

Programação Orientada a Objetos – 2023/1

PROVA DO GRAU B

Resolver as questões com consulta somente ao material apresentado em aula e aos códigos-fonte dos exercícios resolvidos pelo próprio aluno. Não é permitido consultar outros colegas, a Internet, e-mail, drivers virtuais na nuvem, redes sociais ou telefone celular. Responder as questões na linguagem Python 3. Cada questão deve ter seu próprio arquivo-fonte, identificado com o número da questão (1.py, 2.py, etc). **Os arquivos devem ser enviados pelo Canvas até as 22h15.** O código fonte deve estar **corretamente indentado** e comentado (se necessário). Os nomes das variáveis e funções devem ser autoexplicativos. **Os algoritmos não devem ser *hard-coded*.**

1. Uma missão espacial está explorando o sistema solar e coletando informações sobre os corpos celestes que o compõem. Para auxiliar nessa tarefa, você deve implementar um programa utilizando a linguagem de programação de sua escolha.

Crie uma classe base chamada "CorpoCeleste" com os seguintes atributos:

- Nome (string)
- Diâmetro (float)
- Composição (string)

A classe base "CorpoCeleste" deve ter o método **exibirInformacoes()** que exibe na tela as informações gerais do corpo celeste, incluindo o nome, diâmetro e composição.

A partir da classe base "CorpoCeleste", crie as classes derivadas "Planeta", "Satelite" e "Estrela".

A classe "Planeta" deve ter os seguintes atributos adicionais:

- Número de Luas (inteiro)
- Tipo de Atmosfera (string)

A classe "Satelite" deve ter os seguintes atributos adicionais:

- Planeta de Órbita (string)
- Período de Órbita (float)

A classe "Estrela" deve ter os seguintes atributos adicionais:

- Temperatura (float)
- Tipo Espectral (string)

Todas as classes derivadas devem **implementar o método exibirInformacoes()**, para exibir informações específicas do corpo celeste.

Após implementar as classes, crie uma lista chamada **corposCelestes** e instancie objetos das classes "Planeta", "Satelite" e "Estrela", preenchendo as informações necessárias. Adicione os objetos à lista.

Em seguida, percorra a lista **corposCelestes** e chame o método **exibirInformacoes()** para cada objeto. Verifique se as informações estão sendo exibidas corretamente, considerando a especialização dos métodos.

2. No Japão, o Jo ken po é um jogo muito popular, usado em diversas ocasiões para encontrar soluções para pequenos conflitos. A batalha de Jo ken po que simularemos aqui é um jogo de equipes, em que duas equipes competem para acumular 10 pontos antes de sua adversária. Cada equipe é representada pela **classe Equipe**, que possui um nome e um número de vitórias da equipe. Cada equipe é composta por três competidores, representados pela **classe Competidor**, que possuem um nome e um número de vitórias pessoais, que começa em zero.

As regras do jogo são as seguintes:

- 1) No início de cada round, cada equipe escolhe um competidor para representá-la no round atual.
- 2) Os competidores selecionados de cada equipe competem entre si usando o jogo de Pedra, Papel, Tesoura. As opções possíveis são: **Pedra (0)**, **Papel (1)** ou **Tesoura (2)**.
- 3) O resultado do round é determinado da seguinte maneira:
 - Pedra vence Tesoura
 - Tesoura vence Papel
 - Papel vence Pedra
- 4) Se houver um empate entre as escolhas dos competidores, o round é repetido até que haja um vencedor.
- 5) O competidor vencedor do round adiciona uma vitória ao seu número de vitórias pessoais e à pontuação total da equipe.
- 6) A equipe que perdeu o round deve escolher outro competidor, avançando de forma circular na lista de competidores, para representá-la no próximo round.
- 7) O jogo continua com novos rounds até que uma das equipes alcance 10 pontos antes de sua adversária.
- 8) Ao final da batalha, o programa deve retornar uma mensagem informando o nome da equipe vencedora e o nome do competidor com mais vitórias pessoais, que receberá a menção honrosa.

BOA PROVA! 😊

Dica: lembre-se que um problema grande pode ser decomposto em problemas menores.