МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Костромской государственный университет»

(КГУ)

ИАСТ

(наименование института)

Кафедра автоматизированных систем и технологий

(наименование кафедры)

09.03.02

Направление подготовки/Специальность Информационные системы и технологии

(наименование направления подготовки/специальности)

Дисциплина Архитектура ЭВМ

(наименование дисциплины)

Лабораторная работа №10.

Оконная подсистема.

Выполнили студенты Копосов Лев Владимирович

Копосов Владимир Владимирович

(фамилия, имя, отчество)

Группа 22-ИСбо-1б

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_

Кострома

Вопросы:

1. Что такое класс окна, и какую информацию он хранит?

Класс окна Windows - это тип данных, который представляет собой отдельное окно в пользовательском интерфейсе операционной системы Windows. Он содержит информацию:

1. Имя класса – строка, уникальным образом идентифицирующая класс в рамках процесса.
2. Адрес оконной процедуры – обработчик сообщений по умолчанию для окон этого класса.
3. Дескриптор экземпляра приложения – приложение получает дескриптор экземпляра при запуске, он передаётся как один из параметров в точку входа в приложение.
4. Курсор – форма и вид курсора мыши, когда он находится над окном с этим классом.
5. Иконки класса обычно используются для главных и диалоговых окон приложения.
6. Фоновая кисть – описывает вид фона для указанного окна.
7. Меню класса – имя ресурса, содержащего главное меню для указанного окна.
8. Стили класса – позволяют задать стили, присущие всем окнам этого класса.
9. Дополнительные данные – позволяют выделить небольшой объем памяти, связанный с данным классом.

Класс окна определяет поведение окна и его обработку сообщений. Класс окна описывается структурой данных, которая передается в функцию регистрации класса окна.

1. Какие функции позволяют осуществлять поиск/навигацию по дереву окон? Каково назначение каждой из них?

Функция GetDesktopWindow() позволяет получить дескриптор окна рабочего стола.

Функция EnumWindows() перечисляет окна верхнего уровня на текущем рабочем столе.

Функция EnumChildWindows() перечисляет дочерние окна указанного окна.

Альтернативой является функция EnumDesktopWindows(), которая перечисляет окна верхнего уровня на указанном рабочем столе.

1. Какие оконные сообщения обрабатывает ваша программа? Каково их назначение?

Существуют следующие оконные сообщения:

WM\_NCCREATE: Посылается окну сразу после создания дескриптора.

WM\_CREATE: Посылается окну после того, как дескриптор окна был возвращён приложению.

WM\_PAINT: Посылается окну, если оно нуждается в перерисовке содержимого.

WM\_CLOSE: Посылается окну, когда пользователь пытается его закрыть.

WM\_DESTROY: Посылается оконной функции перед закрытием окна. Дочерние окна ещё существуют.

WM\_NCDESTROY: Посылается перед закрытием дескриптора окна. Дочерние окна уже уничтожены.

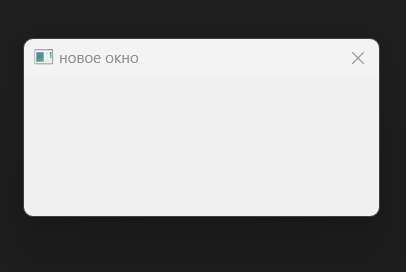
WM\_COMMAND: Посылается при выполнении команды элемента управления, например: кнопка.

Моя программа обрабатывает следующие оконные сообщения: WM\_CLOSE, WM\_DESTROY, WM\_CREATE, WM\_COMMAND, WM\_QUIT.

Вывод в консоль.

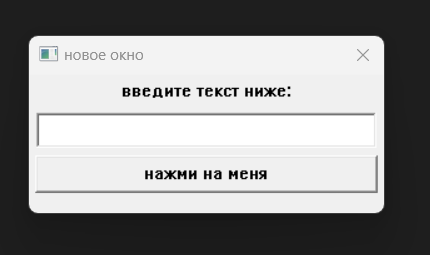
Задание 1.

На базе пустого проекта напишите программу, выводящую простое окно без содержимого. Используйте функцию DefWindowProcA() для реализации поведения по умолчанию. Программа должна завершать работу при закрытии окна.



Задание 2.

Модифицируйте программу так, чтобы она создавала дочерние окна (элементы управления). Примерный внешний вид окна приведён на иллюстрации. По нажатию кнопки программа должна вывести содержимое поля ввода в диалоговом окне.



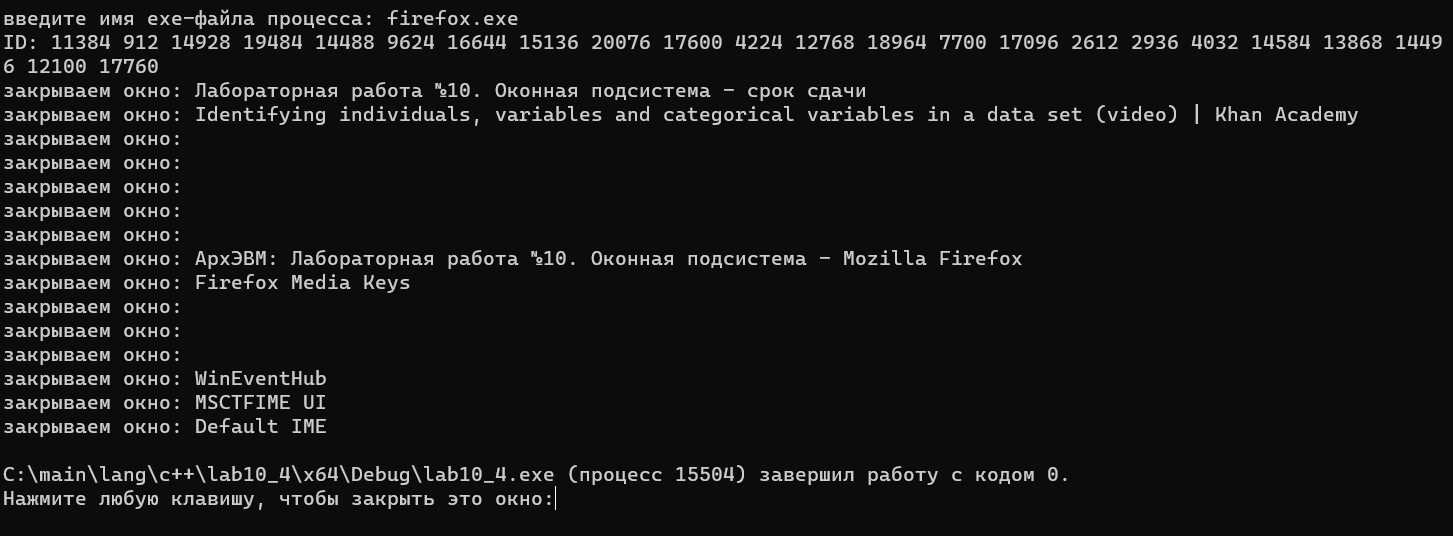
Задание 3.

Модифицируйте программу следующим образом: по щелчку на кнопке, программа должна найти первое окно верхнего уровня, чей текст (заголовок) содержит строку, введённую в поле ввода. Затем она должна послать данному окну сообщение WM\_CLOSE.



Задание 4.

Напишите программу, которая  позволяет ввести со стандартного ввода имя файла. Затем она должна найти все процессы, у которых имя EXE-файла совпадает с введённым, найти их окна, вывести в консоль их заголовки и послать каждому окну сообщение WM\_CLOSE.



Дополнительное задание.

Измените программу пункта 3 следующим образом:

Найдя совпадающее окно, узнайте его положение и размер вызовом GetWindowRect(). Передайте в качестве параметра адрес структуры типа RECT (далее будет называться r).

Создайте прямоугольный регион вызовом CreateRectRgn(). Координаты региона должны быть (0; -30; r.right-r.left, r.bottom-r.top)

Создайте другой прямоугольный регион вызовом CreateRectRgn(). Первую пару координат увеличьте на 100, вторую уменьшите на 100.

Используя функцию CombineRgn(), вычтите (операция RGN\_DIFF) регион п. 3 из региона п. 2, поместив результат в регион из п. 2.

Используя функцию SetWindowRgn(), задайте целевому окну из пункта 1 регион из п. 2. Укажите последним параметром TRUE.

Удалите регион п. 3 вызовом DeleteObject().

