

Universidade Federal do Vale do São Francisco – UNIVASF

Curso: Engenharia da Computação

Disciplina: Programação Orientada a Objetos

Professor: Leonardo Barreto Campos

Lista de Exercícios - III

1. Para que um membro de uma classe-base possam ser acessados por membros da classe derivada, eles devem ser:

- a.** public **b.** protected **c.** private **d.** todas as anteriores

2. Qual a diferença entre derivação pública, privada e protegida?

3. Considere que a classe A seja derivada pública da classe B. Um objeto da classe A pode acessar:

- a.** membros públicos de A **b.** membros protegidos de A **c.** membros privados de A
d. membros públicos de B **e.** membros protegidos de B **f.** membros privados de B

4. Considere que a classe A seja derivada privada da classe B. Um objeto da classe A pode acessar:

- a.** membros públicos de A **b.** membros protegidos de A **c.** membros privados de A
d. membros públicos de B **e.** membros protegidos de B **f.** membros privados de B

5. Considere que a classe A seja derivada protegida da classe B. Um objeto da classe A pode acessar:

- a.** membros públicos de A **b.** membros protegidos de A **c.** membros privados de A
d. membros públicos de B **e.** membros protegidos de B **f.** membros privados de B

6. Os membros de uma classe-base podem acessar:

- a.** membros públicos da classe derivada;
b. membros protegidos da classe derivada;
c. membros privados da classe derivada;
d. nenhuma das anteriores

7. Crie uma classe chamada **Empresa** capaz de armazenar os dados de uma empresa (Nome, Endereço, Cidade, Estado, CEP e Fone). Inclua um construtor sem argumentos e um que receba os dados como argumentos e os inicialize. Escreva duas funções, uma para fazer a interface com o usuário da entrada de dados, **Get()**, e outra para imprimir os dados, **Print()**.

8. Use a classe **Empresa** como base para criar a classe **Restaurante**. Inclua o tipo de comida, o preço médio de um prato, duas funções construtoras, a interface de entrada de dados, **Get()**, e a função que imprima os dados, **Print()**. Construa um programa para testar a classe **Restaurante**.

9. Escreva um programa para armazenar dados de veículos. Primeiramente, crie a classe **Motor** que contém **NumCilindro (int)** e **Potenci(int)**. Inclua um construtor sem argumentos que inicialize os dados com zeros e um que inicialize os dados com os valores recebidos como argumento. Acrescente duas funções, uma para a entrada de dados, **Get()**, e uma que imprima os dados, **Print()**.

10. Escreva a classe **Veiculo** contendo **Peso** em quilos (**int**), **VelocMax** em Km/h (**int**) e **Preco** em R\$ (**float**). Inclua um construtor sem argumentos que inicialize os dados com os valores recebidos como argumento. Acrescente duas funções, uma para a entrada de dados, **Get()**, e uma que imprima os dados, **Print()**.

11. Crie uma classe **CarroPasseio** usando as classes **Motor** e **Veículo** como base. Inclua **Cor (string)** e **Modelo (string)**. Inclua um construtor sem argumentos que inicialize os dados com zeros e uma que inicialize os dados com os valores recebidos como argumentos. Acrescente duas funções, uma para a entrada de dados, **Get()**, e uma que imprima os dados, **Print()**.

12. Crie uma classe **Caminhao** derivada das classes **Motor** e **Veiculo**. Inclua **Toneladas** (carga máxima), **AlturaMax (int)** e **Comprimento (int)**. Inclua um construtor sem argumentos que inicialize os dados com zeros e um que inicialize com os valores recebidos como argumento. Acrescente duas funções, uma para a entrada de dados, **Get()**, e uma que imprima os dados, **Print()**.