|  |  |
| --- | --- |
| sym01_s | **윈도우프로그래밍**  **LAB 09** |
| **분반** | **001** |
| **학번** | **2010044** |
| **이름** | **박진희** |

**# 실습 문제 1**

#1 SDIDoc.h / SDIDoc.cpp 프로그램 소스 및 설명 (추가 및 수정한 부분에 대한 설명)

|  |
| --- |
| SDIDoc.h |
| // SDIDoc.h : interface of the CSDIDoc class  //  #pragma once  class CSDIDoc : public CDocument  {  protected: // create from serialization only  CSDIDoc() noexcept;  DECLARE\_DYNCREATE(CSDIDoc)  // Attributes  public:CString m\_strLines[100];  행을 저장할 문자열과 행의 문자수를 저장하기 위한 int형 변수를 public으로 선언  int m\_nLines;  // Operations  public:  // Overrides  public:  virtual BOOL OnNewDocument();  virtual void Serialize(CArchive& ar);  #ifdef SHARED\_HANDLERS  virtual void InitializeSearchContent();  virtual void OnDrawThumbnail(CDC& dc, LPRECT lprcBounds);  #endif // SHARED\_HANDLERS  // Implementation  public:  virtual ~CSDIDoc();  #ifdef \_DEBUG  virtual void AssertValid() const;  virtual void Dump(CDumpContext& dc) const;  #endif  protected:  // Generated message map functions  protected:  DECLARE\_MESSAGE\_MAP()  #ifdef SHARED\_HANDLERS  // Helper function that sets search content for a Search Handler  void SetSearchContent(const CString& value);  #endif // SHARED\_HANDLERS  }; |

|  |
| --- |
| SDIDoc.cpp |
| // SDIDoc.cpp : implementation of the CSDIDoc class  //  #include "pch.h"  #include "framework.h"  // SHARED\_HANDLERS can be defined in an ATL project implementing preview, thumbnail  // and search filter handlers and allows sharing of document code with that project.  #ifndef SHARED\_HANDLERS  #include "SDI.h"  #endif  #include "SDIDoc.h"  #include <propkey.h>  #ifdef \_DEBUG  #define new DEBUG\_NEW  #endif  // CSDIDoc  IMPLEMENT\_DYNCREATE(CSDIDoc, CDocument)  BEGIN\_MESSAGE\_MAP(CSDIDoc, CDocument)  END\_MESSAGE\_MAP()  // CSDIDoc construction/destruction  CSDIDoc::CSDIDoc() noexcept  {  // TODO: add one-time construction code here  }  CSDIDoc::~CSDIDoc()  {  }  BOOL CSDIDoc::OnNewDocument()  {  if (!CDocument::OnNewDocument())  return FALSE;  // TODO: add reinitialization code here  // (SDI documents will reuse this document)  m\_nLines = 0;  새로운 문서를 만들 시 m\_nLines를 0으로 초기화한다  return TRUE;  }  // CSDIDoc serialization  void CSDIDoc::Serialize(CArchive& ar)  {  if (ar.IsStoring())  저장을 실행하면, 행 별로 ar을 통해 저장한다.  {  // TODO: add storing code here  ar << m\_nLines;  for (int i = 0; i < m\_nLines; i++)  ar << m\_strLines[i];  }  else  {  // TODO: add loading code here  그게 아닌 저장된 파일을 연다면, ar에 저장된 헹들을 m\_strLines로 옮긴다.  ar >> m\_nLines;  for (int i = 0; i < m\_nLines; i++)  ar >> m\_strLines[i];  }  }  #ifdef SHARED\_HANDLERS  // Support for thumbnails  void CSDIDoc::OnDrawThumbnail(CDC& dc, LPRECT lprcBounds)  {  // Modify this code to draw the document's data  dc.FillSolidRect(lprcBounds, RGB(255, 255, 255));  CString strText = \_T("TODO: implement thumbnail drawing here");  LOGFONT lf;  CFont\* pDefaultGUIFont = CFont::FromHandle((HFONT) GetStockObject(DEFAULT\_GUI\_FONT));  pDefaultGUIFont->GetLogFont(&lf);  lf.lfHeight = 36;  CFont fontDraw;  fontDraw.CreateFontIndirect(&lf);  CFont\* pOldFont = dc.SelectObject(&fontDraw);  dc.DrawText(strText, lprcBounds, DT\_CENTER | DT\_WORDBREAK);  dc.SelectObject(pOldFont);  }  // Support for Search Handlers  void CSDIDoc::InitializeSearchContent()  {  CString strSearchContent;  // Set search contents from document's data.  // The content parts should be separated by ";"  // For example: strSearchContent = \_T("point;rectangle;circle;ole object;");  SetSearchContent(strSearchContent);  }  void CSDIDoc::SetSearchContent(const CString& value)  {  if (value.IsEmpty())  {  RemoveChunk(PKEY\_Search\_Contents.fmtid, PKEY\_Search\_Contents.pid);  }  else  {  CMFCFilterChunkValueImpl \*pChunk = nullptr;  ATLTRY(pChunk = new CMFCFilterChunkValueImpl);  if (pChunk != nullptr)  {  pChunk->SetTextValue(PKEY\_Search\_Contents, value, CHUNK\_TEXT);  SetChunkValue(pChunk);  }  }  }  #endif // SHARED\_HANDLERS  // CSDIDoc diagnostics  #ifdef \_DEBUG  void CSDIDoc::AssertValid() const  {  CDocument::AssertValid();  }  void CSDIDoc::Dump(CDumpContext& dc) const  {  CDocument::Dump(dc);  }  #endif //\_DEBUG  // CSDIDoc commands |

#2 SDIView.h / SDIView.cpp 프로그램 소스 및 설명 (추가 및 수정한 부분에 대한 설명)

|  |
| --- |
| SDIView.h |
| // SDIView.h : interface of the CSDIView class  //  #pragma once  class CSDIView : public CView  {  protected: // create from serialization only  CSDIView();  DECLARE\_DYNCREATE(CSDIView)  private:  행의 순서를 나타낼 수인 m\_nPos를 private으로 선언한다.  int m\_nPos;  int m\_nVuSz;  // Attributes  public:  CSDIDoc\* GetDocument() const;  // Operations  public:  // Overrides  public:  virtual void OnDraw(CDC\* pDC); // overridden to draw this view  virtual BOOL PreCreateWindow(CREATESTRUCT& cs);  protected:  virtual BOOL OnPreparePrinting(CPrintInfo\* pInfo);  virtual void OnBeginPrinting(CDC\* pDC, CPrintInfo\* pInfo);  virtual void OnEndPrinting(CDC\* pDC, CPrintInfo\* pInfo);  // Implementation  public:  virtual ~CSDIView();  #ifdef \_DEBUG  virtual void AssertValid() const;  virtual void Dump(CDumpContext& dc) const;  #endif  protected:  // Generated message map functions  protected:  DECLARE\_MESSAGE\_MAP()  public:  virtual void OnInitialUpdate();  afx\_msg void OnEditAppend();  afx\_msg void OnArrowUp();  afx\_msg void OnArrowDown();  };  #ifndef \_DEBUG // debug version in SDIView.cpp  inline CSDIDoc\* CSDIView::GetDocument() const  { return reinterpret\_cast<CSDIDoc\*>(m\_pDocument); }  #endif |

현재 화면에 표시할 행의 수를 정하는 변수

|  |
| --- |
| SDIView.cpp |
| // SDIView.cpp : implementation of the CSDIView class  //  #include "pch.h"  #include "framework.h"  // SHARED\_HANDLERS can be defined in an ATL project implementing preview, thumbnail  // and search filter handlers and allows sharing of document code with that project.  #ifndef SHARED\_HANDLERS  #include "SDI.h"  #endif  #include "SDIDoc.h"  #include "SDIView.h"  #ifdef \_DEBUG  #define new DEBUG\_NEW  #endif  #include "CInputDialog.h"  // CSDIView  IMPLEMENT\_DYNCREATE(CSDIView, CView)  BEGIN\_MESSAGE\_MAP(CSDIView, CView)  // Standard printing commands  ON\_COMMAND(ID\_FILE\_PRINT, &CView::OnFilePrint)  ON\_COMMAND(ID\_FILE\_PRINT\_DIRECT, &CView::OnFilePrint)  ON\_COMMAND(ID\_FILE\_PRINT\_PREVIEW, &CView::OnFilePrintPreview)  ON\_COMMAND(ID\_EDIT\_APPEND, &CSDIView::OnEditAppend)  해당하는 커맨드가 실행되면, 2번째 매개변수에 위치하는 함수를 실행한다.  ON\_COMMAND(ID\_ARROW\_UP, &CSDIView::OnArrowUp)  ON\_COMMAND(ID\_ARROW\_DOWN, &CSDIView::OnArrowDown)  END\_MESSAGE\_MAP()  // CSDIView construction/destruction  CSDIView::CSDIView()  : m\_nPos(0)  {  // TODO: add construction code here  m\_nVuSz = 5;  }  CSDIView::~CSDIView()  {  }  BOOL CSDIView::PreCreateWindow(CREATESTRUCT& cs)  {  // TODO: Modify the Window class or styles here by modifying  // the CREATESTRUCT cs  return CView::PreCreateWindow(cs);  }  // CSDIView drawing  void CSDIView::OnDraw(CDC\* pDC)  {  CSDIDoc\* pDoc = GetDocument();  ASSERT\_VALID(pDoc);  if (!pDoc)  pDoc이 NULL이면 종료  return;  // TODO: add draw code for native data here  for (int i = m\_nPos; i < m\_nPos + m\_nVuSz && i < pDoc->m\_nLines; i++) {  CString str;  행을 화면에 출력한다  str.Format(\_T("%d: "), i);  pDC->TextOutW(10, 10 + 20 \* (i - m\_nPos), str + pDoc->m\_strLines[i]);  }  }  // CSDIView printing  BOOL CSDIView::OnPreparePrinting(CPrintInfo\* pInfo)  {  // default preparation  return DoPreparePrinting(pInfo);  }  void CSDIView::OnBeginPrinting(CDC\* /\*pDC\*/, CPrintInfo\* /\*pInfo\*/)  {  // TODO: add extra initialization before printing  }  void CSDIView::OnEndPrinting(CDC\* /\*pDC\*/, CPrintInfo\* /\*pInfo\*/)  {  // TODO: add cleanup after printing  }  // CSDIView diagnostics  #ifdef \_DEBUG  void CSDIView::AssertValid() const  {  CView::AssertValid();  }  void CSDIView::Dump(CDumpContext& dc) const  {  CView::Dump(dc);  }  CSDIDoc\* CSDIView::GetDocument() const // non-debug version is inline  {  ASSERT(m\_pDocument->IsKindOf(RUNTIME\_CLASS(CSDIDoc)));  return (CSDIDoc\*)m\_pDocument;  }  #endif //\_DEBUG  // CSDIView message handlers  void CSDIView::OnInitialUpdate()  {  CView::OnInitialUpdate();  // TODO: Add your specialized code here and/or call the base class  m\_nPos = 0;  m\_nPos를 0으로 초기화  }  void CSDIView::OnEditAppend()  행 추가 버튼을 누를 시 실행되는 함수  {  // TODO: Add your command handler code here  CInputDialog iDlg;  if (iDlg.DoModal() == IDOK) {  CSDIDoc\* pDoc = GetDocument();  ASSERT\_VALID(pDoc);  if (!pDoc)  return;  pDoc-> m\_strLines[pDoc-> m\_nLines] = iDlg.m\_strLine;  pDoc-> m\_nLines++;  pDoc-> SetModifiedFlag(TRUE);  행이 추가될 시 파일이 수정되었다는(SetModifiedFlag)것을 true로 하여 알려주고,  행을 추가해서 최대 출력 개수에 맞게 표시하도록 한다.  m\_nPos = pDoc-> m\_nLines- m\_nVuSz;  if (m\_nPos < 0)  m\_nPos = 0;  Invalidate();  }  }  void CSDIView::OnArrowUp()  {  // TODO: Add your command handler code here  (m\_nPos > 0) ? m\_nPos-- : 0;  화살표 위키를 누르면, 맨 처음 행이 표시되고 있지 않은 이상 위쪽 행이 보이도록 출력 범위를 이동한다.  Invalidate();  }  void CSDIView::OnArrowDown()  {  화살표 아래키를 누르면, 맨 마지막 행이 표시되고 있지 않은 이상 아래쪽 행이 보이도록 출력 범위를 이동한다.  // TODO: Add your command handler code here  CSDIDoc\* pDoc = GetDocument();  ASSERT\_VALID(pDoc);  (m\_nPos < pDoc->m\_nLines - 1) ? m\_nPos++ : 0;  Invalidate();  } |

#3 InputDialog.cpp 프로그램 소스 및 설명 (추가 및 수정한 부분에 대한 설명)

|  |
| --- |
| CInputDialog.cpp |
| // CInputDialog.cpp : implementation file  //  #include "pch.h"  #include "SDI.h"  #include "CInputDialog.h"  #include "afxdialogex.h"  // CInputDialog dialog  IMPLEMENT\_DYNAMIC(CInputDialog, CDialogEx)  CInputDialog::CInputDialog(CWnd\* pParent /\*=nullptr\*/)  : CDialogEx(IDD\_DIALOG1, pParent)  {  }  CInputDialog::~CInputDialog()  {  }  void CInputDialog::DoDataExchange(CDataExchange\* pDX)  {  CDialogEx::DoDataExchange(pDX);  DDX\_Text(pDX, IDC\_EDIT1, m\_strLine);  }  BEGIN\_MESSAGE\_MAP(CInputDialog, CDialogEx)  END\_MESSAGE\_MAP()  // CInputDialog message handlers |

#4 실행 화면 (텍스트 입력, 텍스트 출력, 문서 이동, 일반 파일 기능 지원에 대한 화면 첨부)















