

Micro atividade 4: Descrever a utilização de encapsulamento em Python

//Disciplina: Nivel4: Conhecendo novos paradigmas

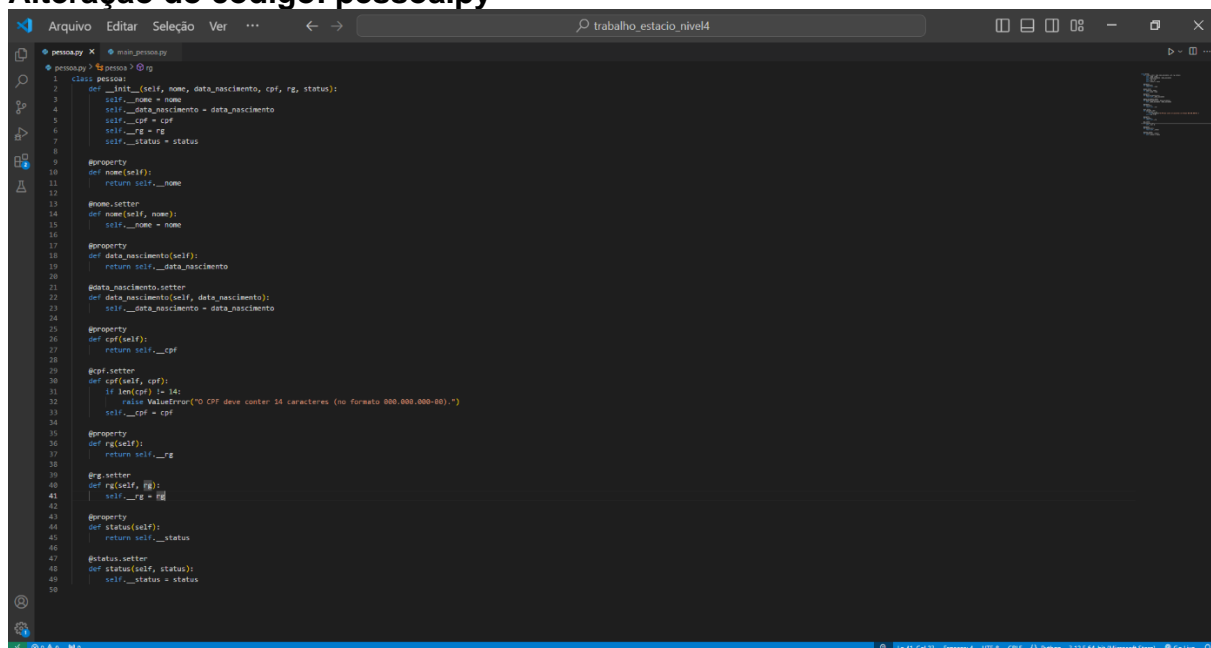
//Professor: Jhonatan Alves

//Aluno: Rafael Lima de Medeiros

//Turma: 2024.2

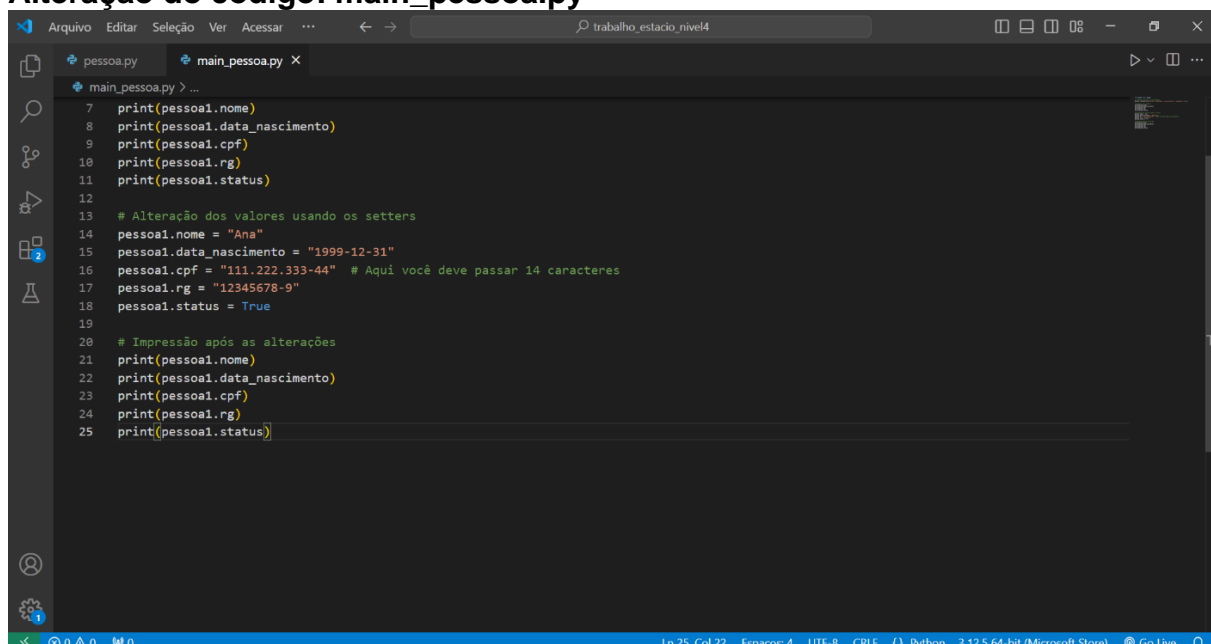
//Data atual: 20/08/2024

Alteração do código: pessoa.py



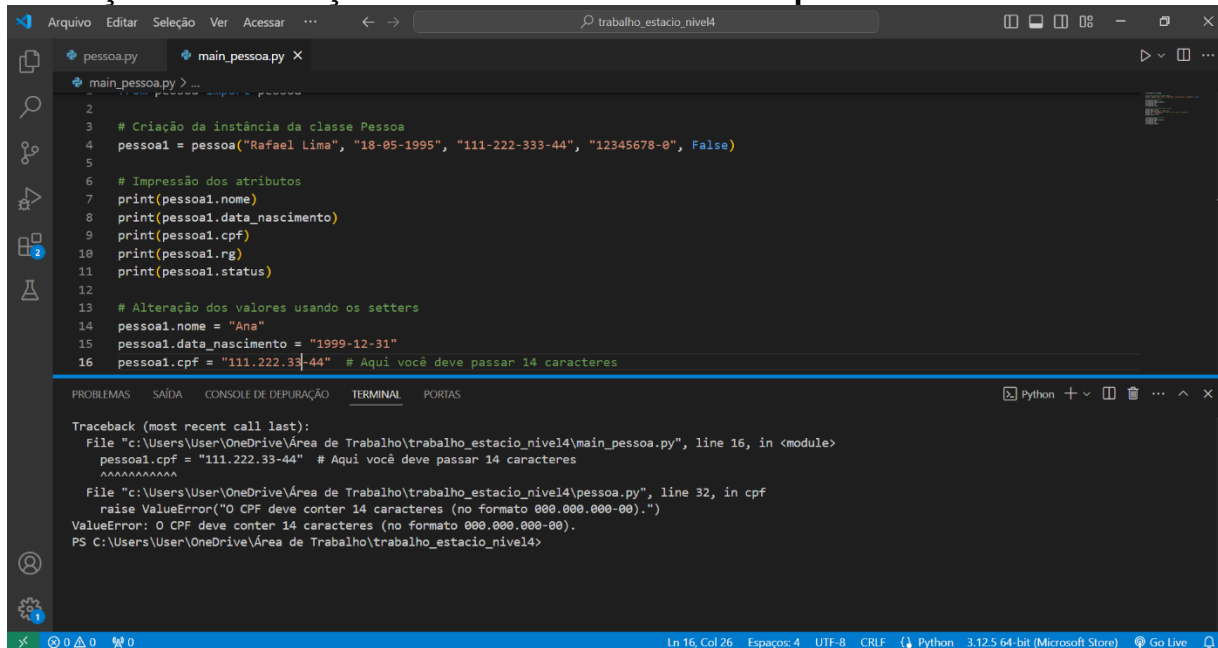
```
1 class Pessoa:
2     def __init__(self, nome, data_nascimento, cpf, rg, status):
3         self.__nome = nome
4         self.__data_nascimento = data_nascimento
5         self.__cpf = cpf
6         self.__rg = rg
7         self.__status = status
8
9     @property
10    def nome(self):
11        return self.__nome
12
13    @nome.setter
14    def nome(self, nome):
15        self.__nome = nome
16
17    @property
18    def data_nascimento(self):
19        return self.__data_nascimento
20
21    @data_nascimento.setter
22    def data_nascimento(self, data_nascimento):
23        self.__data_nascimento = data_nascimento
24
25    @property
26    def cpf(self):
27        return self.__cpf
28
29    @cpf.setter
30    def cpf(self, cpf):
31        if len(cpf) != 14:
32            raise ValueError("O CPF deve conter 14 caracteres (no formato 000.000-000-00).")
33        self.__cpf = cpf
34
35    @property
36    def rg(self):
37        return self.__rg
38
39    @rg.setter
40    def rg(self, rg):
41        self.__rg = rg
42
43    @property
44    def status(self):
45        return self.__status
46
47    @status.setter
48    def status(self, status):
49        self.__status = status
50
```

Alteração do código: main_pessoa.py



```
1 print(pessoa1.nome)
2 print(pessoa1.data_nascimento)
3 print(pessoa1.cpf)
4 print(pessoa1.rg)
5 print(pessoa1.status)
6
7 # Alteração dos valores usando os setters
8 pessoa1.nome = "Ana"
9 pessoa1.data_nascimento = "1999-12-31"
10 pessoa1.cpf = "111.222.333-44" # Aqui você deve passar 14 caracteres
11 pessoa1.rg = "12345678-9"
12 pessoa1.status = True
13
14 # Impressão após as alterações
15 print(pessoa1.nome)
16 print(pessoa1.data_nascimento)
17 print(pessoa1.cpf)
18 print(pessoa1.rg)
19 print(pessoa1.status)
```

Execução com alteração inserindo valor menor do que 14 caracteres:



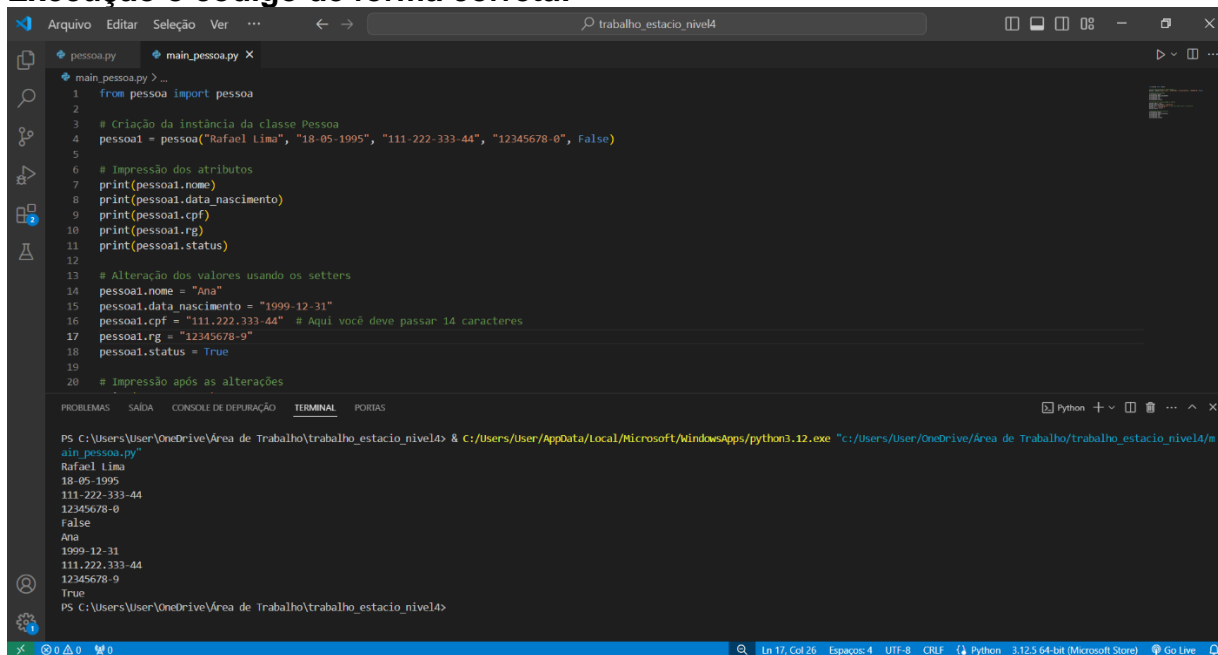
The screenshot shows a Python IDE with a file named `main_pessoa.py`. The code defines a `Pessoa` class and creates an instance `pessoa1`. It then prints the attributes and updates them using setters. The error occurs on line 16, where the `cpf` is set to `"111.222.33-44"`, which is only 13 characters long. The terminal output shows a `ValueError` with the message: "O CPF deve conter 14 caracteres (no formato 000.000.000-00)." (The CPF must contain 14 characters in the format 000.000.000-00).

```
2
3 # Criação da instância da classe Pessoa
4 pessoa1 = pessoa("Rafael Lima", "18-05-1995", "111-222-333-44", "12345678-0", False)
5
6 # Impressão dos atributos
7 print(pessoa1.nome)
8 print(pessoa1.data_nascimento)
9 print(pessoa1.cpf)
10 print(pessoa1.rg)
11 print(pessoa1.status)
12
13 # Alteração dos valores usando os setters
14 pessoa1.nome = "Ana"
15 pessoa1.data_nascimento = "1999-12-31"
16 pessoa1.cpf = "111.222.33-44" # Aqui você deve passar 14 caracteres

PROBLEMAS SAÍDA CONSOLE DE DEPUÇÃO TERMINAL PORTAS

Traceback (most recent call last):
  File "c:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_nivel4\main_pessoa.py", line 16, in <module>
    pessoa1.cpf = "111.222.33-44" # Aqui você deve passar 14 caracteres
    ^^^^^^^^^^^^^
  File "c:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_nivel4\pessoa.py", line 32, in cpf
    raise ValueError("O CPF deve conter 14 caracteres (no formato 000.000.000-00).")
ValueError: O CPF deve conter 14 caracteres (no formato 000.000.000-00).
PS C:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_nivel4>
```

Execução o código de forma correta:



The screenshot shows the same Python IDE with the same code as before, but the `cpf` is now set to `"111.222.333-44"`, which is 14 characters long. The terminal output shows the successful execution of the code, printing the attributes and the updated values.

```
1 from pessoa import pessoa
2
3 # Criação da instância da classe Pessoa
4 pessoa1 = pessoa("Rafael Lima", "18-05-1995", "111-222-333-44", "12345678-0", False)
5
6 # Impressão dos atributos
7 print(pessoa1.nome)
8 print(pessoa1.data_nascimento)
9 print(pessoa1.cpf)
10 print(pessoa1.rg)
11 print(pessoa1.status)
12
13 # Alteração dos valores usando os setters
14 pessoa1.nome = "Ana"
15 pessoa1.data_nascimento = "1999-12-31"
16 pessoa1.cpf = "111.222.333-44" # Aqui você deve passar 14 caracteres
17 pessoa1.rg = "12345678-9"
18 pessoa1.status = True
19
20 # Impressão após as alterações

PS C:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_nivel4> & C:\Users\User\AppData\Local\Microsoft\WindowsApps\python3.12.exe "c:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_nivel4\main_pessoa.py"
Rafael Lima
18-05-1995
111-222-333-44
12345678-0
False
Ana
1999-12-31
111.222.333-44
12345678-9
True
PS C:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_nivel4>
```

Código descrito:

pessoa.py

class pessoa:

```
def __init__(self, nome, data_nascimento, cpf, rg, status):
    self.__nome = nome
    self.__data_nascimento = data_nascimento
    self.__cpf = cpf
    self.__rg = rg
    self.__status = status
```

@property

```
def nome(self):
    return self.__nome
```

@nome.setter

```
def nome(self, nome):
    self.__nome = nome
```

```

@property
def data_nascimento(self):
    return self.__data_nascimento

@data_nascimento.setter
def data_nascimento(self, data_nascimento):
    self.__data_nascimento = data_nascimento

@property
def cpf(self):
    return self.__cpf

@cpf.setter
def cpf(self, cpf):
    if len(cpf) != 14:
        raise ValueError("O CPF deve conter 14 caracteres (no formato 000.000.000-00).")
    self.__cpf = cpf

@property
def rg(self):
    return self.__rg

@rg.setter
def rg(self, rg):
    self.__rg = rg

@property
def status(self):
    return self.__status

@status.setter
def status(self, status):
    self.__status = status

```

main_pessoa.py

```
from pessoa import pessoa
```

```
# Criação da instância da classe Pessoa
```

```
pessoa1 = pessoa("Rafael Lima", "18-05-1995", "111-222-333-44", "12345678-0", False)
```

```
# Impressão dos atributos
```

```
print(pessoa1.nome)
```

```
print(pessoa1.data_nascimento)
```

```
print(pessoa1.cpf)
```

```
print(pessoa1.rg)
```

```
print(pessoa1.status)
```

```
# Alteração dos valores usando os setters
```

```
pessoa1.nome = "Ana"
```

```
pessoa1.data_nascimento = "1999-12-31"
```

```
pessoa1.cpf = "111.222.333-44" # Aqui você deve passar 14 caracteres
```

```
pessoa1.rg = "12345678-9"
```

```
pessoa1.status = True
```

```
# Impressão após as alterações
```

```
print(pessoa1.nome)
```

```
print(pessoa1.data_nascimento)
```

```
print(pessoa1.cpf)
```

```
print(pessoa1.rg)
```

```
print(pessoa1.status)
```

