

Московский государственный университет имени М.В. Ломоносова  
Факультет вычислительной математики и кибернетики  
Кафедра алгоритмических языков

Феофилактов Андрей Дмитриевич

## **Симуляция движения поездов в метро**

Задание по курсу ООП

**Научный руководитель:**  
Большакова Е.И.

Москва, 2019

## Содержание

1	Уточнение постановки задачи	2
2	Диаграмма классов	3
3	Объекты	4
4	Инструментальные средства	5
5	Описание файловой структуры системы	6
6	Пользовательский интерфейс	8

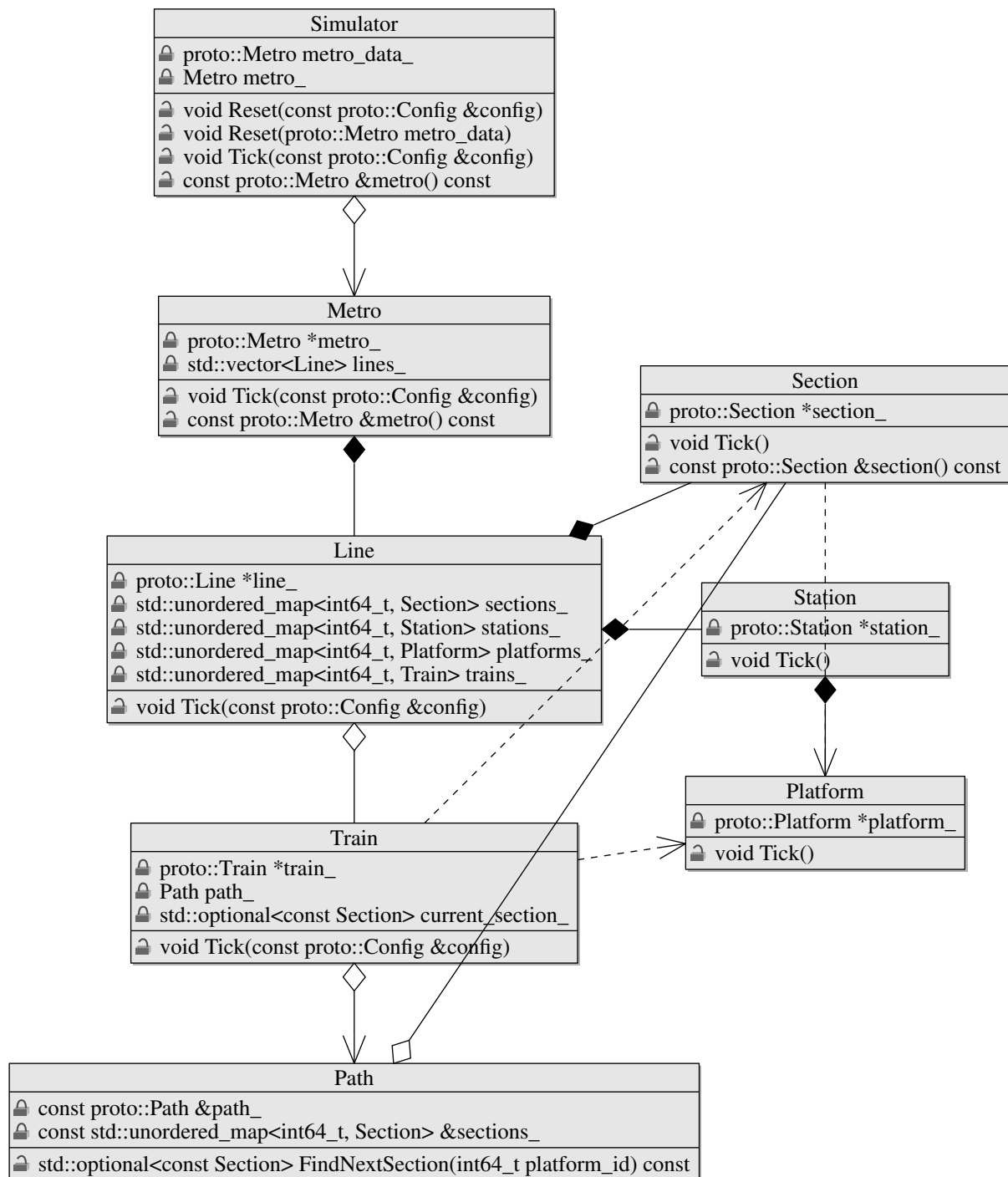
## 1 Уточнение постановки задачи

Требуется реализовать модель работы метро. В данной модели метро представляет из себя набор линий. Каждая из линий состоит из станций, каждая станция может содержать несколько платформ. Тоннели соединяют между собой платформы и однонаправлены. Также каждой линии принадлежит набор поездов, которые по ней движутся. Каждый поезд содержит граф пути, указывающий порядок прохождения секций (тоннелей).

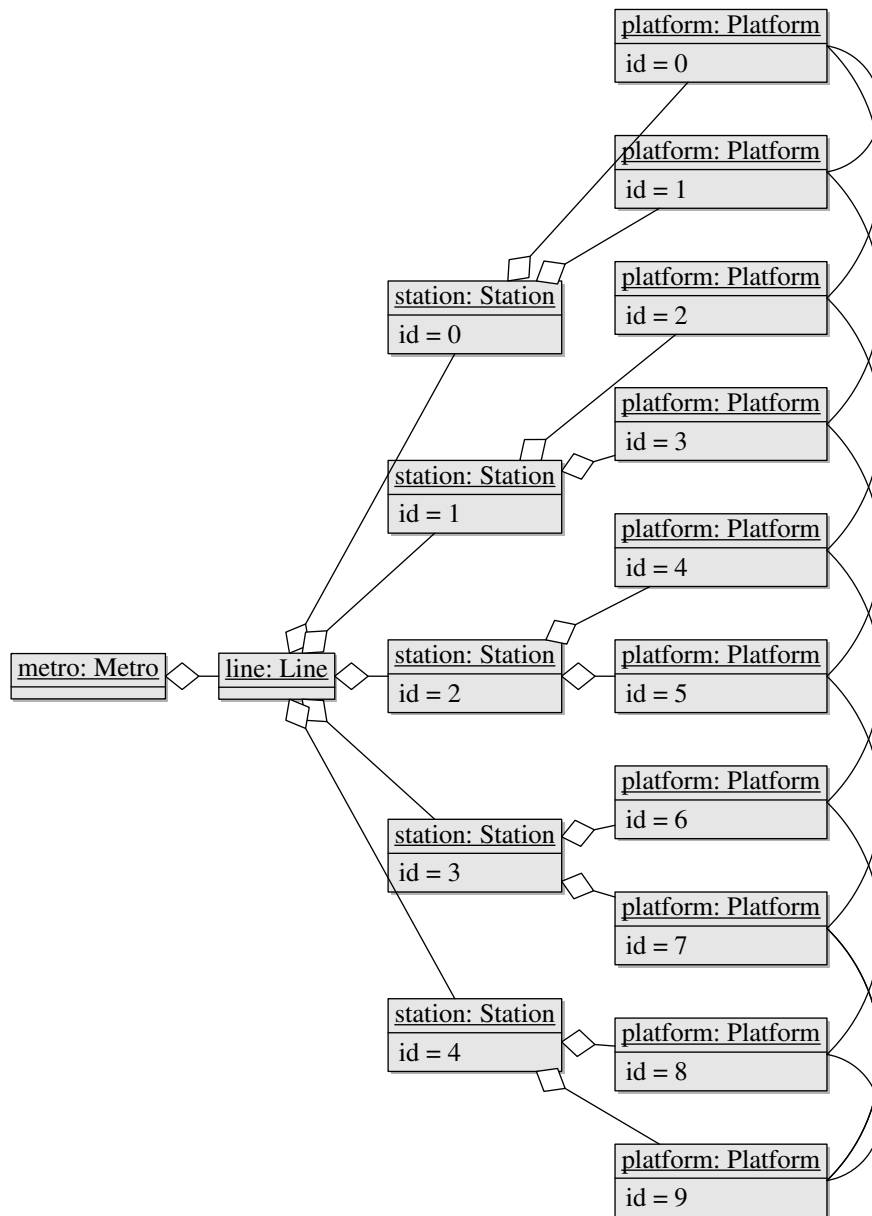
Также требуется реализовать возможность изменения структуры метро (добавление/удаление линий, станций, секций и поездов). Кроме того, нужно сделать графический интерфейс, отображающий положение поездов и структуру метро.

Необходимо, чтобы сама модель не зависела от конкретного графического представления. Для этого будем использовать промежуточное сериализуемое состояние модели.

## 2 Диаграмма классов



### 3 Объекты



## 4 Инструментальные средства

**Язык программирования:** C++17

**Используемые библиотеки:** SDL2, glLoadGen, Google Protobuf, ImGui

**Прочие инструменты:** vim, cmake, make

## 5 Описание файловой структуры системы

### Основные файлы модели

В каждом из следующих заголовочных файлов вида `train.h` определён соответствующий класс `Train`. В соответствующем же ему `train.cpp` находится реализация его методов

- `lib/simulator/train.{h,cpp}`
- `lib/simulator/path.{h,cpp}`
- `lib/simulator/section.{h,cpp}`
- `lib/simulator/station.{h,cpp}`
- `lib/simulator/simulator.{h,cpp}`
- `lib/simulator/line.{h,cpp}`
- `lib/simulator/platform.{h,cpp}`
- `lib/simulator/metro.{h,cpp}`

### Прочие файлы

- `lib/graphics/adapter/sdl_adapter.{h,cpp}` — класс `SDL`. Графический адаптер, отвечающий за отрисовку окна, картинки метро и элементов интерфейса
- `lib/graphics/interface/config_editor.{h,cpp}` — реализация интерфейса редактирования конфига
- `lib/graphics/interface/metro_editor.{h,cpp}` — реализация интерфейса редактирования структуры метро
- `lib/graphics/interface/metro_representation.{h,cpp}` — реализация текстового представления метро
- `lib/graphics/primitives/rectangle.{h,cpp}` — реализация отрисовки прямоугольника с текстурой
- `lib/graphics/shader/shader.{h,cpp}` — класс `Shader`. Обёртка над шейдером `OpenGL`
- `lib/graphics/texture/texture.{h,cpp}` — класс `Texture`. Обёртка над текстурой `OpenGL`

- `lib/util/singleton/singleton.{h,cpp}` — функция Singleton, реализующая одноимённый паттерн
- `bin/main.cpp` — главный цикл программы, синхронизирующий графический и симуляционный модули



## 6 Пользовательский интерфейс

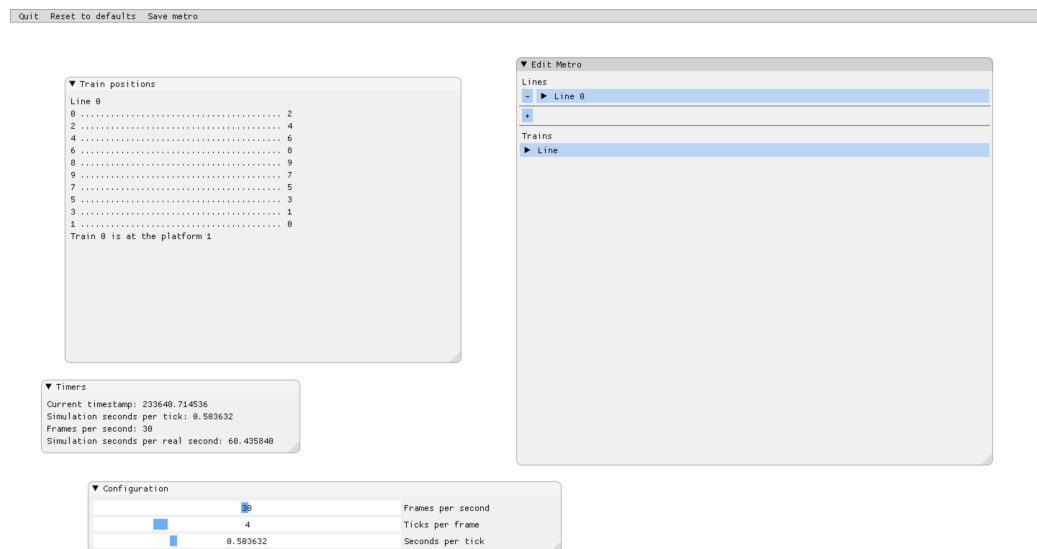


Рис. 1: Общий вид интерфейса

Набор интерфейсов системы (1) состоит из

- Интерфейса отображения состояния симуляции в псевдографике
- Интерфейса редактирования структуры метро
- Интерфейса редактирования хода симуляции
- Интерфейса отображения вспомогательных характеристик системы

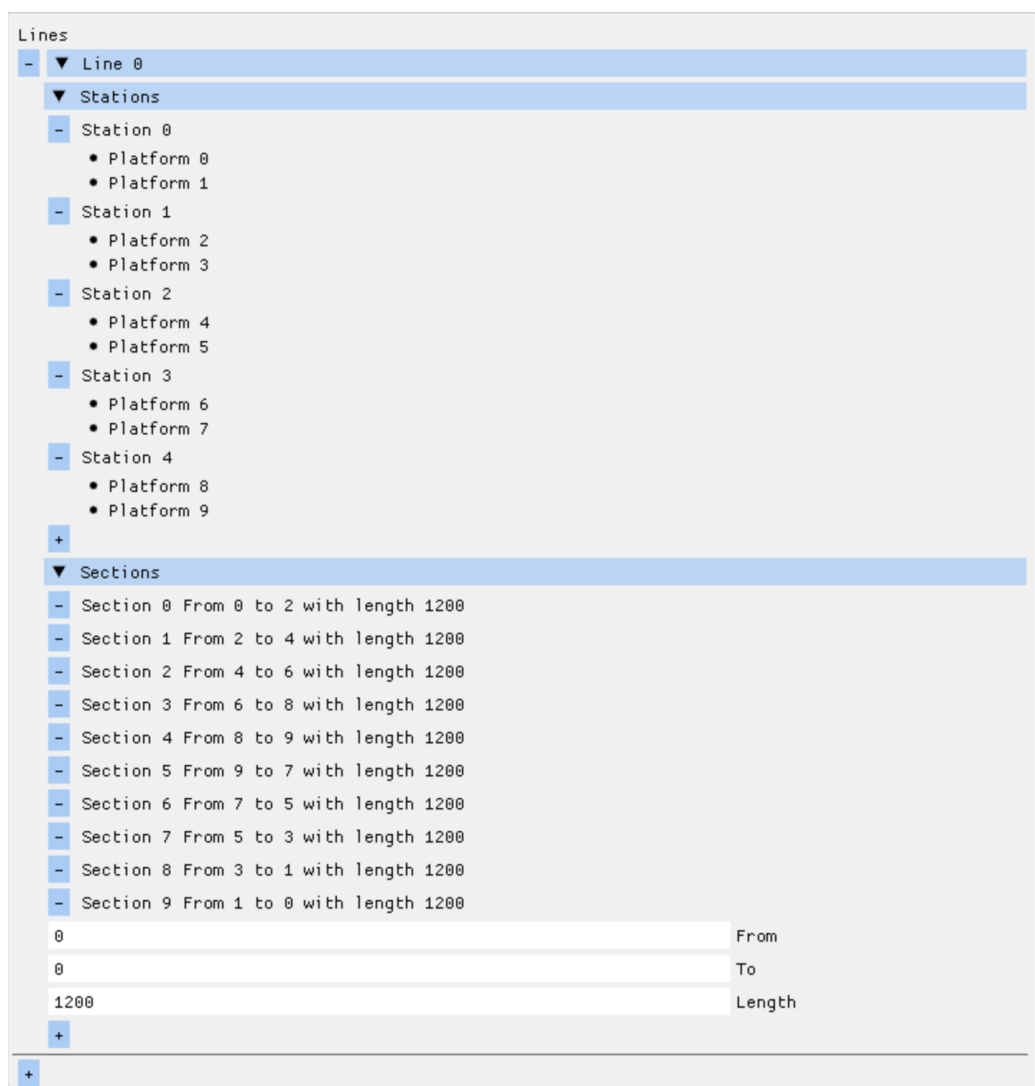


Рис. 2: Интерфейс редактирования линий

Интерфейс редактирования линий (2) позволяет редактировать структуру метро. Добавлять/удалять линии, станции, секции

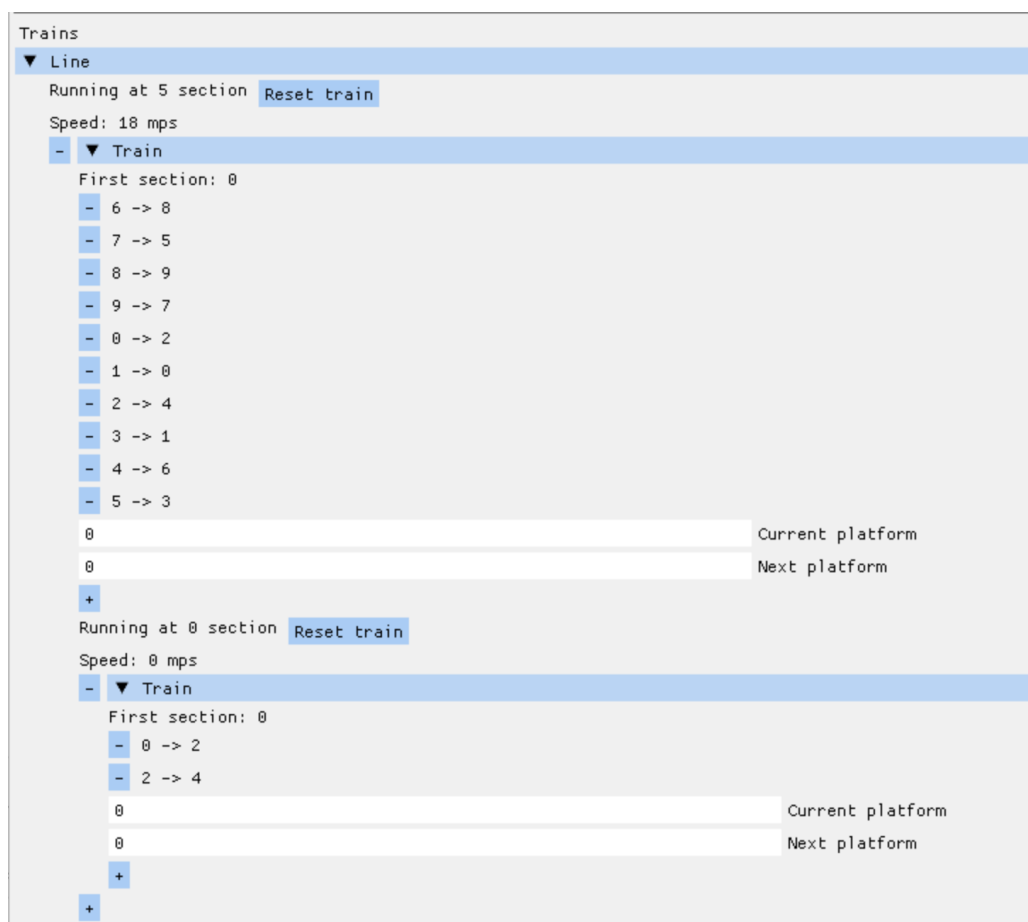


Рис. 3: Редактирование поездов

Интерфейс редактирования поездов (3) позволяет настраивать поезда. Добавлять/удалять, редактировать скорость.