. A matriz transposta de uma matriz qualquer M é representada por Mt.
l) Lidas duas matrizes A, de dimensões mxn e B, de dimensões pxq, calcule a matriz C, produto de A e B.
m) Lida uma matriz quadrada, calcule a somatória dos elementos da diagonal principal.
n) Lidas duas matrizes de tamanho 3x3 verifique se uma é inversa da outra.
o) Lida uma matriz, calcule a porcentagem de elementos nulos desta matriz.

6) Implemente um procedimento que irá imprimir a área de qualquer objeto geométrico de um tipo derivado a partir do tipo ***Object***.

```
package Object is

  type Object is tagged
```

```
    record
     X_Coord: Float;
     Y_Coord: Float;
    end record;
  function Distance(O: Object) return Float;
  function Area(O: Object) return Float;
end Objects;
```

7) Em um mundo tradicional mulheres não tem barbas e homens não amamentam os filhos. Porém, todas as pessoas têm uma data de nascimento. Declare um tipo ***Person*** com o componente comum ***Birth*** do tipo ***Date*** (conforme mostrado abaixo) e então derive os tipos ***Man*** e ***Woman*** que tenham componentes adicionais indicando se eles possuem barba ou não e quantos filhos eles amamentam respectivamente.

```
type Month_Name is (Jan, Feb, Mar, Apr, May, Jun,
                    Jul, Aug, Sep, Oct, Nov, Dec);
type Date is
  record
    Day: Integer range 1..31;
    Month: Month_Name;
    Year: Integer;
  end record;
```

8) Declare procedimentos ***Print_Details*** para ***Person***, ***Man*** e ***Woman*** que fornecem informações a respeito dos valores atuais dos componentes deles. Então declare um procedimento ***Analyze_Person*** que recebe um parâmetro do tipo de uma classe ampla ***Person'Class*** e chama o procedimento apropriado ***Print_Details***.

9) Escreva a especificação e o corpo do pacote ***Queues*** usando uma implementação de lista encandeada.

10) Escreva um pacote ***generic*** para ***Queues*** tal que filas de qualquer tipo possam ser declaradas.

11) Escreva a especificação e o corpo do pacote ***Stacks*** usando uma implementação de lista encandeada.

12) Escreva um pacote ***generic*** para ***Stacks*** tal que pilhas de qualquer tipo possam ser declaradas.

13) Crie um tipo de dado abstrato (TDA) chamado Retângulo. O TDA tem atributos comprimento e largura cada um com valor default igual a 1. Ele tem funções “membro” que calculam o *comprimento*, *largura*, *perímetro* e *área* do retângulo. Forneça as funções *set* e *get*, tanto para o comprimento como para a largura. A função *set* deve verificar se o comprimento e a largura são números de ponto flutuante maiores que 0.0 e menores que 20.0. Além disso, a função *set* especifica se as coordenadas fornecidas de fato especificam um retângulo. Lembre que o comprimento é a maior das duas dimensões. Inclua uma função quadrado, que determina se um retângulo é um quadrado.

14) Crie um TDA Racional para fazer aritmética com frações. Escreva um programa para testar seu TDA. Use variáveis inteiras para representar os dados ***private*** do TDA – o numerador e o denominador. Forneça uma “função” construtor que permita que um objeto deste TDA seja inicializado quando é declarado. O construtor deve conter valores *default* no caso de nenhum inicializador ser fornecido e deve armazenar a fração em formato reduzido. Forneça funções membro para cada um dos seguintes itens:

a) Adição, subtração, multiplicação e divisão de dois números do tipo Racional.
b) Imprimir números do tipo Racional no formato a / b alinhado a esquerda.
c) Imprimir números do tipo Racional em formato de ponto flutuante com 3 dígitos de precisão.

19/12/2013