ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ

Кафедра ВМиК

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

по предмету **«Объектно-ориентированное программирование»**

Выполнил: студент группы МО-203б

Ярошко Е. В.

Проверил:  
доцент каф. ВМиК

Макеев Г. А.

**Уфа 2025 г.**

**Цель лабораторной работы**

Создать GUI-приложение с определённой логикой работы программы. Использовать схему MVC для создания приложения

**Задание**

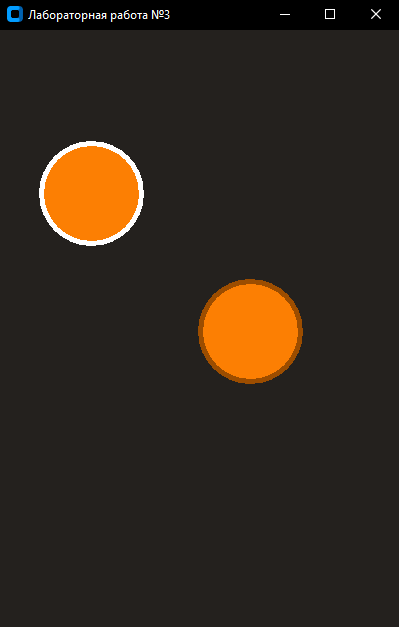
1. Создать GUI-приложение, создающее круги на форме и хранящее их в определённом контейнере. Реализовать функцию выделения кругов и их удаления;
2. Создать GUI-приложение по схеме MVC, имеющее в памяти три числа, работающих по определённым правилам. Реализовать возможность корректировки данных чисел через различные элементы интерфейса.

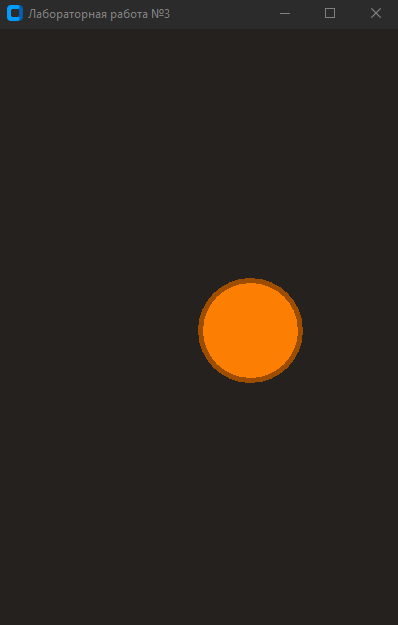
**Ход выполнения лабораторной работы**

Для выполнения данной лабораторной работы использовалась среда разработки, включающая в себя ядро языка Python 3.13.2 с подключённой библиотекой Customtkinter. Редактирование кода выполнялось в программе Visual Studio Code, в качестве компилятора использовался нативный компилятор Python.

При выполнении первого задания был реализован главный класс CCircle, содержащий в себе атрибуты для отрисовки круга на форме, а также координаты положения на форме. В данном классе реализованы методы графической отрисовки самого объекта, определения положения мыши относительно объекта, а также методы выделения и уничтожения. Также был реализован класс Container, который добавляет все объекты класса CCircle в расширяемый список. С помощью класса Container происходит вызов методов при нажатии кнопкой мышки, который, в свою очередь, вызывает методы на проверку и выделение у объектов CCircle. Также в Container реализовано событие Paint, которое перехватывает все вызовы методов у классов и инициирует перерисовку всех фигур в соответствии с изменениями, вызванными данными методами. В главном классе приложения App настраивается окно приложения, а также создаётся главная форма для отображения кругов (Canvas, яв. элементом библиотеки Tkinter, которую включает в себя Customtkinter), после чего происходит привязка кнопок периферийных устройств пользователя к методам объекта Container.

Реализован весь необходимый функционал: создание кругов на форме и в контейнере, выделение кругов (обводка белым цветом), удаление кругов при нажатии Delete (Рисунок 1, 2):

Рисунок 1 — Создание и выделение кругов

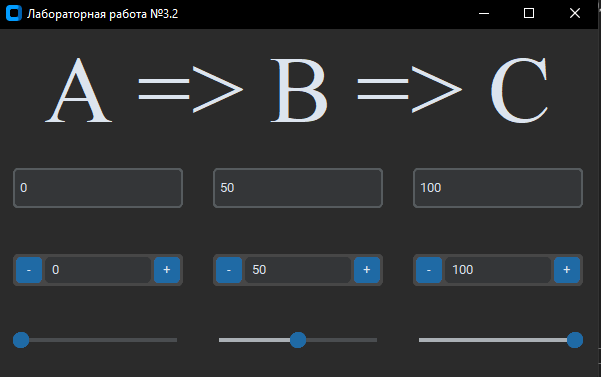
Рисунок 2 — Удаление выделенного круга

При выполнении второго задания лабораторной работы использовалась та же среда разработки, что и в первом задании.

В ходе выполнения был создан класс Model, отвечающий за логику приложения и корректировку трёх значений: A, B и C. Также был создан класс Controller, который отвечает за отрисовку элементов интерфейса и их привязку к методам класса. Т. к. по заданию лабораторной работы необходимо, чтобы данные элементы обновлялись в соответствии с решениями Model, то модель View реализована в классе Controller. В главном классе программы App настраивается главное окно.

Т. к. в библиотеке Customtkinter не реализован элемент Spinbox, то он реализован в отдельном файле spinbox.py, который импортируется в программу.

При компиляции программы высвечивается главное окно приложения (Рисунок 3), позволяющее задавать значения A, B и C. При нарушении правил (A <= B < = C) модель сама корректирует значения и не позволяет пользователю нарушить их.

Рисунок 3 — Главное окно приложения

**Выводы по лабораторной работе**

Создали GUI-приложение с определённой логикой работы программы. Использовали схему MVC для создания приложения.

**Приложение 1**

Исходные коды программ по заданиям

**Задание 1**

Файл lab3.1.py: https://github.com/qvap/OOP--Lab-3.1/blob/main/lab3.1.py

**Задание 2**

Файл lab3.2.py: <https://github.com/qvap/OOP--Lab-3.2/blob/main/lab3.2.py>

Файл spinbox.py: https://github.com/qvap/OOP--Lab-3.2/blob/main/spinbox.py