

Ćwiczenie 3

Cel:

Symulacja pociągów, zabawa z łączeniem wagonów i lokomotyw, ładowaniem towarów do wagonów i zmiana prędkości pociągów.

Na ocenę 3:

Opracować szkielet systemu w oparciu o poniższe klasy i funkcje.

`RailVehicle` -- abstrakcyjna klasa bazowa dla wszystkich pojazdów szynowych. Umożliwia łączenie pojazdów. Umożliwia wyświetlanie stanu i zawartości pojazdów. Funkcje:

```
RailVehicle* ConnectToNext(RailVehicle* nextPtr)
virtual int GetSpeed() const;
virtual void ShowThis(ostream& o) const = 0;
void ShowAll(ostream& o) const;
```

`Engine` - lokomotywa. Umożliwia zmianę prędkości pociągu. Dziedziczy po `RailVehicle`. Ma maksymalną prędkość. Funkcje:

```
int ChangeSpeed(int value)
```

`FreightCar` - wagon towarowy. Dziedziczy po `RailVehicle`.

Zasady łączenia pojazdów:

Pojazdy można połączyć jeśli z danej strony nie są do niczego podłączone. Do łączenia służy jedna funkcja, podłączająca pojazd do następnego w szeregu. Aby połączyć pojazdy, należy podać wskaźnik do następnego pojazdu. Aby rozłączyć pojazdy, należy podać NULL. W przypadku pomyślnej operacji funkcja zwraca wskaźnik do dołączanego pojazdu, w przeciwnym wypadku rzuca wyjątek. Zniszczenie pojazdu powoduje odłączenie go od sąsiadów. Lokomotywę można dołączyć tylko na początku pociągu. Do lokomotywy nie można dołączyć innej lokomotywy.

Zasady zwracania prędkości pociągu:

Każdy pojazd można zapytać o prędkość pociągu. Używając wskaźników do poprzedniego pojazdu dochodzi on do czoła pociągu i jeśli jest tam lokomotywa, zwraca jej prędkość

Inna funkcjonalność:

Funkcje `ShowThis` i `ShowAll` mają odpowiednio wyprowadzać do strumienia informację o pojedynczym pojeździe i całym pociągu.

Zagadka:

Co będzie jeśli w destruktorze `RailVehicle` wywołana zostanie funkcja wirtualna, np. `GetSpeed` w celu sprawdzenia czy pociąg stoi?

Na ocenę 4:

Umożliwić ładowanie i rozładowywanie towarów. Dodać do klasy `FreightCar` funkcje:

```
int AddFreight(int mass)
int RemoveFreight(int mass)
```

Zasady załadunku i rozładunku:

Można przeprowadzać tylko gdy pociąg stoi. Załadowanie powoduje dodanie masy ładunku do masy danego wagonu. Zdjęcie ładunku odejmuje masę ładunku od masy wagonu.

Inna funkcjonalność:

Funkcje `ShowThis` i `ShowAll` mają zostać uzupełnione o wyprowadzanie do strumienia

odpowiednio informacji o masie ładunków w pojedynczym pojeździe i w całym pociągu.

Na ocenę 5:

Uwzględnić masę pociągu w wyznaczaniu jego prędkości. Uzupełnić klasę `RailVehicle` o funkcje:

```
int GetMassAll() const;  
virtual int GetMassThis() const = 0;
```

Zasady zwracania masy danego pojazdu:

Dla wagonu masa jest sumą stałej masy wagonu (przekazywanej w konstruktorze) i masy towarów. Dla lokomotywy jest jej stałą masą (przekazywaną w konstruktorze).

Zasady zwracania masy pociągu:

Każdy pojazd można zapytać o masę całego pociągu. Używając wskaźników do poprzedniego i następnego pojazdu sumuje on masy pojazdów i zwraca te sumę.

Zasady zmiany prędkości pociągu:

W konstruktorze lokomotywy przekazywane są jej maksymalna prędkość bez obciążenia. Maksymalna prędkość pociągu jest ustalana na podstawie maksymalnej prędkości lokomotywy oraz masy pociągu. Dla zerowej masy całkowitej pociągu prędkość maksymalna równa jest maksymalnej prędkości lokomotywy, każda tona masy to spadek prędkości maksymalnej o 1km/h. (uwaga: pociąg nigdy nie będzie jechał z maksymalną prędkością lokomotywy, bo sama lokomotywa ma już pewną masę).

Nie strzelać do prowadzącego!