

Linguagem de Programação II

Atividade Contínua 3

Atividade em Duplas - Data de Entrega: 14/04/2020 até 18h50

Objetivos:

- Utilizar boas práticas de programação na resolução de problemas que envolvem conceitos de POO, classes, e objetos.
- Compreender a dinâmica do uso de mais de uma classe simultaneamente.

Descrição:

A AC consiste na criação de um módulo **controle.py**, contendo a classe Elevador, para armazenar as informações e a lógica de controle de um elevador e a classe Predio, que vai modelar a lógica e fazer o controle dos elevadores em um prédio.

Classe Elevador

A classe deve ter os seguintes atributos, todos privados:

- **andar_atual**: o andar atual onde o elevador se encontra (considere térreo = 0).
- **atendidos**: lista com os andares que o elevador atende (todo elevador tem que atender pelo menos o térreo, obrigatoriamente).
- **capacidade**: capacidade máxima de pessoas no elevador.
- **quantidade_pessoas**: quantidade de pessoas atualmente presentes no elevador.

A classe deve também disponibilizar os seguintes métodos:

- **__init__**: construtor que deve receber como parâmetro a **capacidade** e a **lista de andares atendidos** do elevador. Note que um elevador deve sempre começar no andar **térreo** e **vazio**. Se a lista de andares não possuir o andar térreo, você deve adicioná-lo, afinal o elevador é obrigado a atender o térreo.
- **def entrar(self)**: para acrescentar **1** pessoa no elevador (só deve acrescentar se houver espaço).
- **def sair(self)**: para remover **1** pessoa do elevador (só deve remover se houver alguém dentro dele).
- **def subir(self)**: para subir **ao próximo** andar atendido (não deve subir se estiver no mais alto).
- **def descer(self)**: para descer **ao próximo** andar atendido (não deve descer se estiver no mais baixo).
- **def deslocar_para(self, andar)**: para ir à um andar específico (o andar deve ser informado como parâmetro e, caso o andar não esteja na lista de andares atendidos, o elevador não deve se mover).

Classe Predio

A classe deve ter os seguintes atributos, todos privados:

- **elevadores**: uma lista contendo os elevadores disponíveis no prédio (tem que haver pelo menos um elevador no prédio).
- **andar_alto**: o andar mais alto do prédio.
- **andar_baixo**: o andar mais baixo do prédio (pode ser negativo).

A classe deve também disponibilizar os seguintes métodos:

- **__init__**: construtor que deve receber como parâmetros, o **andar mais alto**, e o **andar mais baixo** do prédio e uma **lista não vazia de elevadores** que o prédio possui. O conjunto de elevadores deve atender o prédio todo, isto é, todos os andares do prédio devem ser atendidos por pelo menos um elevador e nenhum elevador pode atender um andar que não existe. Caso o conjunto de elevadores não satisfaça a condição de atendimento, deve dar um erro, basta simplesmente usar o comando visto na aula de tratamento de erros: **raise ValueError** (que por sinal, é uma classe).
- **def chamar(self, andar)**: realiza um chamado em um andar dado e desloca um elevador para esse andar, seguindo os seguintes critérios:
 - Elevador mais próximo que atenda esse andar.
 - Em caso de empate, o que for descer.
 - Caso ainda esteja empatado, o que estiver mais vazio
 - Finalmente, caso ainda haja um empate, pegue o com menor índice da lista.Após descobrir qual elevador será deslocado, o método faz o deslocamento do elevador e chama um dos métodos privados a seguir.
- **def __embarque(self, indice_elevador)**: toda vez que um elevador chega a um andar acima do térreo ($\text{andar} > 0$), uma pessoa entra nele, se possível.
- **def __desembarque(self, indice_elevador)**: toda vez que um elevador chega ao térreo ou subsolo ($\text{andar} \leq 0$), todas as pessoas saem do elevador.

Atenção :

- Não serão aceitos trabalhos entregues em atraso.
- Caso seja identificado plágio de trabalhos, a nota será zerada.
- Caso algum método presente no arquivo **controle.py** apresente algum erro de execução, a nota da AC será zerada.
- Insira no arquivo um comentário com os nomes completos e RA dos alunos da dupla.
- O módulo **controle.py** não deve ter nenhum código executável, apenas as duas classes. Caso haja código executável nesse arquivo, nada feito, haverá um desconto de **2 pontos** na nota final. Claramente, vocês podem (e devem) fazer um programa lateral para testar as classes de vocês, mas não é para entregar.
- Os nomes do arquivo e dos métodos não devem ser alterados, bem como os parâmetros de cada método. Caso algum nome seja alterado, haverá desconto de **1 ponto** na nota final.
- 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**.

Forma de Entrega: Apenas um dos alunos da duplas postar no Classroom o arquivo **controle.py**.

O aluno da dupla que não fez a entrega, peço que NÃO MARQUE no Classroom como atividade entregue, não é necessário e dificulta a minha correção.