Exercícios

Porto Alegre, maio de 2008.

Florianópolis, outubro de 2012.

Prof. Eduardo Augusto Bezerra, eduardo.bezerra @ eel.ufsc.br

UFSC / CTC / EEL

Classes abstratas, funções virtuais e friends

Exercícios da página 222 do livro "Treinamento em Linguagem C++, Módulo 2", Victorine Viviane Mizrahi, 2a edição, Pearson/Prentice-Hall, 2006.

1. Quais das instruções são válidas para as seguintes declarações:

   ```
   class Base { ... };
   class Derivada : public Base { ... };
   Base bas, * pbas;
   Derivada dv, * pdv;
   ```

   a) pdv = &bas;
   b) pbas = &pdv;
   c) bas = &dv;
   d) dv = bas;

2. Quais das seguintes afirmações estão corretas quando é executada a chamada a um método por meio de um ponteiro para um objeto?

   a) se o método é virtual, a instrução é resolvida levando em conta o tipo do objeto contido no ponteiro;
   b) se o método é não virtual, a instrução é resolvida levando em conta o tipo do ponteiro;
   c) se o método é virtual, a instrução é resolvida após o início da execução do programa;
   d) se o método é não virtual, a instrução é resolvida na compilação do programa.

3. Explique a diferença entre a sobrecarga de funções membros virtuais e não virtuais.

4. Considerando as seguintes declarações:

   ```
   class Base {
   public:
   void xyz() { ... }
   };

   class Derivada : public Base {
   public:
   void xyz() { ... }
   };

   Derivada dv;
   Base *pbas = &dv;
   ```

   a) a instrução *pbas->xyz();* executará a versão de xyz() método de qual classe?

b) se xyz() for declarada virtual, a instrução pbas->xyz(); executará a versão de xyz() método de qual classe?

5. Escreva a declaração do método virtual *xyz()* de tipo *void* e que recebe um argumento do tipo *int*.

6. Resolução dinâmica é o processo de:

   a) associar chamadas a funções e endereços fixos;
   b) associar uma instrução a uma função no momento de sua execução;
   c) criação de funções virtuais;
   d) criação da tabela *v-table*.

7. Uma função virtual pura é uma função que:

   a) não retorna nada;
   b) é parte da classe derivada;
   c) não recebe argumentos;
   d) não tem corpo.

8. Escreva a declaração da função virtual pura chamada *xyz()* que retorna um *float* e recebe dois *ints* como argumento.

9. As classes abstratas:

   a) existem somente para derivação;
   b) contêm métodos para derivação;
   c) não têm corpo de código;
   d) contêm métodos virtuais puros.

10. Quais dos seguintes processos são permitidos com classes abstratas?

    a) declarar objetos;
    b) retornar um objeto de uma função;
    c) enviar um objeto como argumento para uma função;
    d) declarar ponteiros.

11. A classe-base virtual é usada quando:

    a) diferentes métodos nas classes-base e derivada têm o mesmo nome;
    b) uma classe-base aparece mais de uma vez no processo de herança múltipla;
    c) há múltiplos caminhos de uma classe derivada para outra;
    d) a identificação da função da classe-base é ambígua.

12. Uma classe-base é virtual quando:

    a) a palavra *virtual* é colocada na sua declaração;
    b) contém um método virtual;
    c) é especificada virtual na definição da classe derivada;
    d) contém uma função virtual pura.

13. Escreva uma classe-base denominada *Animal* e as classes *Vaca* e *Bufala* derivadas desta. Em seguida, escreva a classe *Terneiro* derivada de *Vaca* e *Bufala*.

14. Verdadeiro ou Falso: todo método pode ser declarado virtual, mesmo sendo um construtor ou um destrutor.

15. Verdadeiro ou Falso: uma função amiga pode acessar os membros privados da classe de quem é amiga, não sendo um método dessa classe.

16. Uma função amiga pode ser usada para:

    a) impedir heranças entre classes;
    b) permitir o acesso a classes das quais não temos acesso ao código fonte;
    c) permitir a uma classe o acesso às classes não documentadas;
    d) aumentar a versatilidade de um operador sobrecarregado.

17. Escreva a declaração da função amiga de *xyz()* de tipo *void* e que recebe um argumento da classe *Yvone*.

18. A palavra-chave *friend* é colocada:

    a) na classe que permite o acesso de outra classe;
    b) na classe que deseja acessar outra classe;
    c) na parte privada da classe;
    d) na parte pública da classe.

19. Escreva uma declaração que torna os métodos da classe *Pedro* funções amigas da classe *Maria*. Indique em que parte do programa essa declaração deverá ser escrita.

20. Defina uma classe denominada *Fracao*. Esta deve armazenar o numerador e o denominador de uma fração em duas variáveis inteiras. Inclua as operações de adição, subtração, multiplicação e divisão entre objetos e entre um número inteiro e um objeto.