

# Инструкция к программе имитации спасательных операций

Михаил Преображенский

September 2024

## Содержание

<b>1</b>	<b>Установка программы</b>	<b>2</b>
1.1	Установка Python и pip . . . . .	2
1.2	Создание директории . . . . .	2
1.3	Создание виртуальной среды . . . . .	3
1.4	Установка сторонних библиотек . . . . .	3
1.5	Запуск программы . . . . .	3
1.6	Использование батника для автоматизации . . . . .	3
<b>2</b>	<b>Инструкция к программе</b>	<b>4</b>
2.1	Выбор метода . . . . .	4
2.2	Параметры метода графов . . . . .	5
2.3	Спасательные станции . . . . .	5
2.4	Запуск эксперимента . . . . .	6
2.4.1	Графики . . . . .	7
2.4.2	Метрики . . . . .	7
2.4.3	Запуск программы . . . . .	8

# 1 Установка программы

Данная программа представляет из себя архив, состоящий из нескольких файлов, написанных на языке Python3 с использованием сторонних библиотек, и папок с нужными ресурсами. Поэтому для правильной установки данной программы нужно выполнить несколько несложных последовательных шагов, описанных в данной главе.

## 1.1 Установка Python и pip

Для начала нужно установить сам язык программирования Python и менеджер пакетов pip.

1. Перейдите на официальный сайт Python: [python.org/downloads](https://python.org/downloads).
2. Скачайте и установите последнюю версию Python для Windows.
3. Во время установки убедитесь, что выбрана опция "Add Python to PATH".
4. После установки Python, убедитесь, что pip также установлен. Для этого откройте командную строку (CMD) и выполните команду:

```
pip --version
```

Если pip установлен, вы увидите версию pip. Если нет, установите pip, выполнив команду:

```
python -m ensurepip --upgrade
```

## 1.2 Создание директории

Так как мы хотим, чтобы наш проект хранился не абы где, а в определенном месте, нам нужно выделить под него отдельную папку. Для примера, я создал папку **RescueApp** в **Документах**. Выглядит это так как показано на (рис. 1).

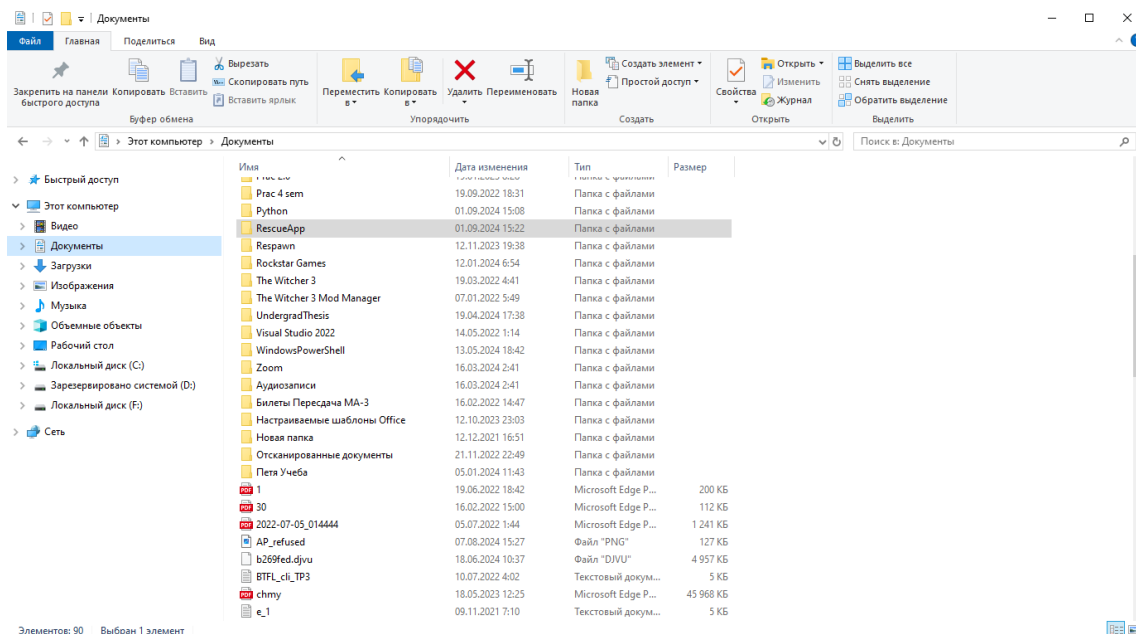


Рис. 1: Новая папка

После создания папки распаковываем архив и перемещаем содержимое в неё.

### 1.3 Создание виртуальной среды

**ВАЖНО! Шаги 1.3 - 1.5 нужны для того случая если не сработает вариант с батником 1.6, это возможно так как я никогда раньше не имел с ними дела и мб где-то накосячил. Так что сначала посмотрите его.**

Для изоляции зависимостей проекта от глобальных библиотек Python, рекомендуется создать виртуальную среду. Для этого откройте командную строку (CMD) и выполните следующие команды:

1. Перейдите в папку проекта:

```
cd Документы\RescueApp
```

2. Создайте виртуальную среду:

```
python -m venv venv
```

3. Активируйте виртуальную среду:

```
venv\Scripts\activate
```

После выполнения этих команд, вы увидите, что в командной строке появится префикс (venv), что означает, что виртуальная среда активирована.

### 1.4 Установка сторонних библиотек

Теперь, когда виртуальная среда активирована, можно установить все необходимые сторонние библиотеки. Для этого выполните следующую команду:

```
pip install -r requirements.txt
```

Файл `requirements.txt` должен находиться в папке проекта и содержать список всех необходимых библиотек.

### 1.5 Запуск программы

После установки всех библиотек, вы можете запустить программу, выполнив следующую команду:

```
python main.py
```

Где `main.py` — это основной файл вашей программы.

### 1.6 Использование батника для автоматизации

Для удобства, можно использовать батник, который автоматически выполнит все необходимые шаги для установки и запуска программы. Для этого я создал файл `run_program.bat`, и поместил его в папку проекта. Итого, для удобного запуска программы достаточно будет просто создать ярлык на рабочем столе, который открывает данный файл. Вот содержимое батника:

```
@echo off
REM Переходим в директорию проекта
cd /d %~dp0

REM Выводим текущий путь для отладки
echo Current path: %cd%

REM Проверяем наличие файла requirements.txt
if not exist requirements.txt (
    echo File requirements.txt not found!
    pause
)
```

```

    exit /b 1
)

REM Создаем виртуальную среду, если она еще не создана
if not exist venv (
    python -m venv venv
)

REM Активируем виртуальную среду
call venv\Scripts\activate

REM Устанавливаем необходимые библиотеки, если они еще не установлены
pip install -r requirements.txt

REM Запускаем программу
python main.py

REM Деактивируем виртуальную среду после завершения работы программы
deactivate

pause

```

Достаточно просто запустить этот батник, чтобы установить все необходимые библиотеки и запустить вашу программу.

**ПЕРВЫЙ ЗАПУСК МОЖЕТ ЗАНЯТЬ МНОГО ВРЕМЕНИ (до 5ти минут)**  
**ЕСЛИ ПРИ ЗАПУСКЕ ДАЖЕ ПО ПРОШЕСТВИИ 5ТИ МИНУТ НИЧЕГО НЕ ПРОИСХОДИТ, МОЖНО ПОПРОБОВАТЬ ЗАПУСТИТЬ ФАЙЛ ОТ ИМЕНИ АДМИНИСТРАТОРА**

**ТАК ЖЕ АНТИВИРУСЫ МОГУТ НАЧАТЬ РУГАТЬСЯ НА БАТНИК, В ЭТОМ СЛУЧАЕ НАДО УЖЕ ПОСТУПАТЬ В ЗАВИСИМОСТИ ОТ КОНКРЕТНОГО АНТИВИРУСА (Касперский, например, долго его рассматривает, после чего помечает как безопасный и разрешает им пользоваться)**

## 2 Инструкция к программе

Сам интерфейс программы представляет из себя следующую менюшку, написанную на PyQt6 (рис. 2). Как видно, она состоит из 4х основных секций. Давайте по порядку рассмотрим каждую из них.

### 2.1 Выбор метода

При реализации программы было рассмотрено два метода нахождения кратчайшего пути от спасательных станций до места ЧП — метод прямых и метод графов.

- **Метод прямых** — быстрый, неточный метод, сильно специализированный под конкретный случай Финского залива. Этот метод был написан мной для того, чтобы свести к минимуму время, затрачиваемое на вычисление одного случая, он создан исключительно для получения приблизительных результатов и может показывать странные результаты для некоторых расстановок СПС (например, для маленького количества спасательных станций наблюдались ошибки, когда траектории спасательных бригад прокладывались задевая сушу). Если честно, не рекомендуется для получения визуальных картинок.
- **Метод графов** — намного более точный, но в то же время ресурсоёмкий метод. Построен на основе алгоритма Дейкстры и дискретизации данного пространства. Этот метод более универсален и намного легче переносится на другие акватории, но в то же время требует больше времени для выполнения.

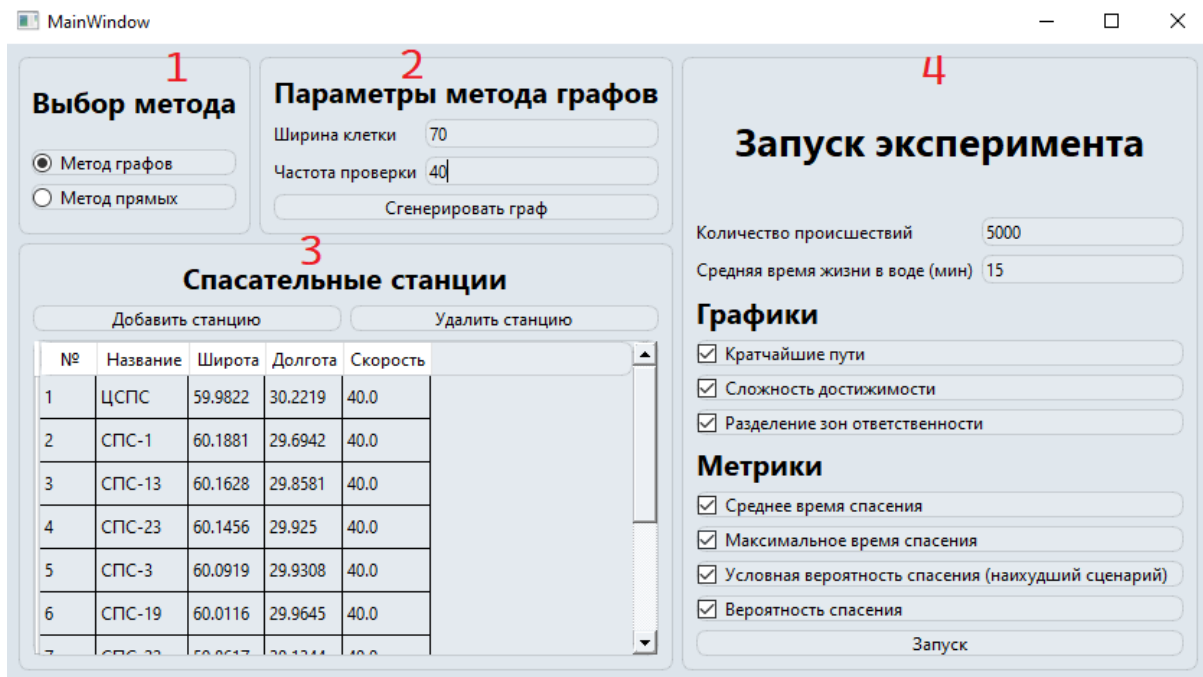


Рис. 2: Интерфейс программы

## 2.2 Параметры метода графов

Данный раздел доступен, очевидно, только при выбранном методе графов и позволяет более тонко настроить некоторые его параметры, такие как ширина клетки дискретизации и частота проверок пересечения суши.

- **Ширина клетки** — при прогоне алгоритма графов, всё пространство дискретизируется клетками определённой ширины (в пикселях, относительно карты "data/canvas\_new\_fish\_gray.png" с шириной 4113 и высотой 3145 пикселей). По умолчанию стоит значение 70, но для конечных расчётов это значение может быть грубовато, так что рабочий диапазон значений — [20, 100], где чем меньше значение, тем точнее результат, но программа работает дольше.
- **Частота проверок пересечения суши** — довольно неестественный параметр. Дело в том, что при формировании графа по всей длине каждой связи проводятся проверки типа "а проходит ли эта связь через кусок суши?". Так что данный параметр стоит трогать (увеличивать), только если на больших значениях ширины клетки (70+) заметны грубые ошибки по типу траекторий, нагло перепрыгивающих через кусок суши. В таком случае можно поднять этот параметр до 60 или даже выше. В противном случае параметр лучше не трогать.

Важная ремарка заключается в том, что расчёт графа с конкретными параметрами (особенно для маленькой ширины клеток) может занимать даже больше времени, чем сам эксперимент. Поэтому все посчитанные графы автоматически сохраняются в отдельную папку и при выставлении новой совокупности параметров, если граф с такими параметрами уже был посчитан, то программа не тратит время на его генерацию, а сразу подгружает его из памяти. Т.е. чтобы программа работала быстрее, нужно использовать уже ранее использованные совокупности параметров, например, {70, 40}, {40, 40} или {20, 40}. Кнопка "сгенерировать граф" запускает расчет графа с указанными параметрами и загружает его в библиотеку уже подготовленных графов без запуска самого эксперимента.

## 2.3 Спасательные станции

В данном разделе всё просто — тут представлен интерфейс для работы с самим расположением спасательных станций. У каждой станции есть несколько параметров: Название, координаты и скорость спасательного средства, доступного станции. По умолчанию в программу вписан список активных на данный момент спасательных станций, предоставленный мне спасательно-поисковой службой. Для добавления и удаления станций предусмотрены две кнопки.

- **Добавление станции:** при нажатии данной кнопки открывается следующее диалоговое окно (рис. 3). Широту и долготу можно ввести двумя способами — вписать дробные значения

Dialog

### Новая станция

Название

Широта  Градусы Минуты Секунды

Долгота  Градусы Минуты Секунды

Скорость (км/ч)  (прим. 40)

Рис. 3: Добавление станции

градусов (через точку, например 60.12) в первые колонки (градусы) оставив остальные поля пустыми, либо воспользоваться стандартным форматом градусы/минуты/секунды. В обоих случаях программа проверит правильность формата введенных данных, и если всё в порядке, добавит новую станцию в таблицу в секции 3.

- **Удаление станции:** при нажатии данной кнопки открывается следующее диалоговое окно (рис. 4). В данной таблице можно нажатием мыши выбрать те станции, которые следует

Выбор станций для удаления

№	Название	Широта	Долгота	Скорость
1	ЦСПС	59.9822	30.2219	40.0
2	СПС-1	60.1881	29.6942	40.0
3	СПС-13	60.1628	29.8581	40.0
4	СПС-23	60.1456	29.925	40.0
5	СПС-3	60.0919	29.9308	40.0
6	СПС-19	60.0116	29.9645	40.0
7	СПС-22	59.8617	30.1344	40.0
8	СПС-10	59.8625	30.0456	40.0
9	СПС-30	59.9092	29.8106	40.0
10	СПС-21	60.0051	29.7167	40.0

Рис. 4: Удаление станции

удалить, после чего нажать кнопку "ОК". Эти станции будут удалены из списка.

## 2.4 Запуск эксперимента

Это главная секция интерфейса, из которой, собственно, и запускается выполнение эксперимента. В ней мы рассмотрим следующие параметры.

- **Количество происшествий:** — число случаев (точек), раскиданных по акватории. В математической статистике принято считать, что для метода Монте-Карло, используемого нами для расчета вероятности спасения, достоверный размер выборки должен составлять не менее 10000 точек, тем не менее так как у нас не самая точная модель, количество точек можно снизить до 2000 без значительной потери в точности, при этом ощутимо снизив время работы программы.
- **Среднее время жизни в воде:** — время, при котором абсолютное большинство людей, помещенных в данные экстремальные условия, погибнет.

### 2.4.1 Графики

Тут можно выбрать, графики какого вида мы хотим увидеть на выходе работы программы. Пока добавлено всего три варианта:

- Кратчайшие пути — главный график (рис. 5), на котором отмечены спасательные станции и места сгенерированных ЧП, а так же изображены оптимальные траектории .

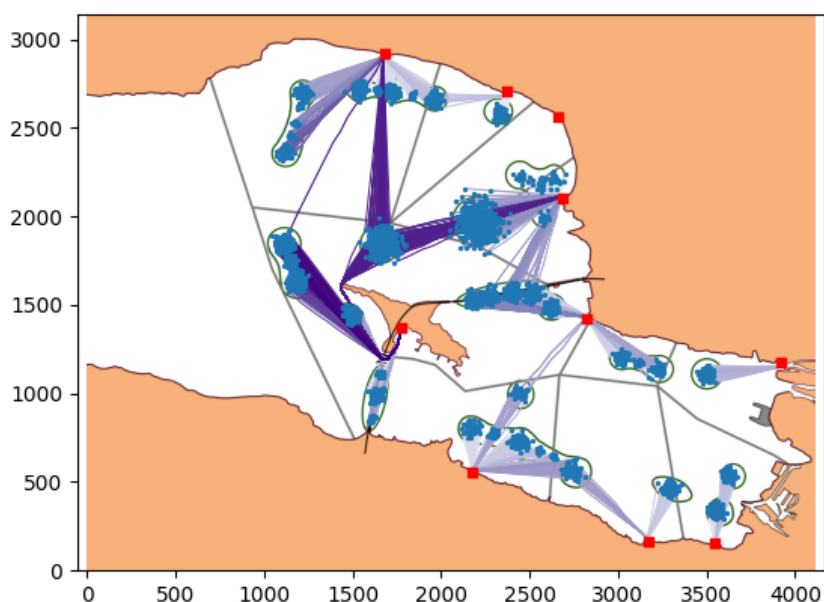


Рис. 5: Пример графика кратчайших путей

- Сложность достижения — вспомогательный график (рис. 6). На нём с помощью тепловой карты отмечены труднодоступные участки акватории (чем темнее участок, тем дольше до него добраться из всех спасательных станций в совокупности).
- Распределение зон ответственности — ещё один вспомогательный график (рис. 7). На нём вся акватория раскрашена в разные цвета радуги, каждый из которых соответствует зоне покрытия конкретной спасательной станции, размещённой на нём. То есть, например, внутри зоны зеленого цвета быстрее всего среагирует именно станция, расположенная в этой зоне.

### 2.4.2 Метрики

Тут уже всё тривиально, нужно выбрать те числовые значения, которые хочется вывести на экран в результате работы программы. Выводимое время реагирования показывается в минутах и **НЕ ВКЛЮЧАЕТ** в себя то время, которое требуется на подготовку экипажу к выходу.

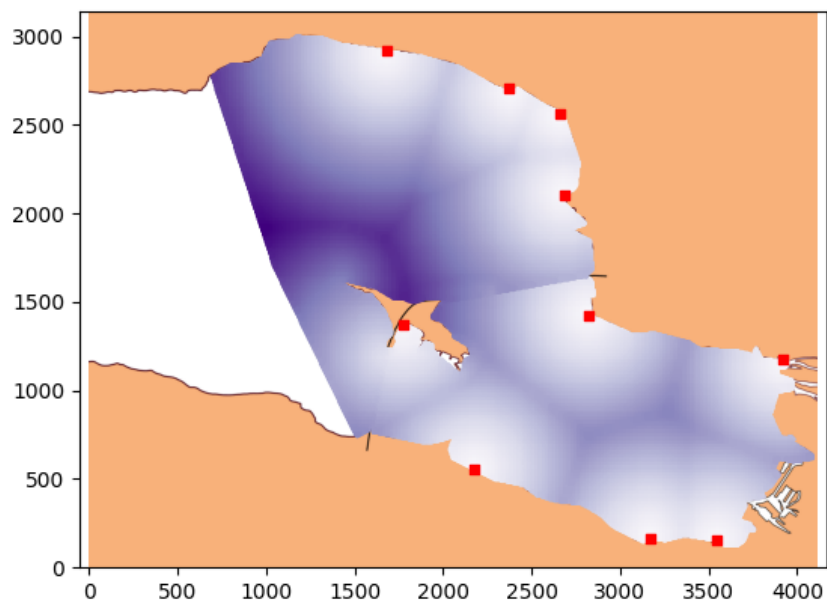


Рис. 6: Пример графика сложности достижимости

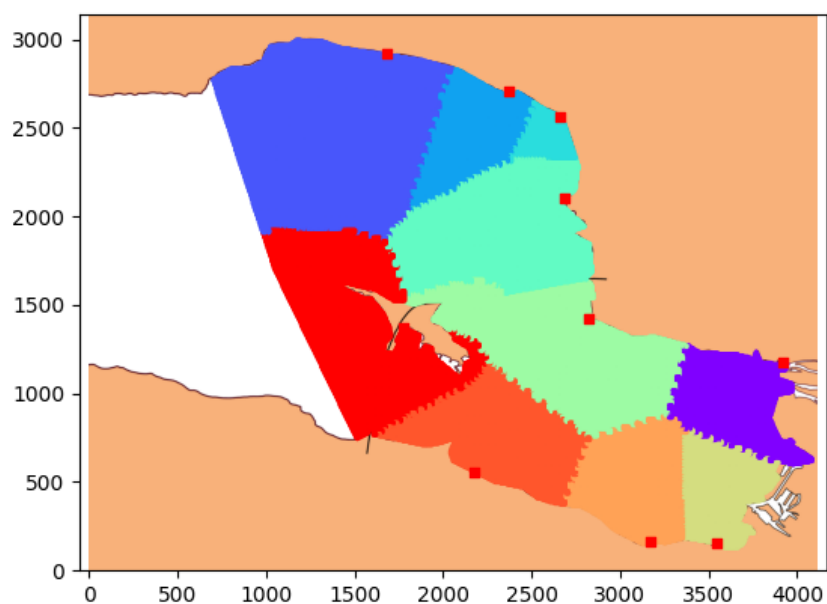


Рис. 7: Пример графика распределения зон ответственности

### 2.4.3 Запуск программы

После нажатия на кнопку "Запуск" начинается выполнение вычислений и вывод полученных графиков. Это может занять некоторое время. К сожалению, я всё ещё работаю над тем, чтобы на время проведения вычислений кнопка "Запуск" меняла свой внешний вид и отключалась, поэтому пока следует обращать внимание на то, что не стоит несколько раз подряд нажимать на эту кнопку, так как это может привести к замедлению процесса (запустится сразу несколько экспериментов).



Окна с полученными результатами экспериментов не закрываются после запуска новых его итераций, так что для большей наглядности можно запустить эксперимент, получить результат, поменять данные (например, изменить параметры графа или добавить ещё станции) и запустить эксперимент снова. В результате будет открыто два окна с результатами, которые можно будет поставить бок о бок и сравнивать друг с другом, что довольно удобно.