Segunda Prova de Técnicas Alternativas de Programação (28/11/2005)

Nome: Assinatura:

Questão 1 (15 PONTOS)

Explique brevemente o que você entende pelas seguinte IDÉIAS (ou princípios) que fazem parte do escopo de programação orientada a objetos: (a) especialização de médodos; (b) abstração.

Questão 2 (25 PONTOS)

O problema clássico do sistema de atendimento de chamadas de elevadores pode ser muito complexo mas vamos adotar várias simplificações. No caso corrente, cada andar possui apenas um botão para indicar chamadas de descida, sempre para o andar 0 (zero). Também por razão de simplicidade, as chamadas são atendidas na mesma ordem em que foram feitas (*i.e.*, sem nenhum planejamento de otimização para economia de energia ou tempo). Finalmente, assumiremos o funcionamento sincronizado de atendimento das chamadas que estão na fila de espera (*i.e.*, quando o atendimento da fila de chamadas se inicia, nenhuma chamada nova pode ser adicionada enquanto todos os atendimentos não terminarem).

Considere o programa INCOMPLETO abaixo, o qual foi escrito em linguagem Flavours (orientado a objetos):

```
flavour elevador;
  ivars andar fila_de_chamadas;
  defmethod chamada_do_andar(n_andar);
    lvars n_andar;
    [ ^^fila_de_chamadas ^n_andar ]
       -> fila_de_chamadas;
    pr('Adicionei uma chamada vinda do andar '
       >< n_andar >< '.\n');
  enddefmethod;
     ...
     ...
endflavour;
```

Complete a definição da classe "elevador" acima para que instâncias desta classe respondam às mensagens abaixo, conforme especificado.

```
: vars unidade_1;
: make_instance([ elevador andar 0
    fila_de_chamadas [] ]) -> unidade_1;
: unidade_1 <- chamada_do_andar(3);
Adicionei uma chamada vinda do andar 3.
: unidade_1 <- chamada_do_andar(2);
Adicionei uma chamada vinda do andar 2.
: unidade_1 <- chamada_do_andar(4);
Adicionei uma chamada vinda do andar 4.
: unidade_1 <- atenda_proximas_chamadas;
Atendendo a chamada do andar 3.
Atendendo a chamada do andar 2.
Atendendo a chamada do andar 4.
```

Questão 3 (25 PONTOS)

Considere o programa que segue.

```
flavour motocicleta;
  ivars idade preco quilometragem;
  defmethod mais_um_ano;
    idade + 1 -> idade;
    preco * 0.9 -> preco;
    pr('O preco da moto agora sera de '
       >< preco >< '.\n');
  enddefmethod;
  defmethod valorizacao;
    pr('Motocicletas comuns se desvalorizam
       10% ao ano.');
  enddefmethod;
endflavour;
```

Suponha que uma instância de "moto_de_trilha" seja criada da seguinte maneira:

```
: vars m;
: make_instance([moto_de_trilha idade 8
  preco 9000]) -> m;
```

Complete da forma que você achar melhor a composição de classes do programa para que instâncias de "moto_de_trilha" passem a reagir exatamente de acordo com o que segue:

```
m <- valorizacao;
Motocicletas de trilha se valorizam 10% ao ano.
m <- mais_um_ano;
O preco da moto agora sera de 9900.
```

Questão 4 (25 PONTOS)

Considere o programa acima e responda aos itens (a) e (b) abaixo:

(a) Cite ao menos 2 (duas) TÉCNICAS de programação Orientada a Objetos que foram utilizadas nas Questões 2 e 3 desta prova (justifique sua escolha).

(b) O que seria necessário alterar na definição da classe "moto_de_trilha" para que instâncias desta classe sejam capazes de a responder à mensagem "quilometragem"?

Questão 5 (10 PONTOS)

Com relação a aspectos semânticos ou de pragmática, cite 1 (uma) semelhança e 1 (uma) diferença entre os paradigmas de programação Orientada a Objetos e de programação em Lógica.

BOA SORTE!!!