МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | |  | |  |  |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | |  | | **Пояснительная записка**  на лабораторную работу №8  по дисциплине «Алгоритмы и структуры данных»  **Тема** «Разработка генератора пятиугольников» | | | | | | |
|  | | |  | |  | | | |
|  | | Исполнитель  студент гр. ИСТбд-21  Салова А.С.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |
|  | | 2024 | | | | | | |

**Введение**

Разрабатываемое приложение представляет собой программу **"Генератор пятиугольников"**, созданную с использованием языка программирования Python и библиотеки tkinter. Целью программы является генерация и управление пятиугольниками через графический интерфейс с возможностью изменения их параметров и визуализации действий. Разработка демонстрирует навыки работы с графическим интерфейсом, взаимодействия с графическими элементами и применения объектно-ориентированного подхода.

**Основания для разработки**

Программа создана в рамках лабораторной работы №8 по дисциплине **"Алгоритмы и структуры данных"**.

**Постановка задачи**

Цель разработки — создать программу для генерации и управления пятиугольниками с возможностью изменения их параметров (цвета, местоположения) и отображения результатов на графическом интерфейсе.

**Структура программы**

**Классы**

1. **Pentagon** — класс для представления пятиугольника.
   * **Атрибуты:**
     + coordinates: координаты вершин пятиугольника.
     + color: цвет пятиугольника.
   * **Методы:**
     + generate\_coordinates(): генерация координат пятиугольника.
     + change\_color(): изменение цвета пятиугольника.
2. **PentagonApp** — класс для управления графическим интерфейсом.
   * **Атрибуты:**
     + root: главное окно приложения.
     + canvas: область для отображения пятиугольников.
     + pentagons: список пятиугольников.
   * **Методы:**
     + create\_widgets(): создание элементов интерфейса.
     + generate\_pentagon(): генерация нового пятиугольника.
     + randomize\_colors(): смена цветов у всех пятиугольников.
     + move\_pentagons(): случайное перемещение пятиугольников.

**Функциональное назначение**

1. Генерация пятиугольников с заданными или случайными параметрами.
2. Изменение цветов пятиугольников на случайные.
3. Перемещение пятиугольников на случайное расстояние.
4. Отображение всех изменений на графическом интерфейсе.

**Проектная часть**

**1.1. Математические методы**

* Используются методы геометрических преобразований для генерации координат пятиугольников.
* Алгоритмы для расчета смещения объектов и изменения их положения на экране.

**1.2. Архитектура и алгоритмы**

**1.2.1. Архитектура**

Архитектура программы включает следующие модули:

* **Модуль генерации пятиугольников:** создание пятиугольников с заданными параметрами.
* **Модуль управления объектами:** изменение параметров пятиугольников (цвет, положение).
* **Графический интерфейс:** взаимодействие пользователя с программой через кнопки и область отображения (Canvas).

**1.2.2. Алгоритмы**

1. **Алгоритм генерации координат пятиугольника:**
   * Расчет вершин пятиугольника на основе его радиуса и центра.
2. **Алгоритм смены цветов:**
   * Применение случайных цветов к каждому пятиугольнику.
3. **Алгоритм перемещения объектов:**
   * Расчет новых координат для всех пятиугольников на основе случайного смещения.

**Стадии и этапы разработки**

1. **Определение задач:** Анализ функциональных требований и проектирование структуры программы.
2. **Проектирование архитектуры:** Создание классов и методов для реализации функциональности.
3. **Кодирование интерфейса:** Создание графического интерфейса с помощью tkinter.
4. **Реализация функционала:** Разработка методов для генерации и управления пятиугольниками.
5. **Тестирование и отладка:** Проверка работоспособности всех компонентов программы.
6. **Документирование:** Подготовка пояснительной записки и описания работы программы.

**Тестирование**

**Цель тестирования**

Проверка корректной работы всех функций программы и удобства взаимодействия пользователя с интерфейсом.

**Методика тестирования**

Тестирование проводилось методом **"черного ящика"** для проверки следующих аспектов:

1. Генерация пятиугольников.
2. Смена цветов пятиугольников.
3. Перемещение объектов.
4. Отображение изменений на графическом интерфейсе.
5. Устойчивость программы к некорректным действиям пользователя.

**Результаты тестирования**

1. **Генерация пятиугольников:**
   * Пятиугольники корректно отображаются на Canvas.
2. **Смена цветов:**
   * Цвета всех пятиугольников изменяются случайным образом.
3. **Перемещение объектов:**
   * Все пятиугольники перемещаются на случайные расстояния без наложения друг на друга.
4. **Интерфейс:**
   * Все кнопки выполняют заданные функции, элементы интерфейса корректно отображаются.
5. **Обработка ошибок:**
   * Программа корректно реагирует на отсутствие объектов (например, при попытке сменить цвет, если пятиугольники еще не сгенерированы).

**Заключение**

Разработанная программа **"Генератор пятиугольников"** соответствует поставленным задачам и демонстрирует следующие возможности:

* Генерация пятиугольников с заданными параметрами.
* Изменение их цвета и положения.
* Взаимодействие с графическим интерфейсом на основе библиотеки tkinter.

Программа обладает интуитивно понятным интерфейсом и может быть использована для обучения работе с графическими библиотеками Python.