

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего  
образования

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Факультет систем управления и робототехники

Отчет по лабораторной работе №3  
по дисциплине «Программирование»  
Вариант № 912345

Выполнил: студент гр. R3135  
Хачатрян Георгий Робертович  
Преподаватель:  
Мартин Райла

Санкт-Петербург, 2020 г.

## Текст задачи

Описание предметной области, по которой должна быть построена объектная модель:

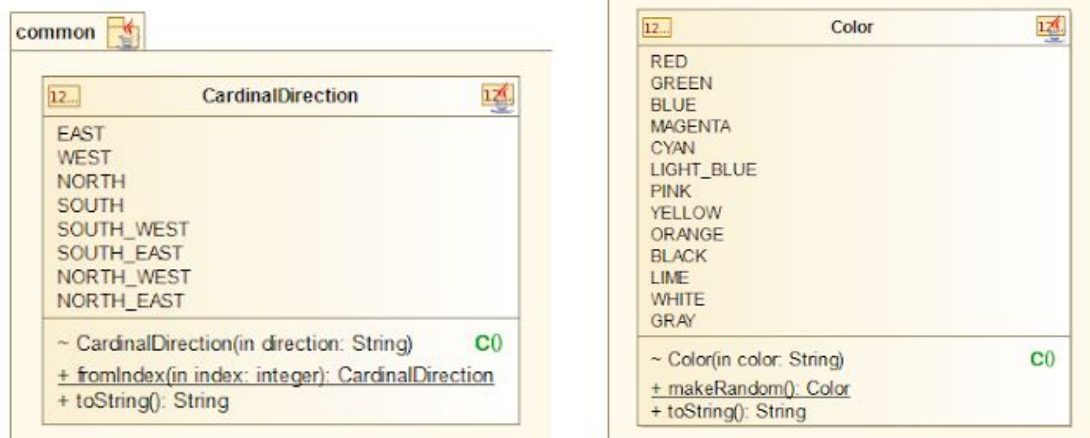
Все засмеялись, а Ворчун сказал: Воздушный шар поднялся еще выше, и весь Цветочный город был виден как на ладони. Дома казались совсем крошечными, а коротышек уж и совсем нельзя было разглядеть. Воздушный шар относил ветром, и скоро весь город виднелся далеко позади.

## Исходный код

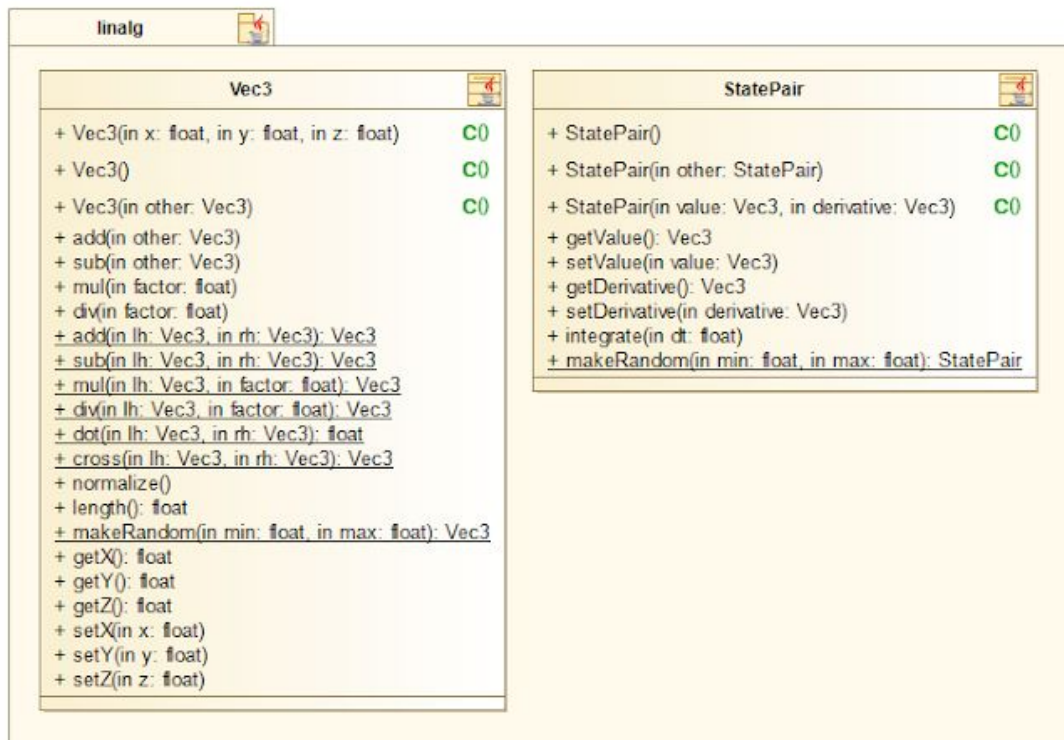
### Структура проекта:

Весь исходный код проекта разбит на 6 пакетов:

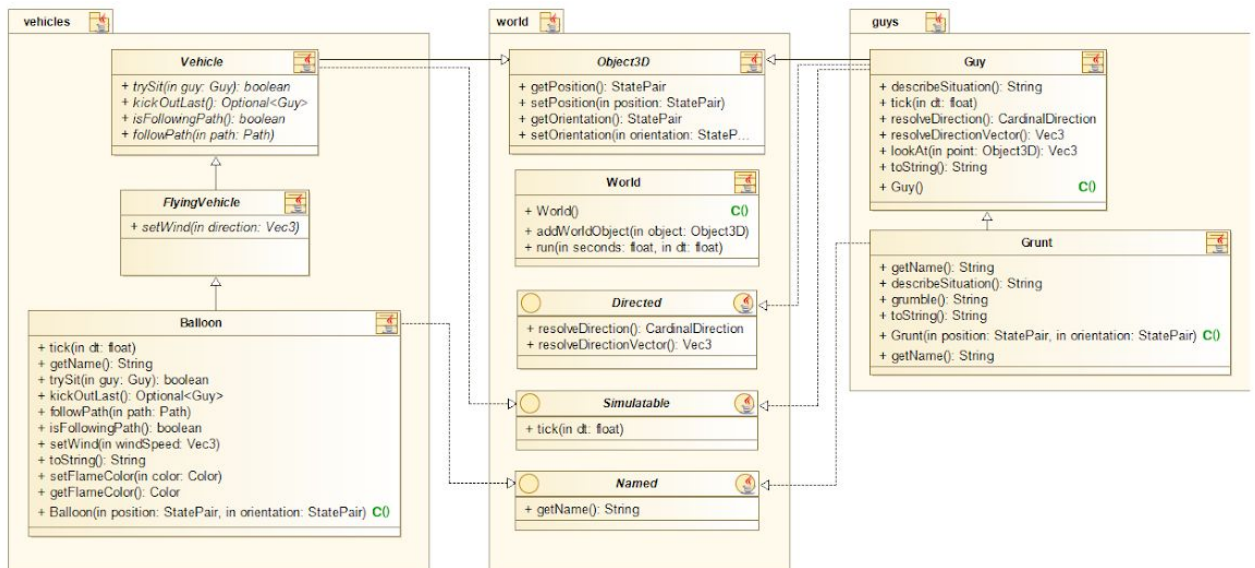
- common – общие enum'ы



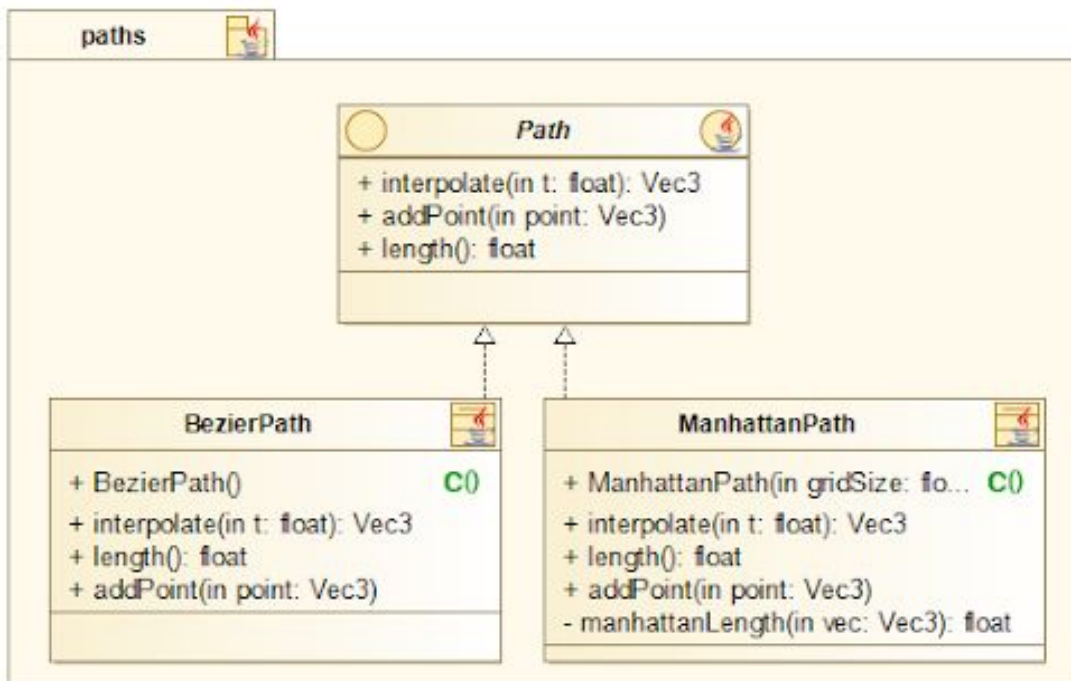
- linalg – базовая линейная алгебра



- world – суперклассы и интерфейсы классов объектов в мире
- guys – классы людей
- vehicles – классы транспортных средств



- paths – классы путей



[Общая диаграмма классов](#)

**Компиляция:**

Команда для компиляции, выполняется в директории “./”

```
javac -sourcepath ./src ./src/Main.java
```

**Создание jar архива:**

Команда для создания jar архива, выполняется в директории “./src”

```
jar cvmf ../Manifest.txt ../Lab3.jar .
```

**Запуск:**

```
java -jar Lab3.jar
```

[Результат выполнения программы](#)**Вывод:****1. Моделирование, UML в частности.**

При разработке приложений с использованием объектно-ориентированных языков часто прибегают к моделированию частей или всего приложения с помощью диаграмм. Предлогом для этого являются “Действенная и эффективная коммуникация” и “Полезная и стабильная абстракция”, а также другие менее существенные причины. Обратной стороной использования подобных практик является сложность или невозможность смены курса разработки приложения при ошибке на первом этапе - моделировании. Ошибку же на этом этапе совершить очень легко так как разработка еще не началась, большинство нюансов и подводных камней не найдено.

**2. Объектно-ориентированное программирование.**

Данная парадигма программирования предоставляет несколько базовых концепций:

- a. Инкапсуляция - объединение данных и методов работы с ними. Часто добавляется еще одно значение - сокрытие, т.е. разделение данных и методов на публичные, приватные и др.
- b. Наследование - создание подклассов, в которых сохраняется функциональность суперкласса.
- c. Полиморфизм - перегрузка и переопределение методов при создании или наследовании классов.

Результатом использования этих концепций является код, в котором все данные являются независимыми, дискретными сущностями, находящимися в случайных местах памяти. Т.е. программа постоянно платит цену плохой локальности данных, постоянных промахов кэша и неверно предсказанных ветвлений. К тому же концепции, которые были на первый взгляд плюсами, в итоге мешают добавлению или изменению функциональности. Сокрытие не дает добраться до нужных, но приватных данных, наследование вытекает в избыточную зависимость подклассов от суперкласса.

3. Результаты повсеместного применения объектно-ориентированного подхода.

Медленно и неверно работающий софт стал нормой. Никого не удивляет то, что текстовые редакторы открывают файлы в пару тысяч строк несколько секунд, IDE загружают свои XML проекты десятки секунд, а полная компиляция относительно небольшой программы может занять минуты.

Говорить о программах от Adobe и MS не вижу смысла. Все это - результат того, что в индустрии никого не волнует как работает их продукт. Проектами руководят люди, которые видят только мнимые плюсы решения и не видят никаких минусов. Последние 20 лет все минусы “инновационных решений” компенсировались прогрессом в железе. Но может ли это продолжаться бесконечно?