

Lista de Exercícios 10

Informações sobre cópias

As questões são individuais. Em caso de cópias de trabalho a pontuação será zero para os autores originais e copiadores. Não serão aceitas justificativas como: “Fizemos o trabalho juntos, por isso estão idênticos”.

Parte A – Programação Orientada a Objetos - Exercícios para serem entregues

Resolva os exercícios a seguir e entregue pelo CANVAS. Cada exercício deve conter um arquivo no **formato .CPP**.

A01. Escreva um programa para cadastrar clientes de uma loja. As informações necessárias são: nome, data de nascimento, endereço e telefone. Devem ser usadas classes para a construção deste cadastro.

Utilize os conceitos de classe, atributos e método construtor. Crie o main, que declara um vetor de clientes e que permite ao usuário adicionar os clientes nesse vetor. O tamanho do vetor deverá ser definido pelo usuário.

A02. João gostaria de desenvolver um sistema de comparação de preços de um tipo específico de eletrodoméstico. Na primeira fase do seu software, gostaria de criar uma classe que contenha o nome da loja, telefone e preço de um eletrodoméstico. Em seguida, irá cadastrar um total de 15 registros e exibir estatísticas desses itens, de modo que possa verificar o melhor preço (menor valor), o preço médio e o preço máximo. Auxilie João a desenvolver esta classe e exiba as informações desejadas.

A03. 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 duas contas e efetue depósito e saque, em uma faça um depósito e na outra um saque.

A04. Crie uma classe para representar uma pessoa, com os atributos privados de nome, idade e altura. Crie os métodos públicos necessários para sets e gets e também um método para imprimir

os dados de uma pessoa. Ilustre o seu funcionamento em um programa *main criando duas pessoas e imprima seus dados*.

A05. Crie uma classe denominada Elevador para armazenar as informações de um elevador dentro de um prédio. A classe deve armazenar o andar atual (0=térreo), total de andares no prédio, excluindo o térreo, capacidade do elevador (em pessoas) e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:

- Inicializa: que deve receber como parâmetros: a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazios);
- Entra: para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
- Sai: para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
- Sobe: para subir um andar (não deve subir se já estiver no último andar);
- Desce: para descer um andar (não deve descer se já estiver no térreo);
- Getters: métodos para obter cada um dos os dados armazenados.