| Uma imagem contendo placar, desenho, relógio  Descrição gerada automaticamente | **ATIVIDADE ADICIONAL PRÁTICA – JAVA** |
| --- | --- |
| JAVA-12 – Herança e polimorfismo |

**Instruções gerais:**

| 1. Utilize o Eclipse ou o STS para desenvolver os algoritmos. |
| --- |
| **Mantenha as entregas das Atividades em dia na Plataforma Canvas** |

**EXERCÍCIOS**

**Boas práticas:**

1. Leia o enunciado do exercício com atenção
2. Observe as indicações de Entrada e Saída esperadas em cada exercício
3. Observe com atenção os desenhos e diagramas inseridos nos exercícios para facilitar a compreensão
4. Utilize o Cookbook, os Vídeos da Plataforma e os Códigos guia como referências para a resolução dos exercícios
5. Caso ainda fique alguma dúvida, consulte os instrutores da sua turma pelo Discord

**Lista de exercícios**

### **Exercícios sobre POO: Herança e Polimorfismo**

1. **Herança Básica: Classe Animal e Subclasses**:
   * Crie uma classe base Animal com os atributos nome e idade, e um método fazerSom(). Crie as subclasses Cachorro e Gato que herdam de Animal, cada uma implementando o método fazerSom() de forma diferente. No método main, crie objetos de Cachorro e Gato e chame o método fazerSom().
2. **Herança e Construtores**:
   * Crie uma classe Veiculo com atributos marca e modelo, e um construtor para inicializar esses atributos. Crie a subclasse Carro que adicione um atributo numeroDePortas. No método main, crie um objeto Carro e exiba todos os atributos.
3. **Polimorfismo com Métodos de Subclasses**:
   * Crie uma classe base Forma com um método calcularArea(). Crie as subclasses Quadrado e Circulo, cada uma com a implementação apropriada do método calcularArea(). No método main, crie uma lista de Formas, adicione objetos de Quadrado e Circulo, e calcule as áreas usando polimorfismo.
4. **Sobrescrita de Métodos**:
   * Crie uma classe Funcionario com um método calcularSalario() que retorna um valor base. Crie a subclasse Gerente que sobrescreva o método calcularSalario() para adicionar um bônus. No método main, crie objetos de Funcionario e Gerente, e calcule os salários.
5. **Classe Abstrata ContaBancaria**:
   * Crie uma classe abstrata ContaBancaria com métodos abstratos depositar(double valor) e sacar(double valor). Crie as subclasses ContaCorrente e ContaPoupanca com implementações específicas desses métodos. No método main, crie objetos de ambas as classes e realize operações bancárias.
6. **Polimorfismo com Interfaces**:
   * Crie uma interface Pagavel com um método pagar(). Implemente essa interface nas classes Funcionario e Fornecedor. No método main, crie uma lista de Pagavel e adicione objetos de Funcionario e Fornecedor, e use polimorfismo para chamar o método pagar().
7. **Hierarquia de Classes: Veiculo, Carro, Moto**:
   * Crie uma hierarquia de classes com Veiculo como classe base e Carro e Moto como subclasses. Adicione métodos específicos a cada subclasse. No método main, demonstre o uso do polimorfismo ao manipular diferentes tipos de veículos.
8. **Classe Base Produto e Subclasses**:
   * Crie uma classe base Produto com atributos nome e preco, e um método exibirDetalhes(). Crie as subclasses Livro e Eletronico, cada uma com atributos adicionais e que sobrescrevam o método exibirDetalhes(). No método main, crie objetos dessas classes e chame o método exibirDetalhes().
9. **Classe Abstrata Pessoa e Implementação Concreta**:
   * Crie uma classe abstrata Pessoa com métodos abstratos obterDocumento() e exibirDados(). Crie as subclasses PessoaFisica e PessoaJuridica, cada uma implementando os métodos abstratos de forma adequada. No método main, crie objetos dessas classes e exiba os dados das pessoas.
10. **Polimorfismo com Coleções**:
    * Crie uma classe base InstrumentoMusical com um método tocar(). Crie as subclasses Guitarra, Piano, e Bateria, cada uma implementando tocar() de forma diferente. No método main, crie uma lista de InstrumentoMusical, adicione objetos de diferentes instrumentos e chame tocar() para todos usando polimorfismo.