

Computação II

Laboratório 2

Entregue todas as classes e métodos em um arquivo chamado `lab2.py`. Critérios de avaliação:

Principalmente (50%) o código funciona e os métodos fazem o que foi pedido.

Adicionalmente (25%) o código é legível, bem organizado, com *docstrings* concisos.

Adicionalmente (25%) o código é eficaz, não faz operações desnecessárias.

1. No link <https://drive.google.com/file/d/1MYM1MzMfmZEZj40QS7oBWS947vdNg08X/view?usp=sharing> você encontra um código, onde estão definidas as classes `Aluno` e `Disciplina`. Para a classe `Disciplina`:

- (a) (1 ponto) Defina o método chamado `consultarVagas` sem parâmetro de entrada, cujo valor de retorno é uma string em formato como no exemplo abaixo:

```
>>> MAB241 = Disciplina("Computação II", 20)
>>> MAB241.consultarVagas()
'Vagas totais: 20. Vagas livres: 20'
>>> fulana1 = Aluno("Eva", 123456)
>>> MAB241.inscreverAluno(fulana1)
>>> MAB241.consultarVagas()
'Vagas totais: 20. Vagas livres: 19'
```

- (b) (3 pontos) Implemente o método especial `__str__` cujo valor de retorno é uma string com o nome da disciplina, lista dos alunos inscritos com os dados deles, e o número de vagas como neste exemplo:

```
>>> MAB241 = Disciplina("Computação II", 20)
>>> print(MAB241)
Computação II, sem alunos inscritos.
Vagas totais: 20. Vagas livres: 20
>>> fulana1 = Aluno("Eva", 123456)
>>> fulana2 = Aluno("Olma", 100000, "trancada")
>>> MAB241.inscreverAluno(fulana1)
>>> MAB241.inscreverAluno(fulana2)
>>> print(MAB241) # chamando o método especial __str__
Computação II, alunos inscritos:
Eva 123456 matricula ativa
Olma 100000 matricula trancada
Vagas totais: 20. Vagas livres: 18
>>> str(MAB241) # outra forma de chamar o método especial __str__
'Computação II, alunos inscritos:\nEva\t123456\tmatricula ativa\nOlma\t100000
\tmatricula trancada\nVagas totais: 20. Vagas livres: 18'
```

2. Crie a classe `Pessoa` que tem os atributos `nome`, `dataNascimento`, `nomeDeMae`, `nomeDePai` e os métodos:

- (a) (1 ponto) O construtor `__init__` com os parâmetros de entrada iguais aos atributos acima listados. A data de nascimento será passada como uma string em formato `"dd/mm/aaaa"`.
- (b) (4 pontos) O método chamado `idade` com o parâmetro `data`. O parâmetro `data` tem o valor *default* igual a `date.today().strftime("%d/%m/%Y")`, ou seja, a data de hoje em formato da seguinte string: `"dd/mm/aaaa"` (importe a classe `date` do módulo `datetime` antes de começar a definir o método `idade`). O método retorna um inteiro que é a idade em anos no dia `data`.
- (c) (1 ponto) O método especial `__str__` que retorna uma string com os dados de um objeto da classe `Pessoa` como no exemplo abaixo:

```
>>> pessoa1 = Pessoa("Paula", '29/04/2000', "Maria", "Joao")
>>> print(pessoa1)
nome: Paula, idade: 20, mae: Maria, pai: Joao
```

Observação: o método `__str__` deve chamar o método `idade` para obter a idade.