

FEMA-IMESA**Curso:** Bacharelado em Ciência da Computação**Coordenadoria:** Informática**Disciplina:** Linguagem de Programação**Professor Responsável:** Dr. Almir R. Camolesi**Lista de Exercícios:** Implementações de Classes

1) Desenvolva utilizando os conceitos de C++, as seguintes classes

O mercado mundial de bebidas tem crescido fortemente em relação a vinhos. Existem pesquisas que o consumo de um bom vinho, consumido em doses adequadas faz bem a saúde. Sabe-se que o sabor de um vinho é constituído pelo sabor da uva que é utilizada em sua fabricação. A Federação dos Produtores de Vinho do Brasil solicitou que sejam elaboradas classes em C++ para controlar os vinhos produzidos, o País de Origem e as respectivas uvas utilizadas.

Classe Uva private: cod: inteiro; nome: string; cor: string; paisOrigem: Pais; público: Uva(); Uva (int, string, string, string, string);	Classe Vinho: herda Uva privado: cod: inteiro; fabricante: Fabricante nome: string; valor: real; público: Vinho();
---	--

Utilizando-se dos conceitos de orientação a objetos apresentados em aula, etc.. Com base nas definições acima, desenvolva as classes Fabricante, País, Uva e Vinho. Para cada Classe desenvolva:

- Métodos construtores: padrão e com argumentos;
- Modificadores (get e set)
- Métodos polimórficos para alterar o valor de um vinho. Um método deve receber o valor do percentual de aumento e realizar tal tarefa. O segundo método deve assumir um aumento de 5% no valor, caso não seja informado nenhum parâmetro;
- função *main* () para demonstrar a utilização das classes e métodos definidos.

Obs: ao desenvolver as classes utilize os conceitos de métodos in line e proteção de atributos e variáveis const.

2) Implemente todas as funções membros das seguintes classes:

<pre>a) Classe Pessoas class Pessoa { public: Pessoa (); Pessoa (string nome, int idade); void lerPessoa(); void impPessoa(); string retNome () const; int retIdade() const; void setNome (string); void setIdade(int); private: string nome; int idade; };</pre>	<pre>b) Classe Funcionário class Funcionario { public: Funcionario (); Funcionario (string funcNome, float funSal): void lerFunc (); void impFunc(); void setSalario (float novoSal); void setNomeFunc(); float ret_Salario () const; string get_NomeFunc () const; private: Pessoa dadosFunc; float salario; };</pre>
---	--

- 3) Implemente uma classe *Address* (*Endereço*). Um endereço possui uma rua, um número e um número de apartamento opcional, uma cidade, um estado e um código postal. Forneça dois construtores um com número de apartamento e outro sem. Forneça um método *print* que imprima o endereço com a rua em uma linha e a cidade, o estado e o código postal na linha seguinte. Forneça uma função no programa principal, *comes_before* (*vem_antes*) que recebe por parâmetro dois objetos *Address*, e verifica qual endereço vem primeiro, ou seja, qual possui código postal menor. Utilize passagem de objetos por ponteiro e constantes para proteger os objetos passados para a função *comes_before*.
- 4) Implemente uma classe *Account* (*Conta*). Uma conta possui um saldo, funções que depositam e retiram dinheiro e uma função para informar o saldo atual. A função retira dinheiro deve cobrar uma multa de R\$5,00, caso seja feita uma tentativa de retirar mais dinheiro do que possível na conta.
- 5) O que são funções membros, e como ele difere de uma função não membro?
- 6) O que acontece se você esquecer de usar *const* em um método de acesso? O que acontece se você acidentalmente coloca *const* em uma função modificadora?
- 7) O quê é um construtor? O que é um construtor default? Qual é a consequência se uma classe não possui um construtor default?
- 8) O que é um método *in line*? Dê exemplos.
- 9) Demonstre o procedimento para iniciação de um construtor *in line*.

- 10) O que é a interface de uma classe? O que é a implementação de uma classe? (1,5)
- 11) Você pode ler e escrever o atributo *salário* da classe *Empregado* com o método de acesso *ObeterSalario()* e o método de alteração de dados *AlteraSalario()*. Cada atributo de uma classe deve ter métodos de acesso e de alteração a dados. Explique por quê sim e por quê não?
- 12) Campos de dados são ocultos na seção privativa de uma classe, mas eles não estão muito bem escondidos. Qualquer um pode ler os dados da seção privativa. Explique até que ponto a palavra chave *private* esconde os atributos privativos de uma classe.
- 13) O que é sobrecarga de operadores? Qual a importância deste conceito relacionado à programação orientada a objetos? Dê exemplos.
- 14) Defina uma classe *Produto*. Um produto possui Código, descrição, valor e fornecedor. O atributo fornecedor pertence à classe *Fornecedor* que possui os seguintes atributos: código, nome, endereço e telefone. Suponha que a classe *fornecedor* já esteja implementada, implemente apenas a classe *Produto*
- Forneça um construtor para iniciar os valores numéricos com zero e os textuais com "".
 - Forneça uma função *imprime* que imprima os dados de um produto e o nome e o endereço de um fornecedor.
 - Forneça uma função que realize a leitura dos atributos da classe *Produto*;
 - Projete dois métodos para atualizar valor do produto: um método que ao ser chamado soma automaticamente 20% ao valor do produto e um outro método que receba por parâmetro o percentual de aumento.
 - Desenvolva um programa principal para demonstrar a utilização da classe.
- 15) O que são construtores e destrutores? Qual a importância dos mesmos para o estilo de programação orientado a objetos?
- 16) Em relação a objetos o que é escopo e tempo de vida? Dê exemplos.

Boa Diversão!!!!