**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Измерительно-вычислительные комплексы»

**Лабораторная работа**

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

**Тема Лабораторная работа №9**

|  |  |
| --- | --- |
| **Подп. и** |  |
| **Инв** |  |
| **Вза** |  |
| **Подп. и** |  |
| **Инв.** |  |

**Пояснительная записка**

**дата**

Р.02069337. <23/710>-<04> ПЗ-<2-зн. номер

редакции> Листов 8

**м. инв. №**

**. № дубл.**

**Исполнитель**: студент гр. ИСТбд-23

**дата**

*Шабаев Р. И*

« 4 » марта 2025 г.

2025

**№ подл.**

**Введение**

Приложение предназначено для визуализации и манипуляции с секторами кругов. Пользователь может загружать данные о секторах из текстового файла, изменять их цвет, проверять пересечения между ними и поворачивать сектора вокруг одной из их границ.

**Проектная часть**

* 1. **Постановка задачи на разработку приложения**

Определяется заданием на курсовую работу. Детализируется в разработанном техническом задании.

* 1. **Математические методы**
* Поворот секторов: При повороте сектора используется тригонометрия для вычисления нового стартового угла. Это позволяет изменять ориентацию сектора в зависимости от движения мыши.
* Поиск пересечений: Метод find\_first\_intersection использует двойной цикл для перебора всех точек на холсте (в заданном диапазоне) и вызывает метод проверки пересечений для каждой точки. Это позволяет находить первую точку пересечения между секторами.
* Функция calculate\_angle: Эта функция использует тригонометрические функции для вычисления угла между центром сектора и курсором мыши. Она применяет арктангенс для нахождения угла, что позволяет определить направление поворота сектора.
  1. **Архитектура и алгоритмы**
     1. Архитектура

Архитектура приложения включает следующие основные структуры данных и функции:

**Классы и методы**

Реализован один класс DraggableArcs, который содержит следующие атрибуты и методы:

**Атрибуты**:

* canvas: холст, на котором рисуются сектора.
* arcs: список, содержащий все сектора.
* offset\_x, offset\_y: смещения для перетаскивания.
* start\_angle, center\_x, center\_y: параметры для поворота сектора.
* current\_arc: текущий выбранный сектор.

**Методы**:

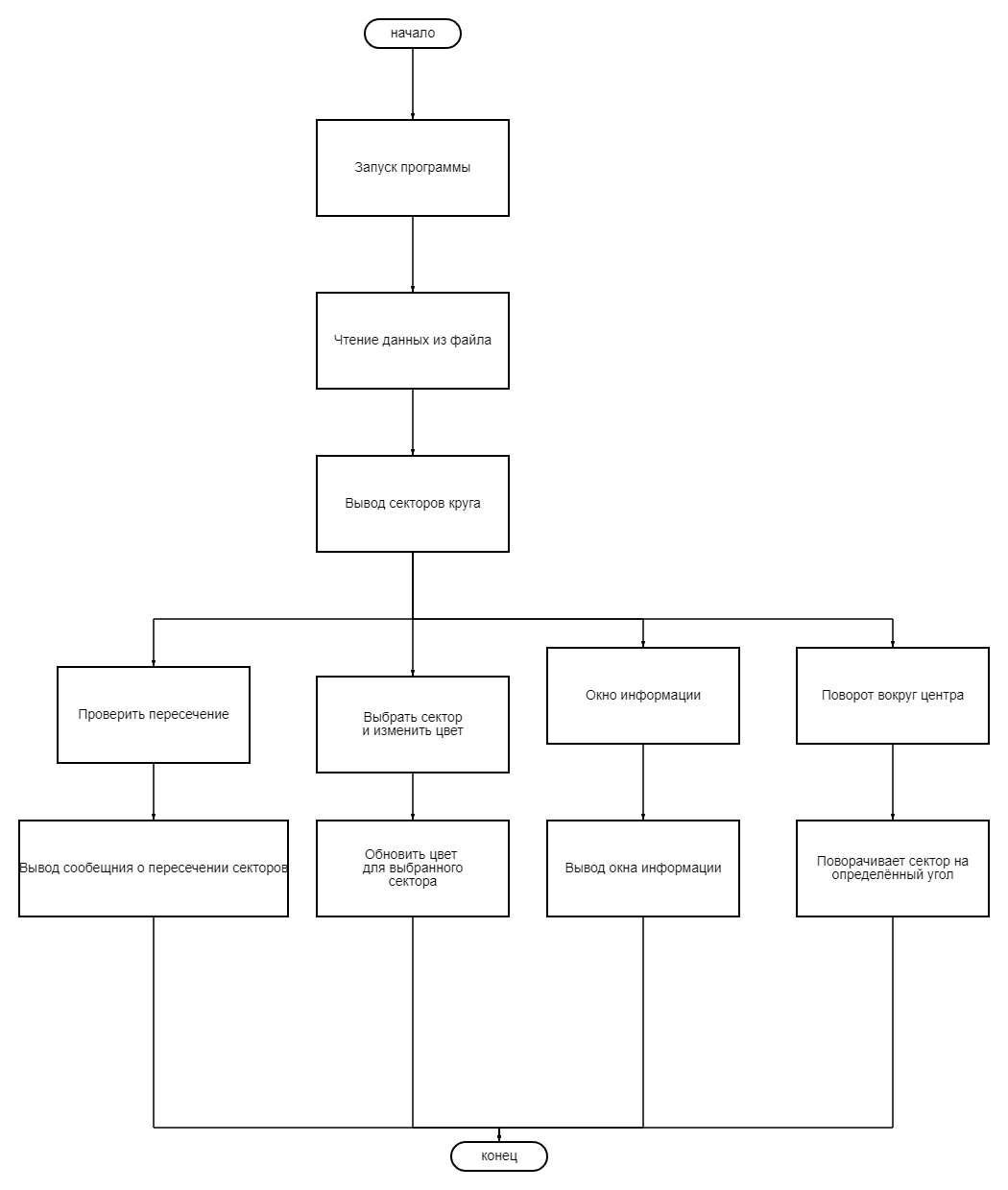
* on\_button\_press: Обрабатывает нажатие левой кнопки мыши для выбора сектора.
* on\_mouse\_drag: Обрабатывает перетаскивание выбранного сектора.
* on\_right\_button\_press: Обрабатывает нажатие правой кнопки мыши для начала поворота.
* on\_right\_mouse\_drag: Обрабатывает поворот сектора.
* open\_file: Загружает данные о секторах из файла.
* show\_information\_about\_program: Отображает информацию о приложении.
* calculate\_angle: Вычисляет угол между центром сектора и курсором мыши.
* find\_first\_intersection: Находит первую точку пересечения между секторами.
* check\_intersection: Проверяет, пересекаются ли сектора.
* show\_intersection: Отображает информацию о пересечении.
* change\_color: Изменяет цвет выбранного сектора.
  + 1. Алгоритм обработки данных

Пример алгоритма, который выполняется при добавлении нового сегмента:

1. Запрос данных о новом сегменте (координаты, цвет, ширина, стартовый и конечный угол).
2. Проверка корректности введенных данных.
3. Если данные корректны:
   * Добавить сегмент в список segments.
   * Обновить Canvas и визуализировать новые сегменты.

Схема алгоритма оформляется в соответствии с ГОСТ 19.701-90, с описанием каждого шага без упоминания языка программирования или имен объектов.

* 1. **Тестирование**



* + 1. Описание отчета о тестировании

Отчет о тестировании предназначен для документирования всех тестовых сценариев и их результатов, а также для выявления проблем и уязвимостей в приложении.

* + 1. Цель тестирования

Проверить работу всех функций приложения, включая загрузку файлов, добавление сегментов и визуализацию данных.

* + 1. Методика тестирования

Методы тестирования включают ручное тестирование интерфейса, автоматизированное тестирование обновлений и стресс-тестирование для проверки устойчивости к нагрузкам.

* + 1. Проведенные тесты

| **Тестовый случай** | **Описание** | **Ожидаемый результат** | **Фактический результат** | **Статус** |
| --- | --- | --- | --- | --- |
| TC1 | Тестирование загрузки файлов | Данные загружены и отображены в таблице. При выборе некорректного файла программа отобразит сообщение об ошибке. | Данные успешно загружены и отображены в таблице. При выборе некорректного файла программа отображает сообщение об ошибке. | Пройден |
| ТС2 | Тестирование добавления сегментов | Сегмент добавлен в таблицу. | Сегмент и данные о нём успешно занесены в таблицу. | Пройден |
| ТС3 | Тестирование смена цвета | Выбрав определённый сектор круга, а также нужный цвет. Сектор меняет свой цвет на новый. | Сектор успешно меняет свой цвет. | Пройден |
| ТС4 | Тестирование сохранения данных | Данные сохраняются корректно и в правильном формате. | Файл успешно сохранён, данные сохранены корректно и в правильном формате. | Пройден |

* + 1. Выводы

Все тестовые случаи были успешно пройдены, что подтверждает корректность работы приложения.

Программа демонстрирует стабильную работу и устойчивость к ошибкам.

Пользовательский интерфейс удобен и интуитивно понятен.

1. **Источники, использованные при разработке**

https://metanit.com/python/tkinter/ Руководство по Tkinter