



Instruções

Implemente cada exercício conforme descrito abaixo. Certifique-se de utilizar o conceito de encapsulamento para proteger os atributos das classes e fornecer métodos adequados para manipulá-los. Adicionalmente, escreva testes de unidade para cada classe utilizando o framework Catch2.

Exercício 1: Classe Produto

Implemente uma classe `Produto` que encapsule os atributos `nome`, `preco` e `quantidade` com métodos `set` e `get`. Adicione métodos para gerenciar o estoque (`adicionarEstoque` e `removerEstoque`).

Requisitos

- Crie um construtor que inicialize os atributos.
- Adicione métodos `set` e `get` para cada atributo.
- Adicione métodos para gerenciar o estoque (`adicionarEstoque` e `removerEstoque`).
- Crie um método `mostrarDados` para exibir as informações do produto.

Código da Classe

Função Main e Saída Esperada

Listing 1 – Main para testar a Classe Produto

```
int main() {  
    Produto p1("Notebook", 2500.0, 10);  
    p1.mostrarDados(); // Nome: Notebook, Preço: 2500, Quantidade: 10  
  
    p1.adicionarEstoque(5);  
    p1.removerEstoque(3);  
    p1.mostrarDados(); // Nome: Notebook, Preço: 2500, Quantidade: 12  
  
    return 0;  
}
```

Exercício 2: Classe Livro

Implemente uma classe `Livro` com os atributos `titulo`, `autor`, `anoPublicacao` e `numPaginas`. Utilize encapsulamento para proteger esses atributos.

Requisitos

- Crie um construtor que inicialize os atributos.
- Adicione métodos `set` e `get` para cada atributo.
- Crie um método `mostrarDados` que exiba as informações do livro.

Código da Classe

Função Main e Saída Esperada

Listing 2 – Main para testar a Classe Livro

```
#include <iostream>

int main() {
    Livro l1("1984", "George Orwell", 1949, 328);
    l1.mostrarDados(); // Titulo: 1984, Autor: George Orwell, Ano: 1949,
                       // Paginas: 328

    Livro l2("Fogo Morto", "Graciliano Ramos", 1933, 229);
    l2.mostrarDados(); // Titulo: Fogo Morto, Autor: Graciliano Ramos,
                       // Ano: 1933, Paginas: 229

    l2.setPaginas(299);
    l2.mostrarDados(); // Titulo: Fogo Morto, Autor: Graciliano Ramos,
                       // Ano: 1933, Paginas: 299

    return 0;
}
```

Exercício 3: Classe ContaBancaria

Implemente uma classe `ContaBancaria` que encapsule os atributos `numeroConta`, `saldo` e `titular`. Inclua métodos para depósito e saque.

Requisitos

- Crie um construtor que inicialize os atributos.
- Adicione métodos `set` e `get` para cada atributo, onde aplicável.
- Crie métodos para `depositar` e `sacar` valores da conta.
- Crie um método `mostrarDados` que exiba as informações da conta.

Código da Classe

Função Main e Saída Esperada

Listing 3 – Main para testar a Classe ContaBancaria

```
#include <iostream>

int main() {
    ContaBancaria c1("12345-6", 1000.0, "Joao Silva");
    c1.mostrarDados(); // Conta: 12345-6, Titular: Joao Silva, Saldo:
                       1000

    c1.depositar(500.0);
    c1.sacar(300.0);
    c1.mostrarDados(); // Conta: 12345-6, Titular: Joao Silva, Saldo:
                       1200

    return 0;
}
```

Exercício 4: Classe Carro

Implemente uma classe **Carro** com os atributos **marca**, **modelo**, **ano** e **quilometragem**. Utilize encapsulamento para proteger esses atributos.

Requisitos

- Crie um construtor que inicialize os atributos.
- Adicione métodos `set` e `get` para cada atributo.
- Adicione métodos para incrementar a quilometragem (`dirigir`).
- Crie um método `mostrarDados` que exiba as informações do carro.

Código da Classe

Função Main e Saída Esperada

Listing 4 – Main para testar a Classe Carro

[illegible]

Exercício 5: Classe Aluno

Implemente uma classe `Aluno` com os atributos `nome`, `matricula`, `curso` e `nota`. Utilize encapsulamento para proteger esses atributos.

Requisitos

- Crie um construtor que inicialize os atributos.
- Adicione métodos `set` e `get` para cada atributo.
- Adicione métodos para calcular a média das notas e verificar a aprovação do aluno (`calcularMedia` e `verificarAprovacao`).
- Crie um método `mostrarDados` que exiba as informações do aluno.

Código da Classe

Função Main e Saída Esperada

Listing 5 – Main para testar a Classe Aluno

```
#include <iostream>

int main() {
    Aluno a1("Maria Silva", 202345, "Engenharia", 7.5);
    a1.mostrarDados(); // Nome: Maria Silva, Matricula: 202345, Curso:
                       Engenharia, Nota: 7.5

    std::cout << "Media: " << a1.calcularMedia() << std::endl; // Media:
                       7.5
    std::cout << "Aprovado: " << (a1.verificarAprovacao() ? "Sim" : "
    N o ") << std::endl; // Aprovado: Sim

    return 0;
}
```