

Prática 5 - Desenvolvimento Orientado a Objetos

Tópicos:

- Análise e Design Orientados a Objetos
- Aplicação de OO na resolução de problemas
- Estilo e documentação

Nota importante: Para todos os exercícios deste guião (e seguintes) deve:

- Começar por analisar o problema adotando OOA;
- Desenhar a solução em termos de classes, suas relações e estruturas de dados necessárias, produzindo um diagrama com as relações entre as classes (usando UML);
- Respeitar o mais possível as indicações PEP 8 relativas a estilo de código, comentários e documentação.

Exercício 5.1 – Carga e descarga de comboio de mercadorias

Pretendemos simular a carga e descarga de comboios de mercadorias. Considere a informação a seguir e desenvolva as classes por forma a conseguir correr pelo menos o programa fornecido ([comboios_teste.py](#)):

- Os comboios são constituídos por uma locomotiva e vários vagões de mercadorias;
- Os vagões têm um número e uma carga máxima que podem transportar (em toneladas). Aos vagões é atribuído o número correspondente à sua posição, sendo o número “1” o primeiro a ser adicionado.;
- As mercadorias a serem carregadas são armazenadas na zona de carga da estação à medida que são entregues para expedição por comboio. Devem ser carregadas por ordem de chegada;
- Cada mercadoria na zona de carga tem a si associada no sistema informático da empresa responsável pelos comboios (ex.: CP) informação com o seu conteúdo, empresa proprietária, e peso (em kg);
- O processo de carga decorre da seguinte forma: (1) separação dos vagões da locomotiva para facilitar o processo; (2) carga dos vagões um a um pela ordem que possuíam na composição original; (3) religação à composição dos vagões à medida que termina a sua carga (no final da composição, sendo o primeiro ligado à locomotiva). Em cada vagão as primeiras mercadorias são colocadas na parte da frente, para facilitar o processo de carga;
- Na descarga, os vagões são processados começando pelo último, sendo desligados da composição à medida que ficam vazios. São descarregados pela parte de trás.

Deve desenvolver funcionalidades para a simulação, incluindo:

- (1) criação um comboio com um determinado número de vagões de mercadoria;
- (2) carga no comboio das mercadorias existentes no armazém da estação de origem, seguindo a informação acima);
- (3) descarga de todo o comboio, mostrando informação do que foi descarregado de cada carruagem e peso total descarregado (por vagão);
- (4) informação sobre as mercadorias que não foi possível transportar.

Programa de teste ([comboios_teste.py](#), disponível no Elearning):

```
from armazem import Armazem
from mercadorias import Mercadoria
from comboio import Comboio

def main():
    armazem_origem = Armazem("Aveiro")

    # receber mercadorias - Mercadoria tem designação, peso (Kg) e dono
    armazem_origem.receber(Mercadoria("Mercedes 200",3000,"Automotor"))
    armazem_origem.receber(Mercadoria("Renault Twingo",2000,"XPTZ"))
    armazem_origem.receber(Mercadoria("BMW",4000,"XPTZ"))
    armazem_origem.receber(Mercadoria("Peças auto",7000,"XPTZ"))
    armazem_origem.receber(Mercadoria("Parafusos",4000,"CP CARGO"))
    armazem_origem.receber(Mercadoria("Cereais",4000,"CP CARGO"))
    armazem_origem.receber(Mercadoria("Motos",5000,"APRILIA"))

    # criar comboio com 3 vagões, todos com carga máxima de 10 toneladas
    comboio = Comboio([10, 10, 10])

    # carregar comboio com o que está em armazém
    comboio.carregar(arrazem_origem)
    print(comboio)

    # fazer viagem e descarregar no destino
    comboio.fazer_viagem()
    comboio.descarregar() # descarrega e mostra

    # mostrar o que ficou por enviar
    print(arrazem_origem)
```

Resultado:

Carregando Mercedes 200, Automotor, 3.0 no vagao Vagão de Mercadorias n.1, Carga máx = 10 toneladas
Carregando Renault Twingo, XPTZ, 2.0 no vagao Vagão de Mercadorias n.1, Carga máx = 10 toneladas
Carregando BMW, XPTZ, 4.0 no vagao Vagão de Mercadorias n.1, Carga máx = 10 toneladas
Necessário novo vagão. Disponíveis 1.0 toneladas e mercadoria pesa 7.0 toneladas.
Carregando Peças auto, XPTZ, 7.0 no vagao Vagão de Mercadorias n.2, Carga máx = 10 toneladas
Necessário novo vagão. Disponíveis 3.0 toneladas e mercadoria pesa 4.0 toneladas.
Carregando Parafusos, CP CARGO, 4.0 no vagao Vagão de Mercadorias n.3, Carga máx = 10 toneladas
Carregando Cereais, CP CARGO, 4.0 no vagao Vagão de Mercadorias n.3, Carga máx = 10 toneladas
Necessário novo vagão. Disponíveis 2.0 toneladas e mercadoria pesa 5.0 toneladas.
AVISO! Não existem mais vagões vazios
COMBOIO:
Vagão de Mercadorias n.3, Carga máx = 10 toneladas
Vagão de Mercadorias n.2, Carga máx = 10 toneladas
Vagão de Mercadorias n.1, Carga máx = 10 toneladas
LOCOMOTIVA
Partida..... viagemChegada.
Início da descarga de Vagão 3 [carga = 8.0, carga max =10

Cereais, CP CARGO, 4.0
Parafusos, CP CARGO, 4.0
Início da descarga de Vagão 2 [carga = 7.0, carga max =10
Peças auto, XPTZ, 7.0
Início da descarga de Vagão 1 [carga = 9.0, carga max =10
BMW, XPTZ, 4.0
Renault Twingo, XPTZ, 2.0
Mercedes 200, Automotor, 3.0
Armazém em Aveiro
Para embarcar:[Motos (APRILIA , 5.0 ton)]

Exercício 5.2 - Agência de Viagens

Pretende-se desenvolver um programa que possibilite a gestão de alguns produtos numa agência de viagens. As entidades principais neste sistema de informação são alojamentos (apartamentos e quartos em hotel) e carros que se podem alugar. Devem ser suportadas as características seguintes:

- A agência de viagens, para além de conter um conjunto de alojamentos e um conjunto de viaturas de aluguer, tem um nome (string) e um endereço (string);
- Um alojamento tem um código (string), nome (string), local (string), preço por noite (float), disponibilidade (boolean) e avaliação (entre 1.0 e 5.0). Deve permitir as operações de check-in e check-out;
- Um apartamento é um alojamento, mas tem adicionalmente informação sobre o número de quartos;
- Um quarto de hotel é um alojamento, mas tem mais um campo que indica o tipo (single, double, twin, triple);
- Um carro tem classe (de 'A' a 'F') e pode ser a gasolina, a diesel ou elétrico. Deve permitir as operações de levantar e entregar.

Represente adequadamente todas estas entidades. Crie construtores, métodos set/get que lhe pareçam adequados, bem como métodos que sejam fundamentais para cada classe.

Teste as classes desenvolvidas usando um programa em que simule a interface com o(a) funcionário(a) da agência (por exemplo, um menu), envolvendo necessariamente a criação dos diversos objetos referidos acima. Simule algumas operações de reserva e entregas e imprima no final a informação atual sobre a agência.

Exercício 5.3- Problema da Loja (de novo) [TPC]

Reutilizando e melhorando as classes que desenvolveu para o exercício 2.3, desenvolva uma solução para gestão da loja, com as seguintes funcionalidades:

- Adicionar produtos ao Stock.
- Adicionar novas aquisições de novos itens de um produto existente.
- Vender produtos, atualizando o Stock. Deve sempre ser vendido um produto que tenha entrado na loja primeiro.
- Listar o conteúdo do Stock
- Calcular o valor total dos produtos em stock

Para além de todas as classes necessárias, desenvolva um programa de teste.