JAVA 프로그래밍

Chapter 11 예외 처리

목차

- 자바의 에러
- 예외 처리
- 예외 클래스들
- 사용자 정의 예외 클래스

자바의 에러

- ❖ 컴파일 타임 에러(Compile-Time Error)
  - 자바의 문법적 오류로 컴파일이 되지 않는 구문상의 오류.

- ❖ 실행 타임 에러(Run-Time Error)
  - 컴파일은 되지만 실행이 되지 않는 로직(Logic) 상의 오류.
  - Error 와 Exception으로 구분

- ❖ 예외(exception)란?
  - 예외(exception)란 컴퓨터 시스템이 동작하는 도중에 예상하지 못한 오류가 발생하는 것을 의미합니다.
  - 이렇게 발생한 예외 상황은 실행되고 있던 프로그램을 비정상적으로 종료시킵니다.
  - 따라서 예외 처리(exception handling)를 통해 이러한 예외 상황을 처리할 수 있도록 코드의 흐름을 바꾸는 행위가 필요합니다.
  - 자바는 언어 차원에서 예외 처리를 설정할 수 있는 문법을 제공하고 있습니다.

예외 처리

❖ 예외 처리(exception handling)

- 프로그램 실행 시 발생할 수 있는 예외의 발생에 대비한 코드를 작성하여 프로그램의 비정상적인 종료를 막고 정상적인 수행을 할 수 있도록 하기 위함.
- 자바에서는 프로그램이 실행되는 도중 발생하는 예외를 처리하기 위해 try-catch-finally 구문을 사용합니다.

```
try{
    // 에러 감지블록
    // 예외 발생 예상 지역
}catch(FileNotFoundException e){
    // 에러 발생시 처리내용;
}finally{
    // 예외가 발생하든 발생하지 않든 처리할 내용;
}
```

- 1. try 블록 : 기본적으로 맨 먼저 실행되는 코드로 여기에서 발생한 예외는 catch 블록에서 처리됩니다.
- 2. catch 블록 : try 블록에서 발생한 예외 코드나 예외 객체를 인수로 전달받아 그 처리를 담당합니다.
- 3. finally 블록 : 이 블록은 try 블록에서 예외가 발생하건 안 하건 맨 마지막에 무조건 실행됩니다.

예외 처리

❖ 자바의 예외처리(Exception Handling)

- try ~ catch ~ finally : 예외 처리
- throw : 프로그래머에 의해 예외 강제 발생
- throws : 예외를 발생시킨 메소드에서 처리하지 않고 호출한 메소드로 예외를 전가

예외의 종류

❖ Checked Exception

- 컴파일러에 의해 체크가 되는 익셉션 .
- 에러 발생의 가능성이 높은 곳에 에러 처리를 의무화 시키기 위함.
- RuntimeException 하위 클래스를 제외한 Exception 클래스의 하위 클래스들.

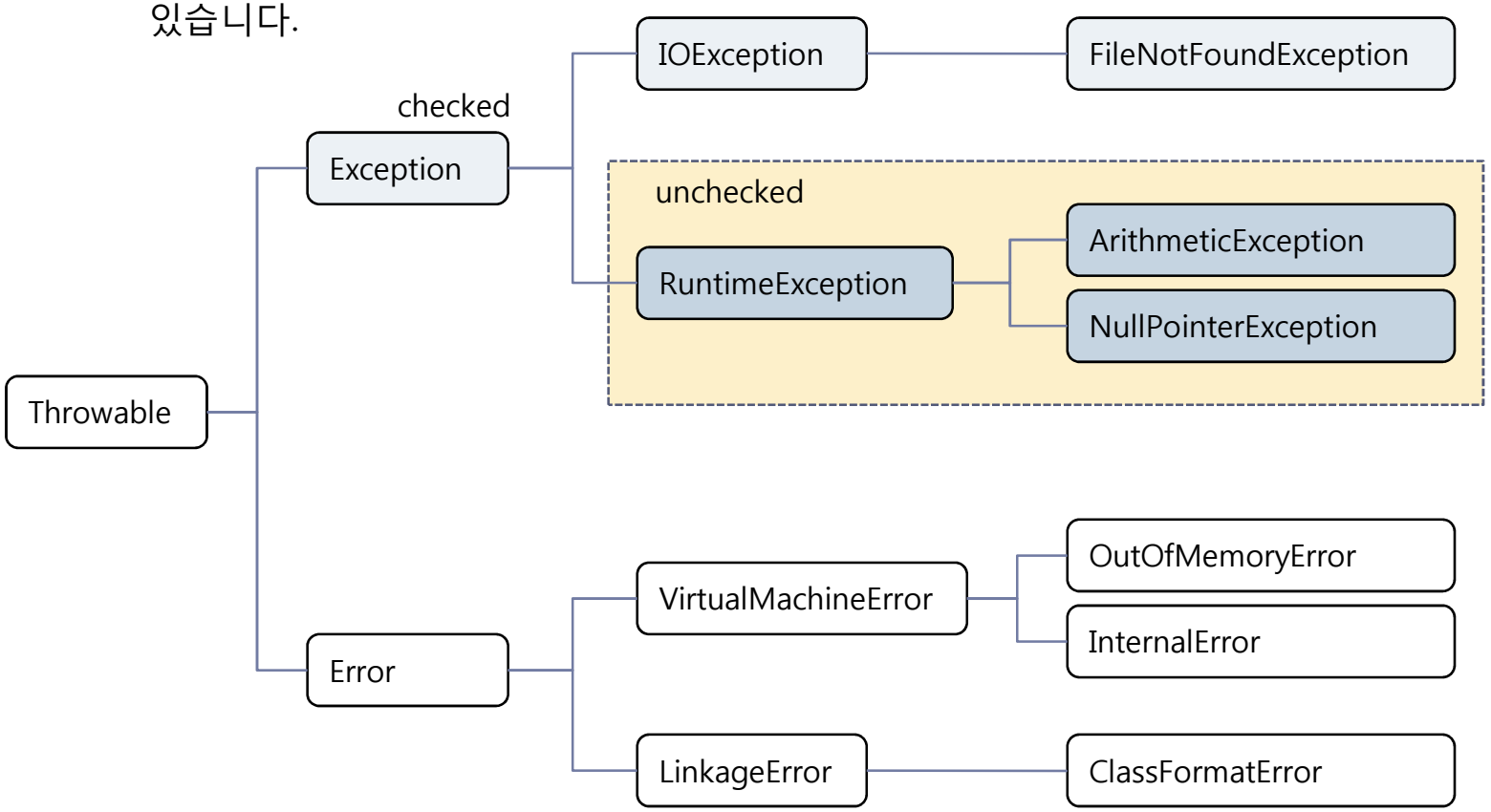
❖ Unchecked Exception

- 런타임시에 체크되는 익셉션.
- RuntimeException 하위 클래스들.

예외 클래스들

❖ Exception 클래스

- 자바에서 모든 예외의 조상 클래스가 되는 Exception 클래스는 크게 다음과 같이 구분할 수 있습니다.



사용자 정의 예외 클래스

❖ 자바에서는 Exception 클래스를 상속받아 자신만의 새로운 예외 클래스를 정의하여 사용할 수 있습니다.

```
class MyException extends Exception{
    MyException(){
        super("내가 만든 에러 입니다.");
    }
    getMessage(){
    }
}
```