

Micro atividade 5: Descrever a reutilização de código através de herança no Python

//Disciplina: Nivel4: Conhecendo novos paradigmas

//Professor: Jhonatan Alves

//Aluno: Rafael Lima de Medeiros

//Turma: 2024.2

//Data atual: 20/08/2024

Código pessoa.py

```
Arquivo  Editar  Seleção  Ver  ...  trabalho_estacio_nivel4

Pessoa.py  x  PessoaFisica.py  PessoaJuridica.py  main_pessoa.py

Pessoa.py > _
1 class Pessoa:
2     def __init__(self, nome, numero_conta, data_abertura_conta, status):
3         self.__nome = nome
4         self.__numero_conta = numero_conta
5         self.__data_abertura_conta = data_abertura_conta
6         self.__status = status
7
8     @property
9     def nome(self):
10         return self.__nome
11
12     @nome.setter
13     def nome(self, valor):
14         self.__nome = valor
15
16     @property
17     def numero_conta(self):
18         return self.__numero_conta
19
20     @numero_conta.setter
21     def numero_conta(self, valor):
22         self.__numero_conta = valor
23
24     @property
25     def data_abertura_conta(self):
26         return self.__data_abertura_conta
27
28     @data_abertura_conta.setter
29     def data_abertura_conta(self, valor):
30         self.__data_abertura_conta = valor
31
32     @property
33     def status(self):
34         return self.__status
35
36     @status.setter
37     def status(self, valor):
38         self.__status = valor
39
40     def imprimir_dados(self):
41         print(f'Nome: {self.__nome}')
42         print(f'Número da Conta: {self.__numero_conta}')
43         print(f'Data de Abertura da Conta: {self.__data_abertura_conta}')
44         print(f'Status: {self.__status}')
45
```

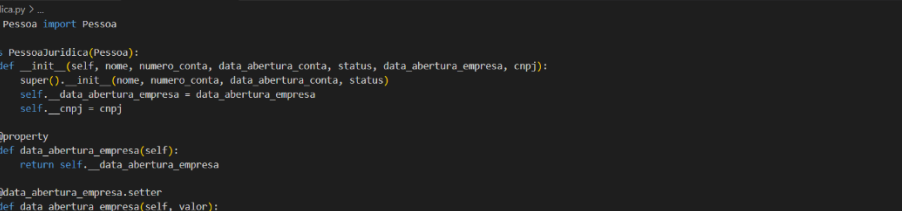
Código PessoaFisica.py

```
Arquivo  Editar  Seleção  Ver  ...  trabalho_estacio_nivel4

Pessoa.py  x  PessoaFisica.py  PessoaJuridica.py  main_pessoa.py

PessoaFisica.py > _
1 from Pessoa import Pessoa
2
3 class PessoaFisica(Pessoa):
4     def __init__(self, nome, numero_conta, data_abertura_conta, status, data_nascimento, cpf, rg):
5         super().__init__(nome, numero_conta, data_abertura_conta, status)
6         self.__data_nascimento = data_nascimento
7         self.__cpf = cpf
8         self.__rg = rg
9
10    @property
11    def data_nascimento(self):
12        return self.__data_nascimento
13
14    @data_nascimento.setter
15    def data_nascimento(self, valor):
16        self.__data_nascimento = valor
17
18    @property
19    def cpf(self):
20        return self.__cpf
21
22    @cpf.setter
23    def cpf(self, valor):
24        if len(valor) == 11:
25            self.__cpf = valor
26        else:
27            raise ValueError("CPF deve ter 11 caracteres.")
28
29    @property
30    def rg(self):
31        return self.__rg
32
33    @rg.setter
34    def rg(self, valor):
35        self.__rg = valor
36
```

Código PessoaJuridica.py



The screenshot shows a code editor with a dark theme. The top bar displays the file path 'trabalho_estacio_nivelM'. The editor has four tabs open: 'Pessoa.py', 'PessoaFisica.py', 'PessoaJuridica.py' (active), and 'main_pessoa.py'. The active tab contains the following Python code:

```

1  from Pessoa import Pessoa
2
3  class PessoaJuridica(Pessoa):
4      def __init__(self, nome, numero_conta, data_abertura_conta, status, data_abertura_empresa, cnpj):
5          super().__init__(nome, numero_conta, data_abertura_conta, status)
6          self._data_abertura_empresa = data_abertura_empresa
7          self._cnpj = cnpj
8
9      @property
10     def data_abertura_empresa(self):
11         return self._data_abertura_empresa
12
13     @data_abertura_empresa.setter
14     def data_abertura_empresa(self, valor):
15         self._data_abertura_empresa = valor
16
17     @property
18     def cnpj(self):
19         return self._cnpj
20
21     @cnpj.setter
22     def cnpj(self, valor):
23         if len(valor) == 18:
24             self._cnpj = valor
25         else:
26             raise ValueError("CNPJ deve ter 18 caracteres.")
27

```

The bottom status bar shows the following information: 'Ln 27, Col 1', 'Espaços: 4', 'UTF-8', 'CRLF', 'Python', '3.12.5 64-bit (Microsoft Store)', and 'Go Live'.

Código main_pessoa.py

```
Arquivo Editar Seleção Ver ... trabalho_estacio_nivel4
Pessoa.py PessoaFisica.py PessoaJuridica.py main_pessoa.py X
main_pessoa.py > ...
1 from Pessoa import Pessoa
2 from PessoaFisica import PessoaFisica
3 from PessoaJuridica import PessoaJuridica
4
5 # Criando instâncias e imprimindo dados
6
7 # Instância de Pessoa
8 pessoa = Pessoa("Rafael Lima", "12345", "18-05-1995", "Ativa")
9 pessoa.imprimir_dados()
10
11 # Instância de PessoaFisica
12 pessoa_fisica = PessoaFisica("Mateus Oliveira", "54321", "2015-05-15", "Ativa", "1990-07-22", "12345678901", "MG-12.345.678")
13 pessoa_fisica.imprimir_dados()
14
15 # Instância de PessoaJuridica
16 pessoa_juridica = PessoaJuridica("Empresa MAXLINE Ltda", "67890", "2020-11-20", "Ativa", "2018-09-10", "12.345.678/0001-95")
17 pessoa_juridica.imprimir_dados()
18
19 # Alterar valores via setter
20 try:
21     pessoa_fisica.cpf = "1234567890" # Valor inválido
22 except ValueError as e:
23     print(e)
24
25 try:
26     pessoa_juridica.cnpj = "12.345.678/0001-9" # Valor inválido
27 except ValueError as e:
28     print(e)
29
30 # Alterar para valores válidos
31 pessoa_fisica.cpf = "12345678901" # Valor válido
32 pessoa_juridica.cnpj = "12.345.678/0001-95" # Valor válido
33
```

Execução do código:

The screenshot displays a Python IDE interface with a file explorer at the top showing three files: `Pessoa.py`, `PessoaFisica.py`, and `PessoaJuridica.py`. The active file is `main_pessoa.py`.

```
1 from Pessoa import Pessoa
2 from PessoaFisica import PessoaFisica
3 from PessoaJuridica import PessoaJuridica
4
5 # Criando instâncias e imprimindo dados
6
7 # Instância de Pessoa
8 pessoa = Pessoa("Rafael Lima", "12345", "18-05-1995", "Ativa")
9 pessoa.imprimir_dados()
10
11 # Instância de PessoaFisica
12 pessoa_fisica = PessoaFisica("Mateus Oliveira", "54321", "2015-05-15", "Ativa", "1990-07-22", "12345678901", "MG-12.345.678")
```

The bottom panel shows the output of the program:

```
Nome: Rafael Lima
Número da Conta: 12345
Data de Abertura da Conta: 18-05-1995
Status: Ativa
Nome: Mateus Oliveira
Número da Conta: 54321
Data de Abertura da Conta: 2015-05-15
Status: Ativa
Nome: Empresa MAXLINE Ltda
Número da Conta: 67890
Data de Abertura da Conta: 2020-11-20
Status: Ativa
CPF deve ter 11 caracteres.
CNPJ deve ter 18 caracteres.
PS C:\Users\User\active\Area de Trabalho\trabalho_estacio_nivel4>
```

Código digitado:

pessoa.py

```
class Pessoa:
    def __init__(self, nome, numero_conta, data_abertura_conta, status):
        self.__nome = nome
        self.__numero_conta = numero_conta
        self.__data_abertura_conta = data_abertura_conta
        self.__status = status

    @property
    def nome(self):
        return self.__nome

    @nome.setter
    def nome(self, valor):
        self.__nome = valor

    @property
    def numero_conta(self):
        return self.__numero_conta

    @numero_conta.setter
    def numero_conta(self, valor):
        self.__numero_conta = valor

    @property
    def data_abertura_conta(self):
        return self.__data_abertura_conta

    @data_abertura_conta.setter
    def data_abertura_conta(self, valor):
        self.__data_abertura_conta = valor

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

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

    def imprimir_dados(self):
        print(f"Nome: {self.__nome}")
        print(f"Número da Conta: {self.__numero_conta}")
        print(f"Data de Abertura da Conta: {self.__data_abertura_conta}")
        print(f>Status: {self.__status}")
```

PessoaFisica.py

from Pessoa import Pessoa

```
class PessoaFisica(Pessoa):
    def __init__(self, nome, numero_conta, data_abertura_conta, status, data_nascimento, cpf, rg):
        super().__init__(nome, numero_conta, data_abertura_conta, status)
        self.__data_nascimento = data_nascimento
        self.__cpf = cpf
        self.__rg = rg

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

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

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

    @cpf.setter
    def cpf(self, valor):
        if len(valor) == 11:
            self.__cpf = valor
        else:
            raise ValueError("CPF deve ter 11 caracteres.")
```

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

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

PessoaJuridica.py

```
from Pessoa import Pessoa
```

```
class PessoaJuridica(Pessoa):
    def __init__(self, nome, numero_conta, data_abertura_conta, status, data_abertura_empresa, cnpj):
        super().__init__(nome, numero_conta, data_abertura_conta, status)
        self.__data_abertura_empresa = data_abertura_empresa
        self.__cnpj = cnpj

    @property
    def data_abertura_empresa(self):
        return self.__data_abertura_empresa

    @data_abertura_empresa.setter
    def data_abertura_empresa(self, valor):
        self.__data_abertura_empresa = valor

    @property
    def cnpj(self):
        return self.__cnpj

    @cnpj.setter
    def cnpj(self, valor):
        if len(valor) == 18:
            self.__cnpj = valor
        else:
            raise ValueError("CNPJ deve ter 18 caracteres.")
```

main_pessoa.py

```
from Pessoa import Pessoa
from PessoaFisica import PessoaFisica
from PessoaJuridica import PessoaJuridica

# Criando instâncias e imprimindo dados

# Instância de Pessoa
pessoa = Pessoa("Rafael Lima", "12345", "18-05-1995", "Ativa")
pessoa.imprimir_dados()

# Instância de PessoaFisica
pessoa_fisica = PessoaFisica("Mateus Oliveira", "54321", "2015-05-15", "Ativa", "1990-07-22", "12345678901", "MG-12.345.678")
pessoa_fisica.imprimir_dados()

# Instância de PessoaJuridica
pessoa_juridica = PessoaJuridica("Empresa MAXLINE Ltda", "67890", "2020-11-20", "Ativa", "2018-09-10", "12.345.678/0001-95")
pessoa_juridica.imprimir_dados()

# Alterar valores via setter
try:
    pessoa_fisica.cpf = "1234567890" # Valor inválido
except ValueError as e:
    print(e)

try:
    pessoa_juridica.cnpj = "12.345.678/0001-9" # Valor inválido
except ValueError as e:
    print(e)

# Alterar para valores válidos
pessoa_fisica.cpf = "12345678901" # Valor válido
pessoa_juridica.cnpj = "12.345.678/0001-95" # Valor válido
```