

IP Programming 4

Jongseok Lee(suk2080@kw.ac.kr)

Yong-Jo Ahn (yjahn@digitalinsights.co.kr)

2018-03-28

Contents

- 4.1 실습 목적
- 4.2 실습 흐름도
- 4.3 실습 따라 하기
- 4.4 실습 과제

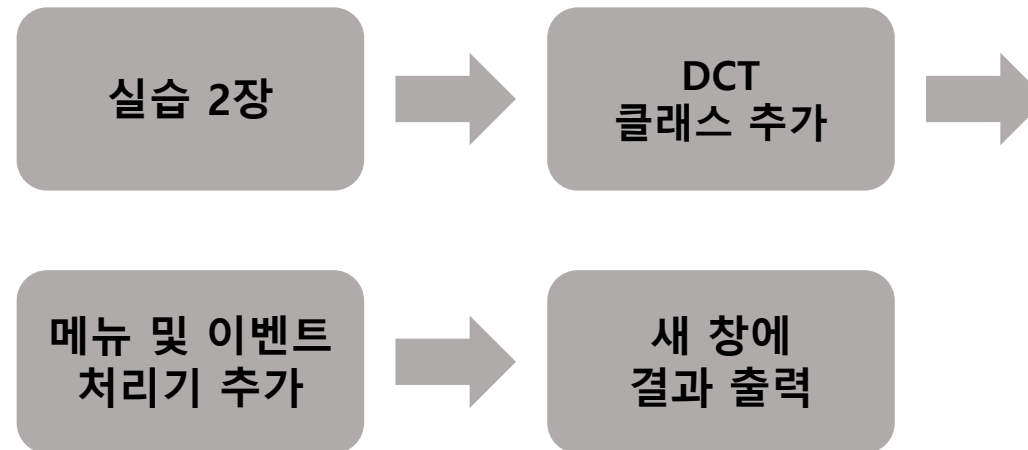
[실습 4.1] 실습 목적

4.1 실습 목적

- 이번 장에서는 이미지를 공간 도메인에서 주파수 도메인으로 변환하여 표현해본다.
- 이 과정을 통해 주파수 도메인에서 나타나는 이미지의 특성을 파악하고 영상의 특성에 따라 영상을 어떻게 개선하면 좋을지에 대해 생각해 본다.

[실습 4.2] 실습 흐름도

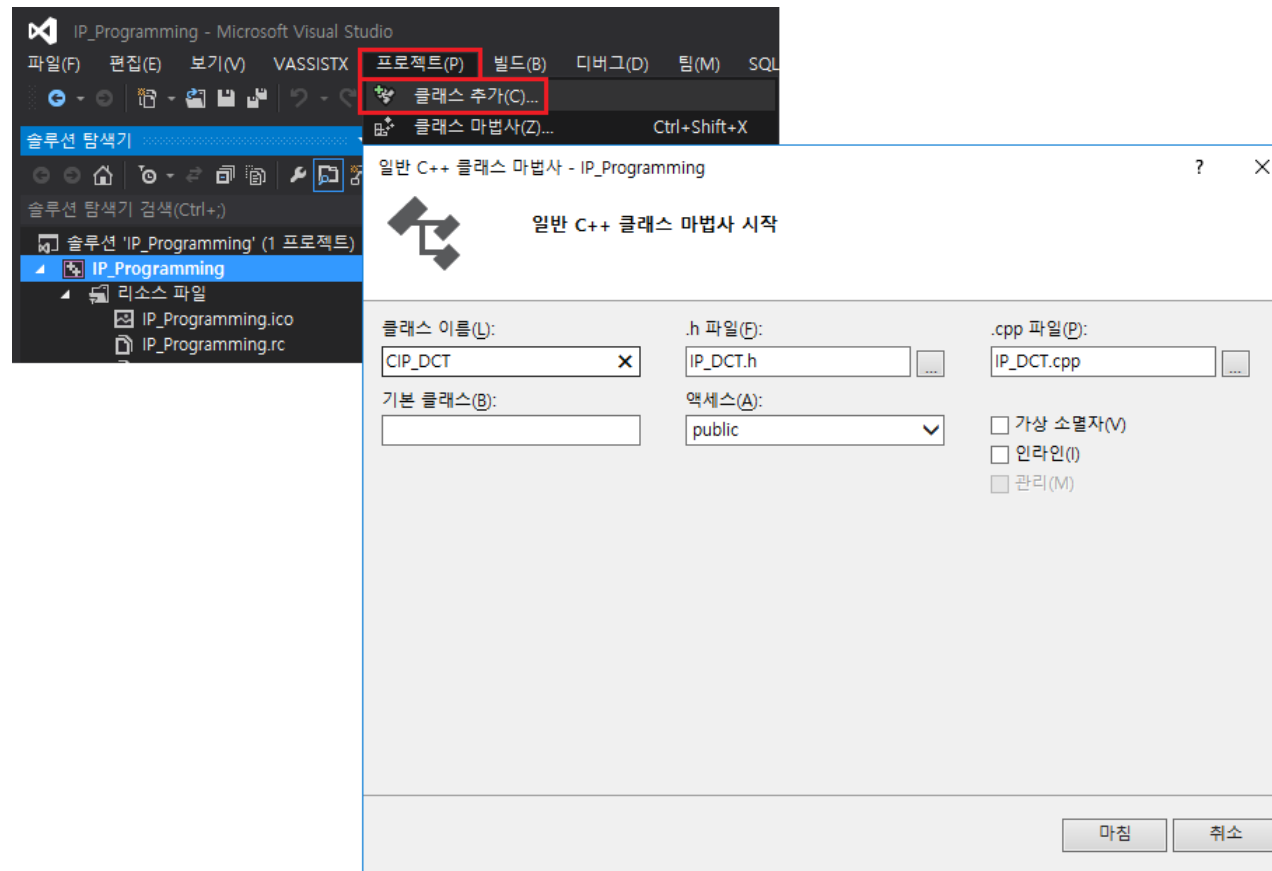
4.2 실습 흐름도



[실습 4-3] 실습 따라 하기

4.3.1 DCT 클래스 추가

- 단계 1 : DCT 클래스 추가
 - [프로젝트] - [클래스 추가]



4.3.1 DCT 클래스 추가

- 단계 2 : DCT 클래스에 멤버변수와 멤버함수 추가


```
IP_DCT.h  + x
(전역 범위)
1  #pragma once
2  class CIP_DCT
3  {
4  public:
5      double** m_pucForwardDCTbuf; //clip DCT 변환 결과 저장 버퍼
6      double** m_pucTempDCTbuf; //DCT 변환 결과 저장 버퍼
7      double** m_pucInverseDCTbuf; //IDCT 변환 결과 저장 버퍼
8
9      UCHAR** FDCTimgbuf; //DCT 이미지 저장 버퍼
10     UCHAR** IDCTimgbuf; //IDCT 이미지 저장 버퍼
11
12 public:
13     CIP_DCT(void);
14     ~CIP_DCT(void);
15
16     void DCT_MakeBlock(int DCT_blocksize, int select, UCHAR** imgbuf, int width, int height); //이미지를 블록 단위로 분할하여 DCT 변환 수행
17     void DCT_ForwardDCT(double** DCT_Block, int DCT_blocksize, int row, int col); //FDCT
18     void DCT_InverseDCT(double** DCT_Block, int DCT_blocksize, int row, int col); //IDCT
19     void DCT_MakeFrequencytoGray(double** imgbuf, UCHAR** buf, int width, int height); //Frequency -> Gray 변환
20     double** memory_alloc2D_D(int width, int height); //2D 메모리 할당
21     UCHAR** memory_alloc2D_UC(int width, int height); //2D 메모리 할당
22 };
```

4.3.1 DCT 클래스 추가

- 단계 2 : DCT 클래스에 멤버변수와 멤버함수 추가

```

IP_DCT.h
(전역 범위)
1  #pragma once
2  class CIP_DCT
3  {
4  public:
5      double**  m_pucForwardDCTbuf; //clip DCT 변환 결과 저장 버퍼
6      double**  m_pucTempDCTbuf; //DCT 변환 결과 저장 버퍼
7      double**  m_pucInverseDCTbuf; //IDCT 변환 결과 저장 버퍼
8
9      UCHAR**   FDCTimgbuf; //DCT 이미지 저장 버퍼
10     UCHAR**   IDCTimgbuf; //IDCT 이미지 저장 버퍼
11
12 public:
13     CIP_DCT(void);
14     ~CIP_DCT(void);
15
16     void DCT_MakeBlock(int DCT_blocksize, int select, UCHAR** imgbuf, int width, int height); //이미지를 블록 단위로 분할하여 DCT 변환 수행
17     void DCT_ForwardDCT(double** DCT_Block, int DCT_blocksize, int row, int col); //FDCT
18     void DCT_InverseDCT(double** DCT_Block, int DCT_blocksize, int row, int col); //IDCT
19     void DCT_MakeFrequencytoGray(double** imgbuf, UCHAR** buf, int width, int height); //Frequency -> Gray 변환
20     double** memory_alloc2D_D(int width, int height); //2D 메모리 할당
21     UCHAR** memory_alloc2D_UC(int width, int height); //2D 메모리 할당
22 };
          
```



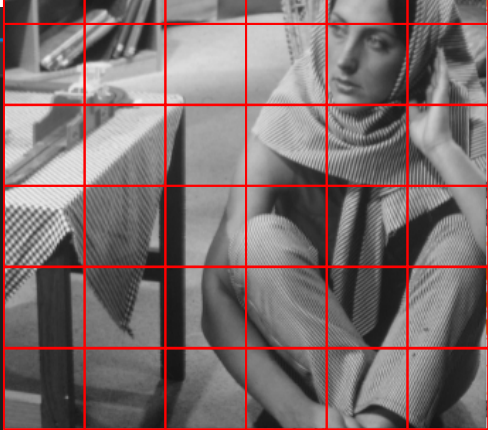
4.3.1 DCT 클래스 추가

- 단계 2 : DCT 클래스에 멤버변수와 멤버함수 추가

IP_DCT.h

(전역 범위)

```
1 #pragma once
2 class CIP_DCT
3 {
4 public:
5     double** m_pucForwardDCTbuf; //clip DCT 변환 결과 저장 버퍼
6     double** m_pucTempDCTbuf; //DCT 변환 결과 저장 버퍼
7     double** m_pucInverseDCTbuf; //IDCT 변환 결과 저장 버퍼
8
9     UCHAR** FDCTimgbuf; //DCT 이미지 저장 버퍼
10    UCHAR** IDCTimgbuf; //IDCT 이미지 저장 버퍼
11
12 public:
13     CIP_DCT(void);
14     ~CIP_DCT(void);
15
16     void DCT_MakeBlock(int DCT_blocksize, int select, UCHAR** imgbuf, int width, int height); //이미지를 블록 단위로 분할하여 DCT 변환 수행
17     void DCT_ForwardDCT(double** DCT_Block, int DCT_blocksize, int row, int col); //FDCT
18     void DCT_InverseDCT(double** DCT_Block, int DCT_blocksize, int row, int col); //IDCT
19     void DCT_MakeFrequencytoGray(double** imgbuf, UCHAR** buf, int width, int height); //Frequency -> Gray 변환
20     double** memory_alloc2D_D(int width, int height); //2D 메모리 할당
21     UCHAR** memory_alloc2D_UC(int width, int height); //2D 메모리 할당
22 };
```

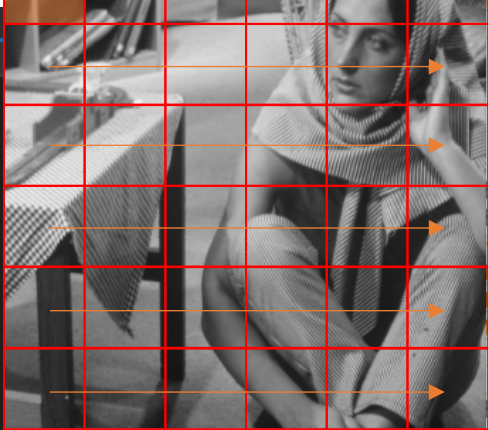


4.3.1 DCT 클래스 추가

- 단계 2 : DCT 클래스에 멤버변수와 멤버함수 추가

```

IP_DCT.h
(전역 범위)
1 #pragma once
2 class CIP_DCT
3 {
4 public:
5     double** m_pucForwardDCTbuf; //clip DCT 변환 결과 저장 버퍼
6     double** m_pucTempDCTbuf; //DCT 변환 결과 저장 버퍼
7     double** m_pucInverseDCTbuf; //IDCT 변환 결과 저장 버퍼
8
9     UCHAR** FDCTimgbuf; //DCT 이미지 저장 버퍼
10    UCHAR** IDCTimgbuf; //IDCT 이미지 저장 버퍼
11
12 public:
13     CIP_DCT(void);
14     ~CIP_DCT(void);
15
16     void DCT_MakeBlock(int DCT_blocksize, int select, UCHAR** imgbuf, int width, int height); //이미지를 블록 단위로 분할하여 DCT 변환 수행
17     void DCT_ForwardDCT(double** DCT_Block, int DCT_blocksize, int row, int col); //FDDT
18     void DCT_InverseDCT(double** DCT_Block, int DCT_blocksize, int row, int col); //IDCT
19     void DCT_MakeFrequencytoGray(double** imgbuf, UCHAR** buf, int width, int height); //Frequency -> Gray 변환
20     double** memory_alloc2D_D(int width, int height); //2D 메모리 할당
21     UCHAR** memory_alloc2D_UC(int width, int height); //2D 메모리 할당
22 };
        
```



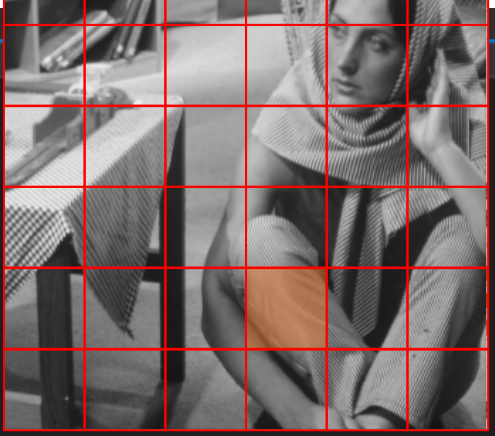
4.3.1 DCT 클래스 추가

- 단계 2 : DCT 클래스에 멤버변수와 멤버함수 추가

```

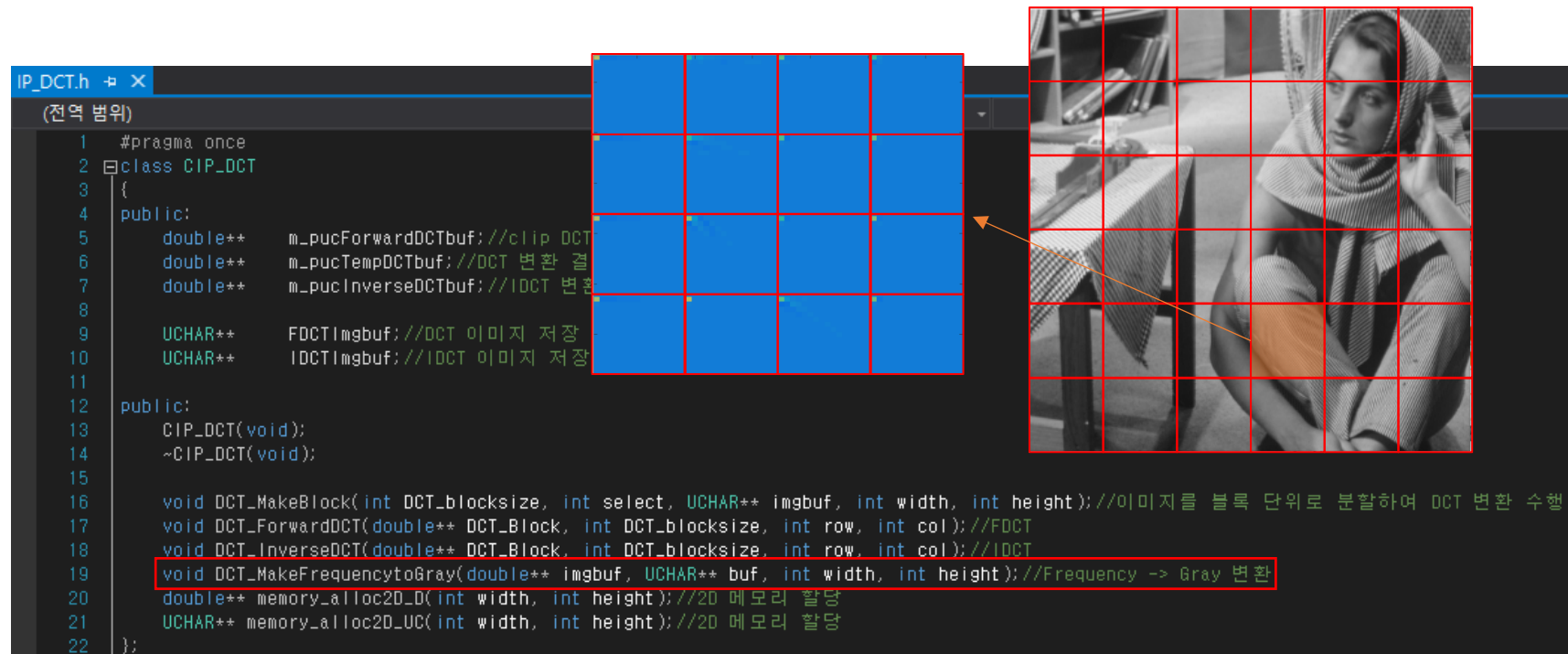
IP_DCT.h
(전역 범위)
1  #pragma once
2  class CIP_DCT
3  {
4  public:
5      double**  m_pucForwardDCTbuf; //clip DCT 변환 결과 저장 버퍼
6      double**  m_pucTempDCTbuf; //DCT 변환 결과 저장 버퍼
7      double**  m_pucInverseDCTbuf; //IDCT 변환 결과 저장 버퍼
8
9      UCHAR**   FDCTimgbuf; //DCT 이미지 저장 버퍼
10     UCHAR**   IDCTimgbuf; //IDCT 이미지 저장 버퍼
11
12 public:
13     CIP_DCT(void);
14     ~CIP_DCT(void);
15
16     void DCT_MakeBlock(int DCT_blocksize, int select, UCHAR** imgbuf, int width, int height); //이미지를 블록 단위로 분할하여 DCT 변환 수행
17     void DCT_ForwardDCT(double** DCT_Block, int DCT_blocksize, int row, int col); //FDDT
18     void DCT_InverseDCT(double** DCT_Block, int DCT_blocksize, int row, int col); //IDCT
19     void DCT_MakeFrequencytoGray(double** imgbuf, UCHAR** buf, int width, int height); //Frequency -> Gray 변환
20     double** memory_alloc2D_D(int width, int height); //2D 메모리 할당
21     UCHAR** memory_alloc2D_UC(int width, int height); //2D 메모리 할당
22 };

```



4.3.1 DCT 클래스 추가

- 단계 2 : DCT 클래스에 멤버변수와 멤버함수 추가



The screenshot shows a C++ code editor with the file name `IP_DCT.h`. The code defines a class `CIP_DCT` with several member variables and functions. The member variables include `m_pucForwardDCTbuf`, `m_pucTempDCTbuf`, `m_pucInverseDCTbuf`, `FDCTimgbuf`, and `IDCTimgbuf`. The functions include `CIP_DCT()`, `~CIP_DCT()`, `DCT_MakeBlock`, `DCT_ForwardDCT`, `DCT_InverseDCT`, `DCT_MakeFrequencytoGray`, `memory_alloc2D_D`, and `memory_alloc2D_UC`. The `DCT_MakeFrequencytoGray` function is highlighted with a red box.

Two visualizations are shown to the right of the code. The first is a 4x4 grid of blue squares, representing a block of the image. The second is a 16x16 grid of squares, representing the entire image. An arrow points from the 4x4 grid to the 16x16 grid, indicating the relationship between the block and the full image.

```

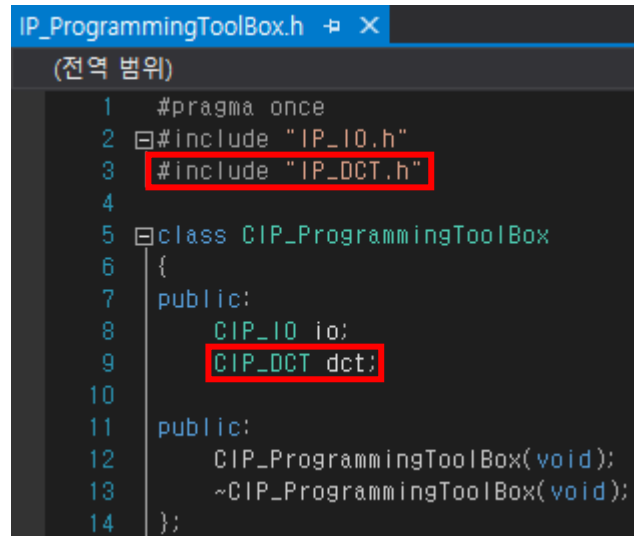
1  #pragma once
2  class CIP_DCT
3  {
4  public:
5      double** m_pucForwardDCTbuf; //clip DCT
6      double** m_pucTempDCTbuf; //DCT 변환 결과
7      double** m_pucInverseDCTbuf; //IDCT 변환
8
9      UCHAR** FDCTimgbuf; //DCT 이미지 저장
10     UCHAR** IDCTimgbuf; //IDCT 이미지 저장
11
12 public:
13     CIP_DCT(void);
14     ~CIP_DCT(void);
15
16     void DCT_MakeBlock(int DCT_blocksize, int select, UCHAR** imgbuf, int width, int height); //이미지를 블록 단위로 분할하여 DCT 변환 수행
17     void DCT_ForwardDCT(double** DCT_Block, int DCT_blocksize, int row, int col); //FDCT
18     void DCT_InverseDCT(double** DCT_Block, int DCT_blocksize, int row, int col); //IDCT
19     void DCT_MakeFrequencytoGray(double** imgbuf, UCHAR** buf, int width, int height); //Frequency -> Gray 변환
20     double** memory_alloc2D_D(int width, int height); //2D 메모리 할당
21     UCHAR** memory_alloc2D_UC(int width, int height); //2D 메모리 할당
22 };
  
```

4.3.1 DCT 클래스 추가

- 단계 3 : DCT 클래스에 멤버함수 작성
 - define 추가
 - 생성자, 소멸자
 - DCT_MakeBlock
 - DCT_ForwardDCT
 - DCT_InverseDCT
 - memory_alloc2D_D
 - memory_alloc2D_UC
 - DCT_MakeFrequencytoGray

4.3.1 DCT 클래스 추가

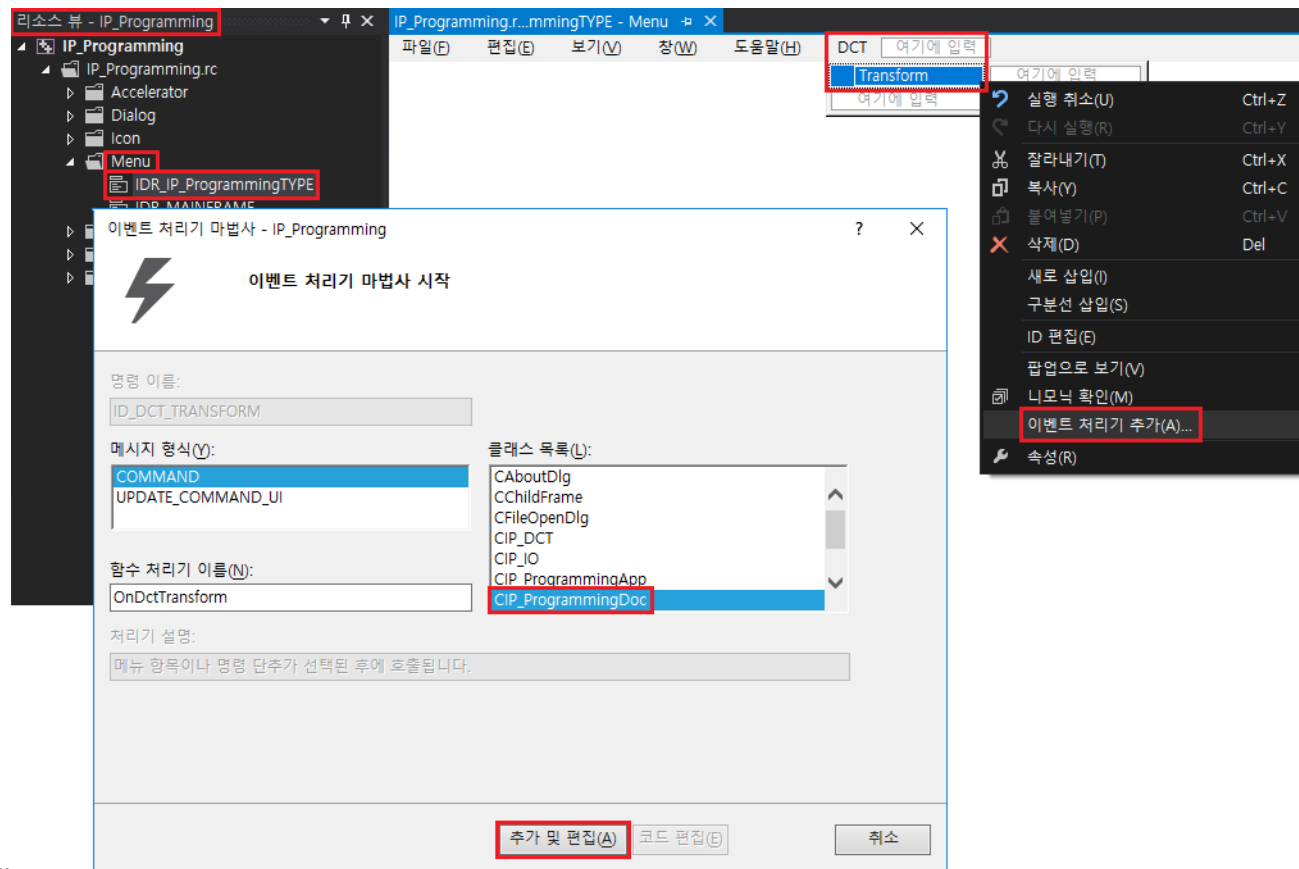
- 단계 4 : CIP_ProgrammingToolbox 클래스에 멤버변수 추가



```
IP_ProgrammingToolBox.h  ➤ ✕  
(전역 범위)  
1  #pragma once  
2  #include "IP_IO.h"  
3  #include "IP_DCT.h"  
4  
5  class CIP_ProgrammingToolBox  
6  {  
7  public:  
8      CIP_IO io;  
9      CIP_DCT dct;  
10  
11  public:  
12      CIP_ProgrammingToolBox(void);  
13      ~CIP_ProgrammingToolBox(void);  
14  };
```

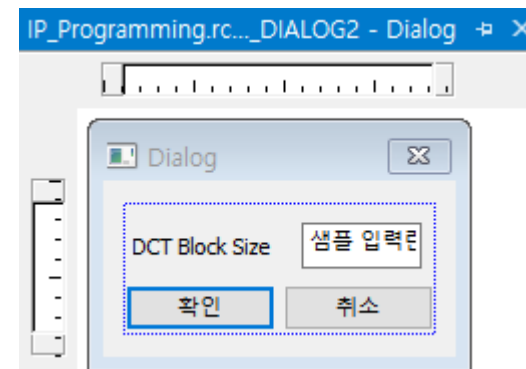
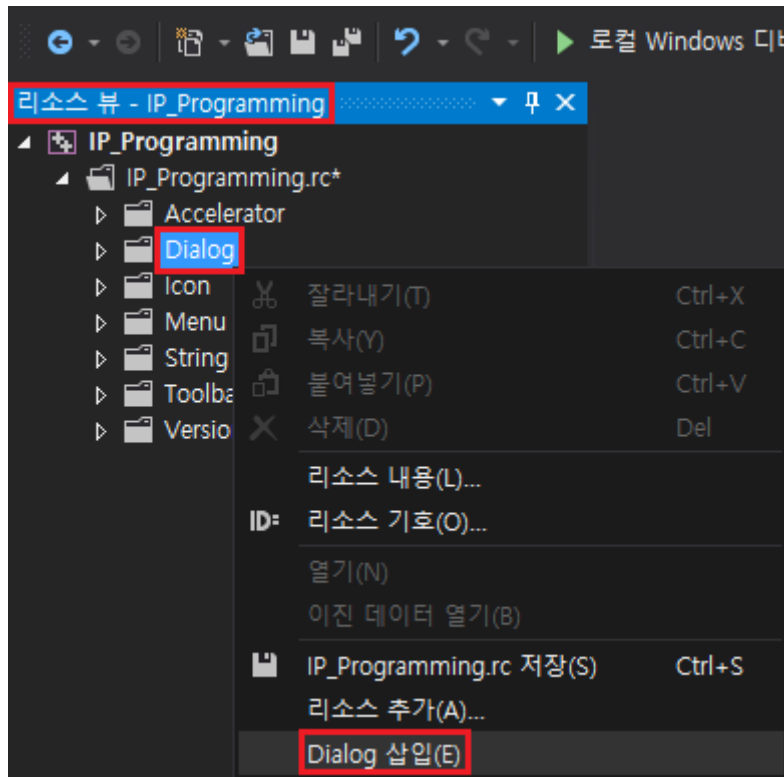

4.3.2 메뉴 및 이벤트 처리기 추가

- 단계 5 : DCT 함수 메뉴 및 이벤트 처리기 추가
 - [리소스 뷰] - [Menu] - [IDR_IP_ProgrammingTYPE]에서 메뉴 추가
 - [DCT] - [Transform] - [이벤트 처리기 추가] - [CIP_ProgrammingDoc]로 이벤트 처리기 추가



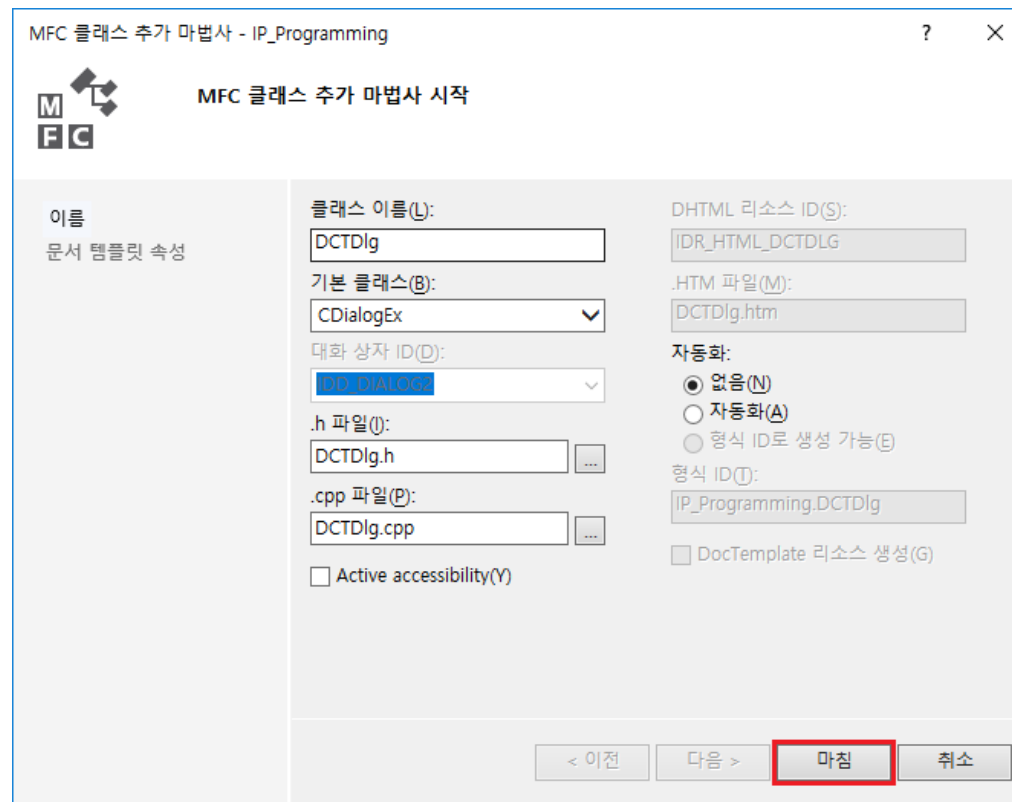
4.3.2 메뉴 및 이벤트 처리기 추가

- 단계 6 : 대화상자 추가
 - [리소스 뷰] - [Dialog] - [Dialog 삽입]



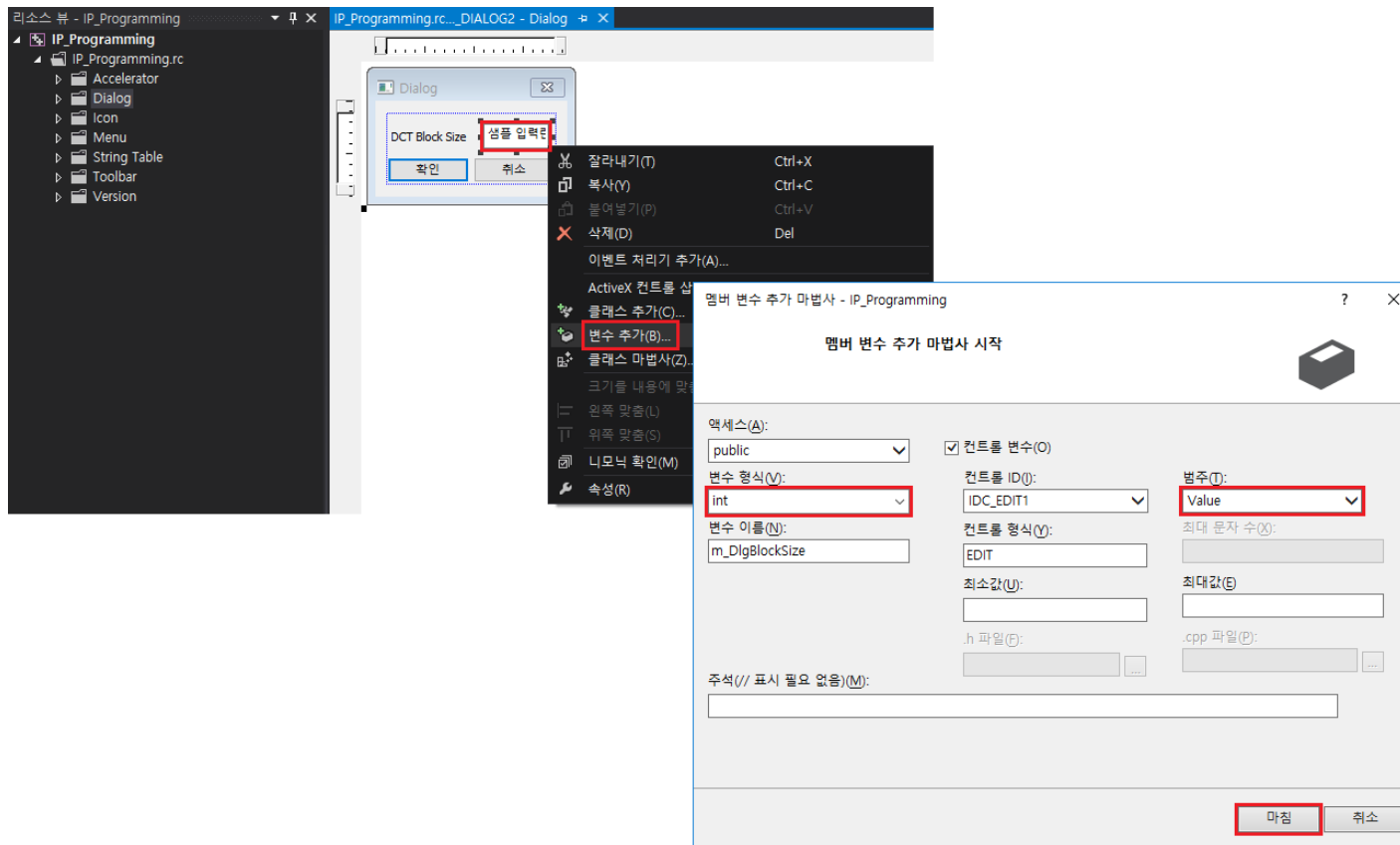
4.3.2 메뉴 및 이벤트 처리기 추가

- 단계 7 : 대화상자 클래스 추가
 - 대화상자 빈공간 더블클릭



4.3.2 메뉴 및 이벤트 처리기 추가

- 단계 8 : 도구상자 변수 추가
 - [Edit Contorl] - [변수 추가] - [범주] - [변수 형식]

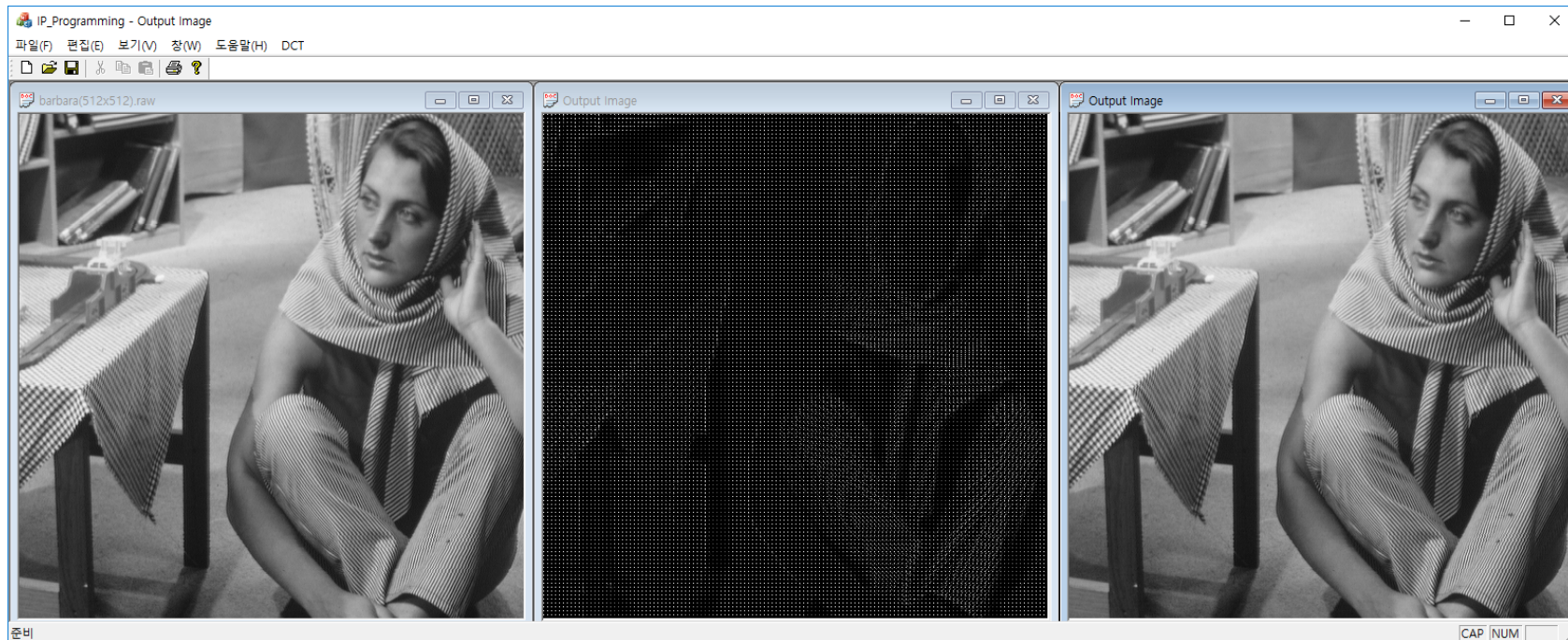


4.3.3 메뉴 및 이벤트 처리기 추가

- 단계 9 : CIP_ProgrammingDoc 클래스에 해더 추가
- 단계 10 : 이벤트 처리기 함수 작성
 - OnDctTransform
- 단계 11 : CIP_Programming 클래스 멤버변수 추가
- 단계 12 : CIP_ProgrammingDoc 클래스 멤버함수 수정

4.3.4 최종 출력 결과

■ 4x4 DCT 변환 및 복원



[실습 4-4] 실습 프로젝트

4.4 실습 과제

- [실습 과제 1] DST 변환 및 복원
 - 입력 받은 영상에 대하여 DST 변환 및 복원을 수행한다.
- [실습 과제 2] DCT 분할
 - 입력 받은 영상에 대하여 DCT 변환 및 복원을 수행한다.
 - 이때, 1차원 DCT를 사용하도록 한다.



< Horizontal 1-D DCT >



< Vertical 1-D DCT >



< 2-D DCT >

END OF PRESENTATION

Q&A