

Data de Entrega: **29/09/2021 (23:59)**

Grupos: máximo de 6 alunos

DESCRIÇÃO:

Crie uma classe que represente um **Clube**. Essa classe possui uma lista de objetos criados a partir de uma classe chamada **Socio**.

- A lista de sócios do clube deve ser inicializada como vazia.

A classe **Socio** contém as seguintes informações sobre o sócio do clube: nome, cpf, data de nascimento (formato DD/MM/AAAA), o mês (1 a 12) e o ano em que a pessoa se associou (com 4 dígitos).

A classe **Clube** deve implementar um método chamado **associar** para poder inscrever novos sócios no clube.

A classe **Clube** também deve possuir um método **numero_de_socios**, que retorna a quantidade de sócios associados ao clube.

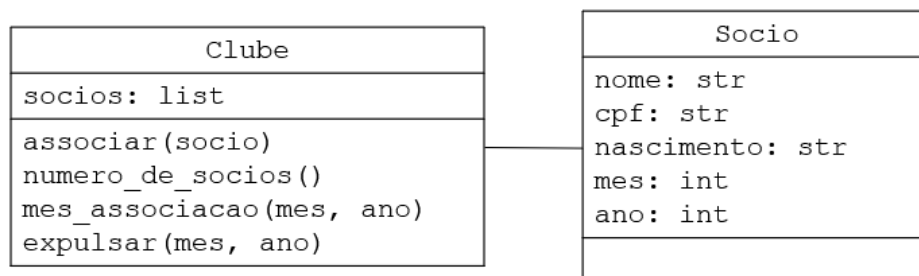
A classe **Clube** deve ter o método **mes_associacao** que retorna quantos sócios se associaram ao clube em um determinado mês e ano.

- Se ninguém se associou no mês/ano informado, o método deve retornar zero.
- Se o mês informado como parâmetro não estiver entre 1 e 12, o método deve gerar uma exceção `TypeError` (*deve apenas gerar a exceção, utilizando `raise`, não deve ser inserido `try/except` na classe*).
- Se o ano informado como parâmetro não conter quatro dígitos, o método deve gerar uma exceção `ValueError` (*deve apenas gerar a exceção, utilizando `raise`, não deve ser inserido `try/except` na classe*).

Defina também um método chamado **expulsar** que elimine todos os sócios que se associaram no mês e ano informado, e retorne uma lista com os nomes dos sócios expulsos, em ordem alfabética.

- Se não houver nenhum sócio para ser expulso no mês/ano informado, deve retornar uma lista vazia.
- Se o mês informado como parâmetro não estiver entre 1 e 12, o método deve gerar uma exceção `TypeError` (*deve apenas gerar a exceção, utilizando `raise`, não deve ser inserido `try/except` na classe*).
- Se o ano informado como parâmetro não conter quatro dígitos, o método deve gerar uma exceção `ValueError` (*deve apenas gerar a exceção, utilizando `raise`, não deve ser inserido `try/except` na classe*).

DIAGRAMA DE CLASSES:



CRITÉRIOS DE AVALIAÇÃO:

- 8.0 pontos: Implementação e execução correta do programa conforme as especificações acima e conforme ilustrado no diagrama de classes.
- 2.0 pontos: Formatação e organização do código de acordo com o padrão PEP8.

ATENÇÃO:

- Não serão aceitos trabalhos entregues em atraso.
- Não serão aceitos trabalhos com mais de 6 alunos no grupo.
- Não deve ser utilizado input dentro das classes.
- Entreguem apenas o arquivo ac02.py, sem renomeá-lo.
 - Pode ser entregue por apenas um aluno do grupo, mas não esqueçam de colocar o nome de todos os integrantes no código do programa.
- Arquivos em outros formatos que não sejam .py, não serão aceitos (doc, pdf, txt, link para git, etc)
- O arquivo ac02_teste.py contém um subconjunto de testes que serão usados na correção da atividade, para que vocês possam testar o funcionamento das classes implementadas e ter uma avaliação prévia do que estão fazendo antes de submeterem a atividade.
 - Outros testes poderão ser realizados na correção da atividade, para verificar se a implementação está de acordo com o solicitado.
- Se for identificada cópia de trabalhos entre os grupos, a nota será zerada.