**Документация разработчика**

**1. Название программы**

**Color Converter** — приложение для интерактивного изменения цветов и их преобразования в RGB, CMYK и XYZ.

**2. Область применения**

Обучение или демонстрация цветовых моделей и преобразований между ними. Подходит для студентов и преподавателей.

**3. Функциональные возможности**

* Выбор цветов из палитры.
* Ввод параметров цвета с помощью ползунков или полей ввода.
* Автоматический пересчет значений цвета в двух других цветовых моделях.
* Поддержка трех цветовых моделей: RGB, CMYK, XYZ.
* Обработка ошибок при вводе некорректных значений с уведомлением пользователя.

**4. Структура программы**

**Основные файлы**

1. main.cpp: содержит точку входа в программу, настройку стилей и запуск основного окна приложения.
2. mainwindow.h и mainwindow.cpp: реализация главного окна, включая логику интерфейса и взаимодействия с модулями цветовых моделей.
3. module.h и module.cpp: реализация модулей для работы с цветовыми моделями, преобразования значений и обработки ошибок.
4. style.qss: файл для настройки внешнего вида приложения.

**Классы и их функции**

**MainWindow**

* Отвечает за создание интерфейса и управление основными компонентами.
* Методы:
  + on\_pbColor\_clicked(): обработка выбора цвета из палитры.
  + changeColor1(), changeColor2(), changeColor3(): обработка изменений параметров цвета в модулях.
  + setwColor(): обновление отображаемого цвета.

**Module**

* Представляет модуль для работы с отдельной цветовой моделью.
* Методы:
  + setType(type t): установка типа цветовой модели (RGB, CMYK, XYZ).
  + setColor(QColor col): задание текущего цвета и обновление значений.
  + leChanged(), sMoved(): обработка изменений в полях ввода и ползунках.
  + outOfRangeError(int code): обработка ошибок при выходе значений за пределы допустимого диапазона.

**Основные алгоритмы**

1. Преобразование между моделями:
   * RGB ↔ CMYK
   * RGB ↔ HLS
2. Обработка ошибок и корректировка некорректных значений.
3. Синхронизация значений между модулями при изменении параметров в одной из моделей.

**5. Используемые библиотеки**

* Qt Core
* Qt Widgets
* Qt GUI

**6. Рекомендации по доработке**

* Добавить поддержку дополнительных цветовых моделей (например, HSV).
* Реализовать экспорт выбранного цвета в формате HEX или RGB.
* Улучшить интерфейс для более плавного взаимодействия пользователя с приложением.

**7. Сопровождение**

* Код прокомментирован для упрощения поддержки.
* Рекомендуется обновлять Qt до последней версии для совместимости.
* Для улучшений и исправлений использовать систему контроля версий Git.