**Департамент образования и науки города Москвы**

**Государственное бюджетное общеобразовательное учреждение**

**Г. Москвы**

**«Школа №1501»**

**«СОЗДАНИЕ ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ С ОПЕРАЦИОННОЙ СИСТЕМОЙ WINDOWS С ПОМОЩЬЮ WINDOWSAPI НА ЯЗЫКЕ С++»**

Автор: Бабич Ростислав Олегович

Класс 11-Б

ГБОУ Школы №1501

Научный руководитель: Муляшов Нил Дмитриевич

Педагог дополнительного образования Центр «На Донской» ГБПОУ «Воробьёвы Горы»

**Г. Москва**

**2022–2023 учебный год**

**Содержание**

**Введение.**

**Ход работы.**

**Вывод.**

**Список источников.**

**Введение**

Истоки идеи для проекта – мой интерес к разработке со взаимодействием с самой системой и компьютером. Мне захотелось создать приложение, с помощью которого можно было бы работать человеку с операционной системой (далее ОС) – более удобное и с новыми идеями по сравнению с обычной панелью управления.

Актуальность:

Данная работа относится к сфере информационных технологий. Рассматривается проблема работы пользователя с операционной системой Windows API (WinAPI). Было бы удобно иметь приложение, которое ускоряло работу с ОС.

В данный момент в этой области практически не проводятся исследования и имеющаяся информация носит сугубо прикладной характер.

Объект:

Приложение для ОС Windows

Предмет:

Работа с интерфейсом и ОС с помощью API Windows.

Цель работы:

С помощью функций WinAPI создать такое приложение, которое ускоряет настройку системы Windows.

Проблема проекта:

Управление своим компьютером может вызвать свои проблемы, поэтому приложение, позволяющие это делать в разы легче, потребовалось бы многим.

Сам план проекта заключался в следующим:

1. Изучить работу Windows API
   1. Работы с интерфейсом
   2. Работа с ОС (операционной системой)
2. Создание самого приложения
   1. Создание приложения
   2. Создание вкладок для работы с ОС

**Ход работы**

**Изучение WinAPI и создание интерфейса**

Есть несколько способов создать приложение для ОС Windows. Одни из самых популярных – QT. Это библиотека для разных языков программирования, которая предоставляет возможность разработать интерфейс для своего приложения. Он заметно упрощает это. Но в отличие от WinAPI, в QT нет функции работы с ОС. У меня был вариант совместить их или использовать только WinAPI. Я выбрал последнее.

Итак, что же такое Windows API? Начнём с того, что такое API вообще. API – это набор функций и параметров, с помощью которых программист может взаимодействовать с определённым продуктом – другой программой, сайтом, в данном случае операционной системой. Соответственно Windows API – это набор функций и параметров для взаимодействия с операционной системой Windows.

В начале были изучены основы работы с Windows API с помощью официальной документации на сайте компании Microsoft(рис.1).

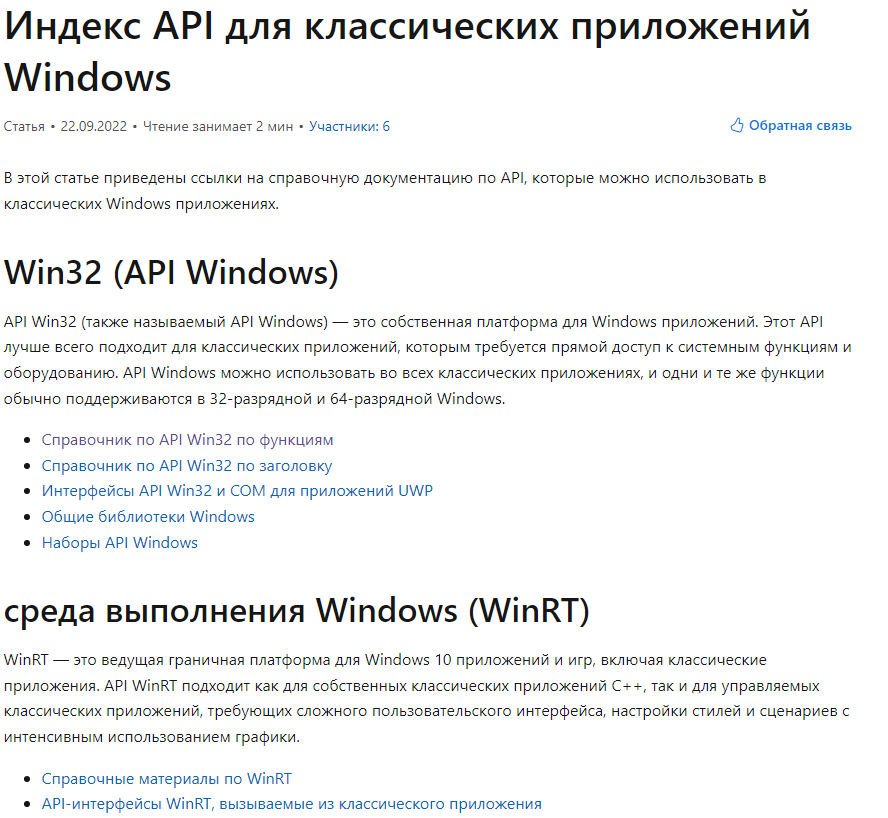


Рис.1

Начнём с основ. Буквально всё, что видит пользователь, есть окна. Сами приложения, текст в них, поля для его ввода – всё это Windows, т. е. окна. Они имеют схожие параметры, например координаты на экране, и различные, например текст имеет переменную, в которой хранится соответственно сам текст, и т. д. То, какие функции имеет то или иное окно, определяется его классом. В API предусмотрены уже сделанные классы. Но для каждого окна можно создать свой собственный класс. При создании прикладного приложения это обязательный этап(рис.2).

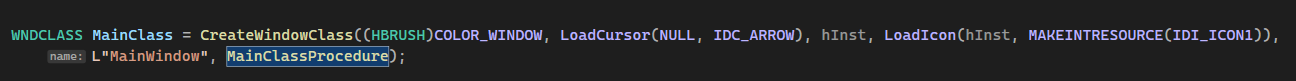


Рис.2 Создание главного класса для главного окна приложения

При запуске программы автоматически вызывается главная функция – WinMain.(рис.3).

Изображение выглядит как текст

Автоматически созданное описание

Рис.3 Основная функция Windows API, которая запускает главное окно приложения.

Здесь нужно рассказать ещё одну главную деталь WinAPI – так называемые сообщения (Messages). Когда пользователь что-либо делает, например нажимает на кнопку мыши, система это выявляет с помощью функции GetMessages (буквально каждое действие пользователя имеет за собой определённый номер) и передаёт в созданную специально программистом функцию, в которой сами сообщения обрабатываются (в зависимости от номера сообщения/действия). В моём случае я назвал данную функцию - MainClassProcedure(Рис.4). Какая именно это будет функция определяет параметр CALLBACK перед названием функции.

Рис. 4

Теперь перейдём к созданию интерфейса. Сам интерфейс представляет из себя совокупность отдельных дочерних окон, их ещё называют виджетами, текст, кнопки, ввод текста, меню (это панель, где находятся “файл”, “справка” и прочее). Каждое окно имеет при себе объект типа HWND (сокращение от header window) – дескриптор, который в себе хранит всю информацию про окно, и, как я раньше уже сказал, номер сообщения при взаимодействии с пользователем. Номера создаёт сам программист с помощью #define «название номера» «сам номер». Для хранения всех hwnd, определений(#define), а также прототипов функций – просто их создание без описания того, чего они должны делать, я сделал отдельный файл.

Переходим к самому главному. Как взаимодействовать с ОС? Для этого используется отдельная функция – SystemParametersInfoW.

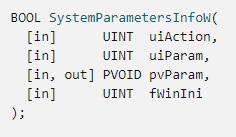


Рис. 5

В неё передаётся 4 аргумента. Из них нам в основном важны только первый, который определяет, что именно мы хотим сделать, и третий. В первый аргумент передаётся код в шестнадцатеричной системе счисления. Все коды уже были созданы Microsoft. Их также можно найти на официальном сайте. В зависимости от кода функция либо берёт данные из третьего аргумента и изменяет что-либо в системе, либо возвращает в переданную в третий аргумент переменную какие-либо данные.

**Создание функция приложения**

Следующим этапом было создание самого приложения с конкретными функциями для настройки ОС. Для начала я взялся за настройку чувствительности мыши.

из приложения(рис.3). Изображение выглядит как текст

Автоматически созданное описание

Рис.

Далее были придуманы пара новых идей, которые будут отличать моё приложение от панели управление.

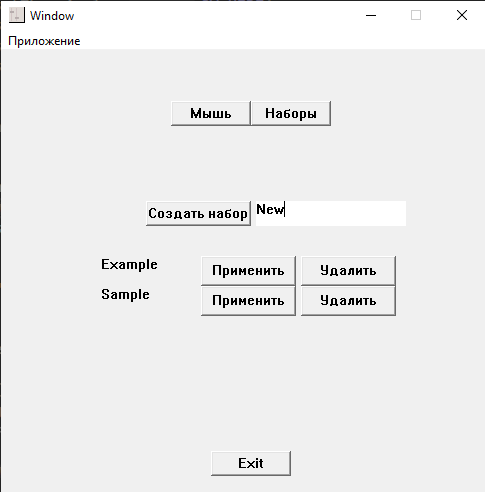
Из реализованных – была создана система наборов, которая позволяет обновлять сразу множество параметров системы одновременно. (рис.4).

Рис.4

Из не созданных – идея с работой с приложениями, запущенными в данный момент в системе.

**Вывод.**

На момент времени создания данного отчёта была создана малая часть функционала приложения.

Одной из идей для проекта была проверка актуальности работы с Windows API в 2022 и далее годах. В данный момент времени для создания приложения под систему Windows достаточно использовать Qt или подобные крупные библиотеки, облегчающие работу. Интерес представляет лишь только работа с самой системой через отдельные функции Windows API, которые были использованы в моём проекте.

**Использован следующий материал:**

* Официальная документация Microsoft (https://www.microsoft.com/ru-ru/)
* Библиотека nlohmann json (https://github.com/nlohmann/json)