-에러(Error) : 코드상의 문제로 컴파일이 불가능한 상태를 말함.(번역 X)

-예외(Exception) : 정상적으로 실행되었으나, 실행중 입력값 등에 의해 발생하는 상태.  
 ex) 배열의 범위가 벗어남. 형변환 타입이 정상적이지 않음 등

-예외처리 : 예외 발생 시 해당시점에서 프로그램 안정성을 위해 처리하는 방법  
1) 기본 형태 : try ~ catch ~ (finally) \*finally는 의미가 없어서 안씀  
 try {  
 내용1; 내용1을 시도한다.  
 } catch(예외클래스 변수명) { 만약 예외클래스에 해당하는 예외가 발생하면  
 내용 2; 내용 2를 실행한다.  
 }  
 \*try catch에서 catch가 여러 개 존재할 경우 예외는 좁은 범위부터 추가하여야 함.  
 (Exception - NumberFormatException 순서로 사용 불가능. 반대는 가능.)  
  
2) try 안에 throw 들어올 경우 : 뒤에 오는 객체에 해당하는 예외를 강제 발생  
 throw new ArrayIndexOutOfBoundsException();  
  
3) throws : 발생할 수도 있는 예외에 대한 정의. 보통 자바외적인 것(디비 등)과 연결처리할 때 사용.  
 DB연결 실패와 같은 상황 발생 시 해당 부분에 대해 예외처리를 해주어야   
 문제가 발생하지 않음. 오버라이딩시에는 권한 자체가 좁아질 수 있음.  
  
  
 접근권한 반환타입 메소드명(인자타입 인자명, …) throws 예외클래스, …. {  
 내용  
 }

printStackTrace() -> java에서 발생한 예외에 대한 내용을 출력하기를 요청(자바)  
toString() -> 예외 발생 내용을 문자열로 취득(사용자가 요청)