



KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
Bilgisayar Mühendisliği Bölümü
2010-2011 Eğitim-Öğretim Güz Yarıyılı
Windows Programlama 1. Arasnav Soruları



- ✓ Soru kâğıdına **adınız, soyadınız** ve **numaranız** dışında başka hiçbir şey yazmayınız.
- ✓ Sorular eşit puanlıdır. Sınav süresi **120** dakikadır.

1. Win32 proje uygulamalarındaki pencere fonksiyonunda iletiler işlenir ve bu fonksiyonun bildirimi aşağıda gibi yapılabilir:
LRESULT CALLBACK WindowProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam);
Bu pencere prosedürünü, CWindow sınıfının bir üye fonksiyonu olarak tanımlamak istersek programda karşılaşılabilecek sorunlar neler olabilir ve bu sorunlar nasıl giderilebilir? Kısaca açıklayınız.

```
class CWindow
{
    //Bazı üye değişkenler ve fonksiyonlar
    WNDCLASSEX m_wc;
protected:
    HWND m_hWnd;
public:
    CWindow(void);
    virtual ~CWindow(void);
    virtual bool CreateEx(DWORD dwExStyle, LPCTSTR lpszClass, LPCTSTR lpszName, DWORD dwStyle,
        int x, int y, int nWidth, int nHeight, HWND hParent, HMENU hMenu, HINSTANCE hInst);
    virtual void GetWndClassEx(/*WNDCLASSEX & wc*/);
    bool RegisterClass(LPCTSTR lpszClass, HINSTANCE hInst);
    virtual WPARAM MessageLoop(void);
    bool ShowWindow(int nCmdShow) const { return ::ShowWindow(m_hWnd, nCmdShow); }
    bool UpdateWindow(void) const { return ::UpdateWindow(m_hWnd); }
    //Pencere fonksiyonu bildirimi ve diğer işlemler...
};
```

2. Bir slayt gösterisi için Win32 proje uygulaması oluşturulmak istenmektedir. Bu işlem için n tane fotoğraf bir diziye aktararak belli aralıklarla uygulama penceresinde gösterilecektir ve aşağıdaki CBitmap sınıfı da bu projede kullanılacaktır. Sadece tek bir Bitmaple ilgili işlemlerde kullanılacak olan CBitmap::Draw(); ve CBitmap::~CBitmap(); üye fonksiyonlarının tanımlamalarını yapınız. Ayrıca uygulamada, belli aralıklarla Bitmap dizisini ekranda göstermek için hangi Windows iletilerinden ve/veya API fonksiyon(lar)ından yararlanılabilir? Kısaca açıklayınız.

```
class CBitmap
{
    HBITMAP m_hBitmap;           //Bitmap tutamağı
    HDC m_hDC;                   //Sürücü içeriğı tutamağı
    int m_iWidth, m_iHeight;     //Bitmap genişliğı ve yüksekliğı
public:
    void Draw(void);             //Bitmapi uygulama penceresinde çiz
    virtual ~CBitmap(void);      //Bitmap nesnesini serbest bırak
    //Bitmapi yükleme gibi ilgili diğer üye değişkenlerin ve fonksiyonların tanımladığı varsayılacak...
};
```

3. Aşağıda verilen MFC uygulamasında CCalcApp sınıfında CCalculator sınıfına ait bir nesne tanımlanarak Dialog kutusu tabanlı bir uygulama oluşturulacaktır. Bu işlem için aşağıda bildirimleri verilen CCalcApp sınıfındaki ilgili üye fonksiyonların tanımlamalarını yapınız.

```
class CCalculator : public CDialog { public: CCalculator(void); /* ... */ };

class CCalcApp : public CWinApp {
    CCalculator *m_pCacl;
public:
    CCalcApp(void);
    virtual ~CCalcApp(void);
    virtual BOOL InitInstance(void);
};
```

4. Menü içeren bir Win32 proje uygulaması çalışırken klavyeden ALT + F4 tuşlarına basıldığında program sonlandırılmıştır. Bu işlem için uygulamanın program kodunda ne gibi düzenlemeler yapılmış olabilir? Gerekli gördüğünüz yerlerde program kod parçalarını da yazarak kısaca açıklayınız.

```
LONG SetWindowLong(HWND hWnd, int nIndex, LONG dwNewLong);
LONG GetWindowLong(HWND hWnd, int nIndex);
HWND CreateWindowEx(DWORD dwExStyle, LPCTSTR lpClassName, LPCTSTR lpWindowName, DWORD dwStyle, int x, int y, int nWidth, int nHeight, HWND hWndParent, HMENU hMenu, HINSTANCE hInstance, LPVOID lpParam);
HACCEL LoadAccelerators(HINSTANCE hInstance, LPCTSTR lpTableName);
int TranslateAccelerator(HWND hWnd, HACCEL hAccTable, LPMSG lpMsg);
HDC CreateCompatibleDC(HDC hdc);
BOOL DeleteObject(HGDIOBJ hgdiobj);
HGDIOBJ SelectObject(HDC hdc, HGDIOBJ hgdiobj);
BOOL DeleteDC(HDC hdc);
DWORD GetTickCount(void);
BOOL BitBlt(HDC hdcDest, int nXDest, int nYDest, int nWidth, int nHeight, HDC hdcSrc, int nXSrc, int nYSrc, DWORD dwRop);
BOOL PatBlt(HDC hdc, int nXLeft, int nYLeft, int nWidth, int nHeight, DWORD dwRop);
BOOL GetMessage(LPMSG lpMsg, HWND hWnd, UINT wMsgFilterMin, UINT wMsgFilterMax);
BOOL KillTimer(HWND hWnd, UINT nID);
UINT SetTimer(HWND hWnd, UINT nID, UINT uElapse, TIMERPROC lpTimerFunc);
LRESULT DefWindowProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam);
```



KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
Bilgisayar Mühendisliği Bölümü
2010-2011 Eğitim-Öğretim Güz Yarıyılı
Windows Programlama 1. Arasınava Soruları



1. Pencere fonksiyonu aşağıdaki gibi **static** olarak bildirilmeli ve iletiler ya bu pencere fonksiyonundaki göstericiye bağlı olarak veya ikinci bir fonksiyon tanımlanarak o fonksiyonda işlenmelidir. Ayrıca pencere oluşturulurken ilave parametreye **this** göstericisi aktarılmalı ve ilgili üye fonksiyonda aşağıdaki gibi düzenlemeler yapılmalıdır.

```
virtual LRESULT WndProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam);
static LRESULT CALLBACK WindowProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam);

LRESULT CALLBACK CWindow::WindowProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam) {
    CWindow * pWindow = NULL;

    if ( uMsg == WM_NCCREATE ) {
        pWindow = (CWindow *) ((LPCREATESTRUCT) lParam)->lpCreateParams;
        SetWindowLong(hWnd, GWL_USERDATA, (LONG) pWindow);
    }
    else
        pWindow = (CWindow *)GetWindowLong(hWnd, GWL_USERDATA);

    if ( pWindow )
        return pWindow->WndProc(hWnd, uMsg, wParam, lParam);
    else
        return DefWindowProc(hWnd, uMsg, wParam, lParam);
}
//...
m_hWnd = CreateWindowEx(dwExStyle, lpszClass, lpszName, dwStyle, x, y, nWidth, nHeight, hParent, hMenu, hInst, this);
```

2. WM_TIMER iletisi kullanılabilir. Süre ayarlaması ise SetTimer(); ve KillTimer(); fonksiyonlarıyla yapılabilir.

```
void CBitmap::Draw(void) {
    if (m_hBitmap != NULL) {
        HDC hMemDC = CreateCompatibleDC(m_hDC);
        SelectObject(hMemDC, m_hBitmap);
        BitBlt(m_hDC, 0, 0, m_iWidth, m_iHeight, hMemDC, 0, 0, SRCCOPY);
        DeleteDC(hMemDC);
    }
}

CBitmap::~CBitmap(void) {
    if (m_hBitmap != NULL) {
        DeleteObject(m_hBitmap);
        m_hBitmap = NULL;
    }
}
```

- 3.

```
CApp::CApp(void) {
    m_pCacl = NULL;
}

CApp::~CApp(void) {
    if (m_pCacl != NULL) {
        delete m_pCacl;
        m_pCacl = NULL;
    }
}

BOOL CApp::InitInstance(void) {
    m_pCacl = new CCalculator;
    m_pCacl->DoModal();
    return FALSE;
}
```

4. Hızlandırıcı tuş tanımlaması yapılmalıdır. Kaynakla ilgili RC ve H uzantılı dosyalar ve ana program için aşağıda bazı program kodları verilmiştir. Gelen IDM_KAPAT iletisi, eğer makro ismi bu şekilde tanımlanmışsa, ilgili pencere fonksiyonunda işlenir.

```
#define IDM_KAPAT 101 //...

CACC ACCELERATORS {

    VK_F4, IDM_KAPAT, VIRTKEY, ALT
}

HACCEL hAccel; HDC hWnd; MSG msg; //...

hAccel = LoadAccelerators(hInstance, "CACC"); //...
while (GetMessage(&msg)) {
    if (!TranslateAccelerator(hWnd, hAccel, &msg)) { /*...*/ }
}
```