

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Брянский государственный технический университет

Утверждаю

Ректор университета

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.Н. Федонин

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г.

**Программирование Windows-приложений**

Обработка сообщений клавиатуры, мыши

**Методические указания**

**к выполнению лабораторной работы №4**

**для студентов очной формы обучения**

**по направлению подготовки**

**02.03.03 «Математическое обеспечение и администрирование информационных систем», профиль «Технология программирования»**

**Брянск 2020**

**УДК 004.01**

Программирование Windows-приложений. Обработка сообщений клавиатуры, мыши [Электронный ресурс]: методические указания к выполнению лабораторной работы №4 для студентов очной формы обучения по направлению подготовки 02.03.03 «Математическое обеспечение и администрирование информационных систем», профиль «Технология программирования» – Брянск: БГТУ, 2020. – 28 с.

Разработал:

Е.А. Белов,

канд. техн. наук, доц.

Рекомендовано кафедрой «Информатика и программное обеспечение» БГТУ (протокол № 1 от 31.01.2020г.)

**Методические издания публикуются в авторской редакции**

**ЦЕЛЬ РАБОТЫ**

Изучить работу с мышью и клавиатурой в приложениях WinAPI 32. В результате выполнения работы студент должен знать события, правила обработки и функции работы с мышью и клавиатурой в приложении.

Продолжительность работы - 2 часа.

**2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

Оконная процедура окна, которое имеет фокус клавиатуры, принимает сообщения о нажатии клавиши, когда пользователь печатает на клавиатуре. Сообщения о нажатии клавиши - **WM\_KEYDOWN**, **WM\_KEYUP**, **WM\_SYSKEYDOWN** и **WM\_SYSKEYUP**. Типичная оконная процедура игнорирует все сообщения о нажатии клавиши кроме **WM\_KEYDOWN**. **Windows** посылает сообщение **WM\_KEYDOWN**, когда пользователь нажимает клавишу.

Когда оконная процедура принимает сообщение **WM\_KEYDOWN**, она должна проверить код виртуальной клавиши, который сопровождает сообщение, чтобы установить, как обработать нажатие клавиши. Код виртуальной клавиши находится в параметре **wParam** сообщения. Как правило, приложение обрабатывает только нажатия клавиши, сгенерированные не символьными клавишами, включая функциональные клавиши, клавиши управления курсором, и клавиши специального назначения типа **INS**, **DEL**, **HOME** и **END**.

Следующий пример показывает общую схему оконной процедуры, которую использует типичная прикладная программа, чтобы принимать и обрабатывать сообщения о нажатии клавиши.

case WM\_KEYDOWN:

switch (wParam) {

case VK\_LEFT:

*.*

*. /\* Обрабатывает клавишу LEFT ARROW (Стрелка влево). \*/*

*.*

break;

case VK\_RIGHT:

*.*

*. /\* Обрабатывает клавишу RIGHT ARROW (Стрелка вправо). \*/*

*.*

break;

case VK\_UP:

*.*

*. /\* Обрабатывает клавишу UP ARROW (Стрелка вверх). \*/*

*.*

break;

case VK\_DOWN:

*.*

*. /\* Обрабатывает клавишу DOWN ARROW (Стрелка вниз). \*/*

*.*

break;

case VK\_HOME:

*.*

*. /\* Обрабатывает клавишу HOME. \*/*

*.*

break;

case VK\_END:

*.*

*. /\* Обрабатывает клавишу END. \*/*

*.*

break;

case VK\_INSERT:

*.*

*. /\* Обрабатывает клавишу INS. \*/*

*.*

break;

case VK\_DELETE:

*.*

*. /\* Обрабатывает клавишу DEL. \*/*

*.*

break;

case VK\_F2:

*.*

*. /\* Обрабатывает клавишу F2. \*/*

*.*

break;

*.*

*. /\* Обрабатывает другие не символьные нажатия клавиш. \*/*

*.*

default:

break;

}

Оконная процедура принимает символьное сообщение тогда, когда функция **TranslateMessage** транслирует код виртуальной клавиши соответствующий клавише знака. Символьными сообщениями являются **WM\_CHAR**, **WM\_DEADCHAR**, **WM\_SYSCHAR**, и **WM\_SYSDEADCHAR**. Типичная оконная процедура игнорирует все символьные сообщения кроме **WM\_CHAR**. Функция **TranslateMessage** генерирует сообщение **WM\_CHAR** тогда, когда пользователь нажимает какую-либо из следующих клавиш.

Любую символьную клавишу

BACKSPACE

ENTER (перевод каретки)

ESC

SHIFT+ENTER (перевод строки)

TAB

Когда оконная процедура принимает сообщение **WM\_CHAR**, она должна проверить код буквы, который сопровождает сообщение, чтобы установить, как обрабатывать символ. Код буквы находится в параметре **wParam** сообщения.

Следующий пример показывает общую схему оконной процедуры, где типичная прикладная программа используется для приема и обработки символьных сообщений.

case WM\_CHAR:

switch (wParam)

{

case 0x08:

*.*

*. /\* Обработка BACKSPACE . \*/*

*.*

break;

case 0x0A:

*.*

*. /\* Обработка перевода строки. \*/*

*.*

break;

case 0x1B:

*.*

*. /\* Обработка Esc. \*/*

*.*

break;

case 0x09:

*.*

*. /\* Обработка табуляции TAB. \*/*

*.*

break;

case 0x0D:

*.*

*. /\* Обработка перевода каретки. \*/*

*.*

break;

default:

*.*

*. /\* Обработка воспроизводимых символов. \*/*

*.*

break;

}

Следующие функции используются, чтобы принимать и обрабатывать ввод информации с клавиатуры.

ActivateKeyboardLayout

EnableWindow

GetActiveWindow

GetAsyncKeyState

GetFocus

GetKeyboardLayout

GetKeyboardLayoutList

GetKeyboardLayoutName

GetKeyboardState

GetKeyNameText

GetKeyState

IsWindowEnabled

keybd\_event

LoadKeyboardLayout

MapVirtualKey

MapVirtualKeyEx

OemKeyScan

RegisterHotKey

SetActiveWindow

SetFocus

SetKeyboardState

ToAscii

ToAsciiEx

ToUnicode

UnloadKeyboardLayout

UnregisterHotKey

VkKeyScan

VkKeyScanEx

Windows посылает сообщения клавиатуры тому окну, которое имеет фокус ввода. Сообщения мыши отличаются: оконная процедура получает сообщения мыши и когда мышь проходит через ее окно, и при щелчке внутри окна, даже если окно не активно или не имеет фокуса ввода.

В Windows для мыши определен набор из 21 сообщения. Однако 11 из этих сообщений не относятся к рабочей области, и программы для Windows обычно игнорируют их.

Если мышь перемещается по рабочей области окна, оконная процедура получает сообщение **WM\_MOUSEMOVE**.

Если кнопка мыши нажимается или отпускается внутри рабочей области окна, оконная процедура получает следующие сообщения:

* о нажатии – **WM\_LBUTTONDOWN, WM\_RBUTTONDOWN, WM\_MBUTTONDOWN**;
* об отпускании – **WM\_LBUTTONUP, WM\_RBUTTONUP, WM\_MBUTTONUP**;
* о двойном щелчке – **WM\_LBUTTONDBLCLK, WM\_RBUTTONDBLCLK, WM\_MBUTTONDBLCLK**.

Для всех сообщений, связанных с рабочей областью, значение параметра lParam содержит положение мыши. Mладшее слово – это **координата Х,** а старшее – **координата Y** относительно верхнего левого угла рабочей области окна.

* Координаты X и Y можно извлечь из параметра lParam с помощью макросов **LOWORD** и **HIWORD**.

Значение параметра **wParam** показывает состояние кнопок мыши и клавиш **<Shift> и <Ctrl>.** Можно проверить параметр wParam с помощью битовых масок, определенных в заголовочных файлах:

* **MK\_LBUTTON** – левая кнопка нажата.
* **MK\_RBUTTON** – правая кнопка нажата.
* **MK\_MBUTTON** – средняя кнопка нажата.
* **MK\_SHIFT** – клавиша <Shift> нажата.
* **MK\_CONTROL** – клавиша <Ctrl> нажата.

Вот как, например, выглядит код, проверяющий, была ли нажата левая кнопка мыши при ее движении по рабочей области окна, и рисующий на этом месте пиксел:

case WM\_MOUSEMOVE:

{

// состояние кнопок мыши

UINT fwKeys = wParam;

// горизонтальная позиция курсора

int xPos = LOWORD(lParam);

// вертикальная позиция курсора

int yPos = HIWORD(lParam);

if(fwKeys & MK\_LBUTTON)

{

HDC hDC=GetDC(hWnd);

SetPixel(hDC,xPos,yPos,0l);

ReleaseDC(hWnd,hDC);

}

}; return 0;

При движении мыши по рабочей области окна, Windows не вырабатывает сообщение WM\_MOUSEMOVE для всех возможных положений мыши.

**3. ЗАДАНИЕ**

На основе простейшего приложения создать приложение, которое демонстрирует обработку сообщений мыши и клавиатуры.

Замечания:

* выводиться в левом верхнем углу окна приложения; курсор мыши - использоваться для визуализации указателя мыши при его прохождении по окну приложения; акселераторы клавиш управления курсором "вверх" и "вниз" - для уменьшения и увеличения интервала между сообщениями WM\_TIMER;
* Список сообщений, обязательных для обработки функцией главного окна: WM\_CREATE, WM\_LBUTTONDOWN, WM\_RBUTTONDOWN, WM\_KEYDOWN, WM\_CHAR, WM\_COMMAND. Остальные сообщения передать на обработку стандартной оконной функции;
* Окно должно выводить информацию о клавиатурных сообщениях мыши и клавиатуры;
* Реализовать рисование траектории движения мыши при нажатой клавише.

Отчет должен содержать исходный текст приложения, снабженный комментариями и объяснениями поведения сообщений того или иного типа.

**4. КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Поясните назначение функции GetMessage.
2. Какие параметры имеет функция GetMessage?
3. Зачем нужна функция PeekMessage?
4. При каком условии завершается цикл обработки сообщений?
5. Какие функции вызываются в теле цикла обработки сообщений минимального приложения?
6. В чем заключается отличие между функциями PostMessage и SendMessage?
7. Какие параметры имеет оконная функция?
8. Чем различаются аппаратные и символьные сообщения клавиатуры?
9. Какие системные и несистемные сообщения Вам известны?
10. Что передается в параметрах сообщений от клавиатурц?
11. Что передается в параметрах сообщений от мыши?
12. Что такое не клиентская область?
13. Как выполнить захват и освобождение мыши?
14. Объясните событие WM\_NCHITTEST?
15. Для чего используется стиль класса CS\_DBLCLKS?
16. Где располагается начало системы координат для курсора мыши?
17. Как определить состояние кнопки мыши и клавиши в связи с обрабатываемым в данный момент сообщением?
18. Сколько сообщений WM\_MOUSEMOVE в секунду поступает от устройства мыши?

**5. СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ**

1. Рихтер, Д. Windows для профессионалов: создание эффективных Win32-приложений (с CD-ROM)./Д. Рихтер. – М.: Русская редакция, 2000. – 752с.
2. Шилдт, Г. "MFC основы программирования". /Г. Шилдт. - Киев: BHV, 1997. – 452с.
3. Румянцев, П.В. Азбука программирования в Win32 API. / П.В. Румянцев. – М.: Радио и связь, 1999. – 242с.
4. Соломон, Г. Внутреннее устройство Microsoft Windows 2000. / Г. Соломон, М. Руссинович. – 4-е изд. – Спб.: Питер, 2001. - 752с.
5. Роббинс, Д. Отладка Windows-приложений (+ CD-ROM)./ Д. Роббинс. - Спб.: BHV, 2001. - 512с.
6. Финогенов, К. Г. Win32. Основы пpогpаммиpования/К.Г. Финогенов. – М.:Диалог - МИФИ, 2002.-416 с.
7. MSDN. <http://msdn.microsoft.com>
8. RSDN. <http://www.rsdn.ru>

Программирование Windows-приложений. Обработка сообщений клавиатуры, мыши [Электронный ресурс]: методические указания к выполнению лабораторной работы №4 для студентов очной формы обучения по направлению подготовки 02.03.03 «Математическое обеспечение и администрирование информационных систем», профиль «Технология программирования».

Белов Евгений Александрович

Научный редактор Д.И. Булатицкий

Компьютерный набор Е.А. Белов

Иллюстрации Е.А. Белов

Подписано в печать \_\_.\_\_.2020г. Формат 60х84 1/16 Бумага офсетная. Офсетная печать. Усл.печ.л. 1,63 Уч.-изд.л. 1,63 Тираж 1 экз.

Брянский государственный технический университет

Кафедра «Информатика и программное обеспечение», тел. 56-09-84

241035, Брянск, бульвар 50 лет Октября, 7 БГТУ