Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №4**

По дисциплине «Проектирование программ в интеллектуальных системах»

Тема: «Типовой каркас оконного mfc-приложения (ТКП). Обработка сообщений. Работа с клиентской областью окна. Стандартные мастера»

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Литвинюк Т. В.

**Проверил:**

Муравьев Г.Л.

Брест 2023

**Цель работы:** 1. Изучение типового каркаса MFC-приложения (ТКП). 2. Изучение организации обработки сообщений. 3. Организация вывода в клиентскую область главного окна, поддержка перерисовки. 4. Изучение элементов автоматизации разработки mfc-приложений.

**Ход работы:**

1. Создать “пустое” приложение (ТКП) в соответствии с § 1.11 и § 2.5. Это каркас приложения без обработки сообщений. Запустить. Изучить его свойства.

При запуске создается пустое окно с базовыми кнопками управления: Закрытие, Свернуть/Развернуть, Спрятать/Показать.

3. Модифицировать приложение, включив чувствительность к сообщению WM\_LBUTTONDOWN (см. §§ 1.6-1.7). Для этого в классе CMainWin включить (разкомментировать) прототип функции-обработчика

Для включения чувствительности необходимо разкомментировать следующие функции:

afx\_msg void OnLButtonDown (UINT flags, CPoint loc); // В классе WINDOW: public CFrameWnd

afx\_msg void WINDOW :: OnLButtonDown (UINT flags, CPoint loc) { CClientDC dc (this); };

ON\_WM\_LBUTTONDOWN() // В очереди сообщений главного окна BEGIN\_MESSAGE\_MAP(WINDOW, CFrameWnd).

4. Создать приложение (на базе ТКП), аналогичное созданному в предыдущем пункте, но включив чувствительность к сообщению WM\_PAINT (см. §§ 1.6-1.7). Для этого в класс CMainWin включить прототип функции-обработчика .

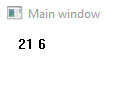
Для включения чувствительности необходимо разкомментировать функции аналогично для WM\_LBUTTONDOWN.

5. Создать приложение для вывода координат курсора “мыши”, фиксируемых по щелчку ее левой клавиши. Для этого: - создать ТКП и включить чувствительность к сообщению WM\_LBUTTONDOWN; - обеспечить вывод координат курсора “мыши”, фиксируемых по щелчку ее левой клавиши, в обработчике сообщения WM\_LBUTTONDOWN (см. § 1.8); - запустить приложение, выполнить манипуляции с окном и анализировать эффект перерисовки.

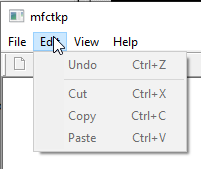
Для фиксации координат мыши используется параметр функции CPoint loc: loc.x и loc.y – координаты по x и по y соответственно. Для вывода координат можно использовать функцию TextBox().

6. Модифицировать приложение, решив проблему перерисовки - ввод координат производить по сообщению WM\_LBUTTONDOWN, вывод по WM\_PAINT.

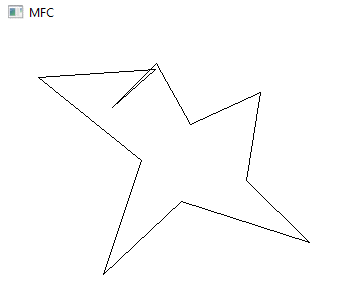
Для принудительной перерисовки можно использовать функцию InvalidateRect(Null,1);

7. Модифицировать приложение для вывода координат в той точке экрана, где был зафиксирован щелчок левой клавиши “мыши”.

Для вывода по тем же координатам можно использовать глобальную переменную типа CPoint и сохранять в ней координаты.

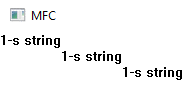
10. Создание каркаса MFC-приложения (ТКП) с помощью мастера. Для этого запустите мастер MFC (например, AppWizard (exe)). В поле Project name введите название проекта приложения и нажмите кнопку ОК. В окне мастера выберите - однодокументное приложение (Single Document Interface - SDI, флажок - Single document) и откажитесь от поддержки архитектуры Документ-Вид (сбросьте флажок – Document/View architecture support). Запустите приложение. Изучите его состав и функциональные возможности.

MFC-каркас приложения включает в себя базовые кнопки управления окном, меню: File->Exit; Edit->Undo, Cut, Copy, Paste; View->Toolbar, Status Bar, About «имя проекта». Присутствуют кнопки: Создание нового файла, вырезать, скопировать, вставить, информация о приложении. В низу окна показывается разная статистика: номер столбца, номер строки и т.п.

11. Создать приложение на базе ТКП для рисования плоских изображений отрезками прямых линий. Координаты точек вводить нажатием левой клавиши “мыши”. Введенные координаты сохранять в динамическом массиве (например, использовать vector). Выводить изображение, путем соединения прямыми линиями соседних точек, по нажатию правой клавиши “мыши”.

Для работы с рисованием на клиентской области окна можно использовать следующий объект : CDC \*cdc= GetDC(); - GetDC(); Для очистки экрана можно использовать следующие команды: CRect cr; GetClientRect(&cr); cdc->FillSolidRect(cr,RGB(255,255,255)) , т.е. мы делаем заливку окна белым цветом. Для хранения координат мыши можно использовать Очередь(queue). Функция cdc->LineTo() позволяет нарисовать линию от текущей координаты кисти в любую другую координату, а cdc->MoveTo() отвечает за передвижение кисти в заданную координату. Функция ReleaseDC() позволяет подтвердить все действия с кистью.

13. Создать приложение на базе ТКП, которое выводит строки, например, как показано на рисунке.

 TEXTMETRIC TextAtttrib; dc.GetTextMetrics(&TextAtttrib); // позволяет получить характеристики шрифта.

dc.GetTextExtent(CString(strs[i].c\_str())).cx;// позволяет узнать сколько по ширине в пикселях введенная строка в параметре функции.

TextAtttrib.tmHeight // узнать высоту шрифта

**Вывод:** изучил типовой каркас MFC-приложения (ТКП), изучил организацию обработки сообщений, организацию вывода в клиентскую область главного окна, поддержка перерисовки, изучил элементы автоматизации разработки mfc-приложений.