***Министерство образования Республики Беларусь***

***Учреждение образования***

***«Брестский государственный технический университет»***

***Кафедра ИИТ***

**Лабораторная работа №5**

**По дисциплине ППвИС за IV семестр**

**Тема: «MFC. ТКП. Сообщения ввода. Работа с клиентской областью»**

**Выполнил:**

Студент группы ИИ-15 (1)

2-го курса

Волк И. А.

**Проверил:**

Муравьев Г.Л.

Брест 2019

Цель:

* Изучить типовой каркас MFC-приложения
* Изучить организации обработки сообщений
* Организация вывода в клиентскую область главного окна, поддержка перерисовки
* Изучение элементов автоматизации разработки mfc-приложений

Состав отчета:

* Описание ТКП
* Листинги программ
* Изображения результатов

Ход работы:

1. Создадим «пустое» приложение на базе MFC.

#include <afxwin.h>

class CMainWindow : public CFrameWnd

{

public:

CMainWindow();

DECLARE\_MESSAGE\_MAP()

};

CMainWindow::CMainWindow()

{

Create(NULL, \_T("Hello world"));

}

class CMyApp : public CWinApp

{

public:

BOOL InitInstance();

};

BOOL CMyApp::InitInstance()

{

m\_pMainWnd = new CMainWindow();

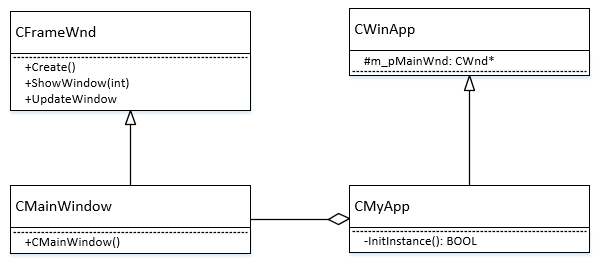
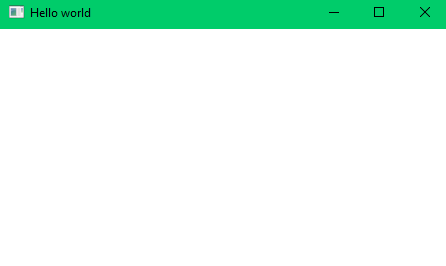
m\_pMainWnd->ShowWindow(m\_nCmdShow);

m\_pMainWnd->UpdateWindow();

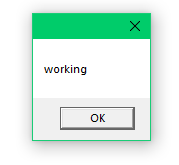
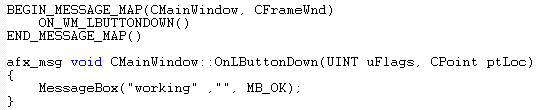
return TRUE;

}

CMyApp app;



1. Добавим чувствительность к WM\_LBUTTONDOWN



Сделаем вывод сообщения в главное окно:

afx\_msg void CMainWindow::OnLButtonDown(UINT uFlags, CPoint ptLoc)

{

CClientDC dc(this);

char szMsg[100] = "Working";

dc.TextOut(10, 10, szMsg, 100);

}



1. Добавим чувствительность к сообщению WM\_PAINT

BEGIN\_MESSAGE\_MAP(CMainWindow, CFrameWnd)

ON\_WM\_PAINT()

END\_MESSAGE\_MAP()

afx\_msg void CMainWindow::OnPaint()

{

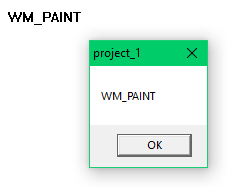
CPaintDC dc(this);

char szMsg[100] = "WM\_PAINT";

dc.TextOut(10, 10, szMsg, 100);

MessageBox("WM\_PAINT");

}



1. Создадим приложение для выведения координат по нажатию клавиши мыши.

afx\_msg void CMainWindow::OnLButtonDown(UINT uFlags, CPoint pt)

{

ptLast = pt;

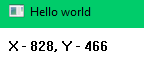
CClientDC dc(this);

char szMsg[100] = "";

wsprintf(szMsg, "X - %d, Y - %d", pt.x, pt.y);

dc.TextOut(10, 10, szMsg, 100);

}



1. Перенесем фрагмент выведения координат в обработчик WM\_PAINT:

class CMainWindow : public CFrameWnd

{

CPoint ptLast;

public:

CMainWindow();

afx\_msg void OnLButtonDown(UINT uFlags, CPoint pt);

afx\_msg void OnPaint();

DECLARE\_MESSAGE\_MAP()

};

BEGIN\_MESSAGE\_MAP(CMainWindow, CFrameWnd)

ON\_WM\_LBUTTONDOWN()

ON\_WM\_PAINT()

END\_MESSAGE\_MAP()

afx\_msg void CMainWindow::OnLButtonDown(UINT uFlags, CPoint pt)

{

ptLast = pt;

Invalidate();

}

afx\_msg void CMainWindow::OnPaint()

{

CPaintDC dc(this);

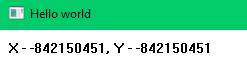
char szMsg[100] = "";

wsprintf(szMsg, "X - %d, Y - %d", ptLast.x, ptLast.y);

dc.TextOut(10, 10, szMsg, 100);

}

С самого начала:



После нажатия левой клавиши:

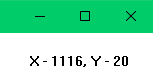


1. Сделаем, чтобы вывод производился на месте, где был произведен клик.

Тогда для вывода:

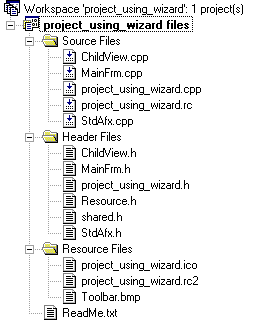
dc.TextOut(ptLast.x, ptLast.y, szMsg, 100);

Пример выполнения:

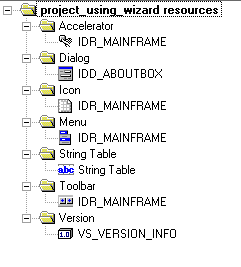


* 1. Создали каркас используя мастер

Содержимое проекта:



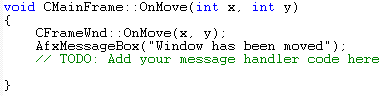
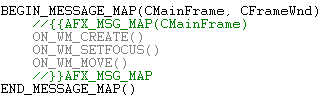
Содержимое ресурсов:



Содержимое классов:



* 1. Добавим обработчик OnMove в CMainFrame используя мастер:



1. Создадим приложение для рисования

class CMainWindow : public CFrameWnd

{

std::vector<CPoint> arrPoints;

public:

CMainWindow();

// Painting

afx\_msg void OnLButtonDown(UINT uFlags, CPoint pt);

// Erasing

afx\_msg void OnRButtonDown(UINT uFlags, CPoint pt);

afx\_msg void OnPaint();

DECLARE\_MESSAGE\_MAP()

};

BEGIN\_MESSAGE\_MAP(CMainWindow, CFrameWnd)

ON\_WM\_LBUTTONDOWN()

ON\_WM\_RBUTTONDOWN()

ON\_WM\_PAINT()

END\_MESSAGE\_MAP()

afx\_msg void CMainWindow::OnLButtonDown(UINT uFlags, CPoint pt)

{

arrPoints.push\_back(pt);

Invalidate();

}

afx\_msg void CMainWindow::OnRButtonDown(UINT uFlags, CPoint pt)

{

arrPoints.clear();

Invalidate();

}

afx\_msg void CMainWindow::OnPaint()

{

CPaintDC dc(this);

if (arrPoints.size() < 2) return;

for(int i = 0; i < arrPoints.size() - 1; i++)

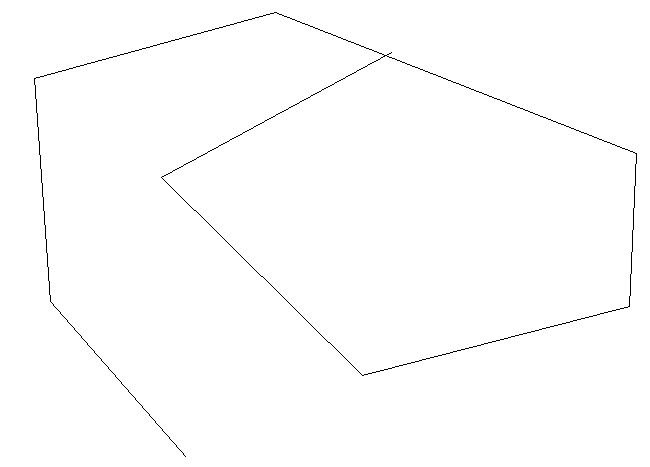
{

dc.MoveTo(arrPoints[i].x, arrPoints[i].y);

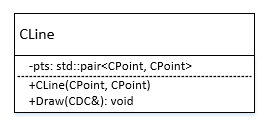
dc.LineTo(arrPoints[i+1].x, arrPoints[i+1].y);

}

}



1. Разработаем класс для хранения линий



class CLine

{

std::pair<CPoint, CPoint> pts;

public:

CLine(CPoint ptFirst, CPoint ptSecond)

{

pts.first = ptFirst;

pts.second = ptSecond;

}

void Draw(CDC &dc)

{

dc.MoveTo(pts.first.x, pts.first.y);

dc.LineTo(pts.second.x, pts.second.y);

}

};

class CMainWindow : public CFrameWnd

{

std::vector<CLine> vecLines;

CPoint ptLast;

public:

CMainWindow();

// Painting

afx\_msg void OnLButtonDown(UINT uFlags, CPoint pt);

// Erasing

afx\_msg void OnRButtonDown(UINT uFlags, CPoint pt);

afx\_msg void OnPaint();

DECLARE\_MESSAGE\_MAP()

};

BEGIN\_MESSAGE\_MAP(CMainWindow, CFrameWnd)

ON\_WM\_LBUTTONDOWN()

ON\_WM\_RBUTTONDOWN()

ON\_WM\_PAINT()

END\_MESSAGE\_MAP()

afx\_msg void CMainWindow::OnLButtonDown(UINT uFlags, CPoint pt)

{

if(ptLast != CPoint())

{

CLine line(ptLast, pt);

vecLines.push\_back(line);

}

ptLast = pt;

Invalidate();

}

afx\_msg void CMainWindow::OnRButtonDown(UINT uFlags, CPoint pt)

{

vecLines.clear();

ptLast = CPoint;

Invalidate();

}

afx\_msg void CMainWindow::OnPaint()

{

CPaintDC dc(this);

for(int i = 0; i < vecLines.size(); i++)

{

vecLines[i].Draw(dc);

}

}

Вывод: по ходу данной лабораторной работы научились работать с типовым каркасом MFC, а также обрабатывать сообщения и работать с мастером.