

Para cada exercício, crie um módulo correspondente contendo as classes e funções solicitadas.

Organize os módulos em um pacote Encapsulamento, criando uma estrutura de diretórios apropriada com arquivo `__init__.py`.

No arquivo principal, importe os módulos e teste as funcionalidades conforme indicado.

Certifique-se de usar a nomenclatura adequada para importação e mantenha o código organizado dentro do pacote.

Exercício 1

Crie a classe Aluno com:

Atributo público nome

Atributo protegido `_matricula`

Atributo privado `__senha_portal`

Adicione:

Um método `exibir_senha()` que imprime a senha.

Exercício 2

Crie um objeto da classe Aluno.

Acesse diretamente o atributo nome.

Tente acessar diretamente `_matricula` e `__senha_portal`.

Use o método `exibir_senha()` para mostrar a senha corretamente.

Adicione um método `get_matricula()` e teste seu uso.

Exercício 3

Modifique a classe Aluno:

Crie o método `get_senha()` para retornar a senha.

Crie o método `set_senha(nova_senha)` que:

Só permite alterar a senha se ela tiver pelo menos 6 caracteres.

Caso contrário, exiba uma mensagem de erro.

Exercício 4

Crie a classe `ContaBancaria` com:

Atributo privado `__saldo`, iniciado em 0.

Adicione os métodos:

`get_saldo()` — retorna o saldo atual.

`depositar(valor)` — só permite depósito se `valor > 0`.

`sacar(valor)` — só permite saque se houver saldo suficiente.

Teste a classe com vários depósitos e saques válidos e inválidos.

Exercício 5

Crie a classe `Funcionario` com:

Atributo público `nome`

Atributo protegido `_cargo`

Atributo privado `__salario`

Adicione os métodos:

`mostrar_detalhes()` — imprime todos os dados.

`get_salario()` — retorna o salário.

`set_salario(valor)` — só aceita valores numéricos positivos.

Exercício 6

Crie a classe Cofre com:

Atributo privado `__segredo`

Atributo protegido `_historico`, uma lista com os segredos antigos

Adicione os métodos:

`alterar_segredo(novo)` — altera o segredo apenas se diferente do atual, e salva o antigo no histórico.

`mostrar_historico()` — imprime todos os segredos antigos armazenados na lista.