

## 자료구조 실습01

Data Structures Lab01

## Notice

### ◎ 주석 작성 방법

☞ Doxygen 사용

- 클래스

```
/**  
 *   클래스 설명  
 *   @author   작성자 이름  
 *   @date     작성일  
 */
```

- 소스코드

Line by line 설명 또는 블록 별 기능 설명

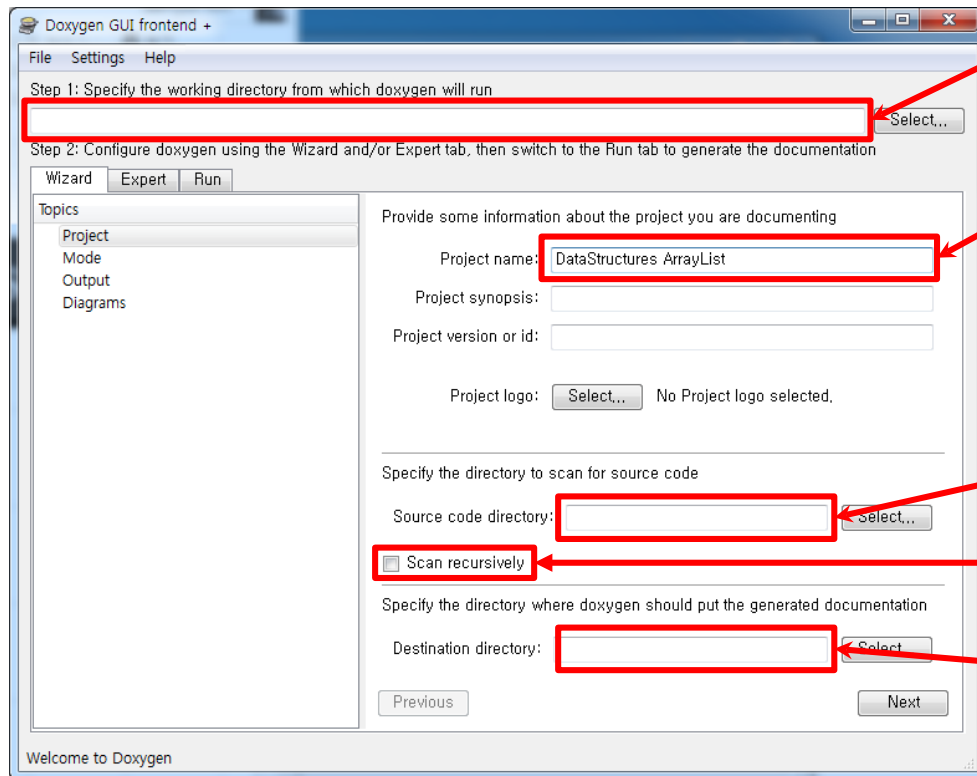
- 참고자료

- 제공된 예제 소스코드 및 html 폴더 확인 (index.html 파일)
- <http://www.stack.nl/~dimitri/doxygen/index.html> (다운로드 및 설명)
- <http://www.slideshare.net/arload/doxygen-33932243>

- 함수

```
/**  
 *   @brief   함수 설명  
 *   @detail  함수의 자세한 설명  
 *   @pre     함수가 동작하기 전의 상태, 조건 등 기술  
 *   @post    함수 동작 후의 상태 변화 등을 기술  
 *   @param   [변수명] 함수의 파라미터 설명  
 *   @return  함수의 return 값 설명  
 */
```

## Doxygen (1/4)



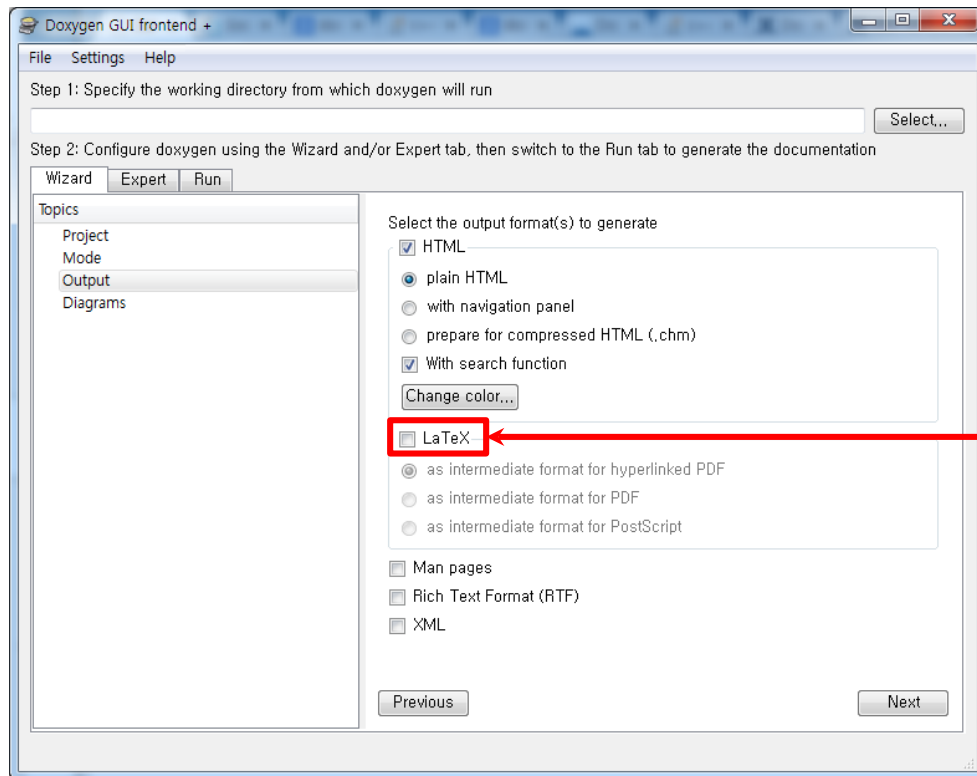
Step 1: Doxygen 으로 문서화 할 프로젝트 경로 삽입

Step 2: Wizard 탭에서 Project name 설정.

Step 3: Wizard 탭에서 Source code directory 에 문서화 할 소스 경로 지정 후 Scan recursively 를 체크. [Step 1] 의 경로와 같은 경로를 삽입해도 됨.

Step 4: Destination directory 에 문서화한 데이터가 저장 될 경로 지정. [Step 1] 의 경로와 같은 경로를 삽입해도 됨.

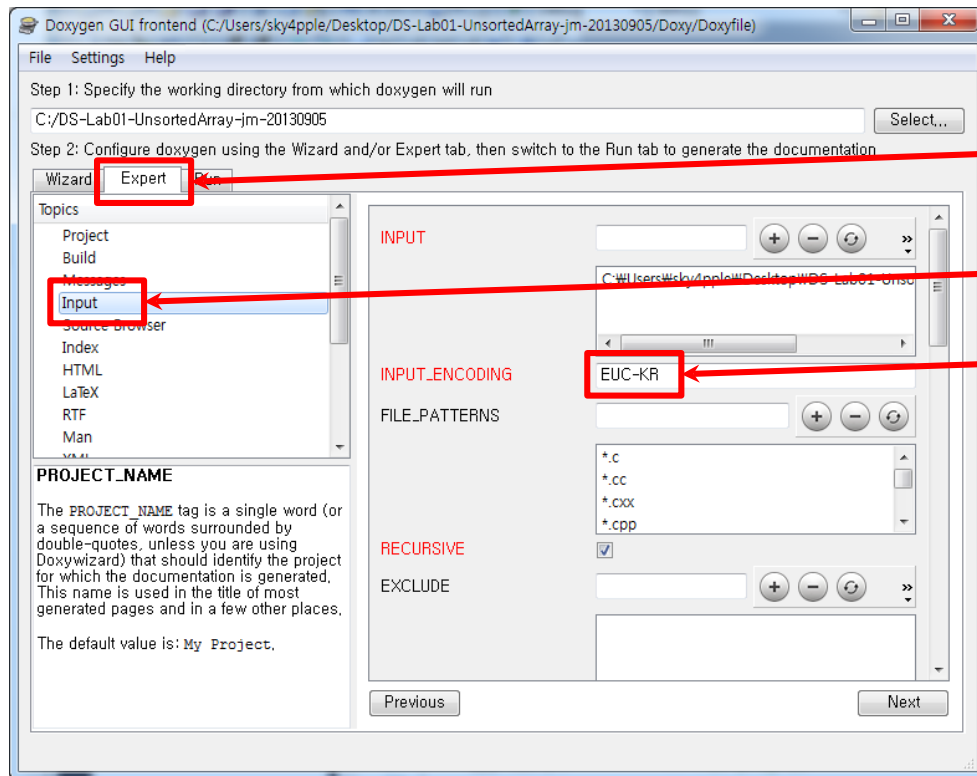
## Doxygen (2/4)



Step 5: LaTeX 체크 해제

## Doxygen (3/4)

- 한글 사용을 위한 추가 설정

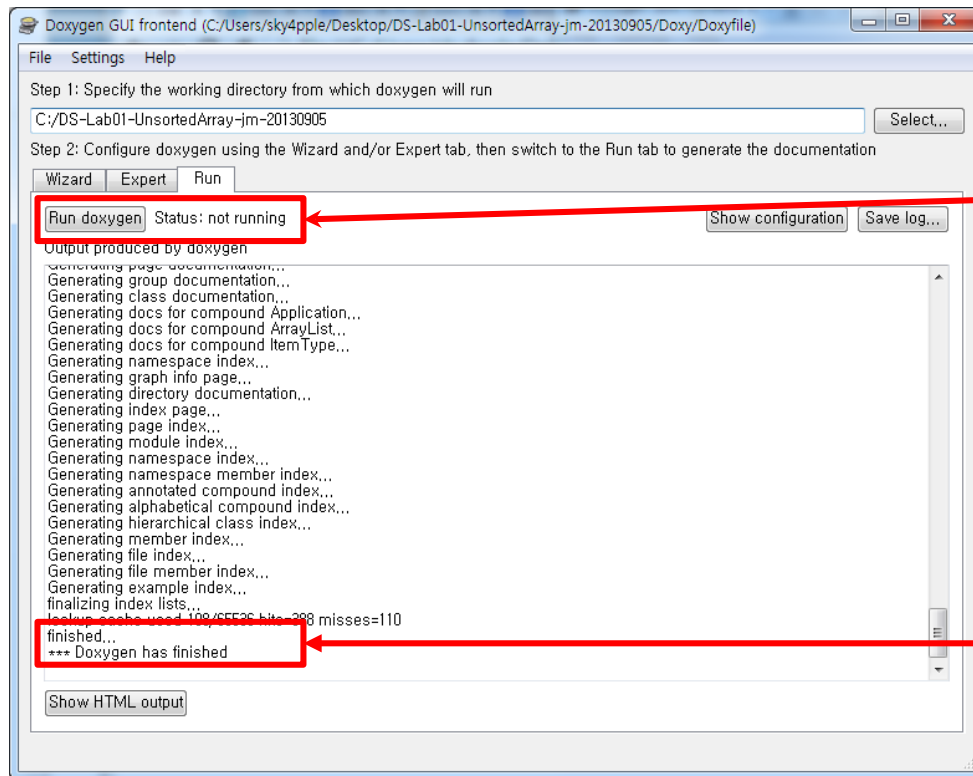


1. Expert 탭으로 이동

2. Input 선택

3. INPUT\_ENCODING의 값을 "EUC-KR"로 변경 후 실행

## Doxygen (4/4)



Step 6: Run doxygen 수행 후, "Doxygen has finished" 확인

## Lab01 예제

◎ 목표: Unsorted List의 ADT 설계 및 구현

◎ 내용:

- ☞ Array를 사용하여 Unsorted List를 구현
- ☞ 과제: iteration을 이용한 삽입, 출력 기능을 갖춘 Unsorted list구현

◎ 방법

- ☞ Array 기반의 Unsorted List 프로그램(Lab01) 분석
- ☞ ADT를 바탕으로 iteration을 이용한 삽입, 출력 기능을 갖춘 Unsorted list를 작성

## 예제: ArrayList

◎ 내용: 다음의 class를 설계

### ☞ Domain:

- ItemType m\_Array[MAXSIZE] // 레코드 배열
- int m\_Length // 리스트에 저장된 레코드 수
- int m\_CurPointer // current pointer

### ☞ Operations:

- void MakeEmpty() // 현재 레코드 모두 삭제
- int GetLength() // 현재 레코드 수 반환
- bool IsFull() // 모든 배열의 사용 여부
- int Add(ItemType data) // 새로운 레코드 추가
- void ResetList() // 레코드 포인터 초기화
- int GetNextItem(ItemType& data) // current pointer를 하나 증가시키고 끝이 아니면 record index를 리턴 끝이면 -1을 리턴



## 예제: ArrayList ADT

```
#define MAXSIZE 5

class ArrayList
{
public:
    ArrayList();           // default constructor
    ~ArrayList();          // default destructor

    void MakeEmpty();      // Make list empty
    int GetLength();       // 레코드 수 반환
    bool IsFull();         // 모든 배열의 사용 여부
    int Add(ItemType data); // 새로운 데이터 추가
    void ResetList();      // 레코드 포인터(current pointer) 초기화
    int GetNextItem(ItemType& data); // current pointer를 하나 증가시키고 끝이 아니면 record index를 리턴 끝이면 -1을 리턴

private:
    ItemType m_Array[MAXSIZE]; // 레코드 배열
    int m_Length;              // 리스트에 저장된 레코드 수
    int m_CurPointer;          // current pointer
}
```

## 예제: ItemType ADT

```
enum RelationType {LESS, GREATER, EQUAL};

class ItemType
{
public:
    ItemType(); // default constructor
    ~ItemType(); // default destructor

    int GetId(); // 학생 ID 반환 함수
    string GetName(); // 학생 이름 반환 함수
    string GetAddress(); // 학생 주소 반환 함수
    void SetId(int inId); // 학생 ID 저장 함수
    void SetName(string inName); // 학생 이름 저장 함수
    void SetAddress(string inAddress); // 학생 주소 저장 함수
    void SetRecord(int inId, string inName, string inAddress); // 학생 정보 저장 함수
    void DisplayIdOnScreen(); // 학생 ID 출력 함수
    void DisplayNameOnScreen(); // 학생 이름 출력 함수
    void DisplayAddressOnScreen(); // 학생 주소 출력 함수
    void DisplayRecordOnScreen(); // 학생 정보 출력 함수
};
```

## 예제: ItemType ADT

```
void SetIdFromKB();           // 키보드로 학생 ID 입력 함수
void SetNameFromKB();        // 키보드로 학생 이름 입력 함수
void SetAddressFromKB();     // 키보드로 학생 주소 입력 함수
void SetRecordFromKB();     // 키보드로 학생 정보 입력 함수
int ReadDataFromFile(ifstream& fin); // 학생 정보를 파일에서 읽는 함수
int WriteDataToFile(ofstream& fout); // 학생 정보를 파일로 출력하는 함수
RelationType CompareByID(const ItemType &data); // primary key (ID)를 기준으로 학생 정보를 비교하는 함수
```

Private:

```
int m_Id;           // 학생 ID
string m_sName;     // 학생 이름 저장 변수
string m_sAddress;  // 학생 주소 저장 변수
}
```

## 예제: console

© UnSorted List를 테스트할 driver를 다음과 같이 작성됨

```
--- ID - Command ----  
1 : Add item  
2 : Print all on screen  
3 : Make empty list  
4 : Get from file  
5 : Put to file  
0 : Quit
```

```
Choose a Command --> _
```

## Lab01 과제

### ◎ 내용

- ☞ 다음 ItemType, ArrayList의 ADT를 사용하여 학생 관리 프로그램 작성
- ☞ 클래스명과 파일명을 통일하여 작성
- ☞ 주석처리 꼭 할 것
  - Doxygen 형식에 맞게 작성
  - 반드시 한글로 작성할 것!!
  - 주석 없거나 Doxygen 결과 파일(html 폴더)이 없을 경우 0점
- ☞ Application class는 예제 파일을 참조하여 작성
- ☞ Main 함수에서는 Application 객체를 선언 후 활성화만 시킬 것
  - 예제소스 참조

## 과제: ArrayList ADT

```
#define MAXSIZE 5
```

```
class ArrayList
```

```
{
```

```
public:
```

```
    ArrayList();
```

```
// default constructor
```

```
    ~ArrayList();
```

```
// default destructor
```

```
    void MakeEmpty();
```

```
// Make list empty
```

```
    int GetLength();
```

```
// 레코드 수 반환
```

```
    bool IsFull();
```

```
// 배열이 가득 찼는지 여부
```

```
    int IsEmpty();
```

```
// 배열이 비었는지 여부
```

```
    int Add(ItemType data);
```

```
// 새로운 레코드 추가
```

```
    void ResetList();
```

```
// 레코드 포인터(current pointer) 초기화
```

```
    int GetNextItem(ItemType& data);
```

```
// current pointer를 하나 증가시키고 끝이 아니면 record index를 리턴 끝이면 -1을 리턴
```

```
    int Get(ItemType& data);
```

```
// Primary key를 기준으로 데이터를 검색하고 해당 데이터를 가져옴
```

```
    int Delete(ItemType data);
```

```
// 기존 레코드 삭제
```

```
    int Replace(ItemType data);
```

```
// 기존 레코드 갱신
```

```
private:
```

```
    ItemType m_Array[MAXSIZE];
```

```
// 레코드 배열
```

```
    int m_Length;
```

```
// 리스트에 저장된 레코드 수
```

```
    int m_CurPointer;
```

```
// current pointer
```

```
}
```