# Генерация графовых задач для ОГЭ / ЕГЭ. Архитектура + описание проекта и mvp.

MVP: будет описан в виде интерфейса в конце документа
ЦА: репетиторы, школьники

## Техническое задание (ТЗ) для разработки программы тренировки к экзамену по графам

### 1. Введение

Целью данного проекта является создание интерфейса, который позволит экзаменаторам тренироваться к экзамену по теме "Графы", а именно 1 задание ЕГЭ, 4,9 задания ОГЭ. Программа будет предоставлять пользователю возможность выбирать различные задания, готовиться по теории, работать с ними и получать обратную связь о правильности своих ответов.

### 2. Основные функции программы

1. Выбор задания:

• Пользователь должен иметь возможность выбрать один из трех типов заданий, связанных с графами.

• Задания могут включать, но не ограничиваться:

• Задание на определение свойств графа.

• Задание на нахождение определенных путей в графе.

• Задание на построение графа по заданным условиям.

2. Генерация задания:

• После выбора типа задания программа должна динамически генерировать задание и отображать его в основном окне.

• Задание должно быть представлено в текстовом формате или в виде графической схемы (при необходимости).
3. Отображение теории:

• Для каждого типа задания должно быть предусмотрено отдельное окно (или диалог), в котором будет представлена теоретическая информация о графах, необходимая для выполнения задания.

• Теория должна быть доступна пользователю перед началом выполнения задания.

4. Ввод ответа:

• Пользователь должен иметь возможность вводить свой ответ в отдельном текстовом поле.

• Поле ввода должно быть достаточно широким для удобного ввода текстовой информации.

5. Проверка ответа:

• После нажатия кнопки "Отправить ответ", программа должна проверить введенный ответ на правильность.

• Если ответ правильный, поле ввода должно подсвечиваться зеленым цветом, если неверный — красным.

• Программа должна также выводить сообщение с результатом проверки (например, "Ваш ответ верный!" или "Попробуйте еще раз.").

6. Переход к следующему заданию:

• После проверки ответа и получения обратной связи от программы, пользователь должен иметь возможность перейти к следующему заданию.

• При этом программа должна заново генерировать новое задание, а также очищать поле ввода для нового ответа.

## 3. Интерфейс пользователя

1. Главное окно:

• Заголовок окна: "Тренировочный экзамен по графам".

• Элементы интерфейса:

• Выпадающий список (ComboBox) для выбора типа задания (3 варианта).

• Кнопка "Начать" для генерации задания.

• Область для отображения задания (QTextEdit или QLabel).

• Кнопка "Теория", открывающая окно с теоретическим материалом.

• Поле ввода для ответа (QLineEdit или QTextEdit).

• Кнопка "Отправить ответ".

2. Окно теории:

• Отдельное окно или диалог, отображающее текст с теоретической информацией о графах и выбранном задании.

3. Обратная связь:

• Подсветка поля ввода (зеленый/красный) в зависимости от правильности ответа.

## 4. Технические требования

1. Язык программирования: Python.

2. Библиотеки: PyQt5 для создания графического интерфейса.

3. Совместимость: Программа должна работать на операционных системах Windows, macOS и Linux.

## 5. Сроки выполнения

• Этапы разработки и тестирования программы должны быть завершены в течение 4 недель с момента начала работы над проектом.

## 6. Заключение

Данная программа будет полезна экзаменаторам, позволяя им эффективно готовиться к экзамену по теме "Графы". Удобный интерфейс и функциональность программы обеспечат качественную подготовку и возможность самопроверки знаний.
Задачи:

## 3 алгоритма

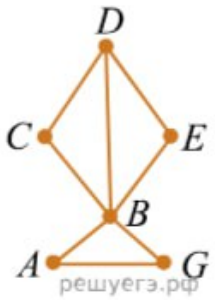
☀️1 задача ЕГЭ 1 вид
☀️4 задача ОГЭ отдельный вид
☀️9 задача ОГЭ от начальной до конечной

- 1 вид

2 категории
- номера этих вершин

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | | * | | * | | |
| 2 | * | | | * | | * |
| 3 | | | | * | * | |
| 4 | * | * | * | | * | * |
| 5 | | | * | * | | |
| 6 | | * | | * | | |

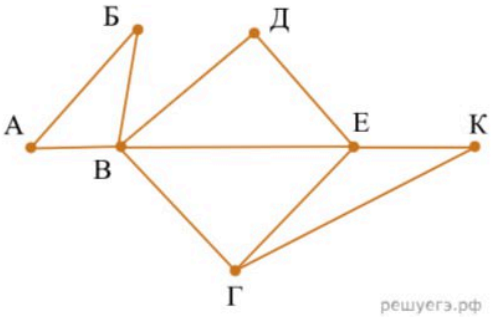
ege.sdamgia.ru



решуегэ.рф

- состояние этих вершин, длину между ними

| | П1 | П2 | П3 | П4 | П5 | П6 | П7 |
|---|---|---|---|---|---|---|---|
| П1 | | 45 | | 10 | | | |
| П2 | 45 | | | 40 | | 55 | |
| П3 | | | | | 15 | 60 | |
| П4 | 10 | 40 | | | | 20 | 35 |
| П5 | | | 15 | | | 55 | |
| П6 | | 55 | 60 | 20 | 55 | | 45 |
| П7 | | | | 35 | | 45 | |



решуегэ.рф

- 2 вид

1 **Тип 4 № 3**

Между населенными пунктами A, B, C, D, E построены дороги, протяженность которых (в километрах) приведена в таблице:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | | 1 | | | |
| B | 1 | | 2 | 2 | 7 |
| C | | 2 | | | 3 |
| D | | 2 | | | 4 |
| E | | 7 | 3 | 4 | |

oge.sdamgia.ru

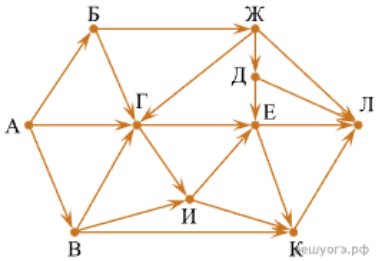
Определите длину кратчайшего пути между пунктами A и E. Передвигаться можно только по дорогам, протяженность которых указана в таблице.

Рассчитать количество путей от начальной до конечной точки через весовые ребра

- 3 вид

1 **Тип 9 № 10258**

На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из пункта А в пункт Л, проходящих через пункт Е?




Разделение обязанностей
Функции + классы

- метод решения

## Краткое описание проекта

Цель нашего проекта - создать десктопное приложение для генерации и решения графовых задач из ОГЭ и ЕГЭ.

Целевая аудитория: школьники, готовящиеся к экзаменам и репетиторы, преподаватели, готовящие к ОГЭ и ЕГЭ.

Пользователь полученного продукта сможет генерировать условия задач 4 и 9 из ОГЭ и задачи 1 из ЕГЭ, а также проверять ответ на правильность и смотреть правильное решение. В придачу получит бесценный блок теории по графам.

# Архитектура проекта

Проект должен быть реализован под архитектуру бек - фронт. Он должен включать в себя:

- Генерацию и решение задач (бек);
- Десктопное приложение (фронт).

При разработке следует придерживаться следующих принципов:

1. SOLID
    a. S - принцип единой ответственности
    b. O - принцип открытости / закрытости
    c. L - принцип заменяемости
    d. I - принцип разделения интерфейсов
    e. D - принцип инверсии зависимостей
2. KISS - пусть всё будет простым до безобразия
3. DRY - не повторяйтесь

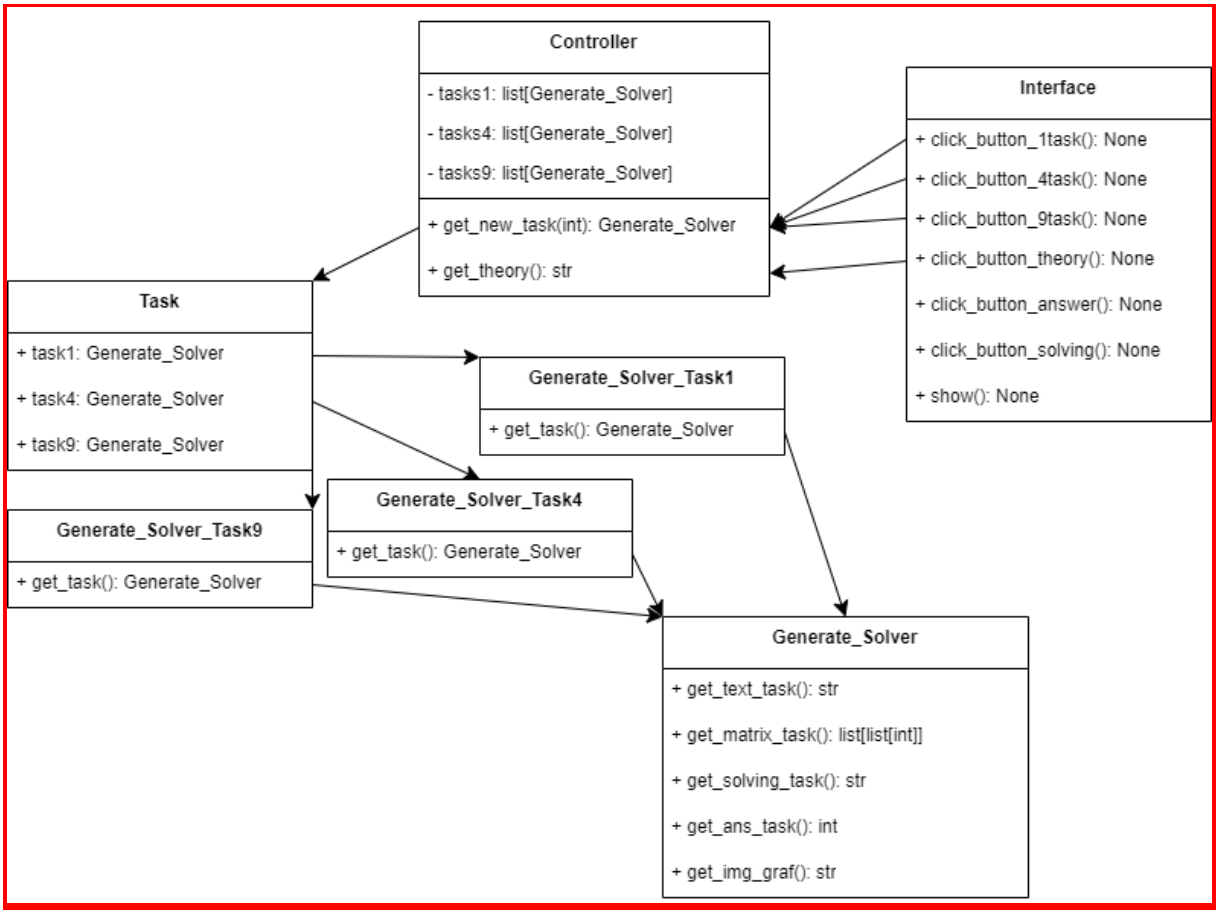
Десктопное приложение должно иметь кнопки, поле для ввода ответа и места для отображения здания и решения, поэтому была выбрана библиотека PyQt на Python.
Для генерации и решения задач необходима простота кода, поэтому выбран Python.

# USM

Доска с задачами:
https://m-yozhka.lukit.ru/project/PFPVUYAS

# Схема классов

Первый класс генерирует текст задачи и матрицу для неё.
Второй класс решает задачу в зависимости от её номера.
Третий класс реализует интерфейс - отображение задач.
Четвертый класс является контроллером - соединяющим интерфейс и бек.




# API

Генерация задач:

- Генерируется условие по шаблону в зависимости от номера 1, 4, 9 (Может сделать Enum?)
- Генерирует матрицу для задачи в зависимости от номера 1, 4, 9

Решение задач:

- Решает задачу по матрице в зависимости от номера 1, 4, 9 и типа условия задачи

Контроллер:

- Связывает работу интерфейса и генератора
- Проверяет повтор заданий
- Возвращает сгенерированную теорию

Интерфейс:

- Отображает вид
- Для задач 1, 9 генерирует красивый вид графа

Роли в команде

Савченко Вероника - отвечает за сборку приложения, Team Lead, Front

Серяков Сергей - Backend, генерация заданий

Корецкая Анна - Интерфейс, документация

Хижняк Юрий - Генерация текстов

Рылкин Максим - Генерация заданий