МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

**Лабораторная работа №9**

**По дисциплине «Алгоритмы и структуры данных»**

**Объектно-ориентированная программа с графическим интерфейсом для работы с геометрическими объектами (квадратами)**

**Пояснительная записка**

Р.02069337. №23/711-Вариант 26

Листов – 6

Исполнитель:

студент гр. ИСТбд-23

Фокин Даниил Сергеевич

«\_\_» \_\_\_\_\_\_\_\_ 2024 г.

Ульяновск 2024 г.

**Введение**

Разрабатываемое приложение — объектно-ориентированная программа с графическим интерфейсом для работы с геометрическими объектами (квадратами). Программа реализована на языке Python с использованием библиотеки Tkinter для графического интерфейса.

Цель проекта — продемонстрировать возможности объектно-ориентированного подхода для обработки и визуализации двумерных графических объектов. Программа позволяет пользователю работать с квадратами: сегментировать их на меньшие части, изменять цвет, поворачивать относительно центра и отображать изменения в графическом интерфейсе.

Подход, использованный при реализации, основан на использовании классов для представления геометрических объектов (квадратов) и управления их состоянием. Такой подход позволяет эффективно обрабатывать данные и интегрировать новые функции при необходимости.

**1. Проектная часть**

**1.1. Постановка задачи на разработку приложения**

Постановка задачи определяется заданием на лабораторную работу. Детализируется в разработанном техническом задании.

**1.2. Математические методы**

Для реализации функциональности приложения использовались математические методы, связанные с преобразованием координат точек на плоскости:

* Симметричная сегментация основана на делении квадрата на четыре равные части с расчётом новых координат для каждого подэлемента.
* Поворот объектов реализован с использованием матриц поворота:

где x, y – исходные координаты, cx, cy – координаты центра поворота,

θ – угол поворота в радианах

* Для проверки корректности ввода данных применялись методы верификации: значения размеров, координат и цвета анализировались на предмет соответствия заданным ограничениям.

**1.3. Архитектура и алгоритмы**

1.3.1. Архитектура

Архитектура приложения состоит из следующих компонентов:

* Класс Square представляет квадрат как объект с атрибутами (размер, цвет, координаты) и методами (отрисовка, сегментация, поворот, изменение цвета).
* Класс SquareApp отвечает за графический интерфейс, обработку событий пользователя и управление объектами.
* Графический интерфейс основан на компоненте Canvas библиотеки Tkinter, который используется для визуализации квадратов.

Обмен данными с пользователем организован через файл, который содержит параметры квадратов в формате размер, цвет, координаты x, y.

1.3.2. Алгоритм сегментации

Алгоритм симметричной сегментации заключается в следующем:

1. Получить координаты центра текущего квадрата.
2. Рассчитать размеры новых квадратов как половину исходного размера.
3. Для каждого нового квадрата рассчитать смещённые координаты с учётом отступа.
4. Создать четыре новых квадрата и добавить их в список для отображения.

1.3.3. Алгоритм поворота

Алгоритм поворота квадрата:

1. Получить координаты всех вершин квадрата.
2. Рассчитать новые координаты вершин с использованием формул для вращения вокруг заданного центра.
3. Обновить положение квадратов на холсте с учётом новых координат.

1.3.4. Алгоритм раскраски

Алгоритм изменения цвета квадратов:

1. Для каждого квадрата в списке:

* Если цвет равен red, изменить его на blue.
* Если цвет равен blue, изменить его на green.
* Для всех остальных цветов установить red.

1. Очистить холст.
2. Перерисовать все квадраты с обновлёнными цветами.

1.3.5. Алгоритм визуализации

Алгоритм отрисовки квадратов:

1. Очистить холст.
2. Для каждого квадрата в списке:

* Рассчитать координаты вершин с учётом размера, положения и угла поворота.
* Если задан поворот, пересчитать координаты с использованием матрицы поворота.
* Отрисовать квадрат на холсте с указанным цветом.

1. Обновить холст, чтобы отобразить изменения.

**1.4. Тестирование**

1.4.1. Описание отчета о тестировании

Отчет о тестировании включает описание методов проверки приложения, результаты выполнения сценариев тестирования и рекомендации по улучшению.

1.4.2. Цель тестирования

Целью тестирования является проверка корректности работы приложения, его устойчивости к ошибкам, соответствия заявленным требованиям и стабильности выполнения.

1.4.3. Методика тестирования

Используются следующие методы:

* Функциональное тестирование для проверки работы всех режимов игры.
* Тестирование интерфейса для оценки удобства использования приложения.
* Нагрузочное тестирование для проверки работы приложения при интенсивных действиях пользователей.

1.4.4. Проведенные тесты

1) Загрузка файла с корректными данными (100,red,50,50).

Результат: квадрат отрисован корректно.

2) Загрузка файла с некорректными данными (abc,red,50,50).

Результат: программа выдала сообщение об ошибке.

3) Сегментация квадрата.

Результат: исходный квадрат разделён на четыре меньших, все корректно отрисованы.

4) Изменение цвета квадратов.

Результат: цвета изменились в соответствии с заданной логикой.

5) Поворот квадратов на 45 градусов.

Результат: квадраты визуально повернулись на указанный угол.

6) Тестирование нагрузки более 1000 данных

Результат: программа работает стабильно, без сбоев.

1.4.5. Выводы

Тестирование показало, что приложение работает стабильно при корректных данных, а ошибки ввода надёжно обрабатываются. Визуализация и графический интерфейс полностью соответствуют требованиям. Рекомендуется в будущем добавить возможность сохранения текущего состояния графического холста в файл.

2. Источники, использованные при разработке

<https://python-scripts.com/tkinter> - Документация tkinter