

Universidade Federal da Paraíba Centro de Informática Departamento de Informática	Laboratório de LP1 Semestre: 2019.2 Professor: Derzu Omaia
---	--

ROTEIRO 2

ASSUNTO: POO – Herança e Composição

Obs: As classes do seu projeto devem ser implementadas separando a sua interface (arquivo de cabeçalho .h) da implementação da classe (arquivo de programa .cpp).

1) Crie um programa em C++ que represente algumas informações para gerenciamento de um restaurante. Esse programa deve conter as classes `Pedido`, `MesaDeRestaurante` e `RestauranteCaseiro`. Algumas recomendações sobre a criação dessas classes são:

- A classe `Pedido` representa o pedido de um único item do cardápio, e contém os seguintes atributos: o número, a descrição, a quantidade comprada de um item e o preço do item;
- A classe `MesaDeRestaurante` deve conter um atributo para representar uma lista (vetor) de instâncias da classe `Pedido`, representando os pedidos feitos na mesa. Ela deve conter também um método `adicionaAoPedido()` que adiciona um novo pedido ao conjunto de pedidos feitos. Se o pedido já existir a quantidade do pedido deve ser adicionada a um pedido já existente na mesa. Se o pedido ainda não existir o pedido é adicionado.
- O método `zeraPedidos()` que cancela todos os pedidos feitos, isto é, faz com que a quantidade de pedidos seja zero para cada item; e o método `calculaTotal()`, que calcula o total a ser pago por aquela mesa.
- A classe `RestauranteCaseiro` deve ter um atributo que é uma lista (vetor) instâncias da classe `MesaDeRestaurante`, para representar suas mesas separadamente.
- A classe `RestauranteCaseiro` também deve ter um método `adicionaAoPedido()` que adicionará um pedido a uma mesa. Esse método deverá chamar o método `adicionaAoPedido()` da mesa à qual o pedido está sendo adicionado.
- A classe `RestauranteCaseiro` também deve conter um método `calculaTotalRestaurante()` que faz o cálculo do total arrecado no restaurante, somando os pedidos realizados em todas as mesas.

Obs: Você pode usar arrays e criar um número predeterminado e imutável de instâncias de `Pedidos` e de `MesaDeRestaurante`.

2) Considere um sistema que possui uma classe Pessoa, uma classe Endereco.

(a) Implemente a classe Endereco, com os atributos rua (String), número, bairro, cidade, estado e CEP (String). Crie um construtor que receba os parâmetro referentes aos atributos. Crie o método toString() que retorna todos os atributos da classe Endereco na forma de string.

(b) Implemente a classe Pessoa que possui os atributos nome (String), endereço (do tipo Endereco), e telefone (do tipo String). Considere também que ela possui dois construtores (um que só recebe o nome e outro que recebe nome, endereço e telefone) e métodos para obter e alterar esses atributos (métodos get e set).

(c) Em seguida, crie um programa com o método main que cria dois objetos do tipo Pessoa e configura os valores de nome, endereço e telefone desses objetos.

3) (a) Crie uma classe Funcionario com os atributos matricula, nome e salário, e os métodos get() e set() de cada atributo. Em seguida, crie uma classe Consultor que herda da classe Funcionario e sobrescreve o método getSalario(), adicionando um percentual de 10% no valor do salário. Implemente também um método getSalario(float percentual), onde o parâmetro "float percentual" determina o percentual a ser acrescido no salário de Consultor.

(b) Crie um programa (função main) para testar as classes Funcionario e Consultor. O programa deve criar um objeto da classe Funcionario e Consultor e testar seus métodos.

4) Considere um programa que lida com Figuras Geométricas. Cada figura tem um nome e é capaz de calcular sua área. Para desenvolver esse programa, utilizou-se uma classe chamada FiguraGeometrica, com o atributo nome e com o método calcularArea(). Implemente essa classe e também algumas classes que herdem de FiguraGeometrica (Triangulo, Quadrado e Circulo, etc). Em seguida, crie um programa (classe) principal para testar essas classes.

5) Crie uma classe Trabalhador e classes derivadas (herança) TrabalhadorPorHora e TrabalhadorAssalariado. Cada trabalhador tem um atributo nome e salário. TrabalhadorPorHora possui um atributo valorDaHora e seu salário é calculado semanalmente. O salário do TrabalhadorAssalariado é referente ao seu salário mensal. No construtor inicie os atributos de cada classe. Para TrabalhadorPorHora inicie o salario mensal com zero. Ele só vai ser alterado pelo método calcularPagamento Semanal().

- Escreva um método float calcularPagamentoSemanal(int horasSemanais) que calcule o pagamento semanal de cada trabalhador e retorna o valor. Neste método calcule também o salário mensal estimado do TrabalhadorPorHora, multiplicando o semanal por 4. No TrabalhadorAssalariado pagamento semanal é o valor do salario mensal dividido por 4.
- O trabalhador assalariado é pago pela carga horária de 40 horas, independentemente de qual seja o número real de horas trabalhadas.

- O trabalhador que ganha por hora é pago, obviamente, de acordo com o número real de horas trabalhadas, sendo horas, no máximo, igual a 40. Se ele trabalhou mais de 40 horas, cada hora excedente é paga como uma hora e meia.

(b) Crie um programa (classe) principal para testar as classes criadas.