Computação I

Lista de exercícios 10 – Introdução à Orientação a Objetos
Atenção! Leia as instruções antes de fazer a lista! - Data de entrega: 20/07/2022
Não é necessário testar se os dados passados por argumento são válidos a menos que pedido na questão. Como de costume, os métodos devem terminar com o número da questão (por exemplo, questao2, questao3, questao4 ...), exceto pelo método construtor (que tem um nome especial __init__), pela questão 6 (que deve criar uma classe com o nome personagem2), e pelo método da questão 8 (pois, nessa questão, é necessário sobrescrever um método herdado). Para que isso funcione, o método da classe herdeira deve ter o mesmo nome do método da classe herdada (no caso, questao5). Ao definir os métodos, preste atenção para o fato de que, além dos argumentos pedidos, um método também recebe um argumento que é uma referência ao objeto (esse argumento é, necessariamente o primeiro argumento do método, ele é um argumento implícito ao fazer a chamada do método e, em geral, ele é chamado de self).

Crie uma classe para representar um personagem de um jogo. O nome da classe deve ser personagem. Essa classe utilizará a função randint da biblioteca random, faça o import antes de definir a classe. Considerando essa classe, defina os seguintes métodos:

1. Método construtor, que define os atributos: nome, país de nascimento, força de ataque, força de defesa e pontos de vida (os nomes dos atributos devem ser: nome, pais, ataque, defesa e vida – atenção a esses nomes para auxiliar a correção!). Os atributos que guardam o nome e país de nascimento são strings e essas strings devem ser passados por argumento. Pontos de vida deve valer 100, ataque deve valer 5 e defesa deve valer 7 (esses três valores são arbitrários e não devem ser recebidos por argumento).

2. Um método de treino, que recebe por argumento uma string com a habilidade (“ataque” ou “defesa”) que deseja treinar. O método deve sortear um número maior ou igual a 0 e menor ou igual a 3 (utilize a função randint, do módulo random importado) e atualizar o atributo de ataque ou defesa (dependendo da string passada por argumento) com o resultado da soma do valor sorteado com o valor que já havia no atributo. O valor final (resultado do sorteio mais o que havia no atributo) também deve ser retornado. Se a string passada por argumento não for nem “ataque” nem “defesa”, retorne -1.

3. Um método para modificar a quantidade de pontos de vida. Esse método recebe por argumento um inteiro positivo ou negativo. O valor passado por argumento deve ser acrescentado ao valor que já existia no atributo de pontos de vida. No entanto, antes de atualizar o atributo, garanta que o resultado é maior ou igual a 0 e menor ou igual a 100. Se o valor resultante for menor que 0, o atributo deve ser definido igual a 0; se o valor resultante for maior que 100, o atributo deve ser definido igual a 100. Além de atualizar o atributo, o método deve retornar a quantidade de pontos de vida após a atualização.

4. Um método para recuperar pontos de vida (esse método não recebe nenhum argumento além do self, necessário para que o Python tenha uma referência para o objeto). O método deve utilizar randint para sortear um número maior ou igual a 1 e menor ou igual a 5, e utilizar o método da questão 3 para acrescentar o valor sorteado ao argumento com os pontos de vida (é o método da questão 3 que modifica o atributo). O valor de retorno deve ser a quantidade de pontos de vida atualizada.

5. Um método para atacar, que recebe como argumento um número inteiro e positivo que representa a defesa de outro personagem. O método deve sortear um valor maior ou igual a 1 e menor ou igual a 5 para o dano causado ao atacar. Se o dano (valor sorteado) somado ao atributo de ataque for maior que a quantidade de defesa passada por argumento, então o ataque foi bem-sucedido e o método deve retornar o dano (isto é, o valor sorteado). Caso contrário (se o dano somado ao atributo de ataque for menor ou igual à quantidade de defesa passada por argumento), retorne 0.

6. Crie a classe personagem2, herdeira de personagem. Considerando essa nova classe, defina os métodos das questões 7 e 8.

7. Um método de apresentação, que retorna uma string com o nome e nacionalidade do personagem em uma frase (esse método não recebe nenhum argumento além do self, necessário para que o Python tenha uma referência para o objeto): 'Ola! Meu nome eh fulano e eu sou do pais_nascimento. ', onde fulano e pais_nascimento devem ser substituídos pelos valores dos respectivos atributos (veja que esses atributos foram herdados, pois o __init__ foi herdado). Veja que a string deve ser retornada (return) e não impressa (print). Por favor, defina a string exatamente (caractere por caractere) como o exemplo acima para facilitar a correção.

8. Sobrescreva o método de atacar (questão 5, atenção ao nome desse método) de forma que o método de atacar da nova classe (personagem2) funcione da seguinte forma: recebe como argumento um número inteiro e positivo que representa a defesa de outro personagem; sorteia um valor inteiro maior ou igual a 1 e menor ou igual à quantidade de ataque (atributo de ataque), onde o valor sorteado é o dano; sorteia um valor inteiro maior ou igual a 1 e menor ou igual à quantidade de defesa (recebida por argumento), onde o valor sorteado é o poder de defesa. Se o dano for maior que o poder de defesa, então o ataque foi bem-sucedido e o método deve retornar o dano (isto é, o valor retornado é valor que foi sorteado anteriormente, entre 1 e o valor do atributo de ataque). Caso contrário, se o dano sorteado for menor ou igual ao poder de defesa sorteado, retorne 0.