

Exercícios - Programação Orientada a Objetos

Disciplina: Programação Orientada a Objetos

Aluno: Bernardo Henrique Lebron Machado

1 a) Modelo, padrão ou estilo de programação suportado por linguagens que agrupam certas características em comum. Exemplos: Estruturado, Orientado a Objetos e Funcional.

b) Ambos são organizados de maneira diferente: o Estruturado é baseado em funções e procedimentos, enquanto o Orientado a Objetos é baseado em classes e objetos. O paradigma estruturado possui vantagens como simplicidade na implementação e eficiência, mas em programas grandes pode haver dificuldade de manutenção. Já o paradigma orientado a objetos possui vantagens como modularização do código, reutilização e maior aplicação no mercado de trabalho, porém pode ser mais complexo de implementar.

c) Existem muitas linguagens de programação diferentes porque cada uma foi criada para resolver tipos específicos de problemas. Assim, para cada desafio, é importante escolher a linguagem mais adequada à situação.

2 a) É o processo de identificar apenas as características essenciais de um objeto do mundo real e ignorar os detalhes desnecessários, representando algo complexo de forma simples. Por exemplo, ao dirigir um carro, o motorista utiliza apenas os comandos principais, sem se preocupar com o funcionamento interno do motor.

b) É o ato de proteger os dados de um objeto, permitindo que sejam acessados e modificados apenas por meio de métodos controlados. Isso é importante porque evita alterações indevidas e mantém a segurança e integridade das informações.

c) Classe é o modelo que define os atributos e comportamentos de algo. Objeto é uma instância real desse modelo. Por exemplo: Classe: "Pessoa"/ Objeto: Nome: "Bernardo", Idade: "21".

3 a) Classe, Atributos, Métodos e Associações.

b) Associação simples: relação entre duas classes que indica um vínculo entre elas.

Associação agregação: relação todo–parte em que a parte pode existir separadamente do todo.

Associação composição: relação todo–parte em que a parte não existe sem o todo.

Generalização: relação em que uma superclasse é especializada em subclasses com características específicas.

Dependência: relação em que uma classe usa outra temporariamente, mas não depende dela o tempo todo.

4 a) Private: acessível somente dentro da própria classe.

Protected: acessível dentro da classe e de suas classes derivadas.

Public: acessível em qualquer classe onde o objeto seja visível.

b) Os modificadores de acesso são importantes porque controlam a visibilidade e o acesso aos dados e métodos, garantindo encapsulamento, segurança e facilidade de manutenção do código.

c)

	private	protected	public	Default (Java)
Mesma classe	sim	sim	Sim	sim
Subclasses	não	sim	sim	não
Classes no mesmo pacote (Java)	não	sim	sim	sim
Classes diferentes	não	não	sim	não

5 a) Herança é a capacidade de compartilhar estruturas e comportamentos em comum entre várias classes derivadas. Polimorfismo é a capacidade de um mesmo método ter diferentes implementações conforme o objeto que o executa.

b) A herança simples ocorre quando uma classe herda de apenas uma superclasse, enquanto a herança múltipla ocorre quando uma classe herda de duas ou mais superclasses.

c) É uma regra usada para verificar se faz sentido aplicar herança entre duas classes. Deve-se usar herança quando uma classe “é um tipo de” outra classe.

d) O polimorfismo de inclusão ocorre quando um objeto de uma subclasse é tratado como se fosse da superclasse, mantendo seu próprio comportamento.

6 a) A sobrecarga de métodos é um recurso que permite criar vários métodos com o mesmo nome, mas com parâmetros diferentes.

b) Para realizar sobrecarga, o nome do método deve ser o mesmo, os parâmetros devem ser diferentes e podem existir assinaturas diferentes para o mesmo método.

c) Em C++, a sobrecarga é mais flexível, pois pode ser aplicada a métodos e operadores. Em Java, é restrita a métodos.
