



UNIVERSIDADE
FEDERAL DO CEARÁ
Dezembro de 2024

Disciplina: Programação Orientada à Objetos
Professor: Filipe Fernandes dos Santos Brasil de Matos

Semestre: 2024.2

Aluno: _____ **Matrícula:** _____

Lista de Exercícios

Herança e Polimorfismo

1. Explique o conceito de herança em Java e sua utilidade no desenvolvimento de software.
2. Defina superclasse e subclasse. Uma superclasse tem acesso aos membros de uma subclasse? E a subclasse pode acessar os membros de uma superclasse?
3. Como impedir que uma subclasse tenha acesso a um membro de uma superclasse?
4. Explique o papel da palavra-chave super em herança. DICA: Descreva a finalidade e a aplicação das duas versões de super discutidas na aula.
5. Pesquise e discuta qual é a diferença entre sobrescrita e sobrecarga de métodos?
6. O que é polimorfismo em Java? Dê exemplos de como ele pode ser aplicado.
7. Implemente uma superclasse Pessoa e duas subclasses, Estudante e Professor, que herdam de Pessoa. Uma pessoa tem um nome e um ano de nascimento. Um estudante tem uma especialização e um professor tem um salário. Escreva as definições de classe e os construtores de todas as classes. Forneça um programa de teste que testa essas classes e os métodos implementados.
8. Crie uma classe Empregado com um nome e salário. Faça uma classe Gerente herdar de Empregado. Adicione um campo de instância, chamado departamento,

do tipo `String`. Forneça um método *toString()* que imprime o nome do gerente, departamento e salário. Faça uma classe `Executivo` herdar de `Gerente`. Forneça os métodos *toString()* apropriados para todas as classes. Forneça um programa de teste que testa essas classes e métodos.

9. Escreva uma superclasse `Trabalhador` e duas subclasses `TrabalhadorAssalariado` e `TrabalhadorHora`. Cada trabalhador tem um nome e um percentual salarial. Escreva um método *computarPagamento(int horas)* que calcula o pagamento semanal para cada trabalhador. Um horista é pago de acordo com o número real de horas trabalhadas, se horas for no máximo 40. Se o horista trabalhar mais de 40 horas, as horas excedentes serão com 50% de acréscimo. O assalariado é pago por 40 horas semanais, independentemente do número real de horas trabalhadas. Forneça um programa de teste que use polimorfismo para testar essas classes e métodos.
10. Crie uma classe `ContaCorrente` com os atributos `nomeCliente`, `numConta` e `saldo` e com métodos *obterSaldo()*, *depositarDinheiro()* e *sacarDinheiro()*. Cada saque gera um débito (a mais) de 0,5% do valor sacado referente à taxa de operação. Em seguida, crie uma subclasse de `ContaCorrente`, chamada `ContaCorrenteEspecial`, que representa uma conta corrente de um cliente especial. Clientes especiais pagam taxas de operação de apenas 0,1% do valor sacado. Faça testes com as duas classes e verifique seus resultados.

Lista de Exercícios

Interfaces e Classes Abstratas

1. Defina os conceitos de interface e classe abstrata em Java. Explique como cada uma delas se distinguem de uma classe comum.
2. Compare as principais características de interfaces e classes abstratas em Java. Liste e explique os aspectos fundamentais de cada uma delas e destaque as suas peculiaridades. Por fim, aponte as principais semelhanças e diferenças entre elas.
3. Como interfaces e classes abstratas são declaradas em Java? Explique a sintaxe utilizada para declará-las através de exemplos.
4. Qual a relação entre classes abstratas e métodos abstratos em Java? Destaque o conceito desses métodos e sua relação com as classes abstratas. É possível ter métodos não abstratos em uma classe abstrata? Esclareça a possibilidade e o propósito de ter métodos concretos em uma classe abstrata.

5. **Pesquise e apresente situações onde as classes abstratas são mais apropriadas que interfaces em Java. Apresente também situações onde as interfaces são mais adequadas e benéficas que as classes abstratas.**
6. **Descreva o conceito de herança de interface e forneça um exemplo. Descreva também como as classes abstratas podem ser estendidas e como a herança é aplicada nesse contexto.**
7. **É possível implementar múltiplas interfaces em uma única classe em Java? Explique como a herança de interface permite que uma classe implemente mais de uma interface.**
8. **Por que não podemos criar objetos de uma classe abstrata? Explique a razão por trás da impossibilidade de criar instâncias diretas de uma classe abstrata.**
9. **Qual é a principal diferença entre métodos concretos e métodos abstratos? Por que usar métodos abstratos em uma classe? Como a presença de métodos abstratos em uma classe afeta a instanciação de objetos dessa classe?**
10. **Projete uma interface Geometrico com métodos para cálculo de área e perímetro. Implemente essa interface em classes como Quadrado e Circulo e realize cálculos para diferentes formas geométricas.**
11. **Elabore uma interface Autenticavel com um método autenticar que recebe um usuário e uma senha. Implemente essa interface em uma classe Autenticador e utilize-a para autenticar usuários.**
12. **Projete duas interfaces, Ordenavel e Pesquisavel, ambas com métodos abstratos. Implemente uma classe ColecaoDados que realize operações de ordenação e pesquisa utilizando ambas as interfaces.**
13. **Crie uma classe abstrata Funcionario com métodos estáticos para calcular o salário líquido e abstratos para definir as regras específicas do cálculo do salário. Implemente subclasses para diferentes tipos de funcionários, como horista e assalariado, por exemplo.**
14. **Desenvolva uma classe abstrata SistemaLog com métodos estáticos para registrar mensagens de log. Adicione métodos abstratos para diferentes níveis de log e destinos de registro.**
15. **Desenvolva uma classe abstrata ControleDeEstoque com métodos estáticos para adicionar e remover itens do estoque, e métodos abstratos para gerenciar inventário. Implemente subclasses para controle de estoque de produtos, materiais, etc.**