

Linguagem de Programação II - Atividade Contínua 5

Atividade em Duplas - Data de Entrega: 19/05/2020 até as 18:50

Objetivos:

Usar todo o arsenal disponível para implementar uma classe que consiga manipular arquivos.

Descrição:

Modelar e implementar uma classe que modele um aluno, que consiga ler seus atributos de um arquivo e escrever a média final e situação (aprovado ou reprovado) em um arquivo, seguindo a seguinte especificação.

Aluno
nome: (string) ra: (int) nota_ac: (list) nota_prova: (float) faltas: (int) media: (float) aprovado: (bool)
__init__(arquivo) calcular_aprovacao() escrever_situacao(arquivo)

Métodos:

A classe prevê a criação dos seguintes métodos públicos:

- **construtor:** usado para ler um arquivo, cujo nome é passado como parâmetro e inicializar os atributos de acordo com o conteúdo do arquivo. A especificação do arquivo estará na próxima parte do enunciado.
- **calcular_aprovacao():** método que calcula a média e a situação do aluno, atribuindo valores aos atributos media e aprovado. Os critérios são os mesmos usados na Faculdade Impacta e o número total de aulas é 60.
- **escrever_situacao():** método que escreve em um arquivo, cujo nome é passado como parâmetro, a situação do aluno. A especificação do arquivo estará na próxima parte do enunciado.

Métodos auxiliares podem ser criados nesta classe, se necessário.

Arquivo de entrada:

É o arquivo que deve ser lido pelo construtor para atribuir valores aos atributos da classe. É um arquivo de texto (.txt) com a seguinte formatação:

Formato	Exemplo
Nome RA Nota AC1 Nota AC2 Nota AC3 Nota AC4 Nota AC5 Nota Prova Faltas	Marcos da Silva 1234567 5.0 9.4 7.9 0.0 5.5 5.5 12

Arquivo de saída:

O É o arquivo que deve ser criado pelo método `escrever_situacao()`. É um arquivo de texto (.txt) com a seguinte formatação:

Formato	Exemplo
RA: Nome Média Final (1 casa decimal) Frequência (%) (1 casa decimal) Situação	1234567: Marcos da Silva 6.2 80.0% Aprovado

Atenção:

- Não serão aceitos trabalhos entregues em atraso.
- A classe, atributos e métodos devem ter, exatamente, os mesmos nomes descritos na especificação.
- Insira no arquivo do programa um comentário com os nomes dos alunos da dupla.
- O arquivo deve seguir as boas práticas de estilo do PEP8. Caso o código não siga o padrão de código, haverá um desconto de 0.5 ponto por erro de padronização.
- A detecção de cópia de parte ou de todo código-fonte, de qualquer origem, implicará em nota zero.
- Erros de execução resultarão em nota zero.

Forma de Entrega:

Postar no Classroom o arquivo `ac05.py`, um por dupla.

O aluno da dupla que não fizer a entrega, por favor não marcar nada no Classroom.