

Universidade Federal do Vale do São Francisco – UNIVASF

Curso: Engenharia da Computação

Disciplina: Programação Orientada a Objetos

Professor: Leonardo Barreto Campos

Lista de Exercícios - IV

1. Uma função virtual pura é uma função virtual que:
 - a. que não retorna nada
 - b. é parte da classe derivada
 - c. não recebe argumentos
 - d. não tem corpo

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

3. As classe abastratas:
 - a. existem somente para derivação
 - b. contém métodos virtuais
 - c. não tem corpo de código
 - d. contém métodos virtuais puros

4. Quais dos seguintes processos são permitidos com classe abstratas?
 - a. declarar objetos
 - b. retornar um objeto de uma função
 - c. enviar um objeto como argumento para uma função
 - d. declarar ponteiros

5. A classe-base virtual é usada quando:
 - a. diferentes métodos nas classes-base e derivada tem o mesmo nome
 - b. uma classe-base aparece mais de uma vez no processo de herança múltipla
 - c. a múltiplos caminhos de uma classe derivada para outra
 - d. a identificação da função da classe-base é ambígua

6. 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
7. Uma função amiga pode ser usada para:
- a. Impedir heranças entre classes
 - b. Permitir acesso a classes das quais não temos acesso ao código-fonte
 - c. Permitir a uma classe o acesso as classes não documentadas
 - d. Aumentar a versatilidade de um operador sobrecarregado
8. Escreva uma classe-base denominada **Animal** e as classes **Vaca** e **Bufalo** derivadas desta. Em seguida, escreva a classe **Bezerro** derivada de **Vaca** e de **Bufalo**.
9. Escreva a classe **Veiculo** contendo **Peso** em quilos (**int**), **VelocMax** em Km/h (**int**) e **Preco** em R\$ (**float**). Inclua um construtor sem argumentos inicialize os dados com zeros e um construtor que inicialize os dados com os valores recebidos como argumentos. Acrescente o método para a entrada de dados **Get()** e um segundo método que imprima os dados **Print()**.
10. Em seguida, crie a classe **CarroPasseio** usando a classe **Veiculo** como base. Inclua cor (**string**) e **Modelo** (**string**). Inclua um construtor sem argumentos que inicialize os dados com zeros e um construtor que inicialize os dados com os valores recebidos como argumentos. Acrescente o método para a entrada de dados **Get()** e um segundo método que imprima os dados **Print()**.
11. Agora, crie a base **Caminhao** derivada da base **Veiculo** inclua **CargaMaxima** em **Toneladas** (**float**), **AlturaMax** (**int**) e **Comprimento** (**int**). Inclua um construtor sem argumentos que inicialize os dados com zeros e um construtor que inicialize os dados com os valores recebidos como argumentos. Acrescente o método para a entrada de dados **Get()** e um segundo método que imprima os dados **Print()**.
12. Faça um programa que declare uma matriz de ponteiros do tipo **Veiculo**. Inclua um laço que pergunte ao usuário sobre o tipo de veículo a ser cadastrado e use o operador new para

criar os objetos conforme o tipo escolhido (**CarroPasseio** ou **Caminhao**). Quando o usuário terminar a entrada dos dados de todos os veículos, imprima os resultados por meio de outro laço.

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 argumnto. 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 constritor sem argumentos que inicialize os dados com os valores recebidos como argumento. Acrescente duas funções, uma para a entrada de daods, **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 pára 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()**.