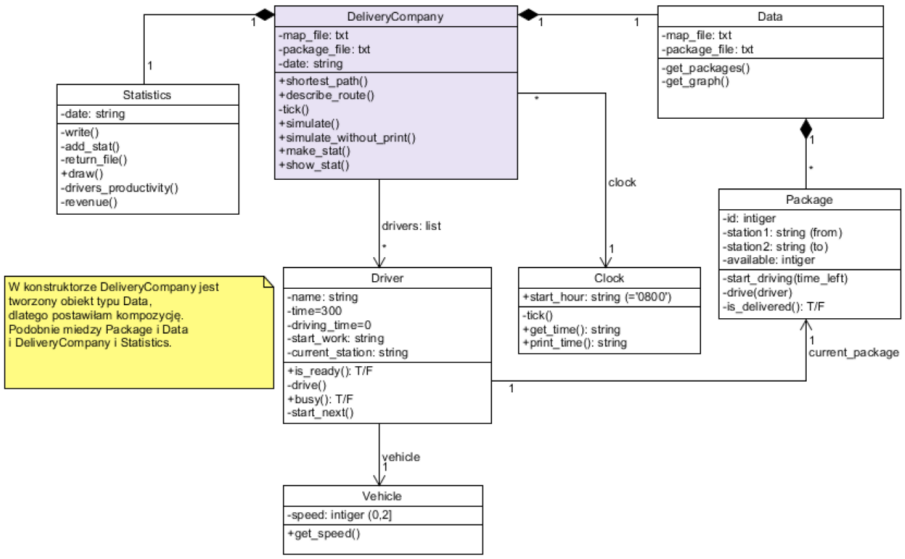
Firma Kurierska

Projekt ten ma na celu zasymulować działanie firmy kurierskiej. Na wejściu użytkownik podaje dwa pliki, jeden z informacjami na temat paczek do rozwiezienia, drugi zawiera informację o stacjach i odległościach między nimi. Użytkownik musi również podać listę pracowników (kurierów). Paczki są rozwożone między stacjami, kurier jednorazowo może przewozić tylko jedną paczkę.

Wszyscy pracownicy zaczynają pracę na ‘stacji 0’. Kiedy kierowca skończy pracę, wraca od razu do domu, obojętnie gdzie się znajduje.

Pracownicy zaczynają pracę o równych godzinach.

Klasy:



1) Delivery Company

Jest to klasa odpowiedzialna za scalenie (tworzenie obiektów) wszystkich pozostałych klas. Na wejściu dostaje 2 pliki:

- Plik z mapą to informacja o odległościach między stacjami zapisanych jako czas dojazdu w minutach. Zapisana jest w takiej postaci: pierwsza

linijka to liczba stacji, poniżej w kolumnie nazwy stacji, a na końcu macierz sąsiedztwa. Kolumny i wiersze odpowiadają kolejności stacji.

- Plik z informacją o paczkach. Pierwsza linijka to liczba paczek do rozwiezienia, poniżej w kolejnych wierszach dane następującej postaci:
  id paczki (max 10 cyfr) stacja startowa stacja docelowa godzina

  godzina = czterocyfrowa liczba, oznacza godzinę, od której paczka jest dostępna (np. 1751 to 17:51)

- Listę złożoną z kierowców, których użytkownik musi wcześniej sam stworzyć.
- String (= dzień, w którym symulacja się dzieje), który będzie nazwą pliku, w którym zapisane będą informacje kto jakie paczki dostarczył.

FUNKCJE:

1. Konstruktor – tworzy Data(), Clock(), Statistics(), pamięta listę kierowców, stos paczek i graf połączeń między stacjami.
2. shortest_path(station1, station2) - znajduję najkrótszą drogę w grafie połączeń ze station1 do staction2 korzystając z algorytmu Dijkstry.
3. describe_route(station1, station2) - wykonuje shortest_path() i zwraca trasę = listę stacji i całkowity czas trasy.
4. tick() - odpowiada za 1 minutę. Na zegarze upływa jedna minuta, jeśli kierowca ma paczkę wywołuje na nim funkcję drive().
5. simulate() - symulacja trwa 10h. W jednej minucie, jeśli stos paczek nie jest pusty, wybieram kierowcę, który zaczął już pracę, biorę paczkę ze stosu, jeśli jest już dostępna i kierowca nie jest zajęty oraz zdąży dowieźć paczkę przed końcem pracy to: albo kurier znajduje się na odpowiedniej stacji, wtedy znajduje drogę do docelowej stacji, albo musi najpierw dojechać do stacji, na której znajduje się paczka i zawieźć ją na docelową stację. Za każdym razem, kiedy paczka jest dostarczona do statystyk dodaje tę informacje. Simulate również wyświetla na bierząco co się dzieje i o jakiej godzinie. Na koniec minuty wykonuje tyknięcie zegara.
6. simulate_without_print() - czasem przydaje się, np. w testach jednostkowych
7. make_stat() - wywołuje na obiekcie Statistics funkcję return_file()
8. show_stat() - wywołuje na obiekcie Statistics funkcję draw()

2) Data

Jest to kluczowa klasa, ponieważ korzysta z plików wejściowych i tworzy z nich obiekty (Package oraz graf reprezentujący połączenia między stacjami). Na wejściu obiekt dostaje dwa pliki wyżej opisane (w pkt 1.).

W konstruktorze tworzą się obiekty Package, które w kolejności malejącej godziny (dostępu), trafiają na stos (najwcześniej dostępne paczki są na szczycie stosu), oraz graf skierowany ważony, w którym węzły to stacje a wagi to odległości (czas dojazdu).

Obiekt ma dwie funkcje, jedna zwraca stos paczek, druga graf połączeń.

3) Clock

Klasa odpowiada za godzinę, potrzebna jest po to by sprawdzić czy dana paczka jest już dostępna.

FUNKCJE:

1. tick() - +1 minuta
2. get_time() - zwraca godzinę w postaci napisu długości = 4
3. print_time() - zwraca godzinę w ładniejszej postaci z ‘:’

4) Driver

Jest to klasa reprezentująca dostawcę. Użytkownik tworząc kierowcę powinien podać jego imię, pojazd jakim będzie się poruszał i godzinę, o której zaczyna pracę (string np. ‘0800’). Za pojazd odpowiada klasa Vehicle, niżej opisana. Driver pamięta imię, pozostały czas pracy w minutach, pojazd i aktualną paczkę, którą przewozi. Każdy pracownik ma 300min pracy dziennie.

FUNKCJE:

1. is_ready() - sprawdza, czy kierowca zaczął już pracę
2. drive() - wywołanie tej funkcji powoduje upływ jednej minuty, czyli przemieszczenie się kierowcy ku stacji docelowej o jedną minutę oraz wywołanie funkcji drive() na paczce, jeśli ją przewozi.
3. busy() - sprawdza czy kierowca jest zajęty, czy można mu dać paczkę
4. start_next() - przypisuje kierowcy nową paczkę

5) Vehicle

Klasa reprezentująca pojazd, od którego zależy prędkość poruszania się kierowcy == pracy. Obiekt Vehicle zapamiętuje prędkość i posiada funkcję get_speed(), która zwraca tę prędkość. Prędkość to liczba rzeczywista z przedziału (0,2].

6) Package

Klasa reprezentująca paczkę, do skonstruowania paczki potrzebny jest numer paczki, stacja, na której znajduje się paczka i docelowa stacja, oraz godzina, od której paczka jest dostępna. Obiekt również zapamiętuje czas (odległość) pozostały do dostarczenia paczki.

FUNKCJE:

1. start_driving() - ustawia nowy 'pozostały czas' do dostarczenia
2. drive(driver) - przemieszczenie się ku stacji docelowej o jedną minutę
3. is_delivered() - sprawdza czy paczka jest już dostarczona => 'pozostały czas' = 0

7) Statistics

Na wejściu dostaje napis, który będzie nazwą pliku wyjściowego. Napis powinien być postaci 'dd_mm_rrrr'

W konstruktorze pamięta słownik, którego klucze to kierowcy a argumenty to listy złożone z paczek, które konkretny kierowca dostarczył danego dnia.

FUNKCJE:

1. write() - zwraca plik o nazwie '*dd_mm_rrrr*_stat', w którym zapisane są dane z powyżej opisanego słownika
2. add_stat() - dodaje kierowcę i paczkę do słownika
3. return_file() - wywołuje funkcję write(), czyli tworzy plik
4. draw() - rysuje wykres słupkowy, kto, ile paczek rozwiózł.

Dodatkowo korzystam z klas z wykładu:

- Stack - stos paczek
- Graph - graf połączeń między stacjami
- Dijkstra – znajdywanie najkrótszej drogi ze stacji do stacji

W przyszłości, dodałabym dodatkowe wykresy. Również przemyślałabym dobór kierowcy do paczki, np. jeśli jest na pobliskiej stacji to ma pierwszeństwo oraz możliwość dodawania nowych paczek w ciągu dnia.