Questão 2 (OO)

Considere a classe Data fornecida no arquivo classeData.py, com a seguinte descrição:

Classe Data: um objeto Data é criado a partir do dia, mês e ano

Atributos: dia, mes e ano

Métodos:

construtor: recebe dia,mes,ano de uma data. Caso os 3 valores não sejam fornecidos a data é criada como a data de HOJE (parâmetro default). Obs: se dia,mes,ano fornecidos não especifiquem uma data correta, é criada a data 01/01/0001
apresentação(exibição): retorna uma string com os valores dos atributos
- recebe uma outra data e retorna a quantidade de dias entre as datas (valor absoluto)
+ Recebe um número de dias e retorna a data após/antes este intervalo de dias
+= Recebe um nº de dias e altera o objeto para a data após/antes este intervalo de dias
== Recebe como parâmetro um outro objeto da classe Data, realizando a operação de comparação equivalente ao operador relacional
== Recebe como parâmetro um outro objeto da classe Data, realizando a operação de comparação equivalente ao operador relacional
> retorna True se for mais recente do que uma data recebida como parâmetro
< retorna True se for mais antiga do que uma data recebida como parâmetro
getDia: retorna o dia da data
getMes: retorna o mês da data
getMesExtenso: retorna o mês da data corrente por extenso
getAno: retorna o ano da data
diaDaSemana: Retorna o dia da semana da data
setDia: recebe um dia e o altera se for válido
setMes: recebe um mês e o altera se for válido
setAno: recebe um mês e o altera se for válido
isMesValido Retorna True se é um valor de mês válido ou False caso contrário'
isDiaValido Retorna True se é um dia válido para o mês ou False caso contrário
isBissexto retorna verdadeiro se o ano da data corrente for bissexto e falso caso contrário (múltiplo de 4 e (não múltiplo de 100 ou múltiplo de 400))
copia: o objeto clona a si próprio, para isto, ele cria um novo objeto da classe Data com os mesmos valores de atributos e retorna sua referência pelo método

Considere a classe Horario fornecida no arquivo classeHorario.py, com a seguinte descrição:

Classe Horario : um objeto Horario é criado a partir de hora, minuto, segundo

Atributos: hora, min, seg

e tempo total(em seg, usado internamente na classe)

Métodos:

construtor: este método constrói um objeto Horario, a partir de hora,minuto e segundo, com os seguintes default: h=0, m=0, s=0 Obs: não verifica se h,m,s incorretos
apresentação: retorna uma string com os valores dos atributos no formato "hh:mm:ss"
- recebe um outro Horario e retorna um novo Horario equivalente a diferença de tempo entre os horários recebidos
+ recebe um outro Horario e retorna um novo Horario equivalente a soma dos tempos dos horários recebidos
+= recebe um outro Horario e atualiza o Horário com a soma dos tempos dos horários recebidos
== Recebe como parâmetro um outro objeto da classe Horário, realizando a operação de comparação equivalente ao operador relacional
!= Recebe como parâmetro um outro objeto da classe Horário, realizando a operação de comparação equivalente ao operador relacional
> retorna True se for maior (cronologicamente) do que um horario recebido como parâmetro
< retorna True se for maior (cronologicamente) do que um horario recebido como parâmetro
>= retorna True se for maior ou igual (cronologicamente) a um horario recebido como parâmetro
<= retorna True se for menor ou igual (cronologicamente) a um horario recebido como parâmetro
getSeg: retorna os segundos do horário
getMin: retorna os minutos do horário
getHora: retorna as horas do horário
getTempo: retorna o tempo total em segundos do horário
clone: o objeto clona a si próprio, para isto, ele cria um novo objeto da classe Horario com os mesmos valores de atributos e retorna sua referência
setSeg: recebe um valor e altera o atributo minuto. Recalcula o tempo decorrido e pode recalcular o valor da hora e do minuto, quando segundos >60
setMin: recebe um valor e altera o atributo minuto. Recalcula o tempo decorrido e pode recalcular o valor da hora, quando minutos >60
setHora: recebe um valor e altera o atributo hora. Recalcula o tempo decorrido
totSegundos: retorna o tempo em s
totMinutos: retorna o tempo em minutos
totHoras: retorna o tempo em horas

2A) Escreva a classe Pedido para representar um pedido em um restaurante a ser entregue em domicílio. (Para facilitar considere que os pedidos só são feitos e entregues no mesmo dia)
Um pedido tem numero (int), cliente(string), valor, horário do pedido (objeto do tipo Horario), data (data do pedido) e horário de entrega (objeto do tipo Horario).
Um pedido é criado sendo fornecidos: numero, nome do cliente, valor, o horário (objeto Horario) em que o pedido foi feito e a data. Caso a data não seja fornecida o pedido é criado com a data de hoje. Todo pedido, no momento da sua criação, tem horário de entrega inicialmente igual ao horário (objeto Horario) correspondente a 00h00m00s (Obs: com esse horário de entrega o pedido é considerado não entregue).

- ⇨ __init__

Ao ser exibido, um pedido exibe uma msg com seu numero,cliente,data, horário e status.
Exemplo : Num:22-Cli:lala-Data:18/09/2023-Horario:14:30:00-Status:EmAndamento

- ⇨ __str__ e __repr__

Um pedido é menor do que o outro se sua data é anterior a do outro. Caso tenham sido feitos na mesma data, um pedido é menor do que outro se seu horário for menor do que o outro.

- ⇨ __lt__

Um pedido pode retornar seu status, retornando 'EmAndamento', caso seu horário de entrega corresponda ainda a 00h00m00s, ou 'Entregue', caso o horário de entrega inicial tenha sido alterado.

- ⇨ método obtemStatus

Um pedido pode registrar o horário em que foi entregue, recebendo para isso um objeto Horario

- ⇨ método registraHorarioDeEntrega

Um pedido pode calcular e retornar seu tempo de espera: se o pedido foi entregue, o tempo de espera é a diferença do horário da entrega e o horário do pedido

- ⇨ método calculaTempoDeEspera

Um pedido pode calcular e retornar seu valor: se o pedido foi feito no mês 9, o valor do pedido retornado será o valor original – 10% (não altere o valor original). Do contrário é retornado o próprio valor original.

- ⇨ método obtemValor

2B) Escreva agora a classe Entregador para representar um entregador do restaurante.
Um entregador tem: nome, tipo de transporte, e uma lista de pedidos (lista de objetos da classe Pedido).
Um entregador é criado sendo fornecidos nome e tipo de transporte. Todo entregador tem uma lista de pedidos inicialmente vazia.

- ⇨ __init__

Um entregador, ao ser exibido com print, exibe seu nome e sua lista de pedidos.

- ⇨ __str__ e __repr__

Um entregador pode incluir um pedido para ser entregue na sua lista de pedidos, recebendo o pedido (objeto do tipo Pedido)

- ⇨ método incluiPedido

Um entregador pode exibir seu nome e seu pagamento. O pagamento de um entregador é de acordo com os pedidos (entregues e não entregues). Há um limite de tempo para a entrega do pedido a partir do horario em que foi feito e que depende do tipo de transporte: se o entregador usa moto o LIMITE é de 1h:30min, do contrário o LIMITE é de 2h:30min.
O entregador recebe 7 reais para pedidos entregues dentro do limite, 5 reais para pedidos entregues fora do limite e é penalizado em 10% (perde) do valor de um PEDIDO não entregue.

- ⇨ método exibePagamento