

라이브러리

학습내용

- 라이브러리 개요
- 사용자 정의 라이브러리

학습목표

- 라이브러리의 정의를 이해하여 표준 라이브러리를 사용할 수 있다.
- 사용자 정의 라이브러리를 만들고 프로그램에 적용할 수 있다.

1

- 🕕 라이브러리 이해
 - 1 정의
 - 01 서브루틴이나 함수들이 저장된 파일들의 모음
 - 02 보통 컴파일된 형태인 목적코드 형태로 존재
 - 03 프로그램에서공통사용가능한가능을포함하고있는오브젝트파일
 - 04 사용자의 프로그램과 링크되어, 실행이 가능한 완전한 프로그램
 - 1 사용 이유



자주 사용되는 특정한 기능을 main 함수에서 분리해 놓음

프로그램 유지

쉬운 디버깅 컴파일 시간 단축



1 라이브러리 이해

2 분류

정적 라이브러리

• 프로그램을 컴파일하여 생성되는 바이너리에 적재

공유 라<mark>이브러리</mark>

- 프로그램이 시작될 때 적재
- 하나의 프로그램에서 적재 후 사용하면 동일 라이브러리를 사용하는 다른 프로그램이 이용

동적 라이브러리

- 바이너리가 실행 후 적재
- 플러그인 모듈 구현에 적합
- 라이브러리 개선 등에 효과적인 형태(정적은 재컴파일)



- 1 작업에 필요한 라이브러리 함수 검색
- 2 에더파일과 함수 원형 파악
- 3 include와 매개변수 기술

C Standard Library Reference Site

https://www.tutorialspoint.com/c_standard_library/https://www.programiz.com/c-programming/library-

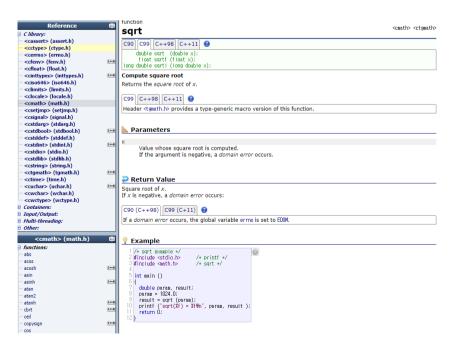
function

http://www.cplusplus.com/reference/clibrary/



ex) 제곱근을 구하는 연산을 수행하는 경우 http://www.cplusplus.com/reference/clibrary/





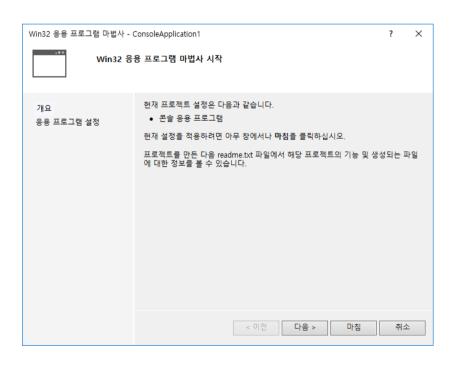


3 C 표준 라이브러리

```
/* sqrt example */
#include \( stdio.h \> /* printf */
#include \( \text{math.h} \rangle \) \( \text{sqrt */} \)
int main ()
 double param, result;
 param = 1024.0;
 result = sqrt (param);
 printf ("sqrt(%f) = %f\foralln", param,
result);
 return 0;
}
```

사용자 정의 라이브러리

- 🕕 Visual Studio에서 만들기
 - 01 정적 라이브러리 프로젝트로 생성
 - 02 헤더파일과 소스코드를 작성
 - 03 lib 빌드

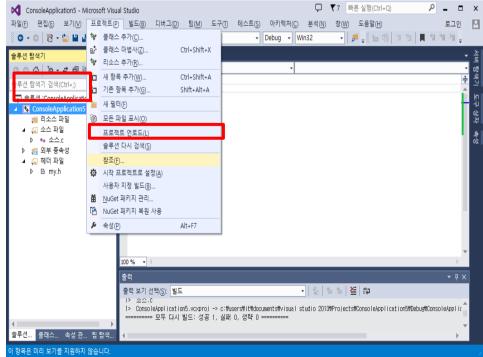


사용자 정의 라이브러리



Visual Studio에서 만들기

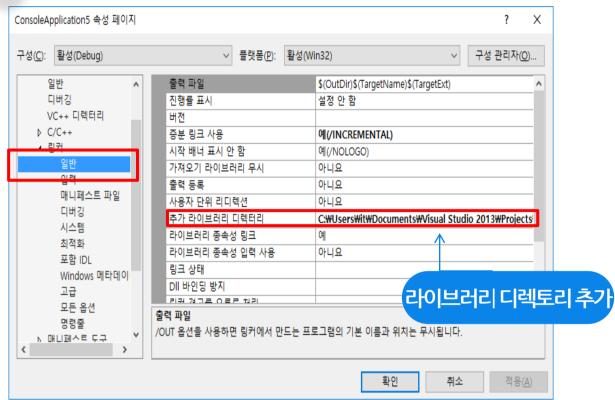


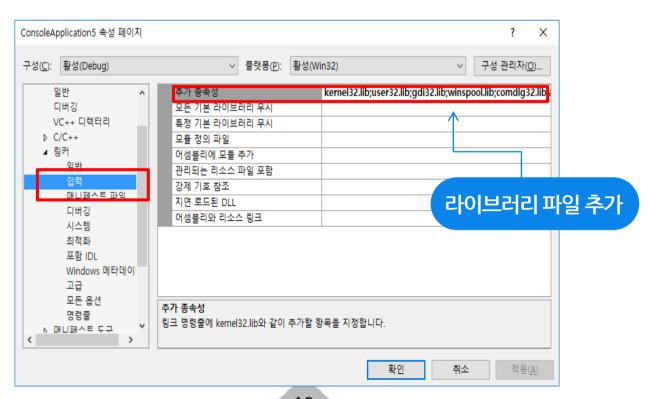


최적화 작업하기



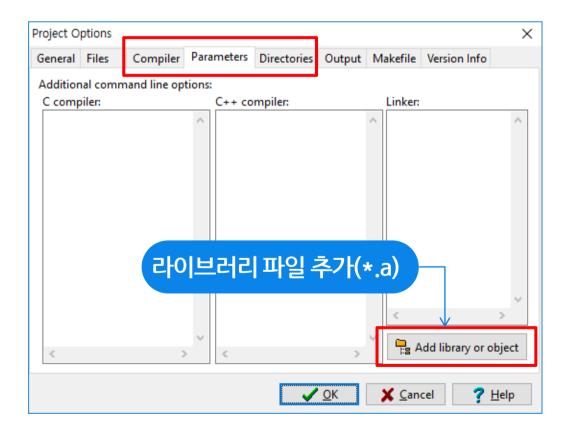
Visual Studio에서 만들기





최적화 작업하기

- ② DEV C++에서 만들기
 - 1 (정적 라이브러리 프로젝트 생성
 - 2 (헤더파일과 소스코드를 작성
 - 3 오브젝트 파일 생성
 - 4 .a 파일 생성



학습정리

1. 라이브러리 개요

- •라이브러리는 함수나 모듈의 집합
- •프로그램 개발 시 자주 사용되는 코드를 라이브러리화하여 재사용할 수 있음
- •정적 라이브러리는 컴파일 시 적재되는 라이브러리
- •공유라이브러리는 하나의 프로그램에서 적재되어 사용되면 이후 필요로 하는 다른 프로그램에서 바로 사용하는 것이 가능함
- •동적 라이브러리는 프로그램이 실행되고 적재되는 라이브러리

2. 사용자 정의 라이브러리

- •프로그래머가 필요에 의해 라이브러리를 만들 수 있음
- •모듈을 만들어 컴파일 후 라이브러리화하여 다른 프로그램에서 호출하여 사용할 수 있음