
Lista de Exercícios

1. Descreva as vantagens que uma linguagem Orientada a Objetos trouxe comparando com linguagens procedimentais, tais como linguagem C.
2. Faça uma lista dos objetos com que cada um dos sistemas a seguir deve lidar:
 - a. Um programa para diagramar um jornal
 - b. Um programa para calcular e armazenar pontos de boliche
 - c. Uma secretaria eletrônica
 - d. Um leitor e gravador de DVD
 - e. Um sistema de entrada de pedidos pelo correio
3. A seguir temos duas listas. A primeira é uma lista de classes que descrevem objetos da implementação. A segunda é uma lista de operações. Para cada uma das classes, selecione as operações adequadas aos objetos daquela classe.

Classes:	Operações:
a) Tabela de Símbolos—uma tabela que mapeia em descritores as palavras-chave de um texto;	i. Append – acrescentar um objeto no final de uma coleção;
b) Conjunto – coleção não-ordenada de objetos sem duplicidades.	ii. Copy – fazer uma cópia de uma coleção;
	iii. Count – indicar o número de elementos de uma coleção;
	iv. Delete – eliminar um componente de uma coleção;
	v. Index–recuperar um objeto posicionado em uma determinada posição de uma coleção;
	vi. Intersec – determinar os membros comuns de duas coleções;
	vii. Insert – colocar um objeto em uma determinada posição de uma coleção;
	viii. Update – acrescentar um componente a uma coleção, escrevendo por cima do que quer que ali esteja.

4. Descreva o que os objetos em cada uma das listas abaixo têm em comum.
 - a. Microscópio eletrônico de varredura, óculos, telescópio, binóculos.
 - b. Bicicleta, veleiro, carro, caminhão, avião, planador, motocicleta, cavalo
 - c. Prego, parafuso, pino
 - d. Tenda, caverna, barraco, garagem, celeiro, casa, arranha-céu
 - e. Raiz quadrada, exponencial, seno, cosseno
5. Modele a classe OperacaoBinaria (matematicamente, uma operação binária é uma função $B: S \times S \rightarrow S$). Na sua modelagem, defina somente os atributos e as assinaturas dos métodos usando a sintaxe da linguagem Java. A classe OperacaoBinaria deve ser uma classe parametrizada com o(s) tipo(s) de elemento(s) que ela vai manipular.
6. Especifique uma classe genérica e abstrata que permita representar e operar sobre uma Arvore Binária. Em seguida faça a implementação de um método denominado eval(), o qual quando aplicado a um nó da arvore some todos os valores de seus nós descendentes. Apresente um exemplo (desenho) de instanciamento de uma arvore instanciada.
7. Discuta sobre o tema “Polimorfismo em linguagens OO”. O seu texto deve obrigatoriamente conter as seguintes palavras-chave: Paramétrico, Inclusão, Sobrecarga e Coerção. Use exemplos em Java para ajudar na sua explicação.
8. Defina os seguintes conceitos: Classe, Objeto, Relacionamentos, e Herança. Para cada um dos conceitos descritos, apresente exemplos usando a linguagem Java.
9. Projete e implemente um programa na linguagem Java para simular casamento entre pessoas. Uma pessoa pode ser do tipo Homem ou Mulher, e deve possuir os atributos nome e idade. Uma pessoa deve possuir o comportamento Casar. As classes Homem e Mulher devem implementar as interfaces Macho e Fêmea respectivamente, as quais deverão ser utilizadas para definir o comportamento da pessoa de acordo com o seu sexo (Dica: pense em um tipo de comportamento específico para homens e mulheres). Crie também uma classe chamada Teste, a qual instancie objetos das classes Homem e Mulher e realize casamentos. A sua implementação deve respeitar as seguintes restrições:

- b. Uma pessoa não pode casar consigo mesma;
 - c. Para uma pessoa casada se casar novamente, ela precisa se divorciar antes do novo casamento;
 - d. Casamento só pode existir entre pessoas de sexo diferentes;
 - e. Pessoas com idade inferior a 18 anos não podem se casar;
10. Especifique e crie uma classe genérica que permita criar grafos. O grafo implementado deve ser direcionado e cada vértice e aresta pode ser anotado com qualquer tipo de dado. Um vértice pode possuir uma quantidade indeterminada de arestas de chegada e arestas de saída. A classe Grafo deve permitir basicamente a criação de vértices e arestas, não é necessário implementar buscas no grafo.
11. Implemente uma classe Java para realizar a verificar uma expressão do tipo “<Operando1> <Operador> <Operando2>”, onde o operandos são números inteiros. Os operadores permitidos são “+” (adição) e “*” (multiplicação), porém suponha que a sua implementação deva facilitar a inclusão de novos operadores. A classe criada deve funcionar conforme o código apresentado abaixo:

```
// (2 + 3) * 4
Operando o1 = new Operando(2);
Operando o2 = new Operando(3);
Operando o3 = new Operando(4);
Operador e1 = new Operador(o1, '+', o2);
Operador e2 = new Operador(e1, '*', o3);
System.println(e2.avaliar()); // Vai imprimir 20
```

12. Uma subclasse pode herdar a 'interface' ou a 'implementação' de uma superclasse. Como as hierarquias de herança projetadas para herdar interface diferem daquelas projetadas para herdar implementação;
13. O Java, ao contrário do C++, não suporta herança múltipla, que ocorre quando uma classe é derivada de mais de uma superclasse direta. Explique como os programadores em Java podem obter muitos dos benefícios da herança múltipla e, ao mesmo tempo, evitar os problemas associados;
14. Comente sobre as quatro maneiras de atribuir referências de superclasse e de subclasse a variáveis de superclasse e a tipos de subclasse. Em qual delas ocorre um erro de compilação?
- a. Atribuir uma referência de superclasse a uma variável de superclasse;
 - b. Atribuir uma referência de subclasse a uma variável de subclasse;
 - c. Atribuir uma referência de subclasse a uma variável de superclasse;
 - d. Atribuir uma referência de superclasse a uma variável de subclasse;
15. Criar uma estrutura hierárquica que contenha as seguintes classes: Veículo, Bicicleta e Automóvel. Em seguida defina os atributos e os métodos com suas assinaturas dessas classes. Explore o uso de polimorfismo e de classe abstrata.