



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

Отчет по заданию практикума по
объектно-ориентированному программированию

Моделирование распространения вирусного заболевания

Выполнил:
студент 425 группы
Маннанов Айрат

Москва, 2025

Содержание

1	Уточнение постановки задачи	2
1.1	Основные правила моделирования	2
1.2	Факторы, влияющие на распространение вируса	2
2	Диаграмма классов	3
3	Текстовые спецификации основных классов	4
3.1	Класс Simulation	4
3.2	Класс Government	4
3.3	Класс City	5
4	Диаграмма объектов	5
5	Средства разработки	5
6	Файловая структура	6
7	Пользовательский интерфейс	7
7.1	Экран настройки параметров	7
7.2	Экран симуляции	8

1 Уточнение постановки задачи

Модель представляет собой симуляцию распространения вирусного заболевания в стране, состоящей из нескольких городов разного типа. Основной целью является минимизация количества заболевших за счёт эффективного распределения вакцин и контроля распространения инфекции.

1.1 Основные правила моделирования

- **Вакцинация и иммунитет:**

- Вакцинированные жители не могут заболеть.
- Вакцина действует моментально и длится три недели.
- Заболевшие не могут вакцинироваться.

- **Распространение инфекции:**

- Если в городе уже есть заболевшие, инфекция распространяется среди населения в зависимости от количества заражённых.
- Если город ещё не заражён, основным фактором распространения вируса является городской коэффициент распространения инфекции, который зависит от заболевших во всей стране.

- **Обозначение заражённых городов:**

- Если в городе начинается распространение вируса, он выделяется голубой рамкой.
- Если процент заболевших превышает 45%, город считается эпидемиологически критическим.

1.2 Факторы, влияющие на распространение вируса

- **Тип города:** мегаполисы, средние города и посёлки имеют разные уровни распространения инфекции.
- **Насыщенность транспорта:** влияет на скорость распространения вируса.
- **Сезонность:** заболеваемость растёт осенью и зимой, а весной и летом снижается.
- **Процент заболевших:** чем выше процент заболевших, тем быстрее инфекция распространяется среди оставшихся здоровых жителей.
- **Процент вакцинированных:** наличие вакцинированных снижает риск распространения заболевания.

2 Диаграмма классов

Диаграмма классов показывает структуру проекта и взаимосвязи между основными объектами системы.

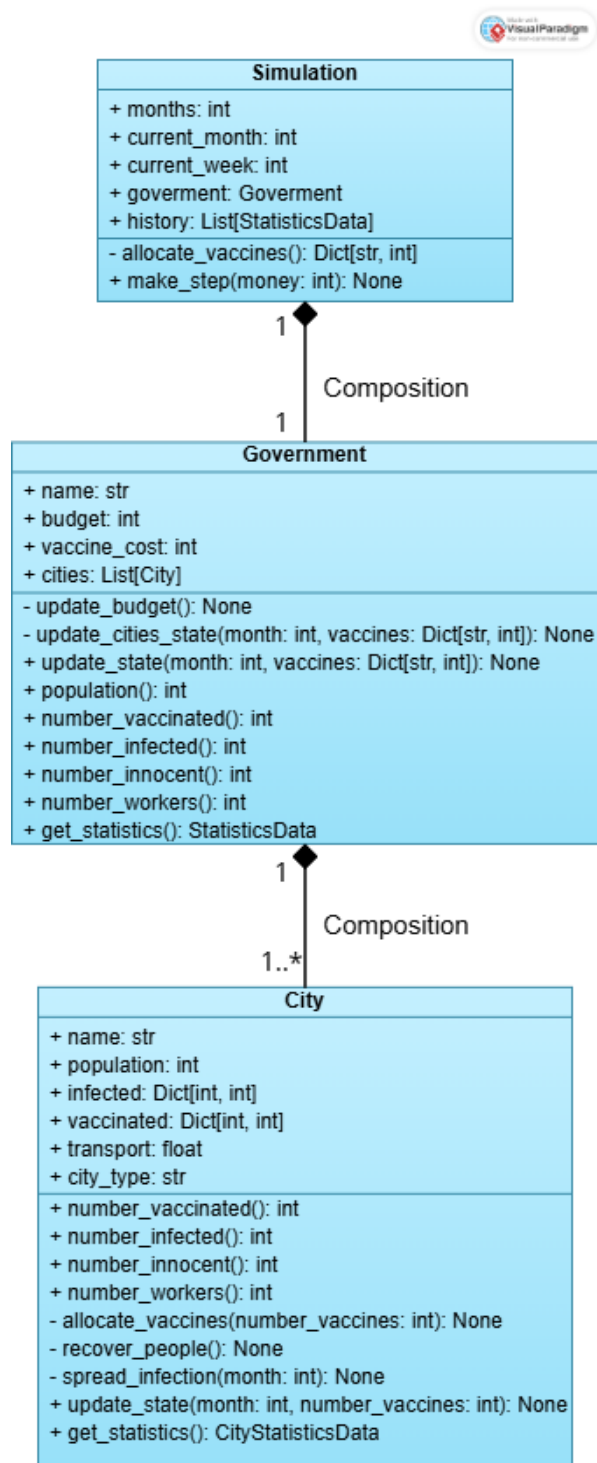


Рис. 1: Диаграмма классов проекта

3 Текстовые спецификации основных классов

В проекте реализованы следующие ключевые классы:

3.1 Класс Simulation

Класс `Simulation` отвечает за управление процессом моделирования.

Атрибуты:

- `months: int` – общее число месяцев моделирования.
- `current_month: int` – текущий месяц симуляции.
- `current_week: int` – текущая неделя.
- `government: Government` – объект, управляющий государственными параметрами.
- `history: List[StatisticsData]` – история изменений состояния системы.

Методы:

- `allocate_vaccines(vaccines: Dict[str, int])` – распределение вакцин.
- `make_step(money: int)` – выполнение шага моделирования.

3.2 Класс Government

Класс `Government` управляет бюджетом, вакцинацией и сбором статистики.

Атрибуты:

- `name: str` – название государства.
- `budget: int` – бюджет государства.
- `vaccine_cost: int` – стоимость одной вакцины.
- `cities: List[City]` – список городов.

Методы:

- `_update_budget()` – обновление бюджета.
- `_update_cities_state(month, vaccines)` – обновление состояния городов.
- `update_state(month, vaccines)` – пересчёт состояния модели.
- `get_statistics()` – получение текущей статистики.

3.3 Класс City

Класс City моделирует отдельный город и его состояние.

Атрибуты:

- `name: str` – название города.
- `population: int` – численность населения.
- `infected: Dict[int, int]` – количество заражённых на разных этапах.
- `vaccinated: Dict[int, int]` – количество вакцинированных.
- `transport: float` – насыщенность транспорта.
- `city_type: str` – тип города (мегаполис, средний, поселок).

Методы:

- `_allocate_vaccines(number_vaccines)` – распределение вакцин в городе.
- `_recover_people()` – обновление данных о выздоровлении.
- `_spread_infection(month)` – моделирование распространения инфекции.
- `update_state(month, number_vaccines)` – обновление состояния города.
- `get_statistics()` – получение статистики по городу.

4 Диаграмма объектов

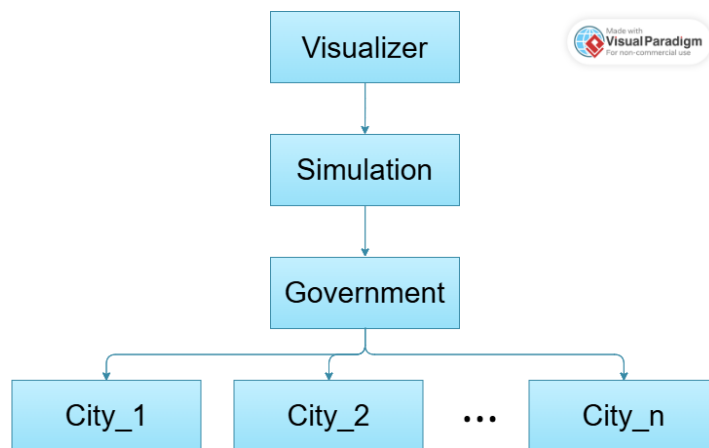


Рис. 2: Диаграмма объектов проекта

5 Средства разработки

- Язык программирования: python

- Модуль для работы с графикой и интерфейсом: `pygame`
- Среда разработки: Visual Studio Code
- Проектирование: [Диаграммы](#), [Прототипы интерфейса](#)

6 Файловая структура

Ниже представлена файловая структура проекта с описанием каждого элемента:

```
Viral-disease-model/
-- models/
--   city.py
--   government.py
--   simulation.py
--   visualizer.py
--   config.py
-- project_plan/
--   images
-- .gitignore
-- README.md
-- main.py
-- requirements.txt
```

- `models/` – директория, содержащая основные модули модели:
 - `city.py` – модуль, описывающий класс `City`, представляющий город в модели.
 - `government.py` – модуль, описывающий класс `Government`, управляющий государственными параметрами и бюджетом.
 - `simulation.py` – модуль, содержащий класс `Simulation`, управляющий процессом моделирования.
 - `visualizer.py` – модуль, отвечающий за визуализацию процесса моделирования с использованием библиотеки `Pygame`.
 - `config.py` – модуль, содержащий параметры по умолчанию для симуляции
- `project_plan/` – директория с изображениями проекта
- `.gitignore` – файл, определяющий файлы и директории, игнорируемые системой контроля версий `Git`.
- `README.md` – файл с общей информацией о проекте, инструкциями по установке и запуску.
- `main.py` – основной скрипт для запуска симуляции.
- `requirements.txt` – файл, содержащий список зависимостей проекта для установки необходимых библиотек.

7 Пользовательский интерфейс

Весь пользовательский интерфейс разбивается на два окна. В каждом окне в левом верхнем угле располагается кнопка выхода.

7.1 Экран настройки параметров

Интерфейс представляет собой графическое меню, разработанное с использованием библиотеки Pygame. На стартовом экране пользователю предлагается задать параметры моделирования:

- Длительность моделирования в месяцах.
- Начальный месяц симуляции.
- Количество городов.
- Начальный бюджет.
- Стоимость вакцины.

В центре интерфейса отображается список доступных городов, при выборе одного из них (выделяется красным) справа появляются соответствующие параметры:

- Тип города (мегаполис, средний, поселок).
- Численность населения.
- Количество зараженных.
- Число вакцинированных.
- Уровень транспортной насыщенности.

После заполнения всех параметров пользователь может нажать кнопку **"Начать симуляцию"**, которая запускает процесс моделирования.

x
Симуляция распространения вирусного заболевания

Укажите длительность моделирования

Число от 1 до 12: 1

Укажите начальный месяц

Число от 1 до 12: 1

Укажите количество городов

Число от 1 до 10: 6

Укажите начальный бюджет

Число до триллиона: 0

Укажите стоимость вакцины

Число до миллиона: 1

Город 1

Город 2

Город 3

Город 4

Город 5

Город 6

Укажите тип города

город

Укажите население

Число до триллиона: 700

Укажите количество заражённых

Число до триллиона: 0

Укажите количество вакцинированных

Число до триллиона: 0

Укажите уровень транспорта

Число от 1 до 100: 50

Начать симуляцию

Рис. 3: Конфигурация симуляции

7.2 Экран симуляции

После запуска моделирования интерфейс переходит в режим симуляции, где отображается текущее состояние городов и ключевые статистические показатели.

Основная часть экрана занята картой городов. Города представлены в виде кругов, размер которых пропорционален численности населения. Цвет круга указывает на уровень зараженности:

- **Зелёный** – низкий уровень заражения.
- **Жёлтый** – средний уровень заражения.
- **Красный** – высокий уровень заражения (эпидемия).

Каждый город подписан, а также обозначен буквой:

- **М** – мегаполис.
- **Г** – город среднего размера.
- **П** – посёлок.

Справа располагается информационная панель, отображающая статистику по всей стране:

- Общее число заболевших.
- Количество вакцинированных.
- Число здоровых жителей.
- Текущий бюджет страны.

- Стоимость одной вакцины.

Ниже расположены элементы управления:

- Кнопка **"Затраты на вакцины"** – позволяет указать количество денег, выделяемых на вакцинацию.
- Кнопки **"Предыдущий шаг"** и **"Следующий шаг"** – перемещение по временной шкале симуляции.
- Кнопка **"Установить новую конфигурацию"** – завершает текущую симуляцию и возвращает к экрану настройки параметров.

В нижнем левом углу отображается текущий месяц и неделя моделирования.

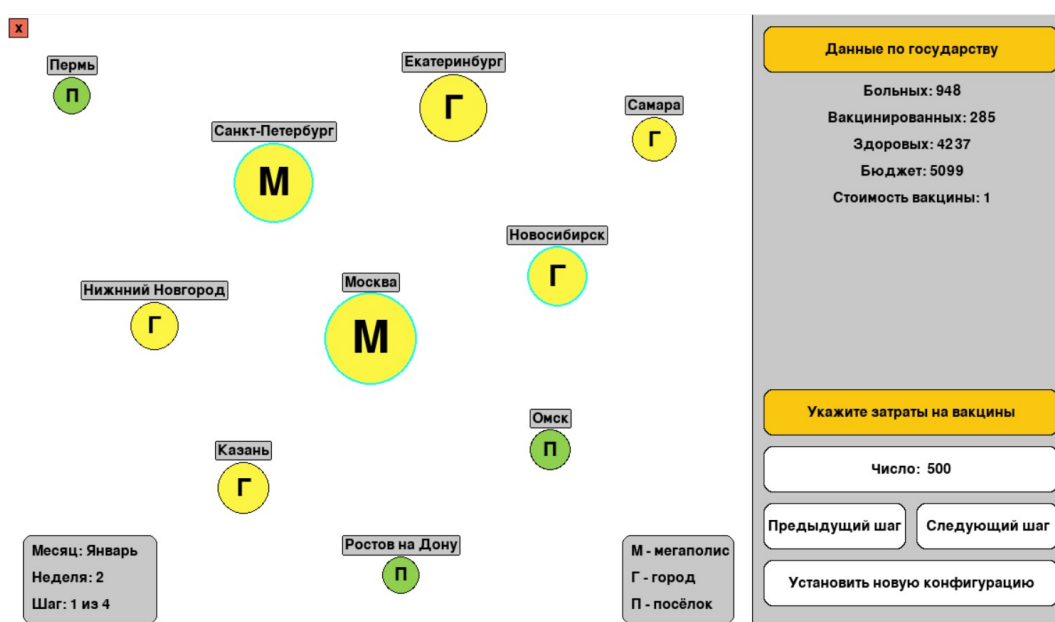


Рис. 4: Симуляция

При нажатии на круг, соответствующий городу, в правой панели отображается дополнительная информация по этому городу. Статистика включает:

- Количество заболевших.
- Число вакцинированных.
- Число здоровых жителей.

Это позволяет пользователю анализировать динамику заболеваемости и эффективность вакцинации в каждом городе в отдельности.

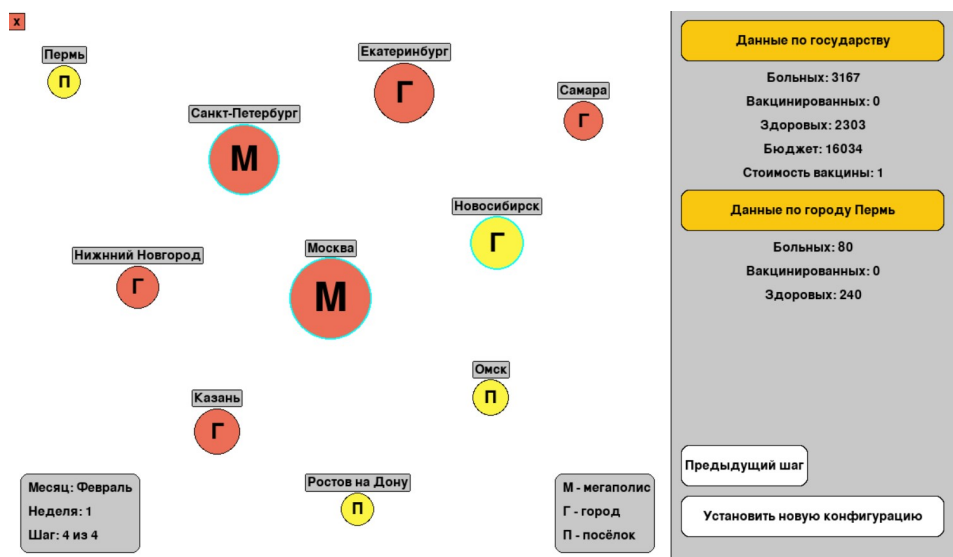


Рис. 5: Симуляция