Politechnika
Wrocławska

PROGRAMOWANIE OBIEKTOWE

LAB/PROJEKT ŚRODA 18:55

Tram traffic simulation

Autor:
Dawid NOWAK

Prowadzący:
mgr Tobiasz PUŚLECKI

1   Team members

Dawid Nowak (Leader)

2   Programming language

C++ 14

3   Description

Program simulates tram traffic in Wrocław, Poland. Each tram, assigned to a specific route, starts its journey at a departure time. Trams may meet each other on tracks which means one is stopping the other one from moving, also there may happen incidents that causes delays; there might be heavy traffic at the intersection making the tram unable to leave the stop, there might be passengers misbehaving, etc. Every tram runs in its own thread which measures actual time of the whole route.

3.1   Traffic manager

There is a "Traffic manager" class that manages whole traffic, assigns trams to lines, begins the simulation, stores data from all objects. To make sure there is only one instance of this, it implements singleton design pattern.

3.2   Tram

Tram class is an abstract class with a virtual method, which is implemented by all particular inheriting tram models. There are 6 trams available for user to choose from: Moderus Gamma LF 07 AC, Moderus Beta MF 24 AC, PESA Twist 146n, PESA Twist 2010Nw, Konstal 105Na and Protram 105 NWr. Each tram model takes different time at the stop (older trams have it longer and breaks down more often).

3.3   Depot

In Wrocław currently there are 3 depots in everyday use. All 3 can be found in this program. Each depot stores different trams. Depot class implements factory design pattern.

3.4   Tram stop

If everything works fine, tram should arrive and leave at the time specified in timetable. But there are incidents likely to happen. Stops have a significant impact on the tram traffic.

3.5   Timetable

It contains all tram lines in our simulation, uses hash (unordered) map structure where the key is a next stop, and its value is departure time from that stop. There are included some of the most popular lines, such as line number 4 or number 2 and more.
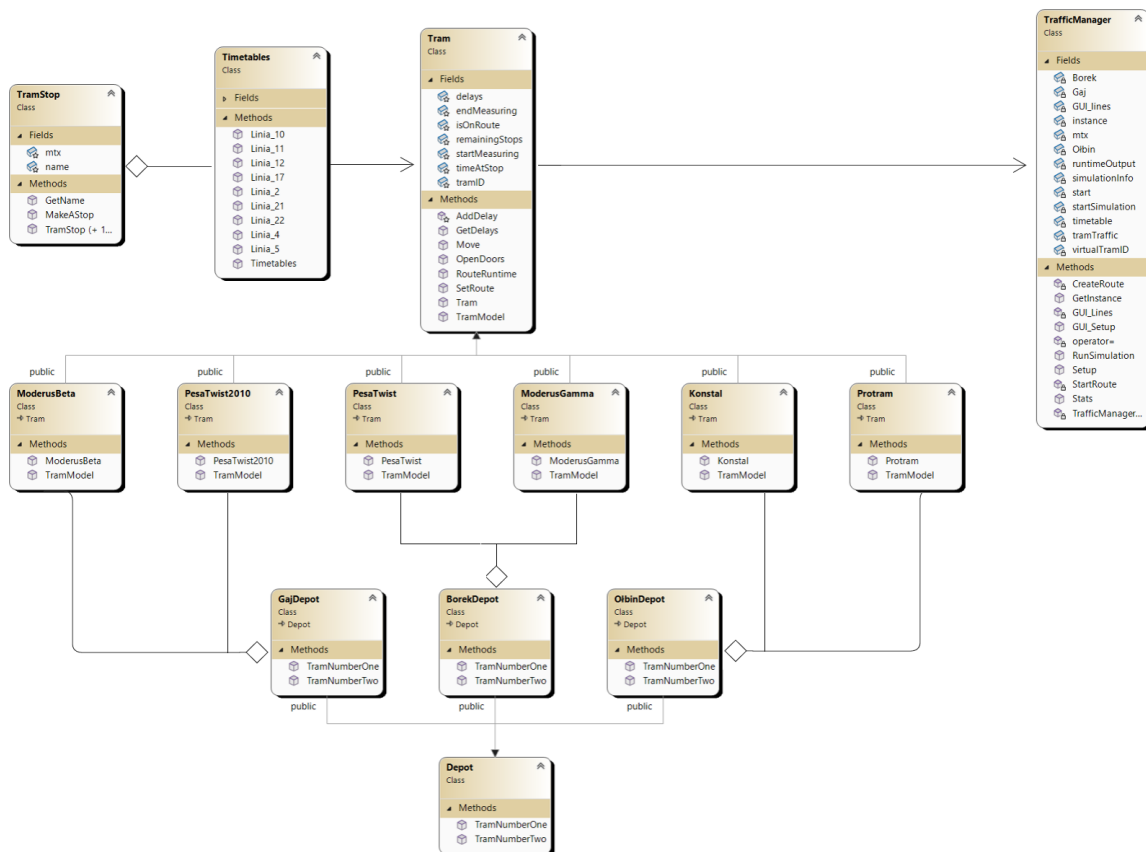
4 In/out data

4.1 Input

Unique tram of user choice assigned to a specific line with a chosen departure time. (there is no limit of possible trams running at once, but their departures have to fit all within one hour)
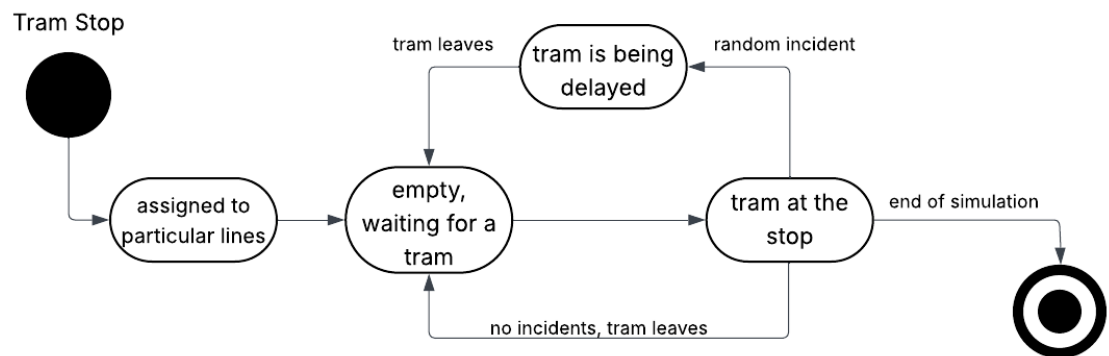
4.2 Output

Logs of each tram journey (expected vs actual moving time, delay causes and places where delays occured.
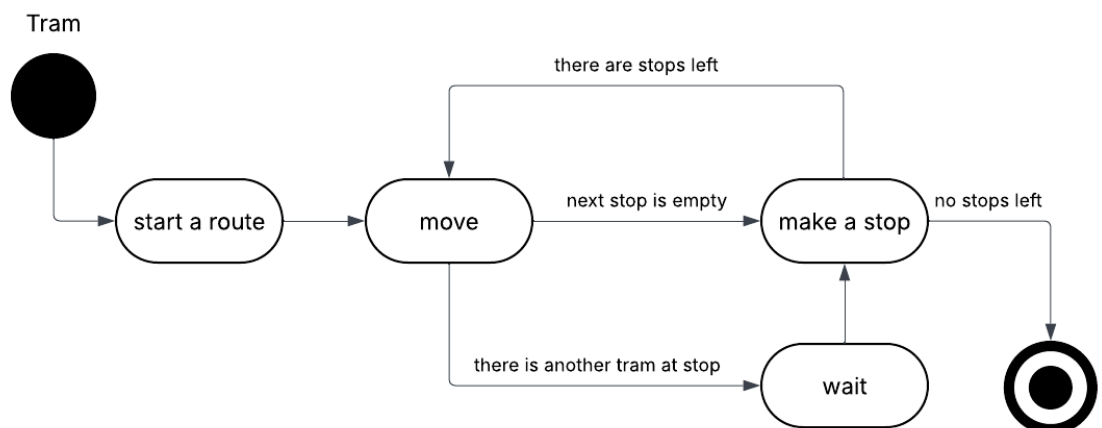
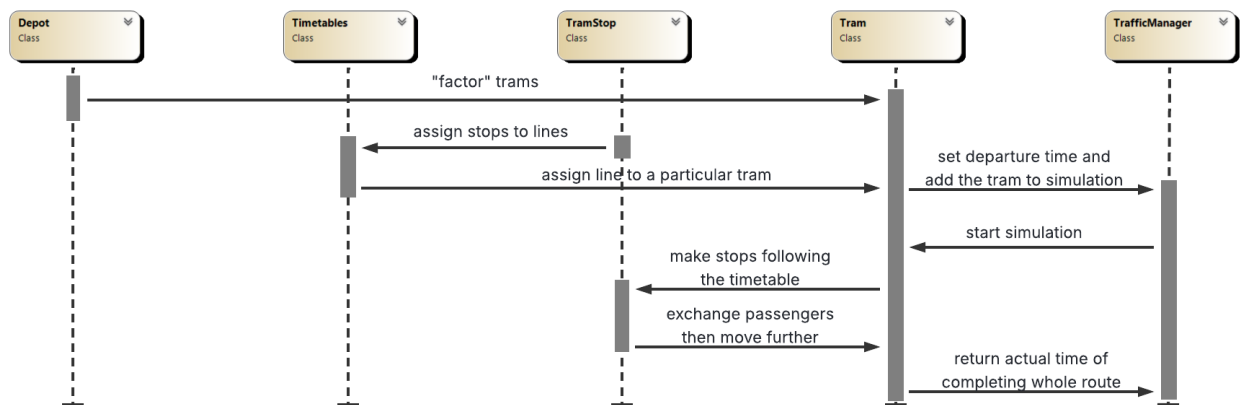# 5 UML diagrams

5.1 class diagram

5.2   tram stop state diagram

Tram Stop

tram leaves

tram is being delayed

random incident

assigned to particular lines

empty, waiting for a tram

tram at the stop

end of simulation

no incidents, tram leaves

5.3   tram state diagram

Tram

there are stops left

start a route

move

next stop is empty

make a stop

no stops left

there is another tram at stop

wait

5.4 sequence diagram

Literatura

N/A