

Data de Entrega: 04/04/2021 (23:59)

Grupos: máximo de 6 alunos

DESCRIÇÃO:

As escolas de magia formam vários bruxos em um determinado ano. Uma escola contém um nome e é formada por várias casas.

As casas também possuem um nome e um animal como símbolo. Em cada casa existe um diretor e um único aluno que é monitor.

Os alunos bruxos são identificados pelo seu nome e data de nascimento. Os alunos também são classificados em três tipos, de acordo com sua descendência, podendo ser *puro-sangue*, *trouxa* ou *mestiço*.

Um aluno bruxo ao entrar na escola é selecionado para morar em uma casa pelo chapéu seletor.

Na escola existe vários professores-bruxos que são identificados pelo seu nome e data de nascimento. Os professores podem ministrar várias disciplinas em um determinado ano.

Cada disciplina possui um nome e uma ementa. E os alunos podem frequentar várias disciplinas.

Durante as aulas os alunos recebem pontos dos professores, que podem ser positivos, quando realizam um *trunfo*, ou negativos, quando realizam um *mau-feito*.

A partir do cenário descrito e do diagrama de classes apresentado, implemente em Python as classes solicitadas, aplicando os conceitos de programação orientada a objetos.

ATENÇÃO:

- Será disponibilizado um arquivo de teste para auxiliar na verificação das classes. Outros testes poderão ser realizados posteriormente para correção da atividade..
- Insira no início do programa um comentário com os nomes dos alunos do grupo.
- Nomes das classes, atributos e métodos devem ser exatamente iguais ao apresentado no enunciado e no diagrama de classes. O programa será corrigido automaticamente e, caso algum nome seja alterado, não será incluído na correção.
- Não serão aceitos trabalhos que não sejam arquivos Python (doc, pdf, txt, links para git, etc.)
- Métodos e atributos adicionais podem ser incluídos caso você ache necessário, desde que os atributos e métodos descritos no enunciado sejam mantidos.
- Caso seja identificada cópia entre trabalhos, a nota dos grupos envolvidos será zerada.
- Não serão aceitos trabalhos entregues em atraso.
- Nesta atividade, não serão descontados pontos por erros de estilo PEP8.

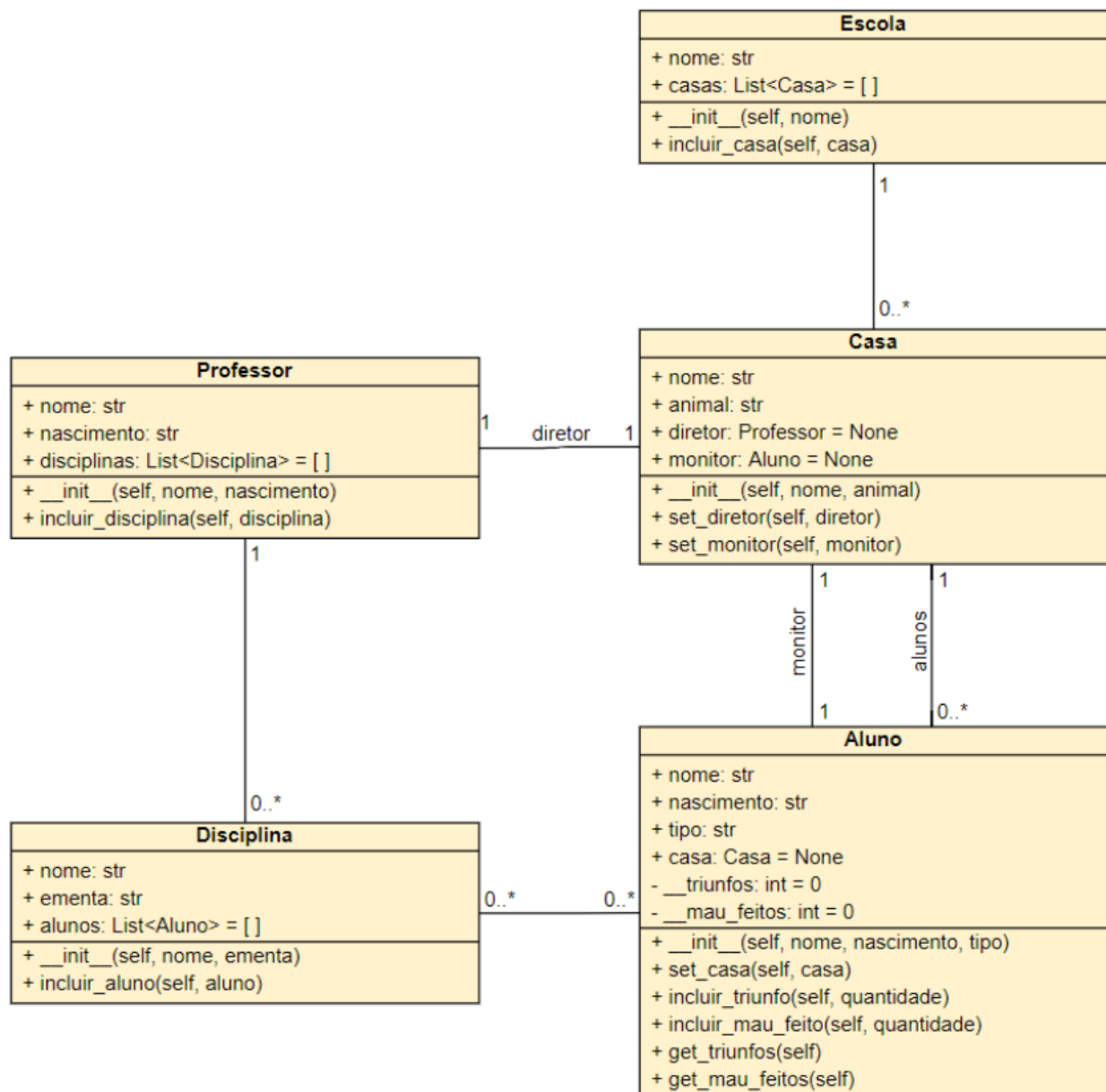
CRITÉRIOS DE AVALIAÇÃO:

- Funcionamento do programa e implementação correta das classes, conforme descrito no enunciado e no diagrama de classes;
- Organização do código respeitando as regras do Padrão PEP8.

FORMA DE ENTREGA:

- Anexar no Classroom os arquivo ac03.py com a implementação solicitada.
- Pode ser enviado por apenas um integrante do grupo (mas não esqueça de colocar o nome de todos os integrantes do grupo no início do arquivo).

DIAGRAMA DE CLASSES:



Classe Escola:

Atributos

- **nome:** nome da escola
- **casas:** lista de objetos da classe Casa. Definido no construtor como uma lista vazia.

Métodos

- **incluir_casa:** recebe como parâmetro um objeto Casa e inclui na lista de casas

Classe Casa:

Atributos

- **nome:** nome da casa
- **animal:** animal símbolo
- **diretor:** objeto da classe Professor. Definido no construtor como None.
- **monitor:** objeto da classe Aluno. Definido no construtor como None.

Métodos

- **set_diretor:** recebe como parâmetro um objeto Professor e o define como diretor da casa.
- **set_monitor:** recebe como parâmetro um objeto Aluno e o define como monitor de casa

Classe Professor:

Atributos

- **nome:** nome do professor
- **nascimento:** data de nascimento no formato AAAAMMDD.
- **disciplinas:** lista de objetos da classe Disciplina. Definido no construtor como uma lista vazia.

Métodos:

- **incluir_disciplina:** recebe como parâmetro um objeto Disciplina e inclui na lista de disciplinas.

Classe Disciplina:

Atributos

- **nome:** nome da disciplina
- **ementa:** ementa da disciplina
- **alunos:** lista de objetos da classe Aluno. Definido no construtor como uma lista vazia.

Métodos:

- **incluir_aluno:** recebe como parâmetro um objeto Aluno e inclui na lista de alunos.

Classe Aluno:

Atributos

- **nome:** nome do aluno
- **nascimento:** data de nascimento no formato AAAAMMDD.
- **tipo:** classificação do aluno em puro-sangue, trouxa ou mestiço.
- **casa:** objeto da classe Casa. Definido no construtor como None.
- **__trunfos:** (*privado*) quantidade de triunfos. Definido no construtor como 0 (zero).
- **__mau_feitos:** (*privado*) quantidade de mau-feitos. Definido no construtor como 0 (zero).

Métodos:

- **set_casa:** recebe como parâmetro um objeto Casa e o define como casa do aluno.
- **incluir_trunfo:** recebe a quantidade de pontos e incrementa os triunfos do aluno.
- **incluir_maufeito:** recebe a quantidade de pontos e incrementa os mau-feitos do aluno.
- **get_trunfos:** retorna a quantidade de triunfos do aluno.
- **get_mau_feitos:** retorna a quantidade de mau-feitos do aluno.