

Estes exercícios devem ser entregues no Google Classroom. Para cada um dos exercícios, crie um arquivo fonte Python com o respectivo nome de acordo com a seguinte regra:

Para esse projeto crie cada classe separada sendo iniciais do seu nome + a classe exemplo usando Hermano Roepke: hrpagamento.py, hrpagamentodinheiro.py, hrpagamentocartao.py, hrpagamentocheque.py e hrteste.

Exercício: Controle de Pagamentos

Imagine que você está desenvolvendo um sistema de controle de pagamentos para uma empresa. Neste sistema, existem diferentes tipos de pagamentos, como pagamento em dinheiro, pagamento com cartão de crédito e pagamento com cheque. Cada tipo de pagamento possui informações específicas.

1. Crie uma classe abstrata chamada **Pagamento** com o seguinte método abstrato:
 - **realizar_pagamento()**: Este método deve ser implementado nas subclasses e simular o processo de pagamento.
2. Crie três subclasses de **Pagamento**: **PagamentoDinheiro**, **PagamentoCartao** e **PagamentoCheque**.
 - Na classe **PagamentoDinheiro**, implemente o método **realizar_pagamento()** para imprimir "Pagamento em dinheiro realizado".
 - Na classe **PagamentoCartao**, implemente o método **realizar_pagamento()** para imprimir "Pagamento com cartão de crédito realizado".
 - Na classe **PagamentoCheque**, implemente o método **realizar_pagamento()** para imprimir "Pagamento com cheque realizado".
3. Crie uma função chamada **processar_pagamento()** na classe teste que aceite um objeto do tipo **Pagamento** como entrada e chame o método **realizar_pagamento()** desse objeto.
4. No bloco principal (classe de teste) do programa, crie instâncias das três subclasses (**PagamentoDinheiro**, **PagamentoCartao** e **PagamentoCheque**) e chame a função **processar_pagamento()** passando cada uma dessas instâncias como argumento.
5. Manipule exceções: Adicione uma cláusula de exceção para lidar com casos em que o método **realizar_pagamento()** não é implementado corretamente nas subclasses. Imprima uma mensagem de erro apropriada nesses casos.

Explicação:

Neste exercício, você está criando um sistema simples de controle de pagamentos com herança e polimorfismo. A classe abstrata **Pagamento** define um método abstrato **realizar_pagamento()**, que é implementado nas subclasses **PagamentoDinheiro**, **PagamentoCartao** e **PagamentoCheque**. O polimorfismo é observado quando a função **processar_pagamento()** é chamada com diferentes tipos de pagamentos, e cada uma delas realiza o pagamento específico. A manipulação de exceções é usada para garantir que as subclasses implementem corretamente o método **realizar_pagamento()**.