

Môn học Lập trình trên Windows
Bài tập thực hành – Phần 1
Thời gian thực hiện: 02 tuần
---oOo---

Chương 1: Lập trình bằng thư viện WIN32 API

1. Làm quen với 1 chương trình C viết trên thư viện API

a. Phần 1:

- Sử dụng VC++ 6, dùng chức năng *New project*, chọn loại project “*Win32 Application*”
- Chọn chức năng “*A typical Hello World !*” application
- Hoàn tất tạo project, biên dịch và chạy thử nghiệm chương trình

b. Phần 2:

- Thay đổi tiêu đề của cửa sổ thành “Chương trình C/Win32 API”
- Thay đổi nội dung hiển thị trên vùng client của cửa sổ thành:
“*Day la chuong trinh dau tien...*
... *viết bằng VC++...*
... *tren HDH Windows...*
... *bằng thư viện API.* “

2. Tài nguyên của chương trình:

Lấy chương trình kết quả của bài tập 1, thực hiện tiếp:

a. Phần 1:

- Thay đổi biểu tượng của chương trình
- Thay đổi con trỏ chuột
- Thay đổi nội dung của hộp thoại “About”

b. Phần 2:

- Bổ sung thêm vào menu “File” các mục sau: New, Open, Save
- Thêm mới 1 menu popup “Edit” với các item sau: Undo, Cut, Copy, Paste, Select all
- Mỗi menu item đều có phím tắt tương ứng

3. Lập trình sự kiện:

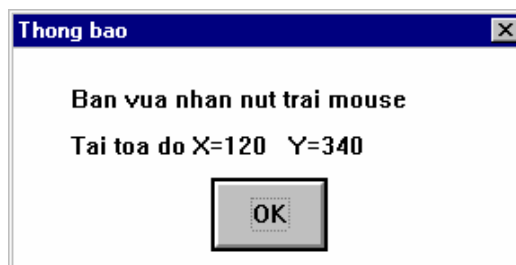
Lấy chương trình kết quả của bài tập 2, thực hiện tiếp:

a. Phần 1:

- Khi user chọn menu item, hiển thị dialog box thông báo:
“Ban vua chon chuc nang XXXX”
- Khi user thoát chương trình, hiển thị dialog box hỏi xác nhận

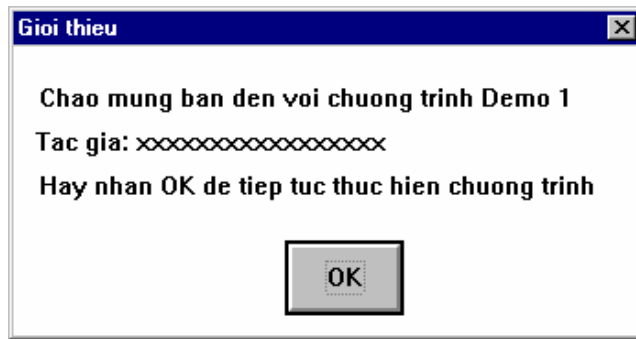
b. Phần 2:

- Khi user click mouse trái/phải, hiển thị dialog box thông báo:



c. Phần 3:

- Trước khi cửa sổ chương trình được hiển thị, hãy thể hiện 1 message-box ghi lời giới thiệu chương trình:



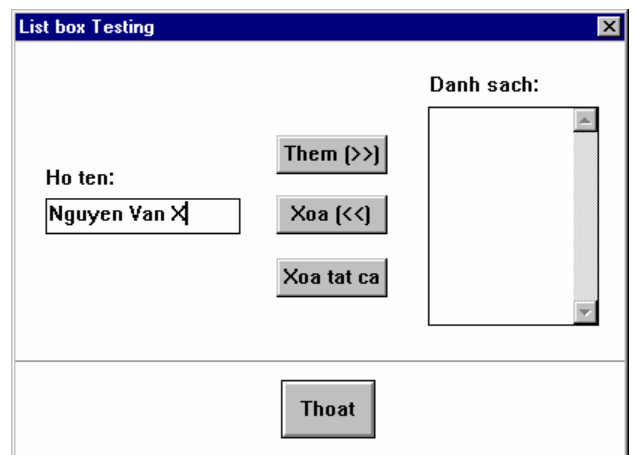
4. Xử lý thông điệp WM_PAINT

- Tạo một “Win32 Application”.
- Vẽ trên vùng client một hình tròn có tâm chính giữa vùng client, bán kính 200 pixel
- Cập nhật lại vùng client thông qua message WM_PAINT để luôn hiển thị hình tròn trên màn hình

5. Xây dựng hộp thoại (Dialog-box)

Thiết kế 1 hộp thoại như hình bên.

- “Danh sách” là 1 list-box, dùng chứa họ tên
- Control “Ho ten” là 1 edit-text. Khi nhập vào control này, sẽ enable nút “Them”. Nếu không có data trong control này, nút “Them” bị disabled
- Control “Them” là 1 push-button. Khi nhấn sẽ add họ tên trong edit-text vào danh sách
- Control “Xoa” là 1 push-button. Bình thường bị disabled. Chỉ được enabled khi chọn 1 phần tử trong list-box “Danh sách”. Nhấn nút “Xoa” sẽ xóa phần tử đang chọn ra khỏi list-box
- “Xoa tat ca” dùng để xóa toàn bộ nội dung list-box



6. Sử dụng hàm Callback

Thiết kế 1 hộp thoại như hình vẽ sau.

Nút “Read font” sẽ đọc tất cả font chữ của hệ thống và add tên font vào trong list-box “List fonts”.

7. Sử dụng bộ định giờ (Timer)

a. Phần 1:

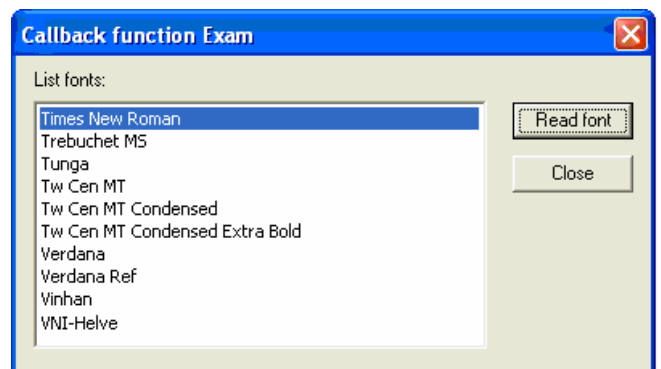
- Viết chương trình có 1 cửa sổ giao diện, kích thước 300x100 pixel, không menu, chỉ có Close button (hình vẽ)

b. Phần 2:

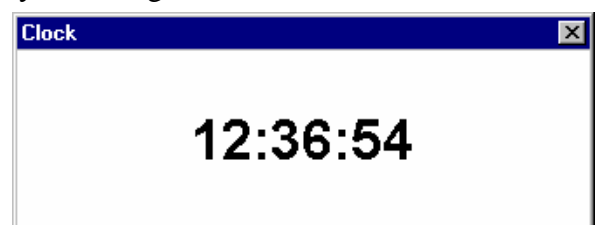
- Định nghĩa 1 bộ định giờ, với chu kỳ 1 giây
- Ứng với thông điệp WM_TIMER, hiển thị giờ hiện tại lên cửa sổ giao diện (hình vẽ)
- Khi kết thúc chương trình (Close window), hủy bộ định giờ

c. Phần 3:

- Tạo lập font chữ, dùng font Arial, size = 28
- Sử dụng font đó để hiển thị giờ (hình vẽ)



- Hết phần 1 -



Môn học Lập trình trên Windows
Bài tập thực hành – Phần 2
Thời gian thực hiện: 02 tuần
---oOo---

Chương 2: Lập trình bằng MFC

1. Làm quen với 1 chương trình C viết bằng thư viện MFC

a. Phần 1:

- Sử dụng VC++ 6, dùng chức năng *New project*, chọn loại project “*MFC AppWizard (exe)*”
- Chọn loại ứng dụng “*Single document*”
- Hoàn tất tạo project, biên dịch và chạy thử nghiệm chương trình

b. Phần 2:

- Thay đổi biểu tượng của chương trình
- Thay đổi con trỏ chuột
- Thay đổi nội dung của hộp thoại “About”

c. Phần 3:

- Khi user click mouse trái/phải, hiển thị dialog box thông báo:
“Ban vừa nhan mouse trai [phai], tai toa do X=, Y=”

2. Hint cho menu item:

Lấy chương trình kết quả của bài tập 1, thực hiện tiếp:

- Tạo Hint cho mỗi menu item trong chương trình
- Các chuỗi Hint được mô tả trong RC

3. Một chương trình soạn thảo văn bản đơn giản

- Tạo ứng dụng MFC ở chế độ “*Single document*”
- Khi user chọn chức năng “New” à tạo 1 cửa sổ soạn thảo EDIT trong vùng client của cửa sổ giao diện chính
- Xử lý WM_SIZE để kích thước của cửa sổ soạn thảo EDIT luôn trùng khớp với kích thước vùng client của cửa sổ giao diện chính
- Thực hiện các chức năng: Open, Save, Save as, Undo, Cut, Copy, Paste (xử lý trên file .TXT)

4. Một chương trình quản lý đơn giản

- Tạo ứng dụng MFC ở chế độ “*Dialog based*”
- Màn hình giao diện gợi ý như hình vẽ

- Chương trình cho phép nhập và lưu thông tin lý lịch của các nhân viên. Dữ liệu được lưu vào file nhị phân dạng record.

- Người dùng nhập thông tin chi tiết cho 1 nhân viên rồi nhấn “*Lưu*” để lưu record vào file. Sau khi lưu xong, xóa các ô nhập liệu để thực hiện cho nhân viên tiếp theo.
- Mục “*Trình độ*” có các lựa chọn sau: tiểu học, trung học, trung cấp, đại học, sau đại học
- Nút “*Chọn hình*” mở hộp thoại Open cho phép chọn 1 file hình ảnh (BMP). Tên file được thể hiện trong ô “*Tên file hình*”, ảnh được hiển thị trên ô trống

- Hết phần 2 -

Môn học Lập trình trên Windows
Bài tập thực hành – Phần 3
Thời gian thực hiện: 02 tuần
---oOo---

Chương 3: DLL và Hook

1. Làm quen với Regular DLL

Cho một DLL có tên PHEP_TINH, bao gồm 2 file: Phep_Tinh.dll và Phep_Tinh.lib

Biết rằng DLL có 4 hàm export:

```
EXPORT int WINAPI Phep_Cong(char *aSh1, char *aSh2, char *aKq);  
EXPORT int WINAPI Phep_Tru(char *aSh1, char *aSh2, char *aKq);  
EXPORT int WINAPI Phep_Nhan(char *aSh1, char *aSh2, char *aKq);  
EXPORT int WINAPI Phep_Chia(char *aSh1, char *aSh2, char *aKq);
```

Mỗi hàm nhận vào 2 số hạng (aSh1, aSh2) **à** chuyển sang số thực **à** thực hiện phép tính **à** trả về kết quả (aKq) ở dạng chuỗi

a. Phần 1:

- Viết 1 ứng dụng (dạng dialog-based) đơn giản cho phép thực hiện 4 phép tính như hình bên.
- Thao tác tính toán được gọi thực hiện thông qua DLL đã cho, áp dụng cách gọi Load-time



b. Phần 2:

- Viết ứng dụng tương tự như phần 1, nhưng áp dụng cách gọi Run-time

2. Làm quen với Extended DLL (MFC)

Cho một DLL có tên EXPORTCLASS, bao gồm 3 file: ExportClass.dll, ExportClass.lib và ExportClass.h

File ExportClass.h chứa mô tả class được Export từ DLL nói trên (SampleClass)

Viết 1 ứng dụng đơn giản, khai báo một thể hiện của SampleClass, gọi phương thức Test() của class

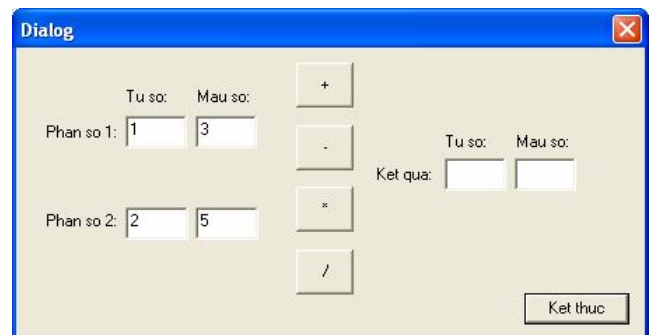
3. Tự xây dựng Extended DLL (MFC)

a. Phần 1:

- Viết một DLL có class PHAN_SO được export
- Class PHAN_SO là class biểu diễn các đối tượng phân số, có các phép toán: cộng, trừ, nhân, chia phân số

b. Phần 2:

- Viết một ứng dụng (dialog-based) như hình bên
- Ứng dụng gọi DLL để thực hiện tính toán



4. Mouse Hook

a. Phần 1:

- Viết một chương trình mouse-hook chặn message WM_LBUTTONDOWN và hiển thị toạ độ mouse lúc đó
- Hook được cài đặt cục bộ (local)

b. Phần 2:

- Làm tương tự phần 1 nhưng cài hook toàn cục (global)

5. Keyboard Hook

a. Phần 1:

- Viết một chương trình keyboard-hook chặn message của keyboard và hiển thị ký tự được nhấn
- Hook được cài đặt toàn cục (global)

b. Phần 2:

- Phát triển chương trình ở phần 1 để có được 1 bộ gõ tiếng Việt
- Chương trình chạy background, hiển thị icon ở system tray
- Hỗ trợ 1 trong các cách gõ: VNI, TELEX
- Hỗ trợ 1 trong các bảng mã: VNI, TCVN3 hay Unicode

--- Hết ---

Môn học Lập trình trên Windows
Bài tập thực hành – Phần 5
Thời gian thực hiện: 02 tuần
---oOo---

Chương 5: Thư viện đồ họa GDI

Ghi chú: bài tập () là những bài tập tương đối khó, cần phối hợp nhiều kiến thức*

1. Làm quen với GDI

- Tạo 1 ứng dụng đơn giản **My1stGDI** bằng Wizard (Win32 Application)
- Viết tiếp **My1stGDI** để thành một chương trình đồ họa đơn giản cho phép sử dụng mouse để vẽ trên vùng client. Các chức năng bao gồm:
 - o Nhấn nút trái mouse → Vẽ hình tròn với tâm là vị trí click, bán kính 50 pixel
 - o Nhấn nút phải mouse → Vẽ hình vuông với tâm là vị trí click, cạnh dài 100 pixel
 - o Ctrl+Click nút trái mouse → tô màu. Vị trí bắt đầu tô là vị trí click mouse, brush màu xanh (blue), giới hạn tô là viền đen.

2. Làm quen với ảnh Bitmap

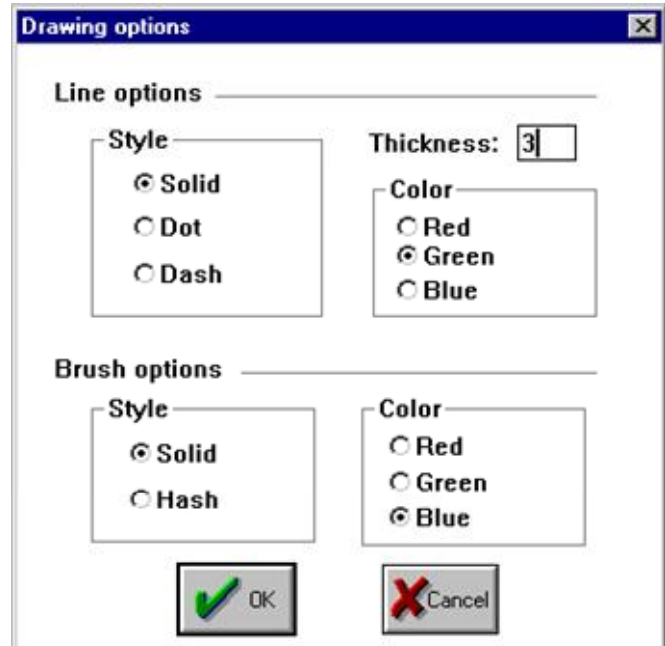
Viết một chương trình với các chức năng như sau:

- **Load bitmap:** load và hiển thị 1 bitmap lên vùng client của cửa sổ giao diện. Bitmap được khai báo trong resource. Bitmap được hiển thị với kích thước thật (tỉ lệ 1:1)
- **Zoom bitmap:** hiển thị lại bitmap trên, với kích thước bằng với kích thước của vùng client

3. Thực tập tiếp với GDI, có sử dụng các đối tượng vẽ (drawing object)

Mở rộng bài [1] bằng cách cho phép user chọn option khi vẽ. Các option được setup thông qua một hộp thoại như hình vẽ.

- Line options: định nghĩa các thuộc tính của nét vẽ (PEN)
- Brush options: định nghĩa các thuộc tính của BRUSH



4. Làm quen với In ấn (printing)

Mở rộng bài [2] với những yêu cầu sau:

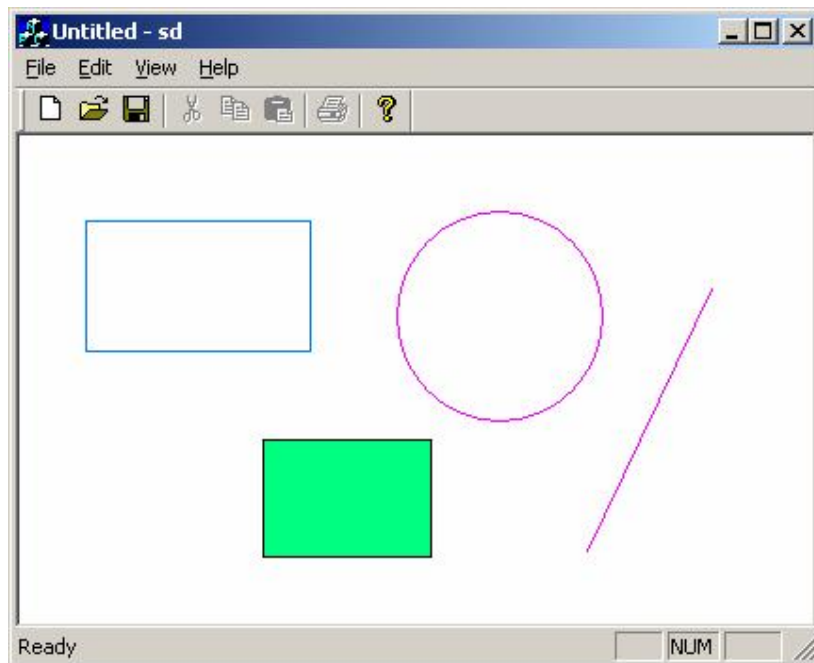
- Chức năng "In": in ảnh bitmap ra máy in. User sẽ chọn máy in trên hộp thoại Print
- Xử lý message WM_PAINT để chương trình tự vẽ lại bitmap

5. Xây dựng chương trình “Simple Drawing” (*)

Chương trình cho phép user dùng mouse để vẽ các đối tượng hình, tô màu. Các thao tác vẽ thể hiện bằng bitmap. (tương tự chương trình Paint)

Mô tả chi tiết:

Giao diện gợi ý như hình vẽ (SDI - Single Document Interface).



a. Các chức năng của chương trình bao gồm:

- File:
 - o New: xóa màn hình, bắt đầu vẽ mới.
 - o Exit: thoát chương trình
- Edit:
 - o Cut: cắt khu vực đang chọn, đưa vào Clipboard (dạng bitmap)
 - o Copy: copy khu vực đang chọn, đưa vào Clipboard (dạng bitmap)
 - o Paste: dán dữ liệu trong Clipboard ra màn hình, tại tọa độ (0, 0)
- Drawing:
 - o Line: chọn chế độ vẽ đường thẳng. User dùng mouse để vẽ đoạn thẳng
 - o Rectangle: chọn chế độ vẽ hình chữ nhật. User dùng mouse để vẽ hình chữ nhật
 - o Circle: chọn chế độ vẽ đường tròn. User dùng mouse để vẽ đường tròn.
 - o Select: chọn chế độ đánh dấu vùng (để Cut/Copy)
 - o Fill: chọn chế độ tô màu. User click mouse để chọn điểm bắt đầu tô màu.
 - o Line Color: mở hộp thoại Color, cho phép user chọn màu vẽ đường.
 - o Fill Color: mở hộp thoại Color, cho phép user chọn màu để tô.
- View:
 - o Toolbar: Xem/ẩn thanh ToolBar
 - o Statusbar: Xem/ẩn thanh StatusBar
- Help:
 - o About: giới thiệu về chương trình

b. Các yêu cầu khác:

- Tất cả các chức năng đều thể hiện trên ToolBar
- Mỗi chức năng đều có ToolTip Text và có hướng dẫn trên StatusBar

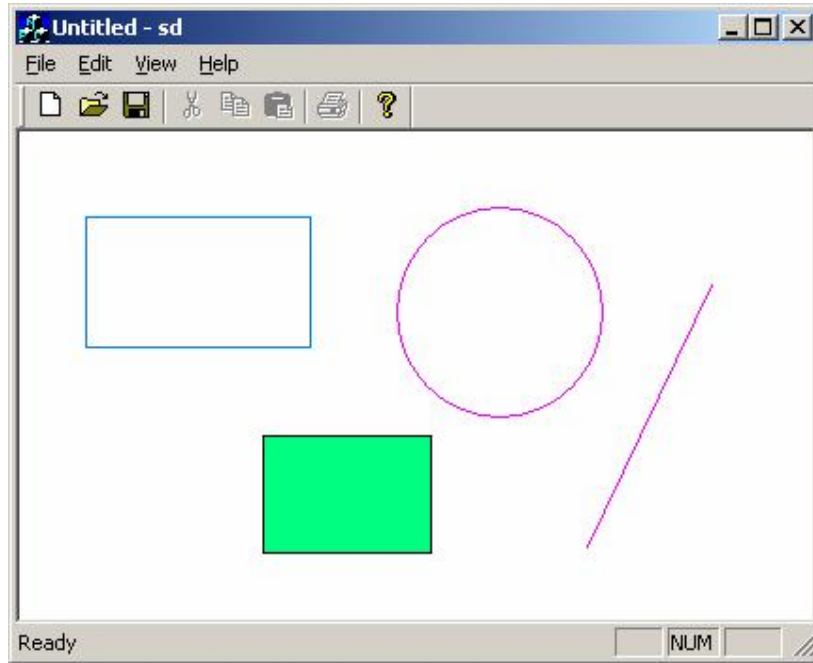
- Tất cả chức năng liên quan đến Clipboard (Cut/Copy/Paste) được tổ chức thành 1 DLL riêng

6. Xây dựng chương trình “Object Drawing” (*)

Chương trình cho phép user dùng mouse để vẽ các **đối tượng hình**, copy, cắt dán các đối tượng. (tương tự chương trình CorelDraw)

Mô tả chi tiết:

Giao diện gợi ý như hình vẽ (SDI - Single Document Interface).



a. Các chức năng của chương trình bao gồm:

- File:
 - o New: xóa màn hình, bắt đầu vẽ mới.
 - o Exit: thoát chương trình
- Edit:
 - o Cut: cắt đối tượng đang được chọn, đưa vào Clipboard (dạng object)
 - o Copy: copy đối tượng đang được chọn, đưa vào Clipboard (dạng object)
 - o Paste: dán đối tượng trong Clipboard ra màn hình, tại tọa độ (0, 0)
- Drawing:
 - o Line: chọn chế độ vẽ đường thẳng. User dùng mouse để vẽ đoạn thẳng. Đối tượng đường thẳng có các thuộc tính: (x1, y1), (x2, y2), nét vẽ (pen, bao gồm: màu (color), độ rộng (thickness), dạng (style)).
 - o Rectangle: chọn chế độ vẽ hình chữ nhật. User dùng mouse để vẽ hình chữ nhật. Đối tượng hình chữ nhật có các thuộc tính: (x1, y1), (x2, y2), nét vẽ (pen, bao gồm: màu (color), độ rộng (thickness), dạng (style)), mẫu tô (brush, bao gồm: dạng (style), màu (color), hatch).
 - o Ellipse: chọn chế độ vẽ đường tròn. User dùng mouse để vẽ đường tròn. Đối tượng ellipse có các thuộc tính: (x, y), bán kính nhỏ, bán kính lớn, nét vẽ (pen, bao gồm: màu (color), độ rộng (thickness), dạng (style)), mẫu tô (brush, bao gồm: dạng (style), màu (color), hatch).
 - o Select: chuyển sang chế độ chọn đối tượng (để Cut/Copy). Đối tượng được chọn sẽ “high-light” và user có thể dùng mouse di chuyển đối tượng.
 - o Settings: mở hộp thoại xác định các thuộc tính vẽ. Hộp thoại cho phép xác định 2 loại thuộc tính sau:

§ Nét vẽ (Pen): màu (color), độ rộng (thickness), dạng (style)

§ Mũi tô (Brush): dạng (style), màu (color), hatch

Các thuộc tính vẽ sẽ được áp dụng cho những đối tượng sau đó.

- View:
 - o Toolbar: Xem/ẩn thanh ToolBar
 - o Statusbar: Xem/ẩn thanh StatusBar
- Help:
 - o About: giới thiệu về chương trình

b. Các yêu cầu khác:

- Tất cả các chức năng đều thể hiện trên ToolBar. Toolbar không thể hiện các chức năng không có trong yêu cầu của đề bài.
- Mỗi chức năng đều có ToolTip Text (trên Toolbar) và có hướng dẫn trên StatusBar.
- Tất cả chức năng liên quan đến Clipboard (Cut/Copy/Paste) được tổ chức thành 1 DLL riêng.

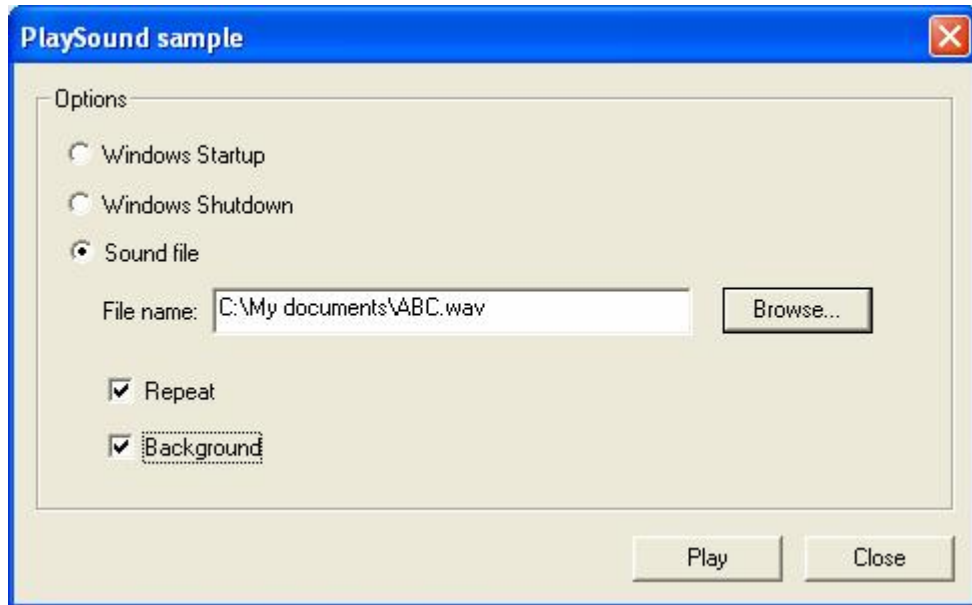
--- Hết ---

Môn học Lập trình trên Windows
Bài tập thực hành – Phần 7
Thời gian thực hiện: 02 tuần
---oOo---

Chương 7: Multi-Media

1. Làm quen với thư viện PlaySound

Viết 1 chương trình (dạng dialog-based) như hình vẽ sau:



Kịch bản:

- User có thể chọn 1 trong 3 option để play:
 - o Windows Startup: play âm thanh khởi động HĐH
 - o Windows Shutdown: play âm thanh shutdown HĐH
 - o Sound file: play 1 file WAV trên đĩa
- Nếu chọn option “Sound file” thì cho phép user chọn “File name” bằng cách dùng hộp thoại Open; ngoài ra, trong option này, user được quyền chọn thêm chế độ play “Repeat” hay “Background”

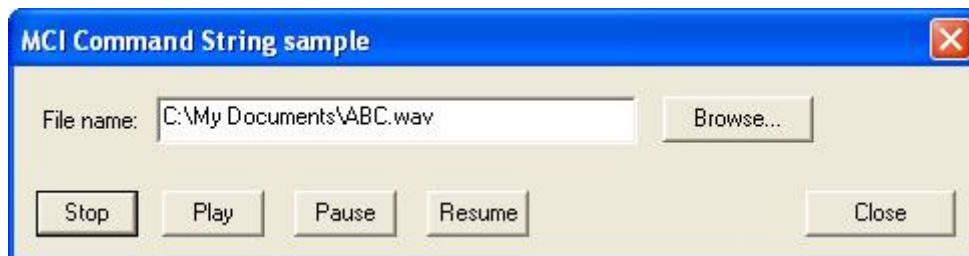
2. Thư viện PlaySound (*)

Viết lại bài [1] với yêu cầu: hàm xử lý cho thao tác “Play” được cài đặt trong một DLL riêng.

3. Làm quen với thư viện MCI

a. Phần 1:

Viết 1 chương trình (dạng dialog-based) như hình vẽ sau:



Kịch bản:

- User có thể chọn 1 file WAV để play; dùng hộp thoại Open để chọn file

- Sau khi chọn file, nút “Play” được Enable; user được nhấn “Play” để play file đã chọn
- Sau khi nhấn Play, các nút “Stop”, “Pause” được Enable
- Khi nhấn “Pause”, nút “Resume” được Enable

Yêu cầu:

- Sử dụng thư viện MCI Command String

b. Phần 2:

Thực hiện tương tự phần 1 nhưng sử dụng thư viện MCI Command Message

4. Audio CD-Player (*)

Dùng thư viện MCI Command Message để viết 1 chương trình CD-Player với những chức năng chính:

- Play, Stop, Pause
- Forward, Backward từng track
- Eject
- Hiện thị số track trên đĩa CD; Track hiện hành; Thời gian của track hiện hành

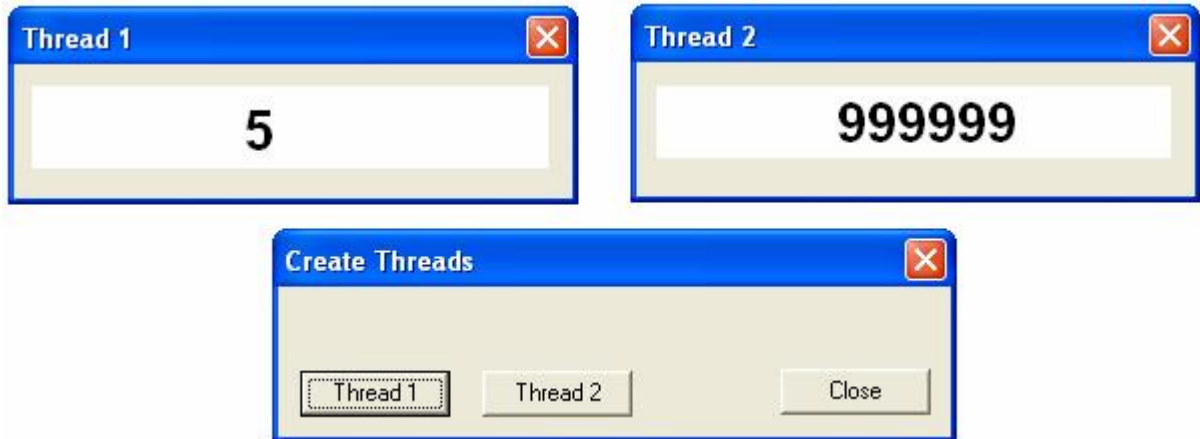
--- Hết ---

Môn học Lập trình trên Windows
Bài tập thực hành – Phần 8
Thời gian thực hiện: 02 tuần
---oOo---

Chương 8: Multi-Thread programming

1. Làm quen với Multi-thread

Viết 1 chương trình dạng dialog-based như hình vẽ:

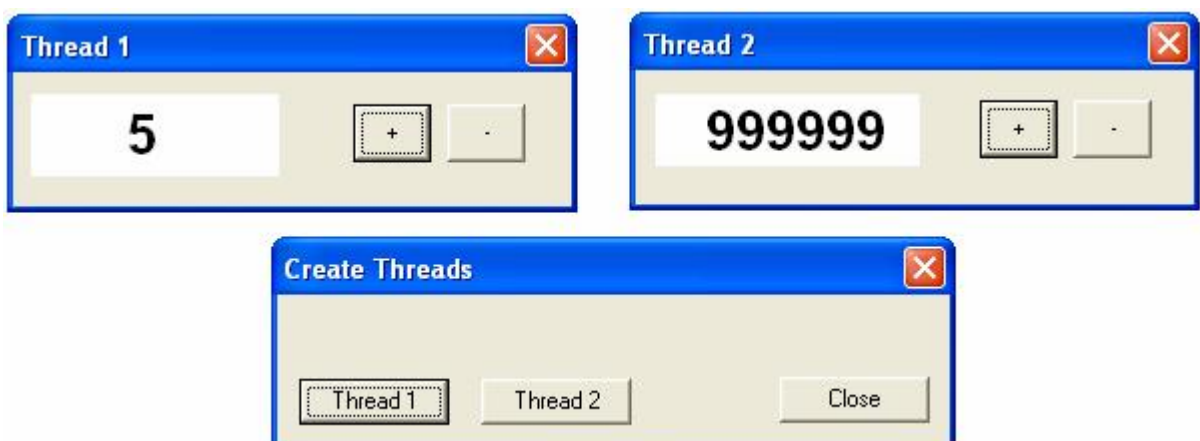


Kịch bản:

- Chương trình chạy với dialog chính “Create Threads”
- User nhấn nút “Thread 1” hay “Thread 2” để tạo ra 2 thread độc lập:
 - o Thread 1: mở 1 cửa sổ, đếm tăng dần từ 1 đến 1,000,000. Thời gian tăng 1 đơn vị là 1ms
 - o Thread 2: mở 1 cửa sổ, đếm giảm dần từ 1,000,000 về 1. Thời gian giảm 1 đơn vị là 1ms
- Quan sát hoạt động của 2 thread. Nhận xét ?

2. Độ ưu tiên của Thread

Phát triển tiếp từ bài 1:



a. Phần 1:

Trên dialog của mỗi thread, tạo 2 button: +, -

- Button +: tăng độ ưu tiên của thread lên 1 bậc (trong 8 bậc ưu tiên của thread)
- Button -: giảm độ ưu tiên của thread xuống 1 bậc (trong 8 bậc ưu tiên của thread)

b. Phần 2:

- Hiển thị mức độ ưu tiên hiện hành của mỗi thread lên dialog của thread đó
- Quan sát hoạt động của 2 thread. Nhận xét ?

3. Độ ưu tiên của Process

Phát triển tiếp từ bài 2, thêm 2 button +/- trên màn hình “Create Thread”:

- Button +: tăng độ ưu tiên của process lên 1 bậc (trong 4 bậc ưu tiên của process: IDLE, NORMAL, HIGH, REALTIME)
- Button -: giảm độ ưu tiên của process xuống 1 bậc (trong 4 bậc ưu tiên của process: IDLE, NORMAL, HIGH, REALTIME)

Quan sát hoạt động của 2 thread. Nhận xét ?

--- Hết ---

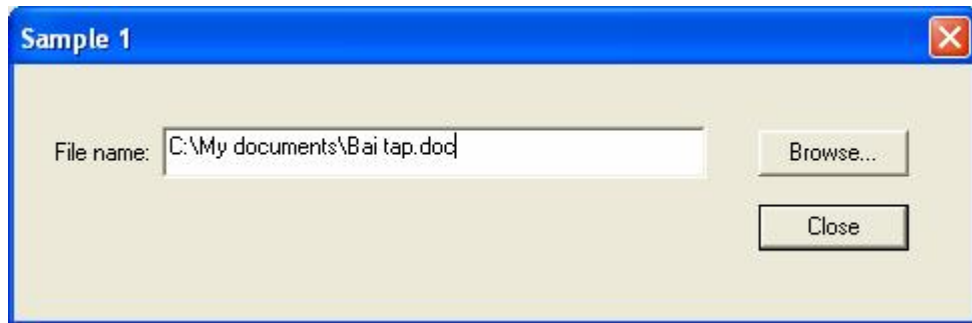
Môn học Lập trình trên Windows
Bài tập thực hành – Phần 9
Thời gian thực hiện: 02 tuần
---oOo---

Phụ lục: Common Dialog boxes

1. Làm quen với các hộp thoại phổ biến

a. Phần 1: Open dialog

Viết một chương trình dạng dialog-based như hình vẽ:



Kích bản:

- Khi nhấn nút “Browse”: mở hộp thoại Open, cho phép user chọn file.
- Nếu user nhấn “Open” thì hiển thị tên file và đường dẫn lên edit-text “File name”; nếu user nhấn “Cancel” thì xóa dữ liệu trong edit-text “File name”
- Nút “Close”: kết thúc chương trình

Yêu cầu:

Hộp thoại Open được mở với các tham số khởi tạo như sau:

- Filter có 4 loại file: .DOC, .XLS, .TXT, .PDF. Loại mặc định được chọn trong filter là .TXT
- Thư mục mặc định là thư mục Windows
- Dùng các flags: OFN_PATHMUSTEXIST, OFN_FILEMUSTEXIST

b. Phần 2: Font dialog

Viết một chương trình dạng dialog-based như hình vẽ:

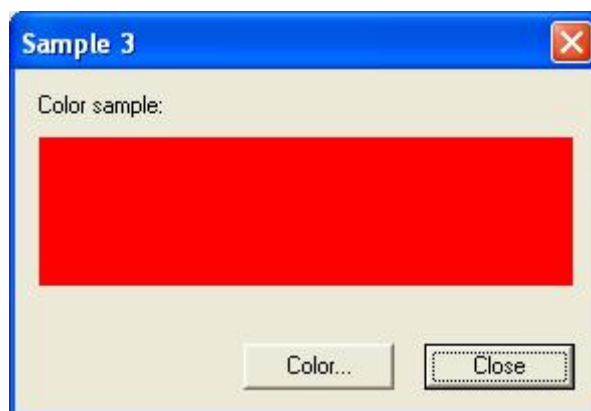


Kích bản:

- Khi nhấn nút “Font”: mở hộp thoại Font, cho phép user chọn font.
- Nếu user nhấn “OK”: hiển thị tên font, size và các tính chất (style) mà user đã chọn trên hộp thoại (như hình minh họa); nếu user nhấn “Cancel” thì xóa tất cả dữ liệu trên hộp thoại.
- Nút “Close”: kết thúc chương trình

c. Phần 3: Color dialog

Viết một chương trình dạng dialog-based như hình vẽ:



Kích bản:

- Khi nhấn nút “Color”: mở hộp thoại Color, cho phép user chọn màu.
- Nếu user nhấn “OK”: dùng màu đã chọn, vẽ 1 hình chữ nhật; nếu user nhấn “Cancel” thì vẽ hình chữ nhật với kiểu tô màu Hatch.
- Nút “Close”: kết thúc chương trình

2. Thay đổi hộp thoại Print

Thực hiện thay đổi hộp thoại Print bằng cách chuyển giao diện của template sang tiếng Việt

3. Thay đổi hộp thoại Open

Chương trình cho phép mở hộp thoại Open chuẩn của Windows với 2 loại file là TXT và BMP. Trên hộp thoại này có một số thay đổi như sau:

- Thêm vào 1 check-box tên là “Preview”, dùng để mở/tắt chế độ Preview nội dung file
- Thêm vào 1 control tên là “View” để hiển thị nội dung file khi user chọn chế độ Preview
- Cách thức hoạt động:
 - § Nếu user chọn chế độ Preview, khi user click chọn vào 1 file nào đó trong list, chương trình sẽ hiển thị nội dung file đó trên control View.
 - § Nếu user không chọn chế độ Preview thì không có gì thay đổi so với hộp thoại Open bình thường
 - § Chỉ support 2 loại file là TXT và BMP.

--- Hết ---