**СОПРОВОДИТЕЛЬНАЯ ДОКУМЕНТАЦИЯ**

**Лабораторная работа 1**

**Цветовые модели**

**Цель работы:**

Целью данной лабораторной работы является разработка приложения для работы с цветовыми моделями. Основной задачей является создание интуитивно понятного пользовательского интерфейса и реализация функциональности для работы с различными цветовыми пространствами.

**Задачи работы:**

Изучить цветовые модели: RGB, CMYK, HSV, HLS, XYZ, LAB, переход от одной модели к другой, исследовать цветовой график МКО. Создать приложение/веб-приложение, позволяющее пользователю выбирать, а затем интерактивно менять цвет, показывая при этом его составляющие в трех моделях одновременно.

**Использованные средства разработки**:

- Фреймворк Qt и язык C++

**Ход работы:**

В данном коде реализовано оконное приложение на основе фреймворка Qt для работы с цветовыми моделями (RGB, CMYK, XYZ, LAB, HSV, HSL). Приложение предоставляет пользователю возможность визуально настраивать цвета и просматривать их значения в разных цветовых моделях. Главное окно приложения наследуется от класса QMainWindow и содержит различные виджеты для отображения и настройки цветов в разных цветовых моделях. Для управления виджетами и хранения текущего цвета используется контейнер, где ключами являются названия цветовых моделей, а значениями - соответствующие виджеты. В коде присутствуют слоты, которые реагируют на изменения значений ползунков и текстовых полей, обновляя при этом цвет и другие значения в соответствии с выбранной моделью.

MainWindow - конструктор класса MainWindow, инициализирует главное окно и его элементы интерфейса. Устанавливает связи между элементами и методами обработки событий.

on\_lineEdit\_textChanged - слот, вызываемый при изменении текста в элементе lineEdit. Отслеживает ввод пользователем значения цвета в формате "#RRGGBB" и обновляет текущий цвет и интерфейс.

updateLineEditFrom(RGB) - метод, обновляющий элемент lineEdit с учетом значений компонентов цвета, например, RGB.

setValue - метод, устанавливающий значение полосы прокрутки (ScrollBar) и блокирующий сигналы, чтобы избежать зацикливания обновления.

on\_(RGB)ScrollBarChanged - слот, вызываемый при изменении значений полос прокрутки, например, RGB. Обновляет текущий цвет, интерфейс и значения полос прокрутки для других цветовых моделей.

updateColorDisplay - метод, обновляющий визуализацию текущего цвета в элементе label и соответствующие значения RGB.

Update(RGB)ScrollBarFromColor - метод, обновляющий полосы прокрутки, например, RGB на основе текущего цвета и обновляющий значения в соответствующих текстовых полях.

LineEdit(RGB)Changed - слот, вызываемый при изменении текстовых полей RGB. Обновляет значения полос прокрутки, например, RGB и текущий цвет.

**Вывод:**

В ходе выполнения данной работы, я:

- разработала приложение для работы с цветами, обеспечивающее конвертацию между различными цветовыми моделями и визуализацию цветов на экране.

- реализовала методы и логику для корректной конвертации цветов, включая обновление интерфейса при изменении цветовых значений.

- применила ограничения для элементов управления вводом, обеспечивая корректный и надежный ввод данных пользователем.

- получила дополнительный опыт работы с системой контроля версий Git.