Chapter. 13
예외 처리

예외와 예외 처리

FAST CAMPUS
ONLINE
자바 기초

강사. 박은종

Chapter. 13

01 예외와 예외 처리

I 오류란 무엇인가?

컴파일 오류 : 프로그램 코드 작성 중 발생하는 문법적 오류

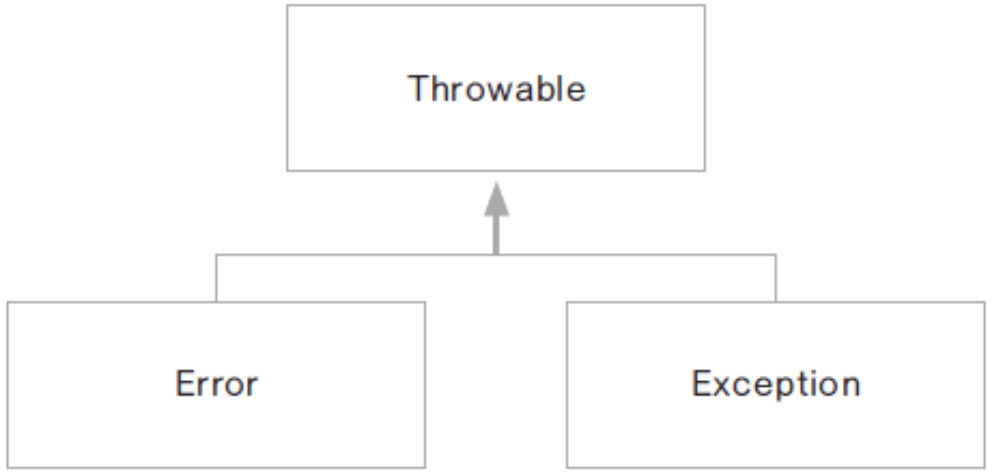
실행 오류 : 실행 중인 프로그램이 의도 하지 않은 동작을 하거나(bug) 프로그램이 중지 되는 오류 (runtime error)

자바는 예외 처리를 통하여 프로그램의 비정상 종료를 막고 log를 남길 수 있음

I 오류와 예외 클래스

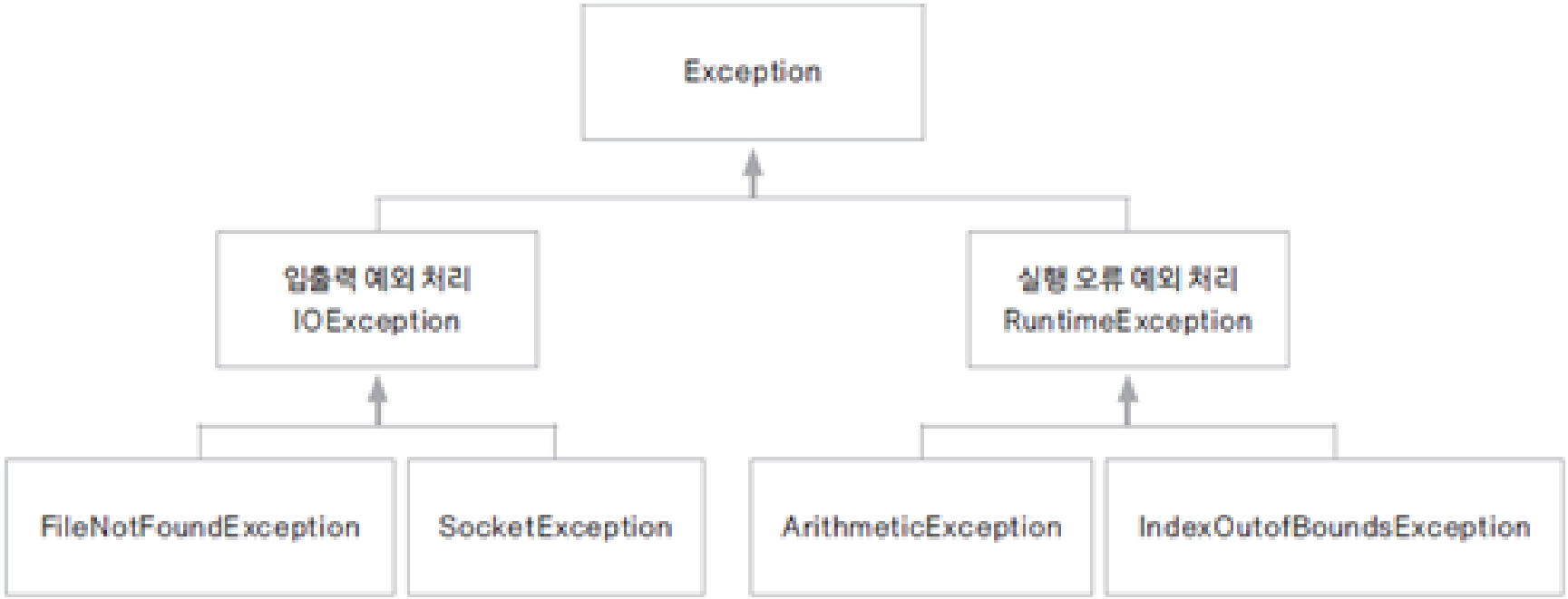
시스템 오류(error) : 가상 머신에서 발생, 프로그래머가 처리 할 수 없음
동적 메모리를 다 사용한 경우, stack over flow등

예외(Exception) : 프로그램에서 제어 할 수 있는 오류
읽으려는 파일이 없는 경우, 네트웍이나 소켓 연결 오류 등
자바 프로그램에서는 예외에 대한 처리를 수행 함



I 예외 클래스

모든 예외 클래스의 최상위 클래스는 Exception 클래스



I try – catch 문으로 예외 처리 하기

```
try {
        예외가 발생 할 수 있는 코드 부분
} catch(처리할 예외 타입 e){
        try블록 안에서 예외가 발생했을 때 수행되는 부분
}
```

try – catch – finally 문으로 예외 처리 하기

```
try {
        예외가 발생 할 수 있는 코드 부분
} catch(처리할 예외 타입 e){
        try블록 안에서 예외가 발생했을 때 수행되는 부분
} finally {
        예외 발생 여부와 상관 없이 항상 수행 되는 부분
        리소스를 정리하는 코드를 주로 씀
}
```

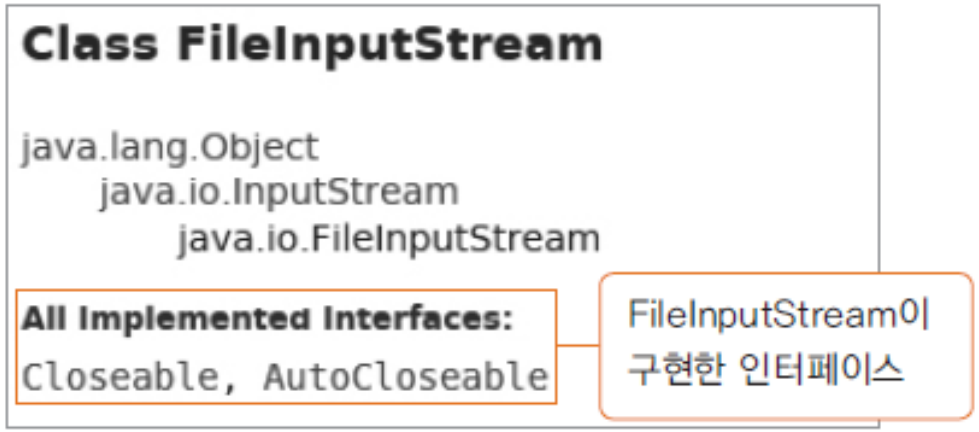
I try – with – resources 문

리소스를 자동으로 해제 하도록 제공해주는 구문

해당 리소스가 AutoCloseable을 구현한 경우 close()를 명시적으로 호출하지 않아도 try{} 블록에서 오픈된 리소스는 정산적인 경우나 예외가 발생한 경우 모두 자동으로 close()가 호출 됨

자바 7부터 제공 됨

FileInputStream 의 경우 AutoCloseable을 구현 하고 있음



AutoCloseable 인터페이스 사용하기

AutoCloseable 인터페이스를 구현한 클래스를 만들고 close()가 잘 호출되는지 확인 해본다

```
public class AutoCloseObj implements AutoCloseable {
  @Override
  public void close( ) throws Exception {
     System.out.println("리소스가 close( ) 되었습니다");
  }
}
```

close( ) 메서드 구현

I 향상된 try – with – resources 문

자바 9에서 제공되는 구문

자바 9 이전

```
AutoCloseObj obj = new AutoCloseObj( );
try (AutoCloseObj obj2 = obj)
  throw new Exception( );
}catch(Exception e) {
  System.out.println("예외 부분입니다");
}
```

다른 참조 변수로 다시 선언해야 함

자바 9 이후

```
AutoCloseObj obj = new AutoCloseObj( );
try(obj) {
  throw new Exception( );
} catch(Exception e) {
  System.out.println("예외 부분입니다");
}
```

외부에서 선언한 변수를 그대로 쓸 수 있음