Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №1**

По дисциплине «Проектирование программ в интеллектуальных системах»

Тема: «Типовой каркас оконного windows-приложения (ТКП). Обработка сообщений. Организация вывода в клиентскую область окна»

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Литвинюк Т. В.

**Проверил:**

Муравьев Г.Л.

Брест 2023

**ЧАСТЬ 1**

**Цель работы:** 1. Ознакомиться со структурой и особенностями программирования оконных windows-приложений. 2. Изучить каркасы windows-приложений. 3. Изучить структуру типового каркаса оконных приложений (ТКП), использование ТКП для создания пользовательских приложений.

**Ход работы:**

1. Cоздать приложение, модифицировав приложение на базе ТКП. Изменить его интерфейс (вид окна, состав элементов окна - кнопок и т.п.). Для этого: - создать приложение на базе ТКП - выполнить отмену опции окна - WS\_OVERLAPPEDWINDOW и обеспечить тот же состав окна с применением опции окна - WS\_OVERLAPPED путем поэлементного включения соответствующих элементов (кнопок и т.п.) через оператор “или” (см. приложение 5 [1]); hWnd = CreateWindow (szWindowStyle, "Karkas ", WS\_OVERLAPPED | WS\_CAPTION | WS\_SYSMENU | WS\_THICKFRAME | WS\_MINIMIZEBOX | WS\_MAXIMIZEBOX,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

(HWND) NULL,

(HMENU) NULL,

hInst NULL );

- выполнить отмену опции окна ТКП - WS\_OVERLAPPED и обеспечить тот же состав окна, но путем поэлементного включения соответствующих элементов (кнопок и т.п.).

hWnd = CreateWindow (szWindowStyle, "Karkas ", WS\_TILED | WS\_CAPTION | WS\_SYSMENU | WS\_THICKFRAME | WS\_MINIMIZEBOX | WS\_MAXIMIZEBOX,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

(HWND) NULL,

(HMENU) NULL,

hInst NULL );

Внести изменения в параметры: - WNDCLASS.

WNDCLASS wcApp;

wcApp.lpszClassName = szWindowStyle;

wcApp.hInstance = hInst;

wcApp.lpfnWndProc = WndProc;

wcApp.hCursor = LoadCursor(NULL, IDC\_CROSS);

wcApp.hIcon = 0;

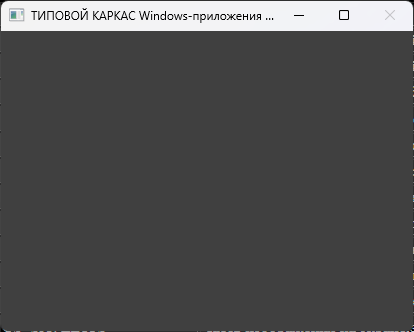
wcApp.lpszMenuName = 0;

wcApp.hbrBackground = (HBRUSH) GetStockObject (DKGRAY\_BRUSH);

wcApp.style = CS\_BYTEALIGNCLIENT|CS\_NOCLOSE |CS\_CLASSDC;

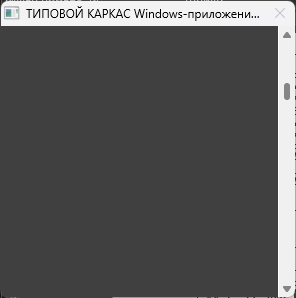
wcApp.cbClsExtra = 0;

wcApp.cbWndExtra = 0;



- функции CreateWindow .Например, добавить в окно рамки, полосы прокрутки, убрать или добавить командные кнопки, внести изменения в координаты и размеры окна, использовав соответствующие параметры функции (x, y, Width, Height);

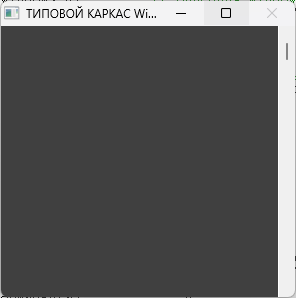
hWnd = CreateWindow (szWindowStyle, "kARKAS ", WS\_BORDER | WS\_CAPTION | WS\_SYSMENU | WS\_VSCROLL,100,200,300,300,(HWND) NULL, (HMENU) NULL,hInst, NULL );



-функции ShowWindow .

ShowWindow (hWnd,SW\_SHOWMINIMIZED);

1. Обеспечить попеременное переключение окна – “спрятано”, “показано”, “свернуто”, “развернуто”.



hWnd = CreateWindow (szWindowStyle, "kARKAS", WS\_BORDER | WS\_CAPTION | WS\_SYSMENU | WS\_VSCROLL|WS\_MAXIMIZEBOX|WS\_MINIMIZEBOX,100,200,300,

300, (HWND) NULL, (HMENU) NULL,hInst,NULL);

**КОНТРОЛЬНЫЕ ЗАДАНИЯ**

1. Создать приложение с ТКП “вручную”, не используя мастер, поменяв размеры окна (200 на 200), заголовок окна, добавив полосы прокрутки.

Решение присутствует в 8 пункте.

1. Добавить к приложению обработчик сообщения WM\_QUIT – осуществить вывод окна сообщения MessageBox о пришедшем сообщении.

case WM\_QUIT:

MessageBox(hWnd,"WM\_QUIT","WM\_QUIT",MB\_OK);

break;

Сообщение **WM\_QUIT** не связано с окном и поэтому никогда не будет получено через процедуру окна. Он извлекается только функциями **[GetMessage](https://learn.microsoft.com/ru-ru/windows/win32/api/winuser/nf-winuser-getmessage)** или **[PeekMessage](https://learn.microsoft.com/ru-ru/windows/win32/api/winuser/nf-winuser-peekmessagea)** .

1. Обеспечить попеременное переключение окна ТКП (“спрятано”-“показано”), использовав для этого обработчик сообщения WM\_LBUTTONDOWN - нажатие левой клавиши “мыши” или нажатие клавиши на клавиатуре.

case WM\_LBUTTONDOWN:

case WM\_KEYDOWN:

ShowWindow(hWnd,SW\_HIDE);

Sleep(1000);

ShowWindow(hWnd,SW\_SHOWNORMAL);

break;

**Часть 2:**

**Цель работы:** 1. Ознакомиться с особенностями организации вывода в оконных windows-приложениях. 2. Ознакомиться с особенностями управления сообщениями.

**Ход работы:**

1. Cоздать приложение на базе ТКП. Добавить строку приветствия (координаты вывода - 0, 0), для чего вставить в обработчик WM\_PAINT пользовательский фрагмент

case WM\_PAINT:

hdc = BeginPaint (hWnd, &ps);

TextOut(hdc,0,0,"Hello PPOIS",11);

ValidateRect (hWnd,NULL);

EndPaint (hWnd, &ps);

Break;

Вывести в столбик для значений Х от 1 до 10 (с шагом 0,5) пары значений: Х - квадрат Х. Запустить приложение. Выполнить свертывание-развертывание окна, перемещение, изменение размеров окна. Убедиться в автоматической перерисовке окна.

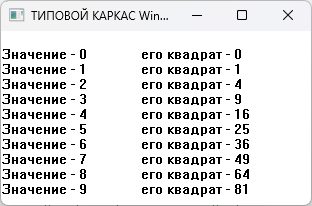
for (i=0; i<10; i++){

sprintf(MyString,"Значение - %d ", i);

TextOut ( hdc, 0, 0 + 15 \* (i + 1), MyString,strlen(MyString));

sprintf(MyString,"его квадрат - %d", i \* i);

TextOut ( hdc, 140, 0 + 15 \* (i + 1), MyString,strlen(MyString)); }



Функция TextOut позволяет выводить текст на экран.

2. Модифицировать предыдущее приложение. Для этого: - изменить формат вывода и увеличить число обрабатываемых чисел до 75; - добавить вывод номера перерисовки (переменная int ReDrawNumber = 0).

++ReDrawNumber;

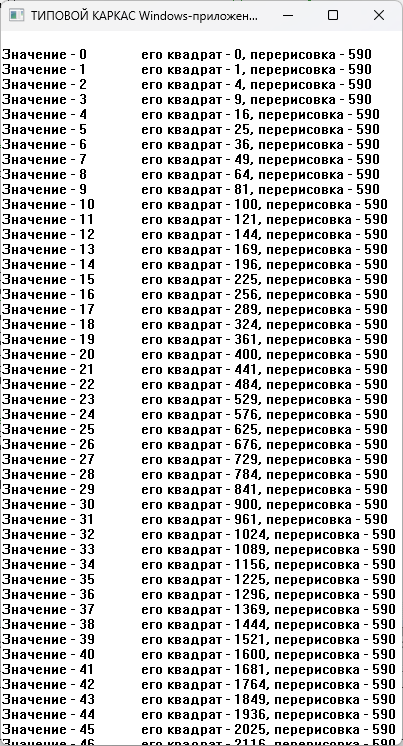
for (i=0; i<75; i++){

sprintf(MyString,"Значение- %d ", i);

TextOut ( hdc, 0, 0 + 15 \* (i + 1), MyString,strlen(MyString));

sprintf(MyString,"его квадрат - %d, перерисовка - %d", i \* i,ReDrawNumber);

TextOut ( hdc, 140, 0 + 15 \* (i + 1), MyString,strlen(MyString));}



3. Cоздать приложение на базе ТКП. Организовать вывод графических данных. Основные графические примитивы рисуются с использованием библиотечных функций: - дуги эллипса Arc, ArcTo; - эллипсы и окружности Ellipse; - линии от текущей точки до точки, указанной в функции LineTo(hdc, x, y); - прямоугольники Rectangle; - полигоны Polygon; - связанные отрезки прямых PolyLine.

hdc = BeginPaint( hWnd, &ps );

hPen = CreatePen(PS\_SOLID,1,RGB (0, 255, 0) );

SelectObject (hdc, hPen);

Ellipse ( hdc, 125, 0, 335, 400 );

MoveToEx(hdc, 50, 50, NULL);

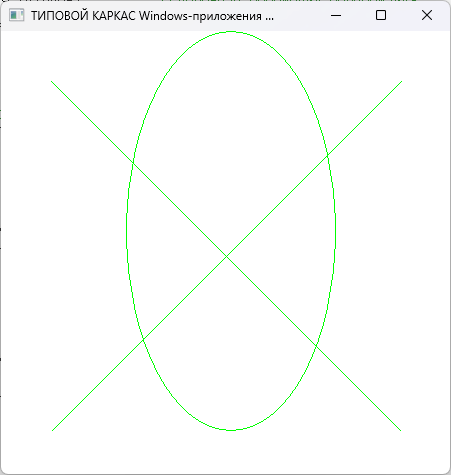
LineTo(hdc, 400,400);

MoveToEx(hdc, 400, 50, NULL);

LineTo(hdc, 50,400);

ValidateRect(hWnd,NULL);

EndPaint(hWnd, &ps);



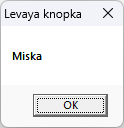
Научился выводить графические примитивы, такие как эллипс, линия, прямоугольник, полигон.

4. Cоздать приложение на базе ТКП. Включить чувствительность к нажатию левой клавиши “мыши” (сообщение WM\_LBUTTONDOWN) - выводить окно MessageBox. Для этого создать в функции-обработчике дополнительную секцию

case WM\_LBUTTONDOWN:

MessageBox(NULL,"Miska","Levaya knopka",MB\_OK);

break;



Повторить п.п. 3-4, выполняя все действия в секции case WM\_LBUTTONDOWN (секция case WM\_PAINT должна быть пустой). Запустить приложение. Выполнить свертывание-развертывание окна, перемещение, перекрытие, изменение размеров окна. Проанализировать характер перерисовки содержимого окна.

case WM\_LBUTTONDOWN:

hdc = BeginPaint( hWnd, &ps );

TextOut(hdc,0,0,"Hello PPOIS",11);

ValidateRect(hWnd,NULL);

EndPaint(hWnd, &ps);

break;

Cвертывание-развертывание окна, перемещение, перекрытие, изменение размеров окна приводит к исчезновению надписи.

5. Создать приложение, в котором подсчитываются события - нажатие левой клавиши мыши. А результат - число нажатий - выводится в секции WM\_PAINT только по нажатии правой клавиши мыши.

case WM\_PAINT:

if(GetKeyState(VK\_RBUTTON)&0x80){

hdc = BeginPaint( hWnd, &ps );

sprintf(MyString,"Kol-vo LBUTTON - %d",ReDrawNumber);

TextOut(hdc,100,100,MyString,256);

ValidateRect(hWnd,NULL);

EndPaint(hWnd, &ps);}

break;

case WM\_LBUTTONDOWN:

ReDrawNumber++;

break;

1. Модифицировать приложение. Результат выводить в секции WM\_PAINT сразу при его обновлении. Для этого инициировать перерисовку окна функцией InvalidateRect ( hWnd, NULL, TRUE ).

case WM\_PAINT:

hdc = BeginPaint( hWnd, &ps );

sprintf(MyString,"Kol-vo LBUTTON - %d",ReDrawNumber);

TextOut(hdc,100,100,MyString,256);

ValidateRect(hWnd,NULL);

EndPaint(hWnd, &ps);

break;

case WM\_LBUTTONDOWN:

ReDrawNumber++;

InvalidateRect(hWnd,NULL,true);

break;

Функция InvalidateRect запускает перерисовку окна, в данном случае при нажатии ЛКМ произойдет обновление экрана.

7. Создать приложение для вывода текста “Работает ТКП”, начиная с позиции X, Y. За начальное значение координат взять 0, 0. По сообщению WM\_LBUTTONDOWN увеличивать координаты точки вывода на 50 единиц и инициировать перерисовку. По сообщению WM\_RBUTTONDOWN уменьшать координаты точки вывода на 50 единиц и инициировать перерисовку.

short x=0,y=0;

---

case WM\_LBUTTONDOWN:

x+=50;y+=50;

InvalidateRect(hWnd,NULL,true);

break;

case WM\_RBUTTONDOWN:

x-=50;y-=50;

InvalidateRect(hWnd,NULL,true);

break;

case WM\_PAINT:

hdc = BeginPaint( hWnd, &ps );

sprintf(MyString,"WorkTKP");

TextOut(hdc,x,y,MyString,256);

ValidateRect(hWnd,NULL);

EndPaint(hWnd, &ps);

break;

1. Создать приложение для вывода координат указателя “мыши” в виде Х = …, У = ….. За начальное значение координат взять 0, 0. По сообщению WM\_LBUTTONDOWN считывать текущие координаты курсора мыши. По сообщению WM\_RBUTTONDOWN инициировать вывод координат. Обеспечивать перерисовку окна. Блокировать обновление вывода по другим причинам перерисовки.

WORD x,y;

----

case WM\_LBUTTONDOWN:

x=LOWORD(lParam);

y=HIWORD(lParam);

break;

case WM\_RBUTTONDOWN:

ReDraw=true;

InvalidateRect(hWnd,NULL,true);

break;

case WM\_PAINT:

hdc = BeginPaint( hWnd, &ps );

if(ReDraw){

sprintf(MyString,"x=%d,y=%d",x,y);}

TextOut(hdc,0,0,MyString,256);

ValidateRect(hWnd,NULL);

EndPaint(hWnd, &ps);

ReDraw=false;

break;

Функции LOWORD и HIWORD позволяют получить координаты мыши.

Из-за блокировки обновления вывода по причинам, не касающихся вывода координат, при перетаскивании окна за пределы экрана, текст координат пропадет и не появится, пока не будет нажата правая кнопка мыши.

**Вывод:** 1. Ознакомился со структурой и особенностями программирования оконных windows-приложений. 2. Изучил каркасы windows-приложений. 3. Изучил структуру типового каркаса оконных приложений (ТКП), использование ТКП для создания пользовательских приложений. 4. Ознакомился с особенностями организации вывода в оконных windows-приложениях. 5. Ознакомился с особенностями управления сообщениями.