

Prova Final de Técnicas Alternativas de Programação (14/12/10)

Questão 1 (25 pontos)

Construir um predicado em Prolog, denominado “n_esimo_elemento”, o qual relaciona os seguintes termos: (1) um número inteiro; (2) uma lista de itens; (3) o item da lista que ocupa a posição correspondente ao número inteiro do primeiro termo. Seu comportamento é o expresso abaixo:

```
?- n_esimo_elemento(3, [a,b,c,d,e], X).  
X = c  
yes
```

Questão 2 (25 pontos)

Construir um predicado em Prolog, denominado “linearizada”, o qual relaciona duas listas. A lista do primeiro termo pode conter strings assim como outras sub-listas como seus elementos. A lista do segundo termo contém todos os strings da lista do primeiro termo, incluindo todos os strings das sub-listas (recursivamente), porém, sem nenhuma sub-lista.

```
?- linearizada([a, [b, [c, d], e], f], X).  
X = [a, b, c, d, e, f]  
yes
```

Questão 3 (25 pontos)

Em programação Orientada a Objetos, diga com poucas palavras o que entende pelos seguintes conceitos:

1. especialização de métodos;
2. efeito do envio de mensagens sobre a sequência do código de um método.

Recomenda-se o uso de exemplos com código na linguagem Flavours em cada um dos casos. Isto pode substituir várias palavras de um texto corrido.

Questão 4 (25 pontos)

No código a seguir, suponha que o programador tenha focado o conceito de polimorfismo na organização

de classes como sendo o princípio mais importante de Orientação a Objetos para programar a solução do problema de desenhar figuras geométricas.

```
flavour figura_geometrica;  
  ivars cor largura_da_linha;  
endflavour;  
  
flavour triangulo isa figura_geometrica;  
  ivars numero_de_lados = 3;  
  defmethod desenhe_se;  
    ...  
    ...  
    ...  
  enddefmethod;  
endflavour;
```

```
flavour retangulo isa figura_geometrica;  
  ivars numero_de_lados = 4;  
  defmethod desenhe_se;  
    ...  
    ...  
    ...  
  enddefmethod;  
endflavour;
```

Ao longo do aprofundamento do entendimento do que o problema realmente requeria, o programador descobriu que precisava detalhar melhor o registro dos dados assim como o comportamento das figuras de quatro lados (quadriláteros). Se quiséssemos re-organizar e estender então a definição do código acima, incluindo a definição de novas classes para abranger a solução do problema de desenhar “quadrados”, “losangos” e “trapézios”, como você faria isso? Por favor, seja claro, explicando isso por meio de texto normal e de código (abreviado) em linguagem Flavours (da mesma forma que foi apresentada no código anterior).

BOA SORTE!