

예외처리

예외처리(Exception Handling)란?

- ▶ 언어상의 문법적인 오류가 아닌, 프로그래머가 의도한, 구현한 프로그램의 논리에 맞지 않는 상황에 대해 특별한 처리를 하는 것
- ▶ 조건문(if)를 통하여 예외 처리를 하는 것도 좋지만 이 것이 예외 처리를 위한 코드인지, 프로그램의 논리를 위한 코드인지 구분이 힘든 단점이 있다

예외처리

- ▶ try
 - 내부의 내용에서 예외가 있는지 확인
- ▶ throw
 - catch로 예외를 던져준다
- ▶ catch
 - throw에서 던져준 값으로 예외에 대한 처리 진행

try 내부의 throw이후는 건너뛴다(skip)

예외처리

```
int main()
{
    std::cout << "==== start ====" << std::endl;
    try
    {
        int a, b;
        std::cout << "분자(a):";
        std::cin >> a;
        std::cout << "분모(b):";
        std::cin >> b;

        if (0 == b) throw b;

        std::cout << "a : " << a << ", b : " << b << std::endl;
        std::cout << "a / b : " << a / b << std::endl;
    }
    catch (int ex)
    {
        std::cout << "Exception!!!" << std::endl;
        std::cout << ex << " is can't used value!!" << std::endl;
    }

    std::cout << "==== End ====" << std::endl;
}
```

예외처리

- ▶ `catch`내부에 특별히 작업을 종료하지 않으면 그 이후의 내용은 정상적으로 처리된다
- ▶ `throw`가 발생하지 않으면 `catch`는 처리되지 않고 `throw`이 후의 내용은 정상적으로 처리 된다
- ▶ `catch (std::exception ex)`
 - `std::exception`은 프로그램에서 지원하는 함수 등에서 발생한 에러를 받아준다
 - `ex.what()`로 발생한 예외의 내용을 알 수 있다