

IFPI

CURSO: TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

DISCIPLINA: PROGRAMAÇÃO ORIENTADA À OBJETOS

PROF.: ROGÉRIO BATISTA

LISTA DE EXERCÍCIOS – Implementação da classe: Estações de rádio

1. Dado a classe RadioFM descrita abaixo:

```
estações = {89.5: 'Cocais', 91.5: 'Mix', 94.1: 'Boa', 99.1: 'Clube'}  
class RadioFM:  
    def __init__(self, vol_max, estações):  
        self.volume_min=0  
        self.volume_max=vol_max  
        self.freq_min=88  
        self.freq_max=108  
        self.estações = estações  
        self.volume=None  
        self.ligado=False  
        self.estaçao_atual=None  
        self.frequencia_atual=None  
        self.antena_habilitada=False  
  
    def ligar(self):  
        pass  
    def desligar(self):  
        pass  
    def aumentar_volume(self):  
        pass  
    def diminuir_volume(self):  
        pass  
    def mudar_frequencia(self):  
        pass
```

- Implementar encapsulamento nesse código
- Implemente todos os métodos descritos acima:
 - No método ligar, atualizar o atributo ligado para True, o volume do rádio deverá ser inicializado com o volume mínimo do rádio. Se a antena estiver habilitada (antena_habilitada=True), a frequencia deverá ser inicializada com a frequencia da primeira estação de rádio definida no dicionário e a estação atual deverá ser inicializada com seu respectivo nome.

- No método desligar, mudar o estado para False, além de atualizar os atributos: volume, frequencia_atual e estação_atual para: None
- O método aumentar_volume deverá receber um atributo opcional com valor inicial igual a 1. Caso este valor seja passado na chamada do argumento, atualizar o volume com o valor do argumento (fazer a crítica para não ultrapassar os valores permitidos para o volume). Caso o argumento fique vazio na chamada, o atributo volume deverá ser incrementado de uma unidade. (Fazer a crítica para não ultrapassar o valor máximo permitido)
- Idem para o método: diminuir_volume
- O método mudar_frequencia deverá receber um atributo opcional com valor inicial igual a 0. Caso seja passado um valor para a frequencia na chamada deste método, atualizar o atributo frequencia_atual para o valor passado no argumento. Se o valor da frequencia estiver presente no dicionário, atualizar o atributo: estação_atual com o nome da respectiva frequencia que se encontra no dicionário, caso contrário atualizar com o valor: 'estação inexistente'. Caso este método seja chamado sem argumentos, atualizar a frequencia_atual com a frequência referente ao próximo elemento do dicionário e atualizar o atributo: estação_atual com seu respectivo nome. Se a frequencia atual for igual ao último elemento do dicionário, mudar os atributos: frequencia e estação para os respectivos valores referentes ao primeiro elemento do dicionário.
- Criar pelo menos 3 objetos e testar os métodos implementados.