**1. Назначение и условия применения программы**

**1.1 Назначение и функции, выполняемые приложением**

Приложение "Редактор треугольников" предназначено для создания, редактирования и анализа треугольников на двумерной плоскости.

Основные функциональные возможности приложения:

* Создание треугольников на холсте
* Выбор и перемещение треугольников
* Поворот треугольников вокруг выбранной вершины
* Изменение цвета треугольников
* Определение и визуализация пересечений между треугольниками
* Сохранение и загрузка созданных треугольников

**1.2 Условия, необходимые для использования приложения**

Для использования приложения необходимы следующие условия:

* Операционная система: Windows 7, Windows 10, Linux
* Python 3.8 и выше
* Библиотеки: tkinter
* Рекомендуемая среда разработки: PyCharm или любая другая IDE с поддержкой Python

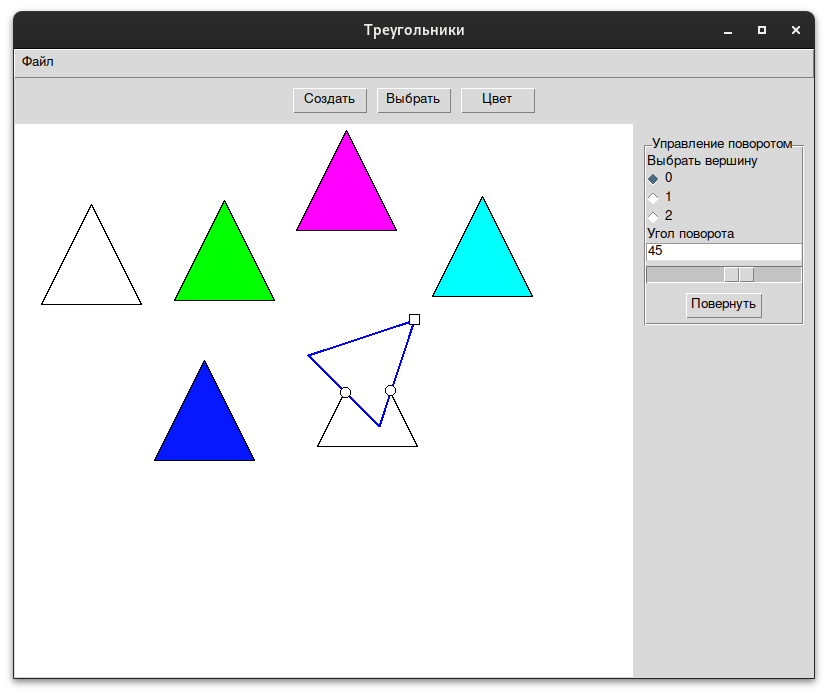
**2. Характеристики программы**

**2.1 Характеристики приложения**

* Количество значимых строк кода: 600
* Количество основных классов: 10
* Количество основных алгоритмов: 5 (создание, перемещение, поворот, определение пересечений, сохранение/загрузка)
* Используемые библиотеки:
* tkinter: для создания графического интерфейса
* math: для математических вычислений
* csv: для работы с CSV-файлами

Работа приложения:

При запуске открывается главное окно с холстом и панелью управления. Пользователь может создавать треугольники, кликая на холст. Треугольники можно выбирать, перемещать и поворачивать. Цвет треугольника можно изменить с помощью цветовой палитры. Пересечения между треугольниками отображаются автоматически. Данные можно сохранить в файл или загрузить из файла.

Рисунок 1 — Интерфейс приложения треугольники

Контроль корректности ввода/вывода:

При загрузке файла проверяется его формат и наличие необходимых данных. При вводе угла поворота проверяется корректность числового значения.

**2.2 Особенности реализации приложения**

Основные структуры данных:

* Vec2: класс для представления двумерных векторов
* Path2D: класс для хранения последовательности точек
* Triangle: класс для представления треугольника
* TriangleManager: класс для управления коллекцией треугольников

Выбор этих структур обусловлен их простотой и эффективностью для решения поставленных задач. Альтернативой могло бы быть использование библиотеки numpy для векторных операций, но это усложнило бы зависимости проекта.

**3. Обращение к программе**

Основные методы и алгоритмы:

Метод rotate(radians): поворот треугольника на заданный угол

def rotate(self, radians):  
 c = self.rotation\_point.get\_center()  
 for p in self.path:  
 if p == c:  
 continue  
 dx = p.x - c.x  
 dy = p.y - c.y  
 p.x = c.x + dx \* math.cos(radians) - dy \* math.sin(radians)  
 p.y = c.y + dx \* math.sin(radians) + dy \* math.cos(radians)

Метод intersection\_lines(): определение пересечения двух отрезков

@staticmethod  
def intersection\_lines(x1, y1, x2, y2, x3, y3, x4, y4):  
 d = (x1 - x2) \* (y3 - y4) - (y1 - y2) \* (x3 - x4)  
 if d == 0:  
 return  
 t = ((x1 - x3) \* (y3 - y4) - (y1 - y3) \* (x3 - x4)) / d  
 u = -((x1 - x2) \* (y1 - y3) - (y1 - y2) \* (x1 - x3)) / d  
 if not (0 <= t <= 1 and 0 <= u <= 1):  
 return  
 x = x1 + t \* (x2 - x1)  
 y = y1 + t \* (y2 - y1)  
 return Vec2(x, y)

Метод App.save(): сохранение данных в CSV-файл

def save(self):  
 with open(self.filename, 'w', newline='') as csvfile:  
 fieldnames = ['coords', 'position\_x', 'position\_y', 'color']  
 writer = csv.DictWriter(csvfile, fieldnames=fieldnames)  
 writer.writeheader()  
 triangles = TriangleManager.get\_instance().all()  
 for \_, item in triangles.items():  
 writer.writerow({  
 'coords': ", ".join(map(str, item.get\_coords())),  
 'position\_x': item.get\_center().x,  
 'position\_y': item.get\_center().y,  
 'color': item.get\_fill()  
 })

Описание использованных библиотек:

* tkinter: создание графического интерфейса
* tk.Canvas: создание холста для рисования
* tk.Button, tk.Entry, tk.Scale: создание элементов управления
* math: математические функции
* math.cos, math.sin: тригонометрические функции для поворота
* csv: работа с CSV-файлами
* csv.DictWriter, csv.DictReader: запись и чтение данных в формате CSV

**4. Сообщения**

Приложение выдает следующие сообщения:

При ошибке загрузки файла: "Невозможно открыть файл! Файл поврежден или отсутствует."

При попытке сохранения файла с неверным расширением: "Невозможно сохранить файл! Неверное расширение файла"

При вводе некорректного значения угла поворота: "Некорректное значение угла. Пожалуйста, введите число."