

Exercițiul 14 – Tren

Să se implementeze clasele `Vagon` (`Car`), `Locomotiva` (`Locomotive`), `VagonDePasageri` (`PassengerCar`¹), `VagonRestaurant` (`DiningCar`) și `Tren` (`Train`), unde `Locomotive`, `PassengerCar` și `DiningCar` sunt derivate din `Car` (direct sau indirect).

Clasa `Car` reprezintă o componentă a unui tren (de exemplu locomotivă, vagon etc.), ce posedă următoarele caracteristici:

- o *greutate* specificată în tone (`double`), o *lungime* specificată în metri (`double`) și un *număr de serie unic*, stabilit de constructor.
- numerele de serie sunt numere naturale consecutive. Prima instanță va primi valoarea 1, corespunzătoare primului număr din serie.
- se vor realiza constructorii corespunzători (constructor cu parametri, constructor de copiere), destructor, operatorul de atribuire supraîncărcat.
- o metodă `toString()` ce returnează un șir de caractere conținând reprezentarea unui obiect de forma `Vagon[#serie=A, greutate=B, lungime=C]` unde A, B și C reprezintă valorile atributelor *număr de serie*, *greutate*, și respectiv *lungime*, corespunzătoare obiectului curent.
- o metodă `connectTo(Car* car)` ce atașează în spatele componentei feroviare curente o altă componentă specificată ca parametru (`car`), astfel încât să se poată forma un tren. Fiecare componentă (feroviară) va conține o referință (pointer) către elementul (vagonul) următor.
- o metodă `next()` ce întoarce adresa următorului element (vehicul feroviar).
- o metodă `clone()` ce întoarce adresa unui obiect nou de tip `Car`, copie a obiectului curent.

Clasa `Locomotive` reprezintă un vehicul feroviar ce posedă un motor:

- o locomotivă va avea, în plus față de proprietățile unui vagon, atributul *putere* specificat în kW (`int`).
- se vor realiza constructorii corespunzători (constructor cu parametri, constructor de copiere).
- o metodă `toString()` ce returnează un șir de caractere conținând reprezentarea unui obiect de forma `Locomotiva[#serie=A, greutate=B, lungime=C, putere=D]` unde A, B, C și D reprezintă

¹ [https://en.wikipedia.org/wiki/Passenger_car_\(rail\)](https://en.wikipedia.org/wiki/Passenger_car_(rail))

valorile atributelor *număr de serie*, *greutate*, *lungime* și respectiv *putere*, corespunzătoare obiectului curent.

Clasa `PassengerCar` reprezintă un vagon de călători:

- acesta este un vehicul feroviar fără motor, în plus față de caracteristicile acestuia prezentând *numărul de locuri* (scaune) (`int`).
- se vor realiza constructorii corespunzători (constructor cu parametri, constructor de copiere).
- o metodă `toString()` ce returnează un șir de caractere conținând reprezentarea unui obiect de forma `VagonDePersoane[#serie=A, greutate=B, lungime=C, #locuri=D]` unde A, B, C și D reprezintă valorile atributelor *număr de serie*, *greutate*, *lungime* și respectiv *număr de locuri pe scaune*, corespunzătoare obiectului curent.

Clasa `DiningCar` reprezintă un vagon-restaurant:

- un vagon în care se poate servi masa, și unde, în plus față de proprietățile unui vagon de călători, trebuie specificat dacă vagonul restaurant este *deschis* sau nu (`boolean`).
- se vor realiza constructorii corespunzători (constructor cu parametri, constructor de copiere).
- o metodă `toString()` ce returnează un șir de caractere conținând reprezentarea unui obiect de forma `VagonRestaurant[#serie=A, greutate=B, lungime=C, #locuri=D, esteDeschis=E]` unde A, B, C, D și E reprezintă valorile atributelor *număr de serie*, *greutate*, *lungime*, *număr de locuri pe scaune* și respectiv *dacă este deschis sau nu*, corespunzătoare obiectului curent.

Datele membre ale claselor menționate nu ar trebui să poată fi modificate după crearea și instanțierea obiectelor.

Clasa `Train` reprezintă un tren, ce poate exista doar dacă are o locomotivă:

- locomotiva trenului va fi specificată în constructor.
- o metodă `add(Car)` ce adaugă la ultimul vagon existent al trenului, următoarea componentă feroviară (folosește metoda `connectTo()` din clasa `Car`). Primul apel pentru adăugarea unui vehicul feroviar va avea drept rezultat adăugarea acestuia (obiectului de tip vehicul feroviar) după locomotivă. De acest vagon va fi legat următorul vagon corespunzător următorului apel de adăugare s.a.m.d.
- o metodă `toString()` ce returnează un șir de caractere reprezentând informații despre tren (de exemplu greutatea totală a trenului). Dacă trenul are un vagon restaurant în componența sa, atunci poate să apară informația “*Has open dining car*”.

Referințe:

Clasele MovablePoint și MovableCircle.

Conținutul fișierului test-train.cpp ar putea fi:

```
#include <iostream>

#include "Car.h"
#include "PassengerCar.h"
#include "DiningCar.h"
#include "Locomotive.h"
#include "Train.h"

int main() {
    Locomotive* locomotive1 = new Locomotive(12.0, 12.0, 12);
    Train train155(locomotive1);

    PassengerCar* w1 = new PassengerCar(1.1, 1.1, 11);
    PassengerCar* w2 = new PassengerCar(2.2, 2.2, 22);
    DiningCar* d1 = new DiningCar(3.3, 3.3, 100, false);
    PassengerCar* w3 = new PassengerCar(4.2, 4.2, 44);

    train155.add(w1);
    train155.add(d1);
    train155.add(w2);
    train155.add(w3);

    train155.hasOpenDiningCar();

    cout << train155.toString() << endl;

    Train* train234 = train155.clone();
    cout << train234->toString() << endl;
    train234->clean();
    delete train234;

    return 0;
}
```