

Programação Orientada a Objetos

Lista de Exercícios: Construtores, Destrutores, Encapsulamento e Métodos Get e Set em C++

Exercício 1: Criar uma Classe Simples com Construtor Padrão

Objetivo: Implementar uma classe `Produto` com os seguintes requisitos:

- A classe deve possuir os atributos:
 - `nome` (string)
 - `preco` (double)
- Implementar o **construtor padrão** que inicializa `nome` com "Produto Desconhecido" e `preco` com 0.0.
- No `main`, criar dois objetos da classe `Produto` e exibir o nome e o preço de ambos.

Dicas:

- Use o construtor padrão para inicializar os atributos.

Exemplo de saída esperada:

```
Nome: Produto Desconhecido, Preço: 0.00
Nome: Produto Desconhecido, Preço: 0.00
```

Exercício 2: Construtor Parametrizado e Destrutor

Objetivo: Criar uma classe `Carro` com as seguintes características:

- Atributos:
 - `marca` (string)
 - `ano` (inteiro)
- Implementar o **construtor parametrizado** que inicializa `marca` e `ano` com os valores passados como argumentos.
- Implementar um **destrutor** que exibe a mensagem "`Destruindo o carro da marca X`" (onde X é o nome da marca).
- No `main`, criar dois objetos `Carro`, exibindo suas informações e, ao final, observe a execução do destrutor.

Dicas:

- O destrutor será automaticamente chamado quando o objeto sair de escopo.

Exemplo de saída esperada:

```
Marca: Toyota, Ano: 2020
Marca: Honda, Ano: 2022
Destruindo o carro da marca Toyota
Destruindo o carro da marca Honda
```

Exercício 3: Encapsulamento e Validação em Setters

Objetivo: Criar uma classe `Pessoa` que respeite o princípio de **encapsulamento**.

- Atributos privados:
 - `nome` (string)
 - `idade` (inteiro)
- Implemente os **métodos get e set** para os atributos.
 - O **setIdade** deve validar que a idade é sempre positiva.
- No `main`, crie um objeto da classe `Pessoa`, defina o nome e a idade usando os **métodos set**, e exiba as informações da pessoa.

Regras de validação:

- Caso o valor de idade seja negativo, exiba a mensagem: `"Idade inválida!"`.

Exemplo de saída esperada:

```
Nome: João, Idade: 25
Idade inválida!
Nome: João, Idade: 25
```

Exercício 4: Construtores com Parâmetros de Valor Padrão

Objetivo: Implementar a classe `Aluno` para trabalhar com construtores que possuem **valores padrão para os parâmetros**.

- Atributos privados:
 - `nome` (string)
 - `nota` (double)
- O **construtor parametrizado** deve ter o nome com valor padrão `"Aluno Desconhecido"` e a nota com valor padrão `0.0`.
- Implemente os **métodos get e set** para os atributos.
- No `main`, crie três objetos de `Aluno`:
 - Um sem passar argumentos (usa os valores padrão).
 - Um passando apenas o nome.
 - Um passando o nome e a nota.
- Exiba as informações dos três objetos.

Dicas:

- Lembre-se de definir os valores padrão da direita para a esquerda.

Exemplo de saída esperada:

```
Nome: Aluno Desconhecido, Nota: 0.00
Nome: Maria, Nota: 0.00
Nome: João, Nota: 9.50
```

Exercício 5: Classe Completa com Construtor, Destrutor, Encapsulamento e Métodos Get e Set

Objetivo: Criar a classe `ContaBancaria`, implementando **construtores**, **destrutores**, **métodos get e set**, e funções para depósito e saque.

1. Atributos privados:

- `titular` (string)
- `saldo` (double)

2. Implemente os seguintes **métodos públicos**:

- **Construtor parametrizado:** inicializa o nome do titular e o saldo inicial. Se o saldo for negativo, ele é definido como zero.
- **Destrutor:** exibe a mensagem: "`Conta de X encerrada`" (onde X é o nome do titular).
- **getSaldo:** retorna o saldo.
- **setTitular:** define o nome do titular.
- **depositar:** soma o valor passado ao saldo. Valores negativos não devem ser aceitos.
- **sacar:** subtrai o valor do saldo, mas apenas se houver saldo suficiente. Caso contrário, exiba a mensagem "**Saldo insuficiente!**".

3. No `main`, crie uma conta para "Ana" com saldo de 1000. Realize as seguintes operações:

- Deposite 200.
- Saque 500.
- Tente sacar 1000 (deve exibir "**Saldo insuficiente!**").

4. Exiba o saldo final.

Exemplo de saída esperada:

```
Saldo atual: 1000.00
Depósito de 200.00
Saque de 500.00
Saldo insuficiente!
Saldo final: 700.00
Conta de Ana encerrada
```