



08장(예외처리)

01. 예외처리

에러(error)

- 프로그램 코드에 의해서 수습될 수 없는 심각한 오류

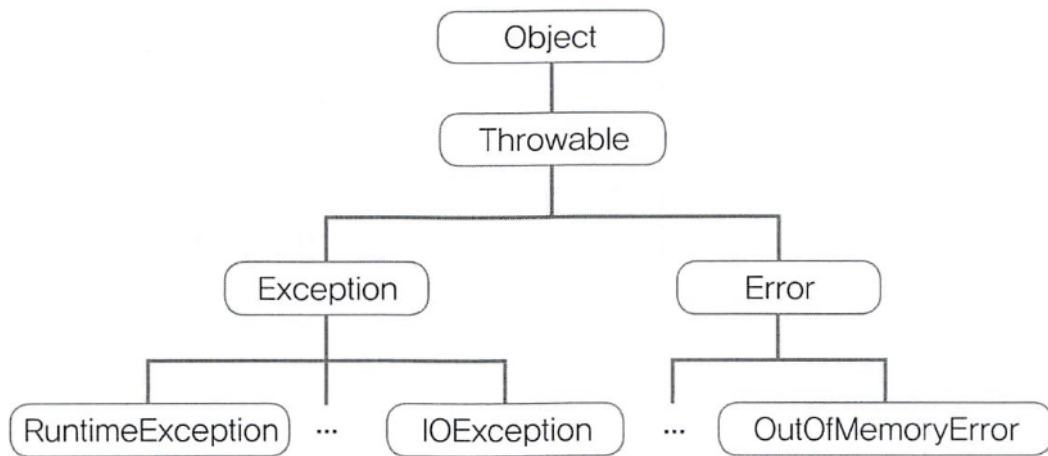
에러의 종류

컴파일 에러	컴파일 시에 발생하는 에러
런타임 에러	실행 시에 발생하는 에러
논리적 에러	실행은 되지만, 의도와는 다르게 동작하는 것

예외(exception)

- 프로그램 코드에 의해서 수습될 수 있는 다소 미약한 오류

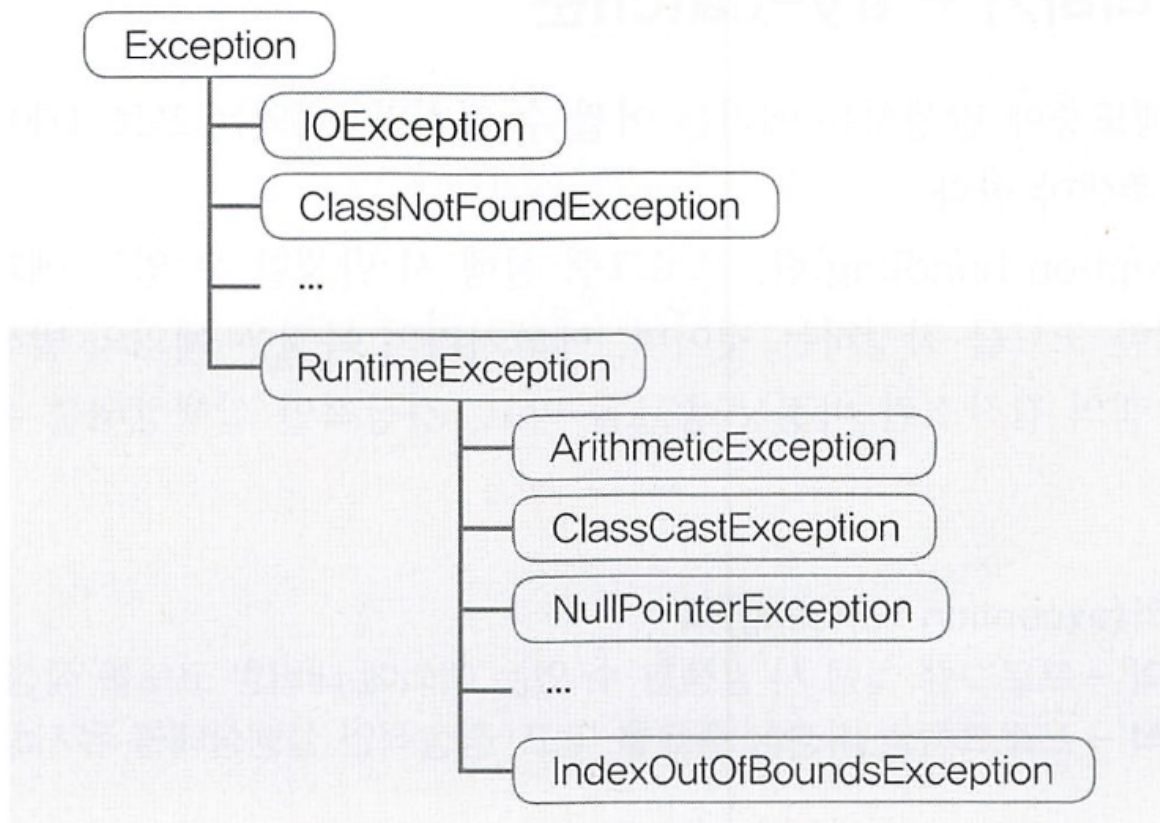
예외 클래스의 계층구조



▲ 그림 8-1 예외클래스 계층도

Exception클래스와 RuntimeException클래스 중심의 상속계층도

- **Exception클래스들** : 사용자의 실수와 같은 외적인 요인에 의해 발생하는 예외
- **RuntimeException 클래스들** : 프로그래머의 실수로 발생하는 예외



▲ 그림 8-2 Exception클래스와 RuntimeException클래스 중심의 상속계층도

try - catch문

- 실행 도중 발생하는 예외를 프로그래머가 미리 처리하기 위해 사용
- **하나 이상의 catch블럭을 만들 수 있음**
- 위에서부터 아래로 예외클래스의 인스턴스에 **instanceof 연산자**를 이용해 검사

```
try{
    ...
}catch(Exception e1){
    ...
}catch(Exception e2){
    ...
}
```

멀티 catch블럭

- **‘|’ (파이프라인) 을 이용하여 중복된 코드를 줄일 수 있음**
- 갯수의 제한이 없음

```
try{
    ...
}catch(ExceptionA | ExceptionB e){
    ...
}
```

예외 발생시키기

- **throw예약어**를 사용하여 프로그래머가 고의로 예외를 발생 시킬 수 있다.
- 인스턴스생성시 ()에 **String**을 적으면 **getMessage()** 를 이용하여 해당 내용을 얻을 수 있음

```
//연산자 new를 이용하여 발생시키려는 예외클래스의 객체를 만들
Exception e = new Exception("어쩌라고");
//키워드 throw를 이용해서 예외를 발생시킴
throw e;
```

TIP!!

- 컴파일러가 예외처리를 확인하지 않는 **RuntimeException**클래스들은 '**unchecked 예외**'라고 부름
- 컴파일러가 예외처리를 확인하는 **Exception**클래스들은 '**checked예외**'라고 부름

메서드에 예외선언

- 메서드에서 예외가 발생시 메서드 명 옆에 **throws**를 사용해서 예외를 선언할 수 있음
- **오버라이딩**할 때에는 단순히 선언된 예외의 개수가 아니라 상속관계까지 고려해야함
- **throws**를 사용하는 것은 예외를 처리하는 것이 아닌, 그 예외를 호출한 곳으로 책임전가 시키는 것

finally블럭

- **try - catch** 문에 마지막에 **finally** 블럭을 추가하는 것
- **예기치 않게 프로그램이 터지지 않는이상 finally는 무조건 수행된다.**
- 소스를 닫는용으로 많이 사용

try-with-resources문

- **try**문에 () 를 사용하여 생성한 소스를 해당 코드 종료시 자동으로 **close()** 시킴

사용자정의 예외 만들기

- 보통 **Exception**클래스, **RuntimeException**클래스를 상속받아 만들지만, 필요에 따라 알맞게 선택 가능

예외 되던지기(exception re=throwing)

- 예외를 처리한 후에 인위적으로 다시 발생시키는 방법
- 하나의 예외에 대해서 예외가 발생한 메서드와 이를 호출한 메서드 양쪽 모두 처리해야 할 작업이 있을 때 사용
- 주의사항

- 예외발생할 메서드에 **try - catch문** 사용해 예외 처리
- 예외 처리 함과 동시에 메서드의 선언부에 발생할 예외를 **throws**에 지정해줘야 함

연결된 예외(chained exception)

- **try - catch 문**에서 예외 발생
- 발생한 예외를 원인으로 다시 지정
- 다른 원인을 발생
- **throw**로 예외 던짐

```

1  try{
2      startInstall();
3      copyFile();
4  }catch(SpaceException e){
5      InstallException ie = new InstallException("설치중 예외발생"); //예외 생성
6      ie.initCause(e);    //InstallException 예외 원인을 SapceException으로 지정
7      throw ie; //InstallException 발생시킴
8  }

```

- **Throwable initCause(Throwable cause)** → 지정한 예외를 원인 예외로 등록
- **Throwable getCause()** → 원인 예외를 반환
- 여러가지 예외를 하나의 큰 부류의 예외로 묶어서 다루기 위해 사용
- **checked예외**를 **unchecked예외**로 처리하기 위해서도 사용