| Uma imagem contendo placar, desenho, relógio  Descrição gerada automaticamente | **ATIVIDADE ADICIONAL PRÁTICA – JAVA** |
| --- | --- |
| JAVA-11 – classe objetos |

**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: Abstração, Classes e Objetos**

1. **Classe Pessoa e Métodos Simples**:
   * Crie uma classe Pessoa com atributos nome e idade. Crie um método dizerOla() que exiba uma mensagem de saudação com o nome da pessoa. No método main, crie um objeto da classe Pessoa e chame o método dizerOla().
2. **Classe ContaBancaria com Operações Básicas**:
   * Crie uma classe ContaBancaria com atributos titular, saldo, e numeroConta. Implemente métodos depositar(double valor), sacar(double valor), e consultarSaldo(). No método main, crie um objeto ContaBancaria, realize depósitos e saques, e consulte o saldo.
3. **Classe Livro com Métodos para Emprestar e Devolver**:
   * Crie uma classe Livro com atributos titulo, autor, e disponivel. Implemente métodos emprestar() e devolver() que atualizam o status de disponibilidade do livro. No método main, crie um objeto Livro, realize empréstimos e devoluções, e exiba o status.
4. **Classe Carro com Métodos para Acelerar e Frear**:
   * Crie uma classe Carro com atributos marca, modelo, e velocidadeAtual. Implemente métodos acelerar(int incremento) e frear(int decremento) para alterar a velocidade do carro. No método main, crie um objeto Carro e simule a aceleração e frenagem.
5. **Classe Produto com Métodos para Aplicar Desconto**:
   * Crie uma classe Produto com atributos nome, preco, e quantidade. Implemente um método aplicarDesconto(double percentual) que reduza o preço do produto pelo percentual dado. No método main, crie um objeto Produto, aplique um desconto e exiba o novo preço.
6. **Classe Aluno e Cálculo de Média**:
   * Crie uma classe Aluno com atributos nome, nota1, nota2, e nota3. Implemente um método calcularMedia() que retorne a média das notas do aluno. No método main, crie um objeto Aluno, calcule a média e exiba o resultado.
7. **Classe Retangulo para Cálculo de Área e Perímetro**:
   * Crie uma classe Retangulo com atributos largura e altura. Implemente métodos calcularArea() e calcularPerimetro(). No método main, crie um objeto Retangulo, calcule e exiba a área e o perímetro.
8. **Classe Funcionario com Cálculo de Salário**:
   * Crie uma classe Funcionario com atributos nome, salarioBase, e horasExtras. Implemente um método calcularSalario() que retorne o salário total considerando as horas extras (cada hora extra é 50% a mais do valor da hora normal). No método main, crie um objeto Funcionario, calcule e exiba o salário.
9. **Classe Quadrado com Método para Verificar se é um Quadrado Perfeito**:
   * Crie uma classe Quadrado com um atributo lado. Implemente um método ehQuadradoPerfeito() que verifique se os lados formam um quadrado perfeito. No método main, crie objetos Quadrado e teste se são quadrados perfeitos.
10. **Classe Círculo com Cálculo de Circunferência e Área**:
    * Crie uma classe Circulo com um atributo raio. Implemente métodos calcularCircunferencia() e calcularArea() para retornar a circunferência e a área do círculo, respectivamente. No método main, crie um objeto Circulo, calcule e exiba a circunferência e a área.