

re S10 _A1 _SL 7 desenvolvimento software

Roteiro de Atividade Prática

Nome: _____ Turma: _____

Atividade 1: Classe Simples - Lampada

Objetivos

Antes de resolver este exercício, é importante entender o que é uma classe e como criar objetos a partir dela. Uma classe é definida usando a palavra-chave `class`, seguida do nome da classe e dois-pontos. Dentro da classe, podemos definir métodos (funções) e atributos (dados).

Descrição do exercício:

Utilizando a linguagem de programação Python, crie uma classe `Lampada` com um atributo `estado` (`ligado/desligado`) e um método para alterar seu estado.

Tempo estimado: 15 minutos

Procedimento experimental

1. Analise o exemplo-base para criação do código:

```
class Lampada:
    def __init__(self):
        self.estado = False # Começa desligada
# Uso da classe
lampada = Lampada()
print(lampada.alterar_estado()) # Liga a lâmpada
print(lampada.alterar_estado()) # Desliga a lâmpada
```

2. Agora, crie uma definição para alterar o seu estado entre ligada e desligada de acordo com o exemplo disponibilizado.

Atividade 2: Encapsulamento com Atributos Privados

Objetivos

Entenda que encapsulamento em Python é mais uma convenção do que uma restrição técnica. Atributos privados são definidos com um sublinhado duplo (__). Eles não são acessíveis diretamente de fora da classe, promovendo o encapsulamento.

Descrição do exercício:

Crie uma classe Contador que mantenha um valor interno privado e tenha métodos para incrementar, decrementar e obter esse valor.

Tempo estimado: 15 minutos

Procedimento experimental

1. Analise o exemplo base para criação do código:

```
class Contador:
    def __init__(self):
        self.__valor = 0

# Uso da classe
contador = Contador()
contador.incrementar()
contador.incrementar()
print(contador.get_valor()) # Deve mostrar 2
contador.decrementar()
print(contador.get_valor()) # Deve mostrar 1
```

2. Agora, a partir do código analisado, crie as definições para incrementar, decrementar e obter esse valor.

Atividade 3: Uso de Getters e Setters

Objetivos

Getters e Setters são métodos usados para obter e definir o valor de atributos privados. Eles são úteis para adicionar lógica adicional durante a obtenção ou a definição de um valor, como validações.

Descrição do exercício:

Crie uma classe Termometro que armazene a temperatura em graus Celsius, mas permita definir e obter a temperatura em Fahrenheit.

Tempo estimado: 10 minutos

Procedimento experimental

1. Analise o exemplo-base para criação do código:

```
class Termometro:
    def __init__(self, temperatura=0):
        self.__temperatura_celsius = temperatura

    def get_temperatura_fahrenheit(self):
        return (self.__temperatura_celsius * 9/5) + 32

# Uso da classe
termometro = Termometro()
termometro.set_temperatura_fahrenheit(68)
print(termometro.get_temperatura_fahrenheit()) # Deve mostrar 68
```

2. Agora, a partir do código analisado, crie as definições para que a temperatura também possa ser obtida em Celsius.