

Управление производственным процессом разработки программного
обеспечения

Документация
Multi Helper App

Работу выполнили:

Пучков Владислав, группа 22933

Кренев Валентин, группа 22933

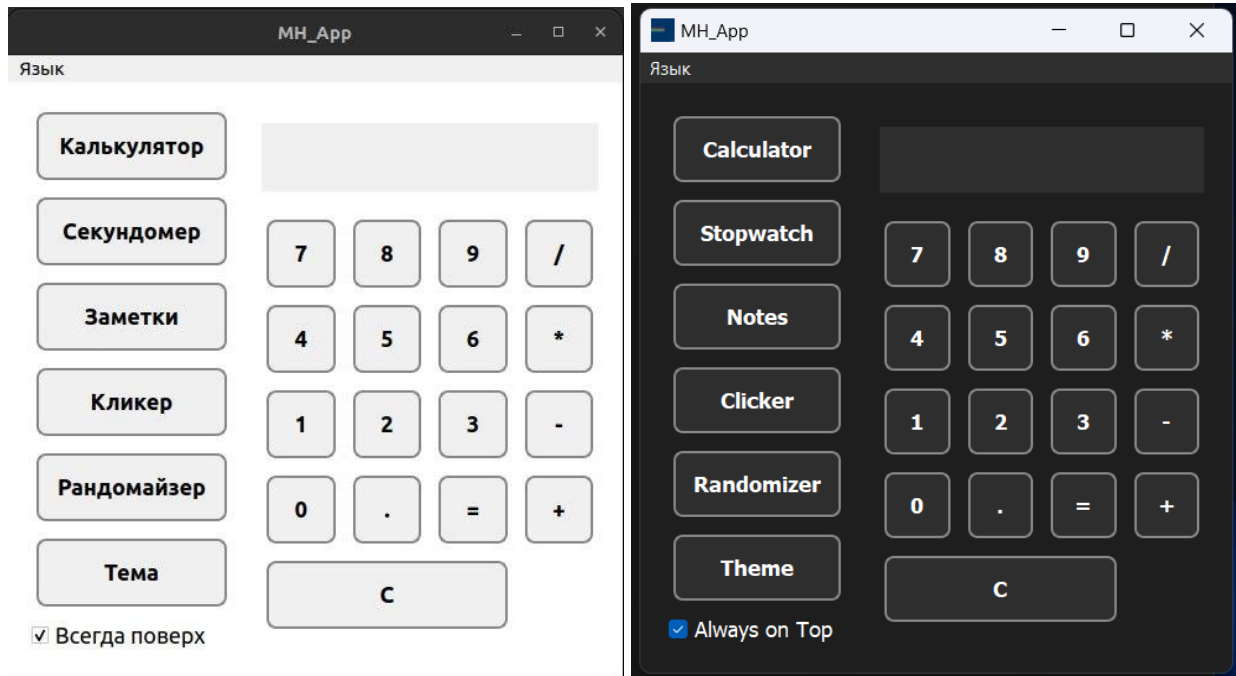
Новосибирский Государственный Университет

2024 год

Введение:

Приложение представляет собой, программу написанную на Python с использованием библиотеки PyQt5

Это приложение служит помощником в мелких компьютерных задачах



Что нужно чтобы запустить?

Windows:

Для обычных пользователей - файл MN_App.exe

Linux (работает и в windows):

Для продвинутых пользователей:

- Python (v3.10 и выше)
- pip install PyQt5
- с командной строки в той же директории:
>python main.py

Как работает код

Этот код - это PyQt5 приложение, которое представляет собой набор мини-приложений в одном окне. Давайте разберем его поэтапно:

1. Импорт библиотек:

- sys, os, random: для стандартных операций с системой, генерации случайных чисел и работы с файловой системой.
- QApplication, QWidget, QPushButton, и другие классы из PyQt5: для создания пользовательского интерфейса.

2. Определение стилей кнопок и окна:

- button_style: содержит стили для кнопок (темные и светлые темы).
- window_style: содержит стили для фона окна (темные и светлые темы).

3. Класс CalculatorWindow:

- Реализует функциональность простого калькулятора.
- Имеет поле для отображения результата (QLineEdit), кнопки с цифрами и операциями.
- Включает методы для обновления стилей и выполнения вычислений.

4. Класс TimerWindow:

- Реализует функциональность секундомера.
- Имеет поле для отображения времени (QLCDNumber) и кнопки управления (старт, стоп, сброс).
- Включает методы для обновления стилей и работы с временем.

5. Класс NotesWindow:

- Реализует функциональность работы с заметками.
- Имеет поле для ввода текста (QTextEdit) и кнопки сохранения и загрузки заметок.
- Включает методы для обновления стилей и сохранения/загрузки заметок.

6. Класс ClickerWindow:

- Реализует функциональность кликера.
- Отображает количество кликов и кнопку для клика.
- Включает методы для обновления стилей и обработки кликов.

7. Класс RandomizerWindow:

- Реализует функциональность генератора случайных чисел и подбрасывания монетки.

- Имеет поля для ввода минимального и максимального значения, кнопку генерации и кнопку подбрасывания монетки.
- Включает методы для обновления стилей и генерации случайных чисел.

8. Класс AppSelectionWidget:

- Содержит кнопки для выбора мини-приложений и кнопку для смены темы.
- Включает методы для обновления стилей и языка интерфейса.

9. Класс AppWindow:

- Главное окно приложения.
- Содержит стек виджет для отображения мини-приложений и меню для выбора языка.
- Включает методы для обновления стилей и языка интерфейса, а также для управления настройками темы и всегда поверх других окон.

10. Основная функция:

- Создает экземпляр QApplication.
- Создает главное окно AppWindow.
- Запускает приложение с главным окном.

В каждом классе есть функции для смены темы и класса.

Каждый класс наследуется от QWidget (QWidget - это базовый класс для всех виджетов в PyQt5. Он представляет собой прямоугольную область на экране, которую можно использовать для отображения пользовательского интерфейса).

Кто что сделал? (Командная работа)

- Владислав – скелет приложения, калькулятор, секундомер, смена, темы, смена языка, заметки.
- Валентин – кликер, рандомайзер, документация, презентация.

Выводы:

В ходе разработки данного проекта было создано многофункциональное приложение на основе библиотеки PyQt5. Приложение включает в себя несколько мини-приложений, таких как калькулятор, секундомер, заметки, кликер и генератор случайных чисел. Каждое из мини-приложений реализует определенную функциональность и обладает интуитивно понятным пользовательским интерфейсом.

Одной из особенностей проекта является возможность переключения между темной и светлой темами оформления, что позволяет пользователю выбрать наиболее комфортный режим отображения. Также пользователь имеет возможность выбора языка интерфейса между английским и русским, что повышает доступность приложения для широкого круга пользователей.

В процессе разработки были использованы принципы объектно-ориентированного программирования, что обеспечило модульность, гибкость и масштабируемость кода. Также были использованы возможности библиотеки PyQt5 для создания графического интерфейса, обработки событий и стилизации виджетов.

В целом, проект успешно реализовал поставленные цели и предоставляет пользователям удобный инструмент для выполнения различных задач, таких как вычисления, ведение заметок, отслеживание времени и генерация случайных чисел. Дальнейшее развитие проекта может включать добавление новых функциональных возможностей, улучшение интерфейса и оптимизацию производительности.