

Programação Orientada a Objetos

Instituto Federal do Ceará - Campus Maranguape

Curso Técnico em Informática - prof. Thomaz Maia

Aluno:

Exercícios - Classes e métodos

Instruções:

1. Crie um arquivo `.py` para cada classe (exceto a questão 3 onde devem ter duas classes no mesmo arquivo).
2. Crie um arquivo `.py` para cada questão importando as classes criadas e executando o programa de cada questão.
3. Deixe seu código o mais organizado possível.
4. Envie tudo em um único código `.zip`.

1. Construa a classe **Filme** em Python que obedeça a descrição abaixo:

- A classe deve possuir dois atributos: `titulo` (str) e `duracao_em_minutos` (int)
- Crie um construtor que receba esses dois parâmetros, cada um representando um dos atributos da classe.
- Crie um método chamado `exibir_duracao_em_horas` que converta o valor em minutos para horas e apresente a mensagem “O filme *TITULO* possui *X* horas e *Y* minutos de duração”.

Por exemplo: dado o filme com título TITANIC que possui 194 minutos de duração, a mensagem que deverá ser exibida para o usuário será: “O filme *TITANIC* possui 3 horas e 14 minutos de duração.”

- Crie um objeto do `um_filme_qualquer` do tipo **Filme**. Utilize o construtor da classe passando os valores “Os Vingadores” para o atributo `titulo` e 142 para o atributo `duracao_em_minutos`.
- Chame o método `exibir_duracao_em_horas` para mostrar quantas horas o filme possui.
- Crie um objeto `meu_filme_favorito` do tipo **Filme**, cujo título deve ser o título do seu filme favorito e a duração deve ser iniciada em 100 minutos.
- Altere o atributo `duracao_em_minutos` do objeto `meu_filme_favorito` para a duração correta do filme.
- Chame o método `exibir_duracao_em_horas` do `meu_filme_favorito` para mostrar quantas horas/minutos o filme possui.
- Exiba a mensagem “Os filmes no catálogo são *X* e *Y*”, onde no lugar de *X* deverá aparecer o título do `um_filme_qualquer` e no lugar de *Y* deverá aparecer o título do `meu_filme_favorito`.

2. Crie a classe `Livro` com a descrição abaixo:

- A classe possui os atributos `titulo` (str), `quantidade_de_paginas` (int) e `paginas_lidas` (int). Esses atributos devem ser marcados como “privado”.
- Crie os métodos de *getters* e *setters* para acessá-los.
- Crie um construtor recebendo os três parâmetros representando os atributos da classe sendo dois deles parâmetros *default* sem valor (igual a 0) definido de modo que o usuário possa criar um objeto apenas com o título do livro.
- Crie ainda o método `verificar_progresso` que deverá calcular a porcentagem de leitura do livro até o momento. Para isso, deverá usar os valores dos atributos `quantidade_de_paginas` e `paginas_lidas` através da fórmula:
$$\text{porcentagem} = \text{paginas_lidas} * 100 / \text{quantidade_de_paginas}.$$

O valor da porcentagem deverá ser mostrado na tela conforme a mensagem “*Você já leu X por cento do livro*”, onde X é o valor calculado pela fórmula apresentado anteriormente.
- Crie um objeto `pequeno_principe` do tipo `Livro`. Utilize o construtor que recebe apenas o título como parâmetro e passe o valor “O Pequeno Príncipe”.
- Altere o atributo `quantidade_de_paginas` para 98. Utilize, para isso, o *setter* desse atributo.
- Escreva na tela a mensagem “*O livro X possui Y páginas*”, onde no lugar de X deverá aparecer o valor do atributo `titulo` e no lugar de Y deverá aparecer o valor do atributo `quantidade_de_paginas` do objeto `pequeno_principe`. Utilize, para tanto, os *getters* desse atributo.
- Altere a quantidade de páginas para 20.
- Chame o método `verificar_progresso`.
- Altere a quantidade de páginas para 50.
- Chame o método `verificar_progresso`.
- Crie um novo objeto chamado `meu_livro`. Utilize o construtor que recebe 3 parâmetros e passe as informações do livro que você está lendo nesse momento (qualquer livro).
- Escreva na tela a mensagem: “*O livro X possui Y páginas*”, onde no lugar de X deverá aparecer o valor do atributo `titulo` e no lugar de Y deverá aparecer o valor do atributo `quantidade_de_paginas` do objeto `meu_livro`.

3. Crie uma classe `Pessoa` com os atributos `nome`, `data_de_nascimento` e `CPF`. Crie um método para calcular a idade da pessoa e um método para exibir as informações dela. Crie uma segunda classe `Agenda` que pode armazenar 10 pessoas e que seja capaz de realizar as seguintes operações:

```
def armazenar_pessoa(self, nome, nascimento, cpf)
def remover_pessoa(self, nome)
def buscar_pessoa(self, nome) # informa em que posição da agenda está a pessoa
def imprimir_agenda(self) # mostra os dados de todas as pessoas na agenda
def imprimir_pessoa(self, index) # mostra os dados da pessoa na posição 'index' da agenda
```

4. Crie uma classe chamada **Elevador** para armazenar as informações de um elevador dentro de um prédio. A classe deve armazenar o andar atual (térreo = 0), total de andares no prédio (desconsiderando o térreo), capacidade do elevador e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:

inicializa - que deve receber como parâmetros a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazio)

entra - para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço)

sai - para remover uma pessoa (só deve remover se houver alguém dentro dele)

sobe - para subir um andar (não deve subir se já estiver no último)

desce - para descer um andar (não deve descer se já estiver no térreo)