



Fundamentos de Orientação a Objetos
Prof. Dr. Edson Tavares de Camargo

Lista de Exercícios 1

1. Quais são as características principais do paradigma imperativo?
2. É possível afirmar que o paradigma orientado a objetos não tem qualquer ligação com o paradigma estruturado? Justifique.
3. De acordo com Bjarne Stroustrup, quais as vantagens e desvantagens do paradigma orientado a objetos? Pesquise outras vantagens e desvantagens em livros ou na Internet.
4. Aponte as quatro características fundamentais para uma linguagem ser classificada no paradigma Orientado a Objetos.
5. Ainda de acordo com Bjarne Stroustrup, qual o paradigma de programação suportado na linguagem C++.
6. No contexto do paradigma de O.O., o que uma classe?
7. No paradigma OO, classes e objetos são um dos principais conceitos.
 - Defina classes e objetos;
 - Aponte a relação entre eles
 - Apresente dois exemplos.
8. É possível afirmar que a classe é um tipo de dado? Justifique.
9. Crie uma classe Retângulo com um método área e perímetro. Crie 3 objetos dessa classe na função main(), atribua diferentes valores de altura e largura e apresente os valores de área e perímetro.
10. Qual a importância da regra dos três? Em que situações ela é aplicável?
11. Definir uma classe que represente um círculo. Esta classe deve possuir métodos privados para:
 - a) calcular a área do círculo;
 - b) calcular a circunferência do círculo;E métodos Públicos para:
 - c) definir o raio do círculo, dado um número real;
 - d) aumentar o raio do círculo, dado um percentual de aumento;
 - e) imprimir o valor do raio;
 - f) imprimir a área do círculo;
 - g) imprimir a circunferência do círculo.
12. Um objeto pode ser criado de forma estática e/ou dinâmica. Qual a diferença em termos de linguagem de programação, ou seja, como se lida com um objeto estático e um objeto dinâmico em C++. E qual a diferença em termos utilização de memória desses objetos?

13. Crie uma classe `Livro` com os seguintes atributos: título, nome do autor, isbn, edição, volume, numero de páginas e página atual. Com exceção do número de páginas, volume e edição, todos os demais atributos são do tipo *string*. Todos os atributos devem ser privados. Inicialize os atributos no construtor padrão. Defina um segundo construtor para receber o nome do livro e o seu número de páginas. Crie métodos do tipo *set* para definir o nome do livro e nome do autor e o número total de páginas. O nome do livro não pode passar de 30 caracteres e o nome do autor não deve exceder 20 caracteres. Crie um método *irParaPágina*, que recebe como parâmetro um número da página. Esse número não pode ultrapassar o número total de páginas. Crie ainda um método *proximaPagina* que incrementa a página atual em 1. Crie também um método para exibir o nome do livro e a página atual.
14. Desenvolva uma classe `Data` em C++ seguindo o paradigma orientado a objetos, que apresenta a data no seguinte formato de data: dd/mm/aaaa. A solução deve observar as seguintes características: objetos alocados dinamicamente, definição da classe separada da sua definição (.h e .cpp), diferentes construtores, um arquivo somente para realizar os testes (.teste.cpp) onde está a função `main()`.

```
1- Informe a data (dd/mm/aaa)
2 – Incrementar ano
3 - Incrementar mês
4 – Incrementar dia
5 – Decrementar ano
6 – decrementar mês
7 – decrementar dia
8 – Ajustar data
9 – Exibir a data
10 – Sair
```

Informe a sua opção:

- O menor ano possível é 1900.
 - Também crie um forma de verificar se o ano é bissexto.
15. Crie uma classe `Conta` que um banco poderia utilizar para representar contas bancárias dos seus clientes. Sua classe deve incluir um membro de dado do tipo *float* para representar o saldo da conta. Deve haver um construtor que recebe o saldo inicial e o utiliza para inicializar o membro de dado. O construtor deve validar o saldo inicial para assegurar que ele seja maior ou igual a 0. A classe deve fornecer três funções membros. A função *credito* deve adicionar uma quantia ao saldo atual. A função *membro débito* deve retirar uma quantia, porém essa quantia não pode levar o saldo inicial a um valor negativo. A função *membro para ver o saldo* deve apresentar o saldo atual da conta.
16. Um determinado jogo verifica qual a pontuação de um jogador de futebol considerando algumas estatísticas do jogo: nome, número de passes certos e errados, quantidade de gols, faltas sofridas e faltas realizadas, minutos jogados. Crie um classe chamada `Jogador` para representar o jogador. Também defina uma fórmula para atribuir uma pontuação de 0 a 10 ao jogador e execute o cálculo em um método chamado *calcularPontos()*. Por exemplo, o número de gols é multiplicado por 3, já sua pontuação diminui 1 ponto a cada 10 faltas realizadas... A classe deve ter pelos menos dois construtores: o construtor padrão

e outro que recebe o nome do jogador. Define *gets* e *sets* para cada atributo. Crie um método para exibir os dados do jogador e sua pontuação.

17. Crie uma classe denominada Elevador para armazenar as informações de um elevador dentro de um prédio. A classe deve armazenar o andar atual (0=térreo), total de andares no prédio, excluindo o térreo, capacidade do elevador, e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:
- a) inicializa: que deve receber como parâmetros: a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazios);
 - b) entra: para acrescentar uma pessoa ou mais pessoas no elevador (só deve acrescentar se ainda houver espaço);
 - c) sai: para remover uma pessoa ou mais pessoas do elevador (só deve remover se houver alguém dentro dele);
 - d) sobe: para subir um andar (não deve subir se já estiver no último andar);•
 - e) desce: para descer um andar (não deve descer se já estiver no térreo);
 - f) get_____: métodos para obter cada um dos os dados armazenados.