



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

2023

Aula 05 – Construtores e Destrutores

Atenção

- Código inicial a ser usado na resolução dos exercícios encontra-se **disponível no e-Disciplinas**.
- Submeta um arquivo comprimido (faça um “.zip” – **não pode ser “.rar”**) colocando apenas os arquivos “.cpp” e “.h”. Não crie pastas no “zip”.

Exercício 01

Baseado na implementação da classe Produto fornecida, modifique os arquivos **Produto.h** e **Produto.cpp** de modo a adicionar um construtor na classe. Abaixo está a assinatura do construtor esperada junto aos seus outros métodos públicos:

```
class Produto {  
public:  
    Produto(string nome, double preco);  
  
    string getNome();  
    double getPreco();  
  
    void imprimir();  
};
```

Seu construtor receberá como parâmetros o nome e o preço do Produto e deve armazená-los nos respectivos atributos.

Foi entregue uma classe Item, a qual representa um item de um Pedido, composto por um produto e uma quantidade daquele produto. Por exemplo, um Item pode ser 2 unidades de uma caneta e outro Item 1 unidade de uma borracha. De forma semelhante ao que precisa ser feito com a classe Produto e baseado na implementação da classe Item fornecida, modifique os arquivos **Item.h** e **Item.cpp** de modo a adicionar um construtor na classe. Abaixo está a assinatura do construtor esperada junto aos seus outros métodos públicos:



```
class Item {  
public:  
    Item(Produto* produto, int quantidade);  
  
    Produto* getProduto();  
    int getQuantidade();  
    double calculaPrecoTotal();  
    void setQuantidade(int quantidade);  
  
    void imprimir();  
};
```

Seu construtor receberá como parâmetros o Produto e a quantidade do Produto e deve armazená-los nos respectivos atributos.

Implemente a função teste1 de acordo com os seguintes passos:

1. Crie um Produto de nome *Cueca* e preço 20.89 (representando 20.89 reais);
2. Imprima o produto *Cueca*;
2. Crie um Item com o Produto *Cueca* e quantidade 5;
3. Imprima o item com *Cueca*;
4. Delete o item com *Cueca*;
5. Delete o produto *Cueca*;

Exercício 02

Implemente agora o destrutor da classe Produto, o destrutor da classe Item e o construtor e destrutor da classe Pedido (Pedido é uma classe que agrega vários itens; o método adicionar cria itens e adiciona a um vetor interno). Com a inclusão dos métodos mencionados, os arquivos de cabeçalho apresentarão os seguintes métodos:



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO
PAULO
Departamento de Engenharia de Computação e Sistemas Digitais

```
class Produto {
public:
    Produto(string nome, double preco);
    ~Produto();

    string getNome();
    double getPreco();

    void imprimir();
};
```

```
class Item {
public:
    Item(Produto* produto, int quantidade);
    ~Item();

    Produto* getProduto();
    int getQuantidade();
    double calculaPrecoTotal();
    void setQuantidade(int quantidade);

    void imprimir();
};
```

```
class Pedido {
public:
    Pedido(int quantidadeMaxima);
    ~Pedido();

    Item** getItens();
    int getQuantidadeItens();
    int getQuantidadeMaximaItens();
    bool adicionar(Produto* produto, int quantidade);
    double calculaPrecoTotal();
};
```



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO
PAULO
Departamento de Engenharia de Computação e Sistemas Digitais

Não altere a definição ou a implementação dos demais métodos da classe Pedido (a implementação já é fornecida). Também não altere os atributos. Defina e implemente apenas o construtor e o destrutor.

O construtor de Pedido recebe como parâmetro a quantidade máxima de itens que o pedido pode ter no vetor itens. O valor passado ao construtor deve ser atribuído ao seu respectivo atributo na classe (quantidadeMaxima). Além disso, **o construtor de Pedido deve alocar dinamicamente o vetor itens**, que deve ter espaço o suficiente para armazenar quantidadeMaxima itens. Inicialize o atributo quantidade com 0 no construtor.

A implementação dos destrutores deve atender às seguintes especificações:

- O destrutor de Produto deve imprimir uma mensagem com o seguinte formato, pulando uma linha ao final:

Produto <nome> destruído

Exemplo: Produto Toalha destruído

- O destrutor de Item deve imprimir uma mensagem com o seguinte formato, pulando uma linha ao final:

Item <nome do Produto> destruído

Exemplo: Item Chinelo destruído

Não destrua o **Produto** ao destruir o **Item**.

- O destrutor do Pedido deve imprimir uma mensagem com o seguinte formato, pulando uma linha ao final:

Pedido com <quantidade> item(ns) destruído

Exemplo: Pedido com 12 item(ns) destruído

Após apresentar essa mensagem, todos os itens contidos no Pedido devem ser destruídos.
Destrua o vetor itens assim como os itens contidos nele.

Implemente a função teste2, para tanto considere os seguintes passos:

1. Crie um Produto de nome *Absorvente* e com preço de 10.40 reais;
2. Crie um Produto de nome *Pringles* e com preço de 12.37 reais;
3. Crie um Produto de nome *Cookies* e com preço de 5.39 reais;



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO
PAULO
Departamento de Engenharia de Computação e Sistemas Digitais

4. Adicione, na ordem de criação, os produtos criados com quantidade 1 a um novo Pedido, de quantidadeMaxima 4;
5. Adicione os produtos *Absorvente* e *Cookies*, com quantidades 1 e 2, respectivamente, a um novo Pedido, de quantidadeMaxima 4;
6. Delete o primeiro Pedido.
7. Delete o segundo Pedido.
8. Delete os produtos na ordem em que eles foram criados.

Testes do Judge

Exercício 1

- Teste construtor e getters de Produto
- Teste construtor e getters de Item
- Teste da função teste1.

Exercício 2

- Construtor de Pedido e getItens
- Mensagem do destrutor de Produto
- Mensagem do destrutor de Item
- Mensagem do destrutor de Pedido
- Destrutor de Pedido destruiu itens
- Teste da função teste2.