



PCS 3111 - Laboratório de Programação Orientada a Objetos para Engenharia Elétrica

2023

Aula 03 – Conceitos Básicos de OO

Atenção

- As definições das classes usadas nos exercícios encontram-se **disponíveis no e-Disciplinas**. Use o código fornecido.
- Os nomes, os atributos, os métodos, e as respectivas assinaturas das classes dadas **devem seguir o especificado** em cada exercício para fins de correção automática.
- A **ordem de declaração** de atributos e métodos fornecidos **não deve ser alterada**. Caso contrário, poderá haver redução automática da nota.
- A função `main` **não deve ser submetida**. Caso contrário, a correção automática retornará um *Compilation Error*.

Exercício 01

Usando a definição fornecida e a especificação a seguir, implemente a classe Produto:

```
class Produto {  
public:  
    string nome = "";  
    double preco = 0.0;  
    double desconto = 0.0;  
    double calcularValorDeVenda();  
    void imprimir();  
};
```

- Um Produto possui, como atributos, um nome, um preço (em reais) e o seu desconto em porcentagem (desconto = 0.1 significa 10% de desconto);
- O valor do desconto será usado para o cálculo do valor de venda, como será explicado a seguir.
- O método `calcularValorDeVenda()` deve retornar o valor do Produto corrigido de acordo com o seu respectivo desconto. Exemplo:
 Se o Produto tem `preco = 10` e `desconto = 0.1`
 `calcularValorDeVenda()` devolve o valor de 9.0 (em double)
- O método `imprimir` deve exibir na tela (usando o `cout`) os dados de cada Produto no seguinte formato:



Produto: <nome> custa <valor de venda>

- Sendo <...> o valor armazenado em cada variável. Por exemplo, para um produto com nome Carregador com preço de 100 reais e com 10% de desconto, a saída seria (pule uma linha ao final):

Produto: Carregador custa 90

- Implemente a função **teste1** declarada no código fornecido seguindo os próximos passos:
 - Crie um produto com nome “**TV**” com preço igual a **1000**;
 - Guarde 0.20 em **desconto** (20%);
 - Chame o método **imprimir**.

Com esses parâmetros, a saída deve ser:

Produto: TV custa 800

- Teste a classe antes de enviar, utilizando a main!
- Não se esqueça de comentar a main ao submeter.

Exercício 02

Usando a definição fornecida e a especificação a seguir, implemente a classe Pedido:

```
class Pedido {
public:
    Produto *produto1 = nullptr;
    Produto *produto2 = nullptr;
    int quantidade1 = 0;
    int quantidade2 = 0;

    bool adicionar(Produto *produto, int quantidade);
    double calcularPrecoTotal();
    void imprimir();
};
```

- Um Pedido é constituído por, no máximo, dois produtos. O atributo quantidade1 e quantidade2 representa a quantidade de cada produto (produto1 e produto2, respectivamente).



- O método `calcularPrecoTotal()` retorna o valor total do pedido (considere os preços e as quantidades). Caso não existam produtos no Pedido, este deve retornar 0;
- O método `adicionar` armazena um produto em `produto1` ou `produto2` (nesta ordem) e retorna `true` caso tenha adicionado. Confira se o produto é `nullptr` para saber em qual das variáveis deve-se armazenar o produto (se `produto1 == nullptr`, é em `produto1`; senão se `produto2 == nullptr`, é em `produto2` senão o pedido está cheio). Caso o produto já tenha sido adicionado ou não seja possível adicionar mais produtos (já existem dois produtos no Pedido), deve-se retornar `false`.
 - Ao testar seu programa, **não** atribua produtos diretamente pelos atributos `produto1` e `produto2`. Use sempre o método `adicionar`.
- O método `imprimir` deve exibir na tela (usando o `cout`) os dados de cada Produto no seguinte formato:

Pedido: Preco total: <Preco Total>
- Implemente a função `teste2()` declarada no código fornecido seguindo os próximos passos:
 - Copie o que foi feito em `teste1()`
 - Crie um produto com nome “**Suporte para TV**” com preço igual a 150 e desconto 0.05 (5%);
 - Chame o método `imprimir` para o “**Suporte para TV**”.
 - Crie um Pedido
 - Adicione dois produtos ao Pedido.
 - **TV** com 1 unidade
 - **Suporte para TV** com 2 unidades
 - Chame o método `imprimir` para Pedido

A saída para o `teste2` deve ser:

Produto: TV custa 800
Produto: Suporte para TV custa 142.5
Pedido: Preco total: 1085

- Teste as classes antes de enviar, utilizando a `main`!
- Não se esqueça de comentar o `main` ao submeter no *judge*.



Testes do Judge

Exercício 1

- Preço dos produtos sem desconto
- Preço dos produtos com desconto
- Teste função teste1

Exercício 2

- Adicionar uma unidade de um produto com pedido vazio sem desconto
- Adicionar várias unidades de um produto com pedido vazio sem desconto
- Adicionar uma unidade de dois produtos com pedido vazio sem desconto
- Adicionar várias unidades de dois produtos com pedido vazio com desconto
- Adicionar mais de dois itens
- Teste função teste2