자바에서는 실행 시 발생할 수 있는 프로그램 오류를 에러와 예외로 구분한다.

에러 : 프로그램 코드로 수습될 수 없는 심각한 오류

예외 : 프로그램 코드에 의해 수습될 수 있는 다소 미약한 오류

-정의 : 프로그램 실행 시 발생할 수 있는 예외의 발생에 대비한 코드 작성

-목적 : 프로그램의 비정상 종료를 막고, 정상적인 실행상태를 유지하는 것

-모든 예외의 최고조상은 Exception클래스

-예외 클래스들은 1)RuntimeException클래스와 그 자손클래스들

2)Exception클래스와 그 자손클래스들 로 두 개의 그룹으로 나눠질 수 있다.

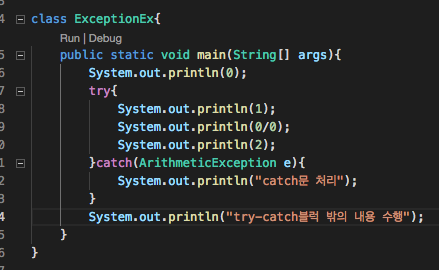
-RuntimeException 클래스들은 프로그래머 실수로 발생될 수 있는 예외들임

-Exception 클래스들은 프로그램 사용자의 실수같은 외부의 영향으로 발생함.

-예외처리를 위해서는 try-catch문을 사용한다.

Try블럭 내에서 예외가 발생하면, 발생한 예외와 일치하는 catch블럭이 있으면 수행하고 catch블럭이 없으면 예외를 처리하지 못함.

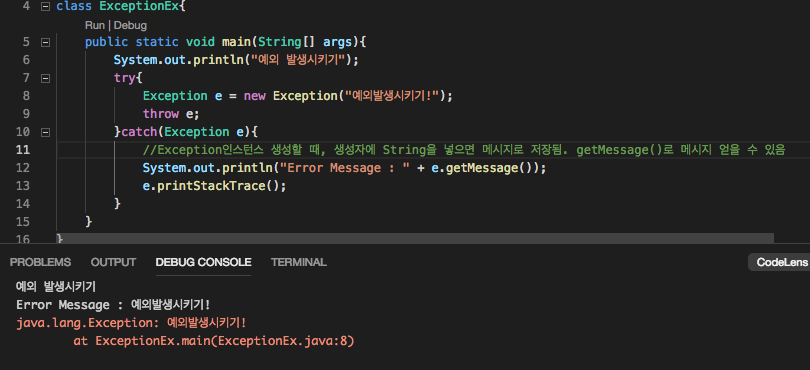
Try블럭 내에 예외가 발생하지 않았으면 catch문은 수행될 필요 없이 전체 블럭을 빠져나감



(Arithmetic Exception 발생시)

-예외 발생시키기

throw를 이용해서 프로그래머가 고의로 예외를 발생시킬 수 있음.

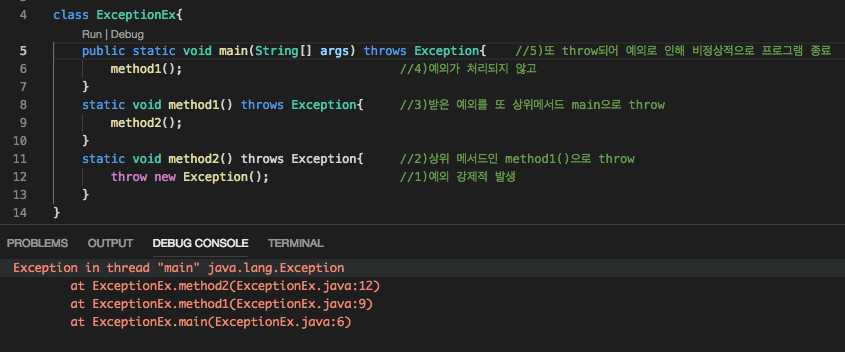


-메서드에 예외 발생시키기

\*\*예외를 발생시키는 키워드 throw

\*\*예외를 메서드에 선언할 때 쓰는 키워드throws

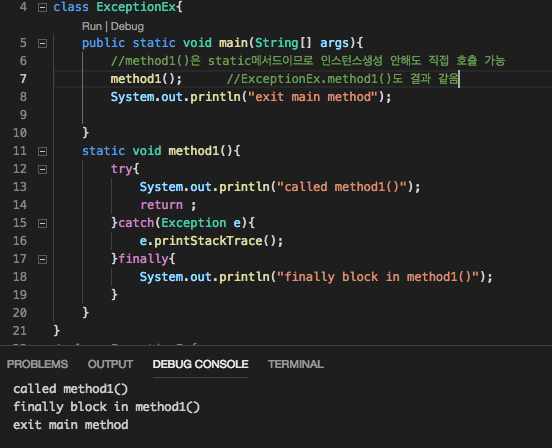
(예외가 발생하면 처리를 상위메서드로 넘김)



-finally 블럭

-try~catch~finally순이며 예외의 발생여부에 상관없이 실행시켜야 하는 메서드가 있을때 사용한다.

-try에서 예외가 발생하면 catch블럭을 지나 finally블럭을 수행하고, try블럭에서 예외가 발생하지 않더라도 finally블럭은 무조건 수행하고 블럭 밖을 수행할 수 있다.



-예외 되던지기(Exception re-throwing)

예외가 여러개 발생하면 몇 개는 메서드 내에서 자체적으로 처리하고, 나머지는 메서드의 선언부에 지정하여 호출한 메서드에서 처리하게 해서, 예외되던지기를 통해 예외를 나눠서 처리할 수 있다.

-하나의 예외에 대해서 예외가 발생한 메서드와 이를 호출한 메서드 모두에서 처리해줘야 할 작업이 있을때 사용됨.

-예외가 발생할 메서드에는 try-catch문으로 예외처리를 해주고, 선언부에 발생할 예외를throws로 지정해줘야함

