

Napisz aplikację, która będzie służyła do symulacji i obsługi logistycznej linii kolejowych z uwzględnieniem stacji kolejowych, połączeń, skrzyżowań linii oraz różnych składów pociągów.

Należy stworzyć wszystkie potrzebne klasy oraz uzupełnić każdą klasę przynajmniej dwoma dodatkowymi polami i metodami o tematyce odpowiadającej klasie (nie metody typu *get* i *set*).

W ramach aplikacji musimy mieć możliwość z poziomu menu (poza innymi niżej opisanymi funkcjonalnościami) stworzenia nowej lokomotywy, wagonów, stacji kolejowych oraz połączeń między stacjami oraz możliwość wykonania zadań takich jak przypisanie wagonu do lokomotywy, załadunek osób/ładunku do wagonów itp. Należy uwzględnić również usuwanie obiektów.

Każdy skład pociągu posiada informacje o lokomotywie oraz o podłączonych do niego wagonach. Każda lokomotywa jest definiowana przez:

- maksymalną liczbę wagonów
- maksymalną wagę (uciąg) transportowanego ładunku
- maksymalną liczbę wagonów wymagających podłączenia do sieci elektrycznej

Każda lokomotywa posiada podstawowe informacje o sobie (nazwę, stacja macierzysta, stacja źródłowa i docelowa transportu) oraz swój unikatowy numer identyfikacyjny nadawany automatycznie podczas tworzenia obiektu.

Dodatkowo lokomotywy posiadają przypisaną prędkość, z którą się porusza, a która co 1 sekundę prędkość będzie się zmniejszała lub zwiększała (losowo) o 3%. Na podstawie tej prędkości należy uzależnić ruch składów pociągów po trasach.

Istnieją różnego typu wagony kolejowe. Każdy z typów wagonów posiada inny zestaw cech (np. nadawca, zabezpieczenia, waga netto, waga brutto, liczba miejsc siedzących itp.). W przypadku każdego typu wagonu należy dodać ponad wymienione co najmniej dwie unikalne cechy do każdego typu wagonu. Każdy wagon posiada swój unikatowy numer identyfikacyjny nadawany automatycznie podczas tworzenia obiektu.

Mamy do dyspozycji między innymi:

- wagon pasażerski, wymagający podłączenia do sieci elektrycznej lokomotywy
- wagon pocztowy, wymagający podłączenia do sieci elektrycznej lokomotywy
- wagon bagażowo-pocztowy
- wagon restauracyjny, wymagający podłączenia do sieci elektrycznej lokomotywy
- wagon towarowy podstawowy
- wagon towarowy ciężki
- wagon chłodniczy, będący rodzajem wagonu towarowego podstawowego, wymagający podłączenia do sieci elektrycznej lokomotywy
- wagon na materiały ciekłe, będący rodzajem wagonu towarowego podstawowego
- wagon na materiały gazowe, będący rodzajem wagonu towarowego podstawowego
- wagon na materiały wybuchowe, będący rodzajem wagonu towarowego ciężkiego
- wagon na materiały toksyczne, będący rodzajem wagonu towarowego ciężkiego
- wagon na ciekłe materiały toksyczne, który jest rodzajem wagonu towarowego ciężkiego, oraz posiada cechy wagonu na materiały ciekłe

W ramach składu pociągu jesteśmy ograniczeni nie tylko maksymalną liczbą wagonów, ale również maksymalnym uciążeniem lokomotywy, w związku z czym przed przyłączeniem kolejnego wagonu musimy sprawdzić, czy kolejny wagon nie przekroczy tych limitów. Jeśli dodanie kolejnego wagonu nie jest możliwe, użytkownikowi zostanie wyświetlony stosowny komunikat bazujący na podniesionym wyjątku.

Składy pociągów poruszają się po ustalonej trasie (wskazanej obiektem stacji startowej i docelowej). Trasę pomiędzy stacjami każdorazowo należy ustalić (nie musi być to trasa najkrótsza, ale każdorazowo musi być ustalona algorytmicznie na bazie grafu połączeń kolejowych).

Skład pociągu staje na każdej napotkanej po drodze stacji. Postój na stacjach pośrednich trwa 2 sekundy. Po dotarciu na stację docelową skład pociągu oczekuje 30 sekund, a następnie pociąg wyrusza w drogę powrotną (ponownie ustalając trasę) i kursuje cały czas w tym cyklu.

Należy zaimplementować również prewencję kolizji, w której pomiędzy dwoma stacjami może poruszać się maksymalnie jeden skład pociągu. Jeśli nastąpi taka sytuacja, że w trasie znajduje się już pociąg, inne oczekują w kolejce na zwolnienie trasy i w kolejności zgłoszenia oczekiwania będą przepuszczane przez pożądaną trakcję pomiędzy stacjami kolejowymi.

Wszystkie kwestie związane z czasem należy zrealizować z uwzględnieniem wątków (nie wolno wykorzystywać klasy *Timer*). Należy zapewnić przemyślaną i poprawną synchronizację wszystkich wątków. Nie można łączyć różnych funkcjonalności w jeden wątek.

Gdy skład pociągu przekroczy prędkość 200km/h, należy poinformować użytkownika aplikacji stosownym komunikatem na konsoli w postaci wyjątku typu ***RailroadHazard*** zawierającego podstawowe informacje dot. składu pociągu przekazane poprzez wiadomość wyjątku.

Musimy mieć możliwość wyświetlenia (po wskazaniu składu pociągu) raport zawierający:

- wszystkie podstawowe informacje o składzie pociągu
- % ukończonej drogi pomiędzy stacją startową i docelową
- zestawieniem informacji o wagonach, liczbie przewożonych osób i przewożonym towarze
- % ukończonej drogi pomiędzy najbliższymi stacjami kolejowymi na swojej trasie.

Stosowna informacja powinna być wyświetlona na konsoli po wyborze odpowiedniej opcji. Ponadto należy automatycznie co 5 sekund dopisywać zestawienie wszystkich istniejących składów pociągów do pliku *AppState.txt* (w formie tekstowej, czytelnej dla użytkownika). Informacje powinny być zapisane z zachowaniem poniższych reguł:

- Wagony w składach pociągów posortowane rosnąco według wagi.
- Składy pociągów posortowane malejąco względem dystansu jaki mają do pokonania.

W programie należy obsłużyć wszystkie powstające wyjątki i zakomunikować użytkownikowi stosownymi komunikatami bez potrzeby przerywania i uruchamiania na nowo programu.

Przygotuj (wygeneruj) na potrzeby symulacji co najmniej 100 stacji kolejowych wraz z połączeniami między nimi, 25 składów pociągów (każdy po losowo 5-10 wagonów).

Dodatkowo poza głównym przebiegiem aplikacji, zrealizuj osobny plik (*Presentation.java*) z dodatkową metodą *main* zawierającą przykłady pokazujące w prosty sposób każdą z funkcjonalności.

Projekt opiera się o materiał z zakresu PPJ oraz GUI.

Uwaga:

- *W przypadku otrzymania projektu ze znacznymi brakami w implementacji/z niezaimplementowaną rozgrywką lub niekompilującego się, wynikiem za taki projekt będzie 0 pkt.*
- *Brak znajomości dowolnej linii kodu lub plagiat skutkować będzie wyzerowaniem punktacji za ten projekt.*
- *W ocenie projektu poza praktyczną i merytoryczną poprawnością będzie brana również pod uwagę optymalność, jakość i czytelność napisanego przez Państwa kodu.*
- *Ważną częścią projektu jest wykorzystanie między innymi: dziedziczenia, kolekcji, interfejsów lub klas abstrakcyjnych, lambda-wyrażeń, typów generycznych, dodatkowych funkcjonalności lub struktur oraz innych elementów charakterystycznych*