Exercício 4: Herança simples com input

Enunciado:
Crie uma `class` chamada `Pessoa`, com os atributos `nome` e `idade`. Adicione um método `def apresentar(self)` que imprime uma mensagem com o nome e a idade.

Depois, crie uma `class Aluno` que herda de `Pessoa` e não sobrescreve nada.

No final, o programa deve ler o nome e a idade pelo `input()`, criar um objeto da `class Aluno` e chamar o método `apresentar()`.

Exercício 5: Herança com `super()` e input

Enunciado:
Crie uma `class Produto`, com os atributos `nome` e `preco`. Crie um método `def detalhes(self)` que imprime essas informações.

Depois, crie uma `class ProdutoImportado` que herda de `Produto` e adiciona um atributo `taxa_importacao`. Use `super()` no `__init__` para aproveitar o da `class` pai.

Sobrescreva o método `detalhes()` para mostrar também a `taxa_importacao`.

Peça ao usuário para inserir `nome`, `preco` e `taxa_importacao` via `input()` e exiba os detalhes com o método sobrescrito.

Exercício 6: Super com lógica adicional

Enunciado:
Crie uma `class Curso`, com os atributos `nome` e `duracao` (em horas). Crie um método `def descricao(self)` que exibe essas informações.

Crie a `class CursoOnline`, que herda de `Curso` e adiciona o atributo `plataforma`. Use `super()` para inicializar os atributos da `class` base e sobrescreva `descricao()` para incluir a `plataforma`.

Solicite os dados via `input()` e exiba a descrição completa.

Exercício 7: Galeria de Arte

Enunciado:
Crie uma classe chamada `ObraDeArte`, com os atributos `titulo` e `ano_criacao`, e um método `mostrar_info()` que imprime esses dados.

Depois, crie uma classe `Pintura`, que herda de `ObraDeArte`, e tem um atributo extra `tecnica` (ex: óleo sobre tela, aquarela, etc).

Implemente o método `mostrar_info()` na classe `Pintura`.

Peça ao usuário o `titulo`, o `ano_criacao` e a `tecnica`, crie o objeto e exiba as informações.

Exercício 8: Personagem de Jogo

Enunciado:
Crie uma classe `Personagem`, com os atributos `nome` e `nivel`, e um método `status()` que imprime esses dados.

Depois, crie a classe `Mago`, que herda de `Personagem`, com o atributo adicional `mana`.

Leia os dados com `input()`, crie o mago e exiba seu status.

Exercício 9: Viagem com conversão de moeda

Enunciado:
Crie uma classe chamada `Viagem`, com os atributos `destino` e `duracao`, e um método `resumo()` que imprime os dados da viagem.

Depois, crie uma classe `ViagemInternacional` que herda de `Viagem` e adiciona os atributos `passaporte`, `moeda_local_em_reais` (valor unitário da moeda em reais. Exemplo: 1.0 Dólar () = `7.0` Reais) e `quantidade_dinheiro_local` (quantidade de dinheiro que o usuário possui no valor local. Exemplo: 6.0 dólares).

Adicione à classe `ViagemInternacional` um método `converter_para_reais()` que retorna o valor convertido para reais (Exemplo: 7.0 * 6.0 - ou seja, `moeda_local * quantidade_dinheiro_local`).

A classe também deve sobrescrever o método `resumo()` para incluir as novas informações.

Solicite ao usuário os seguintes dados via `input()`:

- destino
- duracao
- passaporte (string)
- moeda_local_em_reais (ex: valor de 1 dólar para reais, valor de 1 euro para reais)
- quantidade_dinheiro_local (float qualquer)

Depois, exiba o resumo da viagem e o valor convertido para reais.