

중간고사

담당교수: 단국대학교 멀티미디어공학전공 박경신

- 답은 반드시 답안지에 기술할 것. 공간이 부족할 경우 반드시 답안지 몇 쪽의 뒤에 있다고 명기한 후 기술할 것. 그 외의 경우의 답안지 뒤쪽이나 연습지에 기술한 내용은 답안으로 인정 안 함. 답에 는 반드시 네모를 쳐서 확실히 표시할 것.
- 답안지에 학과, 학번, 이름 외에 본인의 암호를 기입하면 성적공고시 학번 대신 암호를 사용할 것임.

1. 다음 각각의 상황에서 스택 (Stack)이 유용할 것 같으면 S, 큐 (Queue)가 유용할 것 같으면 Q, 일반적인 리스트 (List)가 유용할 때에는 L로 답하고, 일반적인 맵 (Map)이 유용할 때는 M으로 답하시오. (20점)

- 1) 대형 마트에서 오래된 재고품을 먼저 반품 처리한다. Q
- 2) 주소 목록을 관리하기 위한 프로그램을 작성한다. L
- 3) 근무연한을 기준으로 가장 최근에 고용된 사원이 우선적으로 일시해고된다. S
- 4) 미로에서 출구를 찾는 문제에서 막다른 길에 다다르면 바로 직전 위치로 되돌아간다. S
- 5) 미로에서 출구를 찾는 문제에서 막다른 길에 다다르면 처음 위치로 되돌아간다. Q
- 6) 워드프로세서에서 Undo (가장 최근에 실행시켰던 명령부터 차례대로 취소화시키는 방법) / Redo(가장 최근에 Undo했던 명령어부터 차례대로 다시 실행시키는 방법) 기능을 구현한다. S
- 7) 컴퓨터 프린터에 출력물이 대기하고 있다. Q
- 8) 도시명과 인구수의 쌍을 관리하는 프로그램을 작성한다. M
- 9) 회원정보 처리 프로그램을 작성한다. L
- 10) 윈도우 응용프로그램의 메시지 처리 구현 Q

2. 객체지향 프로그래밍의 상속 (Inheritance)의 목적을 적어도 3가지 기술하시오. (5점)

1. 상위 클래스의 인터페이스를 재사용하기 위한 것이다.
2. 상위 클래스에서 구현된 내용 중 일부를 재사용하기 위한 것이다.
3. 하위 클래스에서는 상위 클래스에 없는 작업만 정의하기 위한 것이다.
4. 상위 클래스의 private 상태 변수에 접근하기 위한 것이다.
5. 비슷한 속성을 여러 번 중복 구현하는 일이 없어진다.

3. 다음은 C++의 Standard Template Library 의 조합 컨테이너 클래스인 set, multiset, map, multimap 의 차이를 간단히 설명하라. (5점)

컨테이너	특성	
set	키가 곧 원소	동일한 값을 갖는 키 허용 안됨
multiset		동일한 값을 갖는 키 허용
map	키와 데이터가 분리	동일한 값을 갖는 키 허용 안됨
multimap		동일한 값을 갖는 키 허용

4. MFC의 CObject 기반 클래스가 제공하는 대표적인 서비스 4가지를 적고 간단히 설명하라. (10점)

1. 실시간 클래스 정보 (run-time class information) 프로그램 실행 중에 객체의 정보 (클래스 타입, 크기 등)를 알 수 있게 해줌.
2. 동적 객체 생성 (dynamic object creation) 객체를 동적으로 생성할 수 있게 해줌
3. 직렬화 (serialization) 객체를 저장하거나 읽어 들일 수 있게 해줌
4. 실시간 객체 상태 검사 (run-time object diagnostics) 객체 상태를 점검하게 해줌

5. MFC 도큐먼트/뷰 구조의 주요 클래스인 CWinApp, CFrameWnd, CView, CDocument를 간단히 설명하라. (5점)

CWinApp - 응용 프로그램 관리, 구동 클래스로 인스턴스를 초기화하고 메시지 루프를 설정

CFrameWnd - 윈도우 이동, 크기 조절 등 윈도우의 프레임을 관리

CView - 윈도우의 클라이언트 영역을 관리하는 클래스로 데이터를 화면에 보여주는 기능, 클라이언트 영역에서 발생하는 마우스, 키보드 메시지 처리를 위한 메시지 핸들러 오버라이딩

CDocument - 프로그램에서 사용되는 데이터 구조, 파일을 관리 (읽기, 저장, 새로운 파일 생성, 파일 닫기, 변경된 데이터를 뷰 객체에 알려주는 기능 등)

6. MFC의 디바이스 컨텍스트와 GDI 객체 등을 제공하는 그리기와 페인팅 관련 클래스를 5개 이상 적어라 (5점)

CDC, CPaintDC, CClientDC, CWindowDC, CMetaFileDC

CGdiObject, CBitmap, CBrush, CFont, CPalette, CPen, CRgn

7. 다음은 Win32 SDK 윈도우 응용프로그램을 보여주고 있다. WinMain 함수와 WinProc 함수의 빈 칸에 프로그램의 구조를 자세히 설명하라. (10점)

```
#include <windows.h>
```

```
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
```

```
int APIENTRY WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
```

```
LPSTR lpCmdLine, int nCmdShow)
```

```
{
```

```
    HWND hwnd;
```

```
    MSG msg;
```

```
    WNDCLASS wndclass;
```

```
    // 1. __윈도우 클래스 구조를 정의하고 운영체제가 등록_____
```

```
    wndclass.style = CS_HREDRAW | CS_VREDRAW;
```

```
    wndclass.lpfnWndProc = WndProc;    // 2. __윈도우 프로시저 함수 주소 정의_____
```

```
    wndclass.cbClsExtra = 0;
```

```
    wndclass.cbWndExtra = 0;
```

```
    wndclass.hInstance = hInstance;
```

```
    wndclass.hIcon = LoadIcon(NULL, IDI_APPLICATION);
```

```
    wndclass.hCursor = LoadCursor(NULL, IDC_ARROW);
```

```
    wndclass.hbrBackground = (HBRUSH)GetStockObject(WHITE_BRUSH);
```

```
    wndclass.lpszMenuName = NULL;
```

```
    wndclass.lpszClassName = "HelloClass";
```

```
    if(!RegisterClass(&wndclass)) return -1;
```

```
    // 3. __메인 윈도우 생성_____
```

```
    hwnd = CreateWindow("HelloClass", "HelloSDK Application",
```

```
        WS_OVERLAPPEDWINDOW,
```

```
        CW_USEDEFAULT, CW_USEDEFAULT,
```

```
        CW_USEDEFAULT, CW_USEDEFAULT,
```

```
        NULL, NULL, hInstance, NULL);
```

```
    // 4. __메인 윈도우 출력_____
```

```
    ShowWindow(hwnd, nCmdShow);
```

```
    UpdateWindow(hwnd);
```

```
    // 5. __메시지 루프 - 윈도우 시스템으로부터 들어오는 메시지를 분석하고 처리_____
```

```
    while(GetMessage(&msg, NULL, 0, 0) > 0){
```

```

        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return msg.wParam;
}

LRESULT CALLBACK WndProc(HWND hwnd, UINT message,
    WPARAM wParam, LPARAM lParam)
{
    HDC hdc;
    PAINTSTRUCT ps;
    char *str = "Hello, SDK";

    switch(message){
    case WM_CREATE: // 1. ____윈도우가 생성될 때 특별히 처리하지 않음____
        return 0;
    case WM_PAINT: // 2. ____윈도우가 다시 그려질 때 화면에 Hello, SDK 출력____
        hdc = BeginPaint(hwnd, &ps);
        TextOut(hdc, 100, 100, str, strlen(str));
        EndPaint(hwnd, &ps);
        return 0;
    case WM_LBUTTONDOWN: // 3. ____왼쪽 마우스버튼이 눌렸을 때 메시지박스가 뜬다____
        MessageBox(hwnd, "마우스 버튼이 눌렸습니다", "메시지", MB_OK);
        return 0;
    case WM_KEYDOWN:
        if (wParam == VK_ESCAPE) // 4. ____ESC 키가 눌렸을 때 윈도우 프로그램이 종료____
            DestroyWindow(hwnd);
        return 0;
    case WM_DESTROY: // 5. ____윈도우가 파괴될 때 종료메시지를 보내어 메시지루프를 빠져나옴____
        PostQuitMessage(0);
        return 0;
    }
    return DefWindowProc(hwnd, message, wParam, lParam);
}

```

8. 내장 객체 (Stock GDI Object)는 무엇인가? 내장 객체의 예를 5개 이상 (이름과 용도를 함께 나타내어) 써라. 그리고 내장 객체의 사용하여 “경계선이 없고 내부가 회색으로 채워진 사각형”을 그리는 간단한 MFC 프로그램의 OnPaint() 함수 부분을 작성하여 보여라. (10점)

```
void CChildView::OnPaint()
{
    CPaintDC dc(this); // device context for painting

    // 내장객체 예제 - 경계선이없고 내부가 회색으로 채워진
    // 사각형(시작점 x=50, y=50, width=200, height=200)을 그린다. Rectangle 함수를 사용한다
    ____dc.SelectStockObject(NULL_PEN); ____
    ____dc.SelectStockObject(GRAY_BRUSH); ____
    ____dc.Rectangle(50, 50, 250, 250); ____
}
```

내장 객체란 윈도우 운영체제가 미리 만들어서 제공하는 GDI 객체로, 내장 객체는 생성과정을 생략하고 MFC의 CDC::SelectStockObject() 함수를 사용하여 DC에 선택한다.

내장 객체의 예로는 BLACK_PEN (폭이 1픽셀인 검정색 펜), WHITE_PEN (폭이 1픽셀인 흰색 펜), NULL_PEN (투명 펜), BLACK_BRUSH (검정색 브러시), DKGRAY_BRUSH (어두운 회색 브러시), GRAY_BRUSH (회색 브러시), LTGRAY_BRUSH (밝은 회색 브러시), SYSTEM_FONT (윈도우 운영체제가 사용하는 폰트), 등등

9. 다음은 간단한 메시지 처리 프로그램 예제를 보여주고 있다. 아래의 네모 안에 OnPaint, OnSize, OnTimer, OnCreate, OnDestroy, OnLButtonDown, OnMouseMove, OnKeyDown 메시지 핸들러 함수를 간단히 설명하고 (예, 언제 불리는지 어떤 일을 하는지 등), 이 프로그램의 전체적인 실행 결과를 써라. (20점)

```
--CChildView.h
#pragma once

// CChildView window
class CChildView : public CWnd
{
// Construction
public:
    CChildView();
```

```
// Attributes
public:

// Operations
public:

// Overrides
protected:
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);

// Implementation
public:
    virtual ~CChildView();

    // Generated message map functions
protected:
    afx_msg void OnPaint();
    DECLARE_MESSAGE_MAP()
public:
    CPoint m_nPt;
    CString m_str, m_str2;
    afx_msg void OnSize(UINT nType, int cx, int cy);
    afx_msg void OnTimer(UINT_PTR nIDEvent);
    afx_msg void OnDestroy();
    afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
    afx_msg void OnLButtonDown(UINT nFlags, CPoint point);
    afx_msg void OnMouseMove(UINT nFlags, CPoint point);
    afx_msg void OnSetFocus(CWnd* pOldWnd);
    afx_msg void OnKillFocus(CWnd* pNewWnd);
    afx_msg void OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);
};
```

--CChildView.cpp

```
#include "stdafx.h"
#include "SimpleMsg.h" // 메인응용프로그램
#include "ChildView.h"

#ifdef _DEBUG
```

```
#define new DEBUG_NEW
#endif

// CChildView
CChildView::CChildView()
{
    m_nPt.x = m_nPt.y = 0;
}

CChildView::~CChildView()
{
}

BEGIN_MESSAGE_MAP(CChildView, CWnd)
    ON_WM_PAINT()
    ON_WM_SIZE()
    ON_WM_TIMER()
    ON_WM_DESTROY()
    ON_WM_CREATE()
    ON_WM_LBUTTONDOWN()
    ON_WM_MOUSEMOVE()
    ON_WM_KEYDOWN()
END_MESSAGE_MAP()

// CChildView message handlers
BOOL CChildView::PreCreateWindow(CREATESTRUCT& cs)
{
    if (!CWnd::PreCreateWindow(cs))
        return FALSE;

    cs.dwExStyle |= WS_EX_CLIENTEDGE;
    cs.style &= ~WS_BORDER;
    cs.lpszClass = AfxRegisterWndClass(CS_HREDRAW|CS_VREDRAW|CS_DBLCLKS,
        ::LoadCursor(NULL, IDC_ARROW), reinterpret_cast<HBRUSH>(COLOR_WINDOW+1), NULL);

    return TRUE;
}
```

```
void CChildView::OnPaint()
{
    CPaintDC dc(this); // device context for painting

    // TODO: Add your message handler code here

    // background
    CRect rect;
    GetClientRect(&rect);
    rect.InflateRect(-30, -30);
    CBrush brush(HS_DIAGCROSS, RGB(0, 255, 0));
    dc.SelectObject(&brush);
    dc.Rectangle(&rect);

    // size
    dc.TextOut(10, 10, m_str2);

    // mouse
    dc.SetBkMode(TRANSPARENT);
    dc.TextOut(m_nPt.x, m_nPt.y, m_str);

    // timer
    CTime time;
    time = CTime::GetCurrentTime();
    CString str = time.Format("현재 시간: %H시%M분%S초");
    GetClientRect(&rect);
    dc.SetTextColor(RGB(255, 0, 255));
    dc.SetBkColor(RGB(125, 125, 125));
    dc.DrawText(str, &rect, DT_CENTER|DT_VCENTER|DT_SINGLELINE);

    // Do not call CWnd::OnPaint() for painting messages
}

void CChildView::OnSize(UINT nType, int cx, int cy)
{
    CWnd::OnSize(nType, cx, cy);

    // TODO: Add your message handler code here
```



```
m_str2.Format("윈도우크기: 가로=%d, 세로=%d", cx, cy);
Invalidate(false);    // 변경된부분만다시그림
}

void CChildView::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: Add your message handler code here and/or call default
    if (nIDEvent == ID_TIMER)
        Invalidate();

    CWnd::OnTimer(nIDEvent);
}

void CChildView::OnDestroy()
{
    CWnd::OnDestroy();

    // TODO: Add your message handler code here
    KillTimer(ID_TIMER);
}

int CChildView::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CWnd::OnCreate(lpCreateStruct) == -1)
        return -1;

    // TODO: Add your specialized creation code here
    SetTimer(ID_TIMER, 1000, NULL);

    return 0;
}

void CChildView::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    m_nPt = point;
    m_str.Format("왼쪽마우스버튼: x=%d, y=%d", point.x, point.y);
}
```

```
    Invalidate(false);

    CWnd::OnLButtonDown(nFlags, point);
}

void CChildView::OnMouseMove(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    m_nPt = point;
    m_str.Format("마우스이동: x=%d, y=%d", point.x, point.y);
    Invalidate(false);

    CWnd::OnMouseMove(nFlags, point);
}

void CChildView::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    // TODO: Add your message handler code here and/or call default
    switch(nChar) {
        // 가상키코드값이
        case VK_LEFT: // 왼쪽화살표키를누를때
            m_nPt.x--; // 왼쪽으로1 이동
            break;
        case VK_RIGHT: // 오른쪽화살표키를누를때
            m_nPt.x++; // 오른쪽으로1 이동
            break;
        case VK_UP: // 위쪽화살표키를누를때
            m_nPt.y--; // 위쪽으로1 이동
            break;
        case VK_DOWN: // 아래쪽화살표키를누를때
            m_nPt.y++; // 아래쪽으로1 이동
            break;
    }

    CRect rect;
    GetClientRect(&rect);
    if(m_nPt.x < 0)
        m_nPt.x = 0; // m_ptNow.x = 0으로초기화
    else if(m_nPt.x > rect.Width())
```

```

        m_nPt.x = rect.right;
    if(m_nPt.y < 0)
        m_nPt.y = 0; // m_ptNow.y = 0으로초기화
    else if(m_nPt.y > rect.Height())
        m_nPt.x = rect.bottom;
    Invalidate(false);

    CWnd::OnKeyDown(nChar, nRepCnt, nFlags);
}
    
```

OnPaint : 다시 그려야할 때

클라이언트영역에 약간 -30 작은 초록색 해쉬 다이아고날 배경 사각형을 그리고 (10, 10) 위치에 “윈도우크기: 가로=xxx, 세로=xxxx”을 출력하고
 현 마우스 위치에 “왼쪽마우스버튼: x=xx, y=xx”또는 “마우스이동중: x=xx, y=xx” 출력
 이 배경 사각형의 가운데 위치에 “현재시간: %H시%M분%S초”를 출력한다.

OnSize : 윈도우 크기가 변경될 때 새로운 윈도우 크기 cx, cy를 받고

그 값을 “윈도우크기: 가로=%d, 세로= %d”로 m_str 에 넣어준다

OnTimer : 지정된 타이머 이벤트가 불릴 때 (매 1초 단위)

Invalidate()를 사용하여 OnPaint를 호출한다

OnCreate : 프로그램의 윈도우가 생성될 때

타이머의 셋팅을 지정 (매 1초단위로 OnTimer를 호출) 한다

OnDestroy : 프로그램의 윈도우가 제거될 때

타이머와 관계된 리소스를 반납하게 된다

OnLButtonDown : 왼쪽 마우스 버튼이 눌렸을 때 현재 마우스 위치를 받고

그 값을 "왼쪽 마우스 버튼: x=%d, y=%d"로 화면의 현재 마우스 위치에 출력한다

OnMouseMove : 마우스가 움직일 때 현재 마우스 위치를 받고

그 값을 "마우스 이동 중: x=%d, y=%d"로 화면에 현재 마우스 위치에 출력한다

OnKeyDown : 왼쪽 화살표키, 오른쪽 화살표키, 위쪽 화살표키, 아래쪽 화살표키 누를 때

각각 왼쪽, 오른쪽, 위쪽, 아래쪽으로 1픽셀씩 이동하고

0, 0, 윈도우 크기 보다 넘치면 다시 초기화한다

전체적인 실행 결과 : 화면 윈도우에 초록색 해쉬 다이아고날 (윈도우의 크기보다 20작은)

사각형 위에, “윈도우 크기: 가로=xxx, 세로=xxx”란 검정색 글씨가 (10, 10) 위치에 출력되고, 배경 초록색 사각형의 가운데 위치에 “현재시간: 11시 29분 31초”란 회색사각형 배경에 보라색 글씨가 보인다. 시간은 매1초단위로 갱신되고, 윈도우의 크기가 바뀔 때 m_str (즉, 윈도우 크기::가로..)값이 갱신된다. 마우스가 움직이거나 왼쪽 마우스버튼이 눌렸을 때 "마우스 이동 중: x=%d, y=%d" 또는 "왼쪽 마우스 버튼: x=%d, y=%d"가 화면의 현재 마우스 위치에 출력한다.

10. 다음은 CArray, CList와 CMap 클래스의 활용 예제를 보여주고 있다. 아래의 네모 안에 프

로그래밍의 실행 출력 결과를 적어라. (10점) 그리고, 다음 예제 프로그램 소스코드에 주석을 달아라. (extra 10점)

(힌트: CObArray와 CObList는 CObject로부터 파생된 객체의 배열과 리스트이고 CMapStringToOb는 CString과 CObject로부터 파생된 객체의 맵)

--ManObject.h

```
#ifndef __MANOBJECT_H
#define __MANOBJECT_H
class CManObject : public CObject
{
public:
    CString m_strName;
    int m_nAge;
    CString m_strAddr;
public:
    CManObject();
    CManObject &operator=(const CManObject& manObject);
};
#endif
```

--ManObject.cpp

```
#include "stdafx.h"
#include "ManObject.h"

CManObject::CManObject()
{
    m_strName = _T("");
    m_nAge = 0;
    m_strAddr = _T("");
}

CManObject &CManObject::operator=(const CManObject& manObject)
{
    m_strName = manObject.m_strName;
    m_nAge = manObject.m_nAge;
    m_strAddr = manObject.m_strAddr;
    return *this;
}
```

--main.cpp

```
#include "stdafx.h"
#include "main.h"
#include "ManObject.h"
#include <afxtempl.h>

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// The one and only application object
CWinApp theApp;

using namespace std;

int _tmain(int argc, TCHAR* argv[], TCHAR* envp[])
{
    int nRetCode = 0;

    // initialize MFC and print and error on failure
    if (!AfxWinInit(::GetModuleHandle(NULL), NULL, ::GetCommandLine(), 0))
    {
        // TODO: change error code to suit your needs
        _tprintf(_T("Fatal Error: MFC initialization failed\n"));
        nRetCode = 1;
    }
    else
    {
        // TODO: code your application's behavior here.
        CManObject *manObject;
        CObArray manArray;
        CObList manList;
        CMapStringToOb manMap;

        // 데이터생성
        manObject = new CManObject;
        manObject->m_strName = "둘리";
    }
}
```

```
manObject->m_nAge = 7;  
manObject->m_strAddr = "서울 성동구";  
manArray.Add(manObject);  
manList.AddTail(manObject);  
manMap[CString("둘리")] = manObject;
```

```
manObject = new CManObject;  
manObject->m_strName = "희동이";  
manObject->m_nAge = 3;  
manObject->m_strAddr = "서울 광진구";  
manArray.Add(manObject);  
manList.AddTail(manObject);  
manMap[CString("희동이")] = manObject;
```

```
manObject = new CManObject;  
manObject->m_strName = "도우너";  
manObject->m_nAge = 20;  
manObject->m_strAddr = "서울 강남구";  
manArray.Add(manObject);  
manList.AddTail(manObject);  
manMap[CString("도우너")] = manObject;
```

```
manObject = new CManObject;  
manObject->m_strName = "또치";  
manObject->m_nAge = 200;  
manObject->m_strAddr = "서울 서초구";  
manArray.Add(manObject);  
manList.AddTail(manObject);  
manMap[CString("또치")] = manObject;
```

```
manObject = new CManObject;  
manObject->m_strName = "마이콜";  
manObject->m_nAge = 30;  
manObject->m_strAddr = "서울 강동구";  
manArray.Add(manObject);  
manList.AddTail(manObject);  
manMap[CString("마이콜")] = manObject;
```

```

// 데이터추가/삭제/검색/출력
CManObject *pManObject;

// Array
manArray.RemoveAt(3);

for(int i = 0; i < manArray.GetSize(); i++){
    pManObject = dynamic_cast<CManObject *> (manArray.GetAt(i));
    printf("Array : Name = %s, Age = %d, Addr = %s\n",
        pManObject->m_strName, pManObject->m_nAge, pManObject->m_strAddr);
}

// List
CManObject *aNewObject;
aNewObject = new CManObject;
aNewObject->m_strName = "고길동";
aNewObject->m_nAge = 35;
aNewObject->m_strAddr = "서울 광진구";

pManObject = dynamic_cast<CManObject *> (manArray[1]);
POSITION pos;
if(pos = manList.Find(pManObject)) {
    printf("FIND : Name = %s, Age = %d, Addr = %s\n",
        pManObject->m_strName, pManObject->m_nAge, pManObject->m_strAddr);
    manList.InsertBefore(pos, aNewObject);
}

pos = manList.GetTailPosition();
while(pos != NULL) {
    pManObject = dynamic_cast<CManObject *> (manList.GetPrev(pos));
    printf("List : Name = %s, Age = %d, Addr = %s\n",
        pManObject->m_strName, pManObject->m_nAge, pManObject->m_strAddr);
}

// Map
CString strKey;
pos = manMap.GetStartPosition();
while(pos) {

```

```

        manMap.GetNextAssoc(pos, strKey, (CObject *)&pManObject);
        printf("Map : Name = %s, Age = %d, Addr = %s\n",
            pManObject->m_strName, pManObject->m_nAge, pManObject->m_strAddr);
    }

    // search
    CManObject *aManObject;
    aManObject = new CManObject;
    if(manMap.Lookup(CString("희동이"), (CObject *)&aManObject))
        printf("LOOKUP : Name = %s, Age = %d, Addr = %s\n",
            aManObject->m_strName, aManObject->m_nAge, aManObject->m_strAddr);
    }

    return nRetCode;
}

```

