예외 처리

오류의 종류

에러 : 하드웨어의 오동작 또는 고장으로 인해 응용프로그램 실행 오류가 발생하는 것

JVM 위에서 실행되는 프로그램 수정으로 해결X

예외 : 사용자의 조작이나 잘못된 코딩으로 발생하는 프로그램 오류

프로그램에서 제외할 수 있는 오류

예외(Exception)

자바는 안정성이 중요한 언어로 대부분 프로그램에서 발생하는 오류에 대해

문법적으로 예외를 처리함

자바에서는 예외를 클래스로 관리함

예외 발생 => JVM이 해당 예외 클래스로 객체를 생성

=>예외처리코드에서 예외 객체를 이용할 수 있도록 해줌

모든 예외 클래스는 java.lang.Excepion을 상속받음

일반 예외는 RuntimeException을 상속 받지 않음 : JVM의 실행 예외 판단 여부

일반예외(컴파일 예외)

자바 소스를 컴파일 하는 과정에서 예외 처리 코드가 필요한지 검사하기 때문에

컴파일 오류라고도 함

프로그램 코드 작성 중 발생하는 문법적 오류

ex) IOException, ClassNotFoundException, InterruptedException

실행 예외

자바 컴파일러가 체크하지 않기 때문에 개발자가 직접 예외 처리 코드 삽입해야 함

실행 예외 발생하면 바로 프로그램 종료

ArithmeticException : 정수를 0으로 나눈 경우 발생

NullPointerException: null 값을 갖는 참조 변수로 객체에 접근하려 했을 때

ArrayoutOdBoundsException : 배열에서 인덱스 범위를 초과하여 사용할 경우

NumberFormatException : 포장(Wrapper) 클래스를 이용하여 문자열 숫자를

숫자로 리턴 할 때 숫자로 변환할 수 없는 문자가 있다면 발생

ClassCastException : 상위- 하위 클래스 관계가 아닌 타입변환(Casting)을 하려

하는 경우

예외 처리

try catch finally문

- finally 블럭에서 파일을 닫거나 네트웍을 닫는 등의 리소스 해제 구현을 해야 함

try{

예외가 발생할 수 있는 코드 부분

} catch(처리할 예외 타입 e){

try 블록 안에서 예외가 발생했을 때

예외를 처리하는 부분

} finally{

예외발생 여부와 상관 없이 항상 수행되는 부분

리소스를 정리하는 코드를 주로 씀

}

다중 catch문

catch문은 다중으로 작성 가능함

하위클래스가 위에 위치해야 함.

상위 클래스의 catch 블록이 실행되면 하위 catch블록은 실행되지 않음

자동 리소스 닫기(try-with-resource)

자바7에 추가

예외 발생 여부와 상관 없이 close() 호출

Autocloseable 인터페이스 구현해야 함

java9부터 리소스 try()외부에서 선언하고 변수만 try(obj)와 같이 사용할 수 있음

예외 떠넘기기

throws

리전타입 메소드 throws 예외클래스1, 예외클래스2….

리전타입 메소드 throws Exception으로 한 번에 처리 가능- 좋지 않은 방법

JVM main() myMethod1() mymethod2()

<- <- <- 예외 발생

main()에서 throws 사용하면 JVM에서 처리됨

사용자 정의 예외 클래스

직접 만들어야 하는 예외가 있을 때 사용

일반 예외로 선언할 경우Exception, 실행예외일 때는 RuntimeException

예외 발생시키기

throw

자바 IO 패키지

IO 패키지

(키보드,파일,프로그램) 프로그램 (모니터,파일,프로그램)

<=입력스트림=> <=출력스트림=>

자바의 입출력을 위한 I/O 스트림

java.io에서 제공

분류

바이트 기반 스트림 : 문자, 그림 , 멀티미디어 등 모든 종류 멀티미디어 가능

최상위 클래스

입력스트림 : InputStream

출력스트림 : OutputStream

하위클래스

입력스트림 : FileInputStream , BufferInputStream

출력스트림 : FileOutputStream , BufferOutputStream

문자 스트림 : 문자만, 2바이트씩

최상위 클래스

입력스트림 : Reader

출력스트림 : Writer

하위클래스

입력스트림 : FileReader , BufferReader

출력스트림 : FileWriter , BuffeWriter

InputStream

- int read()

- 입력 스트림으로부터 한 바이트의 자료를 읽어들임

- 읽은 자료의 바이트 수를 반환함

- int read(byte[] b)

- 입력 스트림으로부터 b[] 크기의 자료를 b[]에 읽어들임

- int read(byte[] b, int off, int len)

- 입력 스트림으로부터 b[] 크기의 자료를 off변수 위치부터 len만큼

읽어들임

-void close()

- 입력 스트림과 연결된 대상 리소스 닫음