

LISTA 2 DE EXERCÍCIOS – POO

1. HERANÇA MÚLTIPLA

- 1.1. O que é herança múltipla e em que situações ela é útil?
- 1.2. Explique o "Problema do Diamante" na herança múltipla.
- 1.3. Como o C++ resolve o Problema do Diamante?
- 1.4. Por que o Java não suporta herança múltipla de classes?
- 1.5. Qual a diferença entre herança múltipla e implementação de múltiplas interfaces?
- 1.6. Crie o sistema de biblioteca com Material, Livro, DVD e AudioLivro.

2. POLIMORFISMO E PARÂMETROS DEFAULT

- 2.1. O que é polimorfismo de sobrecarga?
- 2.2. Como funcionam os parâmetros default em C++?
- 2.3. Por que em Java não existem parâmetros default?
- 2.4. Quais as regras para uso de parâmetros default em C++?
- 2.5. Implemente construtores sobrecarregados para a classe Pessoa, que pode ser inicializada com os atributos: idade, nome, altura, peso, cor do cabelo, ou nenhum destes.

3. TEMPLATES

- 3.1. O que são templates em C++?
- 3.2. Qual a diferença entre template de função e template de classe?
- 3.3. Quais as vantagens e desvantagens do uso de templates?
- 3.4. Como funciona a instanciação de templates em tempo de compilação?

4. STL - VECTOR, LIST, QUEUE, STACK

- 4.1. Qual a principal vantagem do vector em relação a arrays comuns?
- 4.2. Explique as diferenças entre vector, list e queue.
- 4.3. Quando usar stack ao invés de queue?
- 4.4. O que é um iterator e qual sua importância na STL?
- 4.5. Use vector para armazenar e manipular uma lista de números.
- 4.6. Implemente uma pilha usando stack para verificar palíndromos.
- 4.7. Use queue para simular um sistema de atendimento.

5. TRATAMENTO DE EXCEÇÕES

- 5.1. O que é uma exceção e quando devemos usá-la?
- 5.2. Explique o funcionamento de try, catch e throw.
- 5.3. Qual a diferença entre exceções verificadas e não verificadas em Java?
- 5.4. O que é o bloco finally e quando ele é executado?
- 5.5. Implemente tratamento de exceções para divisão por zero.
- 5.6. Crie um sistema de cadastro com validação de dados usando exceções.

6. QUESTÕES DE IMPLEMENTAÇÃO

- 6.1. Implemente um sistema de gerenciamento de universidade usando:
 - Herança de Pessoa para Professor, Aluno, Funcionário
 - Tratamento de exceções personalizadas para dados inválidos
 - STL para armazenamento de dados