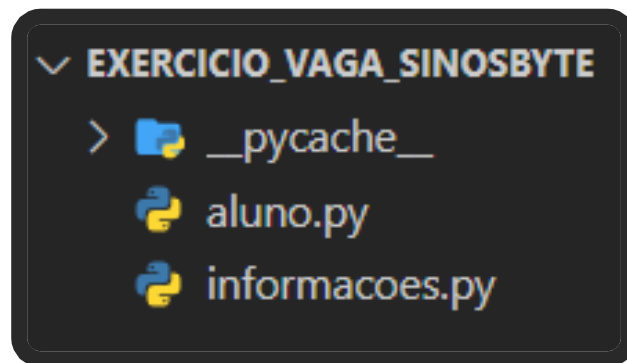


Nome: Hassan Augusto Bittencourt de Oliveira

Data: 27/01/2022

Exercício de Programação Sinosbyte

1. Primeiramente, criamos a pasta “exercicio_vaga_sinosbyte” e nela criamos dois arquivos (“aluno.py” e “informações.py”) em linguagem Python para melhor organização e resolução do problema.



2. No arquivo “aluno.py” temos a classe chamada “Aluno”, a qual temos alguns métodos. O primeiro método é a “__init__” que tem o propósito de inicializar as propriedades da classe para um objeto específico, “self” representa o objeto que herdará essas propriedades.

```
aluno.py > ...
1 class Aluno:
2
3     def __init__(self: object, codigo: int, nome: str, nota1: float, nota2: float) -> None:
4         self.__codigo: int = codigo
5         self.__nome: str = nome
6         self.__nota1: float = nota1
7         self.__nota2: float = nota2
8
```

3. Adiante, temos alguns métodos property para cada parâmetro do objeto e para métodos auxiliares no desenvolvimento da questão. A função property cria e retorna um objeto de propriedade (ao invés de criar métodos getter(), setter() e delete()), economizamos linhas com o property.

```
9      @property
10     def codigo(self: object) -> int:
11         |         return self.__codigo
12
13     @property
14     def nome(self: object) -> str:
15         |         return self.__nome
16
17     @property
18     def nota1(self: object) -> float:
19         |         return self.__nota1
20
21     @property
22     def nota2(self: object) -> float:
23         |         return self.__nota2
24
```

4. Os métodos auxiliares ditos acima foi o “media” e o “aprovado”. O método “media” calcula a média das notas e a retorna; e o método “aprovado” verifica e retorna se foi aprovado ou não de acordo com a média das notas.

```
25     @property
26     def media(self: object) -> float:
27         |         media: float = ((self.__nota1 + (2*self.__nota2))/3)
28         |         return media
29
30     @property
31     def aprovado(self: object) -> str:
32         |         if self.media >= 6:
33         |             return 'SIM'
34         |         if self.media < 6:
35         |             return 'NÃO'
```

5. Por último, temos o método “__str__” o qual retorna uma representação de string do objeto.

```
37 def __str__(self) -> str:
38     return f'Código: {self.codigo} \nNome: {self.nome} \nNota 1: {self.nota1} \nNota 2: {self.nota2} \nMédia: {self.media} \nAprovado: {self.aprovado}'
39
```

6. Finalizando o arquivo “aluno.py”, agora vamos para o arquivo principal “informações.py” o qual primeiramente contém a lista chamada “alunos”, e para cada índice dessa lista temos um dicionário com informações de cada aluno.

```
4 alunos = [
5     {'codigo': 1001, 'nome': 'Hassan Oliveira', 'nota1': 9.6, 'nota2': 9.1},
6     {'codigo': 1002, 'nome': 'Anderson Rodrigues', 'nota1': 8.3, 'nota2': 8.7},
7     {'codigo': 1003, 'nome': 'Joana Osasco', 'nota1': 6.7, 'nota2': 7.5},
8     {'codigo': 1004, 'nome': 'Gabriela Nunes', 'nota1': 9.8, 'nota2': 9.2},
9     {'codigo': 1005, 'nome': 'João Vitor', 'nota1': 4.2, 'nota2': 6.3},
10 ]
```

7. Adiante, criei uma lista vazia chamada “Lista_de_alunos”, a qual pela estrutura de repetição “for”, usarei para guardar as informações do objeto (inserindo no banco de dados) com as informações de cada aluno, por isso a criação dos dicionários.

```
12 Lista_de_alunos = []
13
14 for aluno in alunos:
15     Lista_de_alunos.append(Aluno(aluno['codigo'], aluno['nome'], aluno['nota1'], aluno['nota2']))
16
```

8. Agora, temos o método “solicitacao” a qual recebe como parâmetro o valor do “codigo” que o usuário digita na linha 25. Por meio de outra estrutura de repetição “for” e um “if” indentado, percorremos aluno por aluno no banco de dados e comparamos o “codigo” digitado com o “codigo” dos alunos. Não encontrando o “codigo”, o programa solicita novamente que entre com outro valor. E assim, na linha 26 imprimimos as informações do aluno em que o “codigo” de entrada é igual ao “codigo” do aluno.

```
17 def solicitacao(codigo: int) -> aluno:
18     for aluno in Lista_de_alunos:
19         if codigo == aluno.codigo:
20             return aluno
21         else:
22             codigo: int = int(input('Digite o código do aluno (1001 a 1005): '))
23             return solicitacao(codigo)
24
25     codigo: int = int(input('Digite o código do aluno (1001 a 1005): '))
26     aluno = solicitacao(codigo)
27     print(aluno)
```