

Curso Rápido de C++ – Lista de Exercícios

1) Crie uma classe que representa um ponto no plano cartesiano. Em seguida, crie uma classe que representa um triângulo, reusando a classe anterior por composição. Finalmente, escreva um programa que receba do usuário as coordenadas dos vértices do triângulo e imprima seu perímetro.

2) Crie uma classe que representa um funcionário, registrando seu nome, salário e data de admissão. Em seguida, crie uma classe que represente um departamento de uma empresa, registrando o nome e os funcionários que nele trabalham (para uso de vetores, considere um máximo de 100 funcionários). Por fim, crie uma classe que representa uma empresa, registrando seu nome, CNPJ e departamentos (considere um máximo de 10 departamentos). Faça um programa que:

- Crie uma empresa;
- Adicione a esta empresa alguns departamentos;
- Adicione aos departamentos alguns funcionários;
- Dê aumento de 10% a todos os funcionários de um determinado departamento;
- Transfira um funcionário de um departamento para outro.

É esperado que seu código seja bem encapsulado. Por exemplo, para adicionar um departamento em uma empresa (ou um funcionário a um departamento), não se deve acessar o vetor (ou lista) de departamentos diretamente, mas sim ter um método na classe que representa a empresa para adicionar um departamento.

3) Defina uma classe base pessoa que contenha informação de propósito geral comum a todas as pessoas, como por exemplo o nome, a idade e a altura. Esta classe deverá conter:

- um construtor por default, que inicializa os atributos com "Indefenido", 0 e 0.0;
- um construtor alternativo (que receba dados para os atributos da classe);
- o destruidor (vazio);
- métodos públicos setter/getter para acesso aos membros dados privados.

Escreva um programa que exemplifique a utilização da classe criada no ponto anterior (main.cpp), por exemplo, um cadastro de pessoas com operações de cadastrar e listar.

4) Utilizando a classe base Pessoa criada no exercício anterior, projete a classe derivada Professor, mediante implementação de um mecanismo de herança pública. A classe deve obedecer aos seguintes requisitos:

- não permitir invocar o construtor por default da classe derivada;
- construir o objeto fornecendo apenas o nome da disciplina que leciona o professor;
- construir o objeto recorrendo a um construtor inicializador em herança, que permita definir os membros dados da classe base.

Escreva um programa que exemplifique a utilização das classes criadas no ponto anterior, não esquecendo de deixar clara a utilização dos vários construtores, bem como o acesso aos membros públicos da classe base.

5) Crie uma classe para representar uma conta corrente, com métodos para depositar uma quantia, sacar uma quantia e obter o saldo. Para cada saque será debitada também uma taxa de operação equivalente à 0,5% do valor sacado. Crie, em seguida, uma subclasse desta classe anterior para representar uma conta corrente de um cliente especial. Clientes especiais pagam taxas de operação de apenas 0,1% do valor sacado. Faça testes com as duas classes e verifique seus resultados.



6a) Crie uma hierarquia de classes de domínio para uma loja que venda livros, CDs e DVDs. Sobrescreva o operador `<<` para que imprima:

- Para livros: nome, preço e autor;
- Para CDs: nome, preço e número de faixas;
- Para DVDs: nome, preço e duração.

Evite ao máximo repetição de código no construtor e no operador sobrescrito, acessando a implementação da superclasse. Em seguida, crie uma classe `Loja` com o método `main()` que adicione 5 produtos diferentes (a sua escolha) a um vetor e, por fim, imprima o conteúdo do vetor.

6b) Modifique o código do programa anterior, da seguinte forma:

- a) Adicione um atributo que represente o código de barras do produto (é um valor obrigatório e, portanto, deve ser pedido no construtor);
- b) Sobrescreva o operador `==` retornando `true` se dois produtos possuem o mesmo código de barras;
- c) Na classe `Loja`, implemente um simples procedimento de busca que, dado um produto e um vetor de produtos, indique em que posição do vetor se encontra o produto especificado ou imprima que o mesmo não foi encontrado;
- d) No método `Loja.main()`, após a impressão do vetor (feita na questão 2a), escolha um dos 5 produtos e crie duas novas instâncias idênticas a ele: uma com o mesmo código de barras e outra com o código diferente. Efetue a busca deste produto no vetor utilizando as duas instâncias e verifique o resultado.

7) Crie a seguinte hierarquia de classes:

- Uma interface para representar qualquer forma geométrica, definindo métodos para cálculo do perímetro e cálculo da área da forma;
- Uma classe abstrata para representar quadriláteros. Seu construtor deve receber os tamanhos dos 4 lados e o método de cálculo do perímetro já pode ser implementado;
- Classes para representar retângulos e quadrados. A primeira deve receber o tamanho da base e da altura no construtor, enquanto a segunda deve receber apenas o tamanho do lado;
- Uma classe para representar um círculo. Seu construtor deve receber o tamanho do raio.

No programa principal, pergunte ao usuário quantas formas ele deseja criar. Em seguida, para cada forma, pergunte se deseja criar um quadrado, um retângulo ou um círculo, solicitando os dados necessários para criar a forma. Todas as formas criadas devem ser armazenadas em um vetor. Finalmente, imprima: (a) os dados (lados ou raio); (b) os perímetros; e (c) as áreas de todas as formas.