



Power Java

제15장 예외 처리



© 2009 인피니티박스 All rights reserved



이번 장에서 학습할 내용

- 예외 처리란?
- 예외처리기의 개요
- 예외의 종류
- 예외와 메소드
- 예외 생성하기

예외는 오류가
발생하더라도
오류를
우아하게
처리하게
합니다.

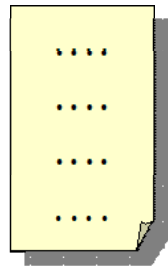


© 2009 인피니티박스 All rights reserved



예외란?

- 예외(exception): 잘못된 코드, 부정확한 데이터, 예외적인 상황에 의하여 발생하는 오류
 - (예) 0으로 나누는 것과 같은 잘못된 연산이나 배열의 인덱스가 한계를 넘을 수도 있고, 디스크에서는 하드웨어 에러가 발생할 수 있다.



실행 오류 발생



예외 객체 발생



예외 처리기

그림 15.1 자바에서는 실행 오류가 발생하면 예외가 생성된다.



예외의 예

```
public class DivideByZero {
    public static void main(String[] args)
    {
        int x = 1;
        int y = 0;
        int result = x / y; // 예외 발생!
        System.out.println("이 문장이 출력될까요?");
    }
}
```

실행되지
않음!

실행결과

Exception in thread "main" java.lang.ArithmeticException: / by zero
at ExceptionTest.main(ExceptionTest.java:5)



중간 점검 문제

1. 예외는 어떤 경우에 발생하는가?
2. 예외를 처리하는 경우와 처리하는 않은 경우를 비교하여 보라. 장점은 무엇인가?.

© 2009 인피니티박스 All rights reserved



예외 처리기의 개요

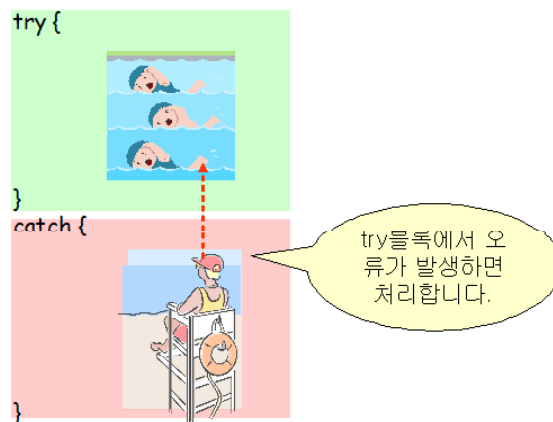


그림 15.2 try블록은 예외가 발생할 수 있는 위험한 코드이다. catch 블록은 예외를 처리하는 코드이다.

© 2009 인피니티박스 All rights reserved



try/catch 블록에서의 실행 흐름

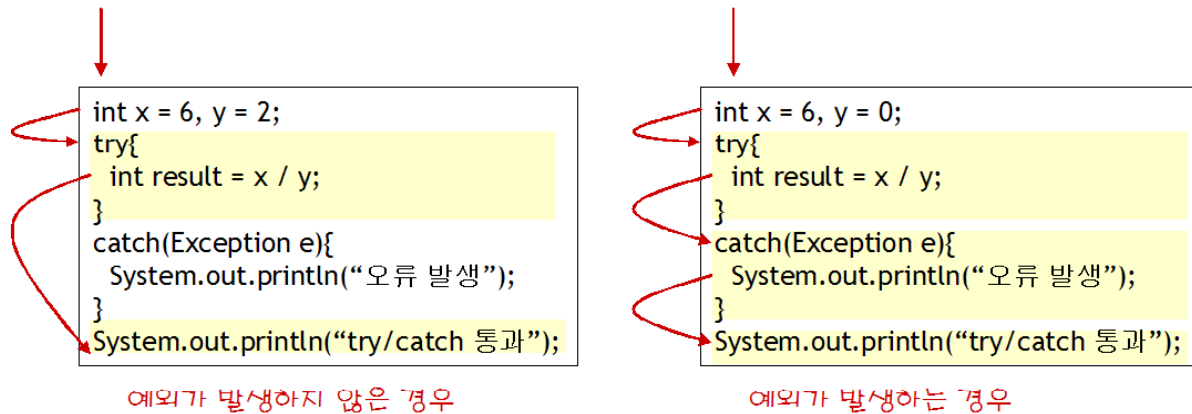


그림 15.3 try/catch 블록에서의 실행 흐름



예제

DivideByZeroOK.java

```

public class DivideByZeroOK {
    public static void main(String[] args) {
        int x = 1;
        int y = 0;
        try {
            int result = x / y;           // 예외 발생!
        } catch (ArithmeticException e) {
            System.out.println("0으로 나눌 수 없습니다.");
        }
        System.out.println("프로그램은 계속 진행됩니다.");
    }
}
    
```

실행결과

0으로 나눌 수 없습니다.
프로그램은 계속 진행됩니다.



예제

InputError.java

실행결과

정수를 입력하시오: one
정수가 아닙니다.

```
import java.util.*;

public class InputError {
    static Scanner scan = new Scanner(System.in);

    public static void main(String[] args) {
        int i = 0;
        System.out.print("정수를 입력하시오: ");
        try {
            i = scan.nextInt();
            System.out.println("입력 받은 정수의 값: " + i);
        } catch (InputMismatchException e) {
            System.out.println("정수가 아닙니다. ");
        }
    }
}
```

© 2009 인피니티박스 All rights reserved



finally 블록

- 오류가 발생하였건 발생하지 않았건 항상 실행되어야 하는 코드는 finally 블록에 넣을 수 있다.

```
try {
    turnTVOn();
    watchTV();
} catch (TVException e) {
    System.out.println("TV를 볼수 없습니다");
} finally {
    turnTVOff();
}
```

© 2009 인피니티박스 All rights reserved



중간 점검 문제

1. 배열에서 인덱스가 범위를 벗어나도 예외가 발생된다. 크기가 10인 배열을 생성하고 11번째 원소에 0을 대입하여 보라. 이 예외를 처리하는 try-catch 블록을 만들어 보라.
2. 1번 문제에 배열 참조 변수에 null을 대입하여 배열을 삭제하는 문장을 finally 블록으로 만들어서 추가하여 보라.

© 2009 인피니티박스 All rights reserved



예외의 종류

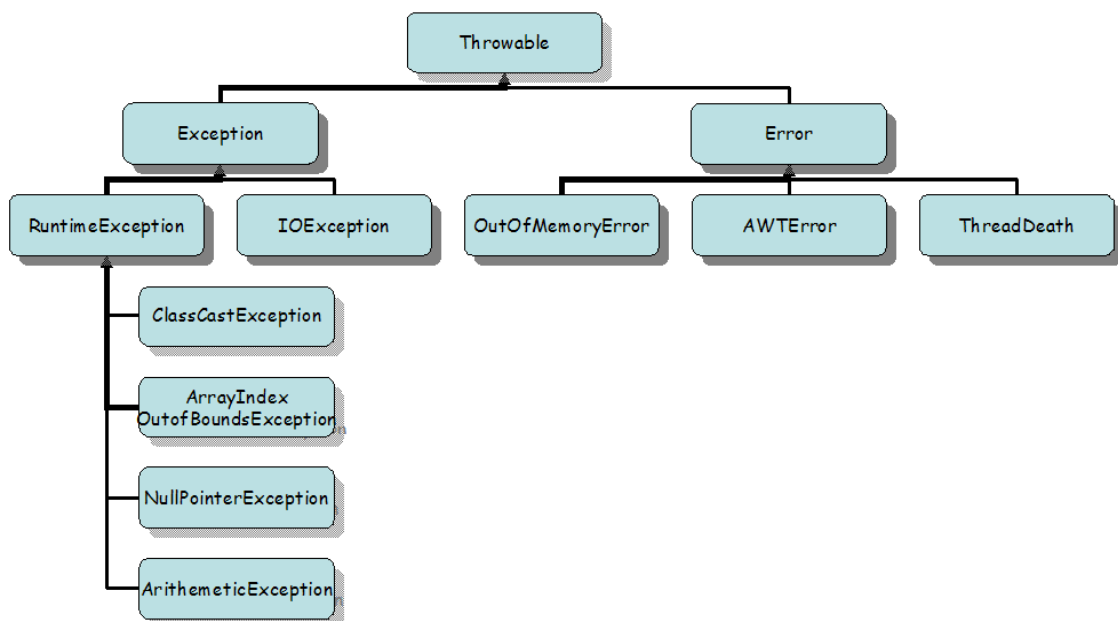
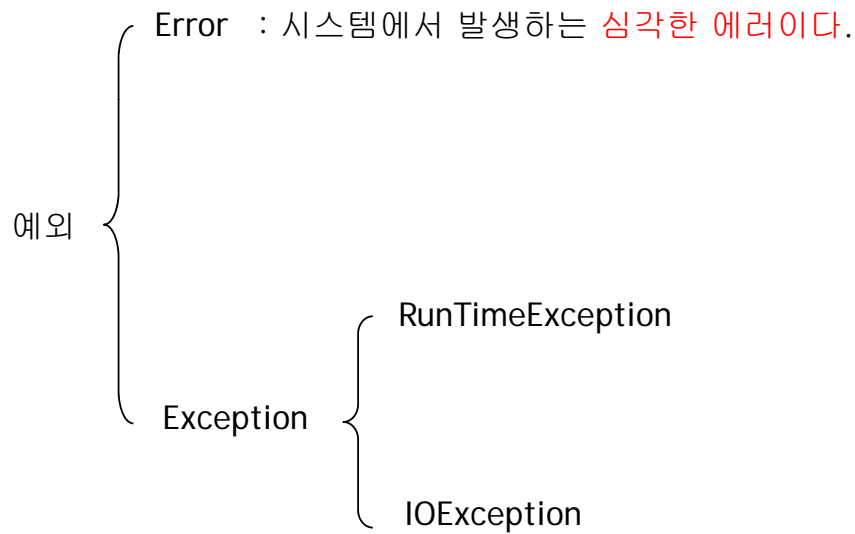


그림 15.4 일부 예외 관련 클래스의 계층도

© 2009 인피니티박스 All rights reserved



Error와 Exception



© 2009 인피니티박스 All rights reserved



RuntimeException

분류	예외	설명
RuntimeException	<u>ArithmeticException</u>	어떤 수를 0으로 나눌 때 발생한다.
	<u>NullPointerException</u>	널 객체를 참조할 때 발생한다.
	<u>IncompatibleClassChangeException</u>	클래스 내부의 변수의 선언이 static 에서 non-static 또는 반대로 변경되었는데 다른 클래스가 이 변수를 참조하는 경우
	<u>ClassCastException</u>	적절치 못하게 클래스를 <u>형변환</u> 하는 경우
	<u>NegativeArraySizeException</u>	배열의 크기가 음수값인 경우
	<u>OutOfMemoryException</u>	사용 가능한 메모리가 없는 경우
	<u>NoClassDefFoundException</u>	원하는 클래스를 찾지 못하였을 경우
	<u>IncompatibleTypeException</u>	인터페이스의 <u>인스턴스</u> 를 생성하려고 시도하는 경우
	<u>ArrayIndexOutOfBoundsException</u>	배열을 참조하는 인덱스가 잘못된 경우
	<u>UnstatisfiedLinkException</u>	자바가 아닌 다른 프로그램 언어로 작성된 <u>메소드</u> 를 연결하는데 실패했을 경우

© 2009 인피니티박스 All rights reserved



IOException

- 입출력과 관련된 오류

IOException	IOException	파일의 끝을 지나쳐서 읽으려고 하는 경우나 잘못된 URL을 사용하는 경우
-------------	-------------	---

© 2009 인피니티박스 All rights reserved



체크 예외와 비체크 예외

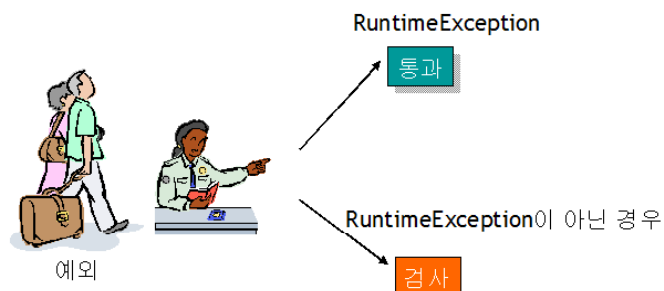


그림 15.5 자바 컴파일러는 RuntimeException은 확인하지 않는다. 그밖의 예외는 모두 처리를 확인한다.

© 2009 인피니티박스 All rights reserved



다형성과 예외

- 다음과 같은 예외 상속 계층도를 가정하자.

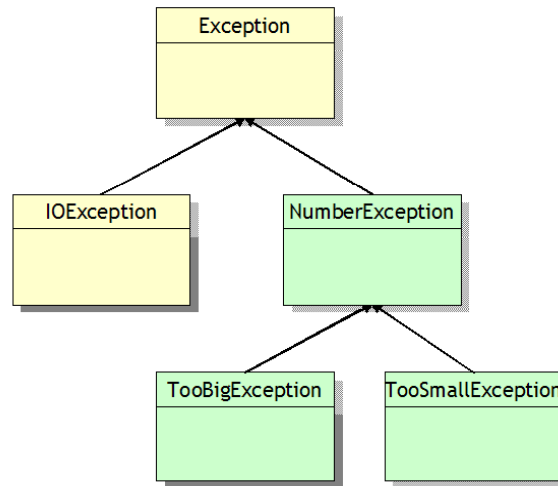


그림 15.5 오류 클래스 상속 계층도

© 2009 인피니티박스 All rights reserved



다형성과 예외

```
try {
    getInput();
}
catch(NumberException e) {
    // NumberException의 하위 클래스를 모두 잡을 수 있다.
}
```

```
try {
    getInput();
}
catch(Exception e) {
    //Exception의 모든 하위 클래스를 잡을 수 있으나 분간할 수 없다.!!
}
```

© 2009 인피니티박스 All rights reserved



다형성과 예외

```
try {
    getInput();
}
catch (TooSmallException e) {
    // TooSmallException만 잡힌다.
}
catch (NumberFormatException e) {
    // TooSmallException을 제외한 나머지 예외들이 잡힌다.
}
```

```
try {
    getInput();
}
catch (NumberFormatException e) {
    // 모든 NumberException이 잡힌다.
}
catch (TooSmallException e) {
    // 아무 것도 잡히지 않는다!
}
```

© 2009 인피니티박스 All rights reserved



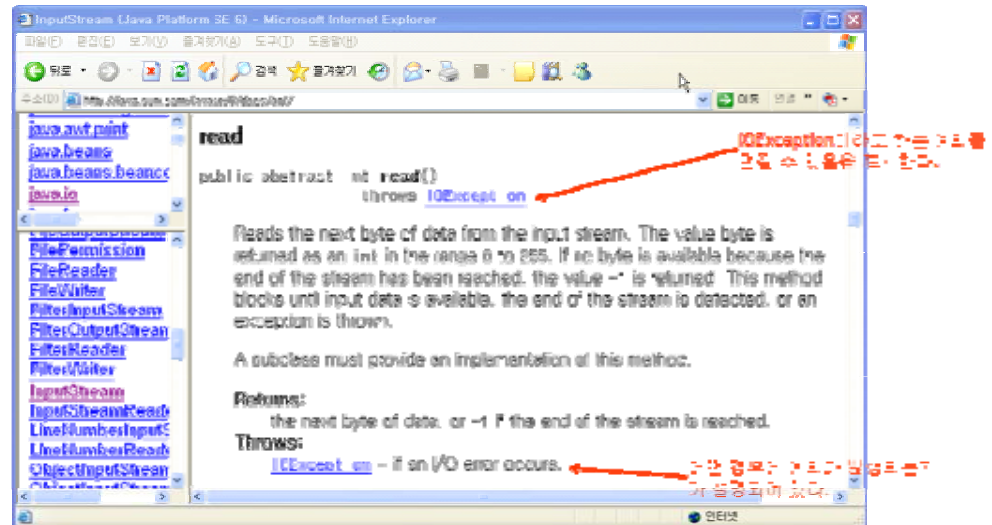
중간 점검 문제

1. Error와 Exception은 어떻게 다른가?
2. 자바 코드에서 반드시 처리하여야 하는 예외는 어떤 것들인가? 어떤 클래스에서 상속되는가?
3. RuntimeException을 처리했는지를 왜 컴파일러에서는 검사하지 않는가?



예외가 발생하는 메소드

- 예외를 발생하는 메소드



- 처리 방법
 - 예외를 try/catch로 처리하는 방법
 - 예외를 상위 메소드로 전달하는 방법

© 2009 인피니티북스 All rights reserved



try/catch 블록 이용

```
import java.io.IOException;

public class Test {
    public static void main(String[] args) {
        System.out.println(readString());
    }

    public static String readString() {
        byte[] buf = new byte[100];
        System.out.println("문자열을 입력하십시오:");
        try {
            System.in.read(buf);
        } catch (IOException e) {
            System.out.println(e.getMessage());
            e.printStackTrace();
        }
        return new String(buf);
    }
}
```

예외를 그 자리
에서 처리

© 2009 인피니티북스 All rights reserved



상위 메소드로 전달

```
import java.io.IOException;

public class Test {
    public static void main(String[] args) {
        try {
            System.out.println(readString());
        } catch (IOException e) {
            System.out.println(e.getMessage());
            e.printStackTrace();
        }
    }

    public static String readString() throws IOException {
        byte[] buf = new byte[100];
        System.out.println("문자열을 입력하시오:");
        System.in.read(buf);
        return new String(buf);
    }
}
```

예외를 상위 메
소드로 전달

© 2009 인피니티박스 All rights reserved



예외 생성하기

- 예외는 throw 문장을 이용하여 생성한다.

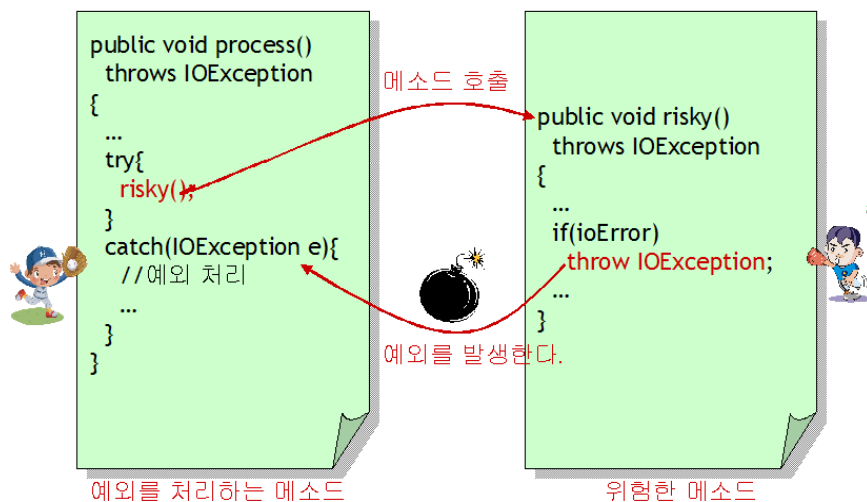


그림 15.7 예외를 던지고 받기

© 2009 인피니티박스 All rights reserved



중간 점검 문제

- 1. 만약 호출한 메소드가 예외를 발생할 가능성이 있다면 어떻게 하여야 하는가?
- 2. 예외를 발생할 가능성이 있는 메소드는 어떻게 정의되는가?
- 3. throws가 의미하는 것은 무엇인가?

© 2009 인피니티박스 All rights reserved



예외 생성의 예

```
void allocateMemory()  
{  
    ...  
    if(memory == null)  
        throw new AllocateMemoryFailException();  
    ...  
}
```

예외 객체 생성

© 2009 인피니티박스 All rights reserved



사용자 정의 예외

- 사용자 정의 예외: Exception 클래스의 서브 클래스를 생성시켜서 만든다.

```
public class MyException extends Exception {
    ...
}
```

© 2009 인피니티박스 All rights reserved



예제

```
class DivideByZeroException extends ArithmeticException {
    public DivideByZeroException()
    {
        super( "0으로 나눌수는 없음." );
    }
}

public class ExceptionTest {
    public static void main(String[] args)
    {
        double result;
        try {
            result = quotient(1,0);
        }
        catch ( DivideByZeroException e ) {
            System.out.println(e.toString());
        }
    }

    public static double quotient( int n, int d )
        throws DivideByZeroException
    {
        if ( d == 0 )
            throw new DivideByZeroException();
        return ( double ) n/ d;
    }
}
```

실행결과

DivideByZeroException: 0으로 나눌수는 없음.



여러 개의 예외를 주고 받는 방법

```
class NetworkFile {
    public void saveFile( ) throws NetworkException, DiskException
    {
        // 네트워크에서 데이터를 받아서 파일에 저장한다.
    }
}
```

```
public class Test {
    public void process() {
        NetworkFile f = new NetworkFile();
        try {
            f.saveFile();
        }
        catch(NetworkException e){
            // 네트워크 오류 처리 코드
        }
        catch(DiskException e){
            // 디스크 오류 처리 코드
        }
    }
}
```

© 2009 www.kyongho.com All rights reserved.



예제

```
class MyException extends Exception {
    public MyException()
    {
        super( "사용자 정의 예외" );
    }
}

public class ExceptionTest1 {
    public static void main( String args[] )
    {
        try {
            method1();
        }
        catch ( MyException e )
        {
            System.err.println( e.getMessage() +
                               "\n호출 스택 내용:" );
            e.printStackTrace();
        }
    }
    public static void method1() throws MyException
    {
        throw new MyException();
    }
}
```

실행결과

사용자 정의 예외

호출 스택 내용:

MyException: 사용자 정의 예외

at ExceptionTest.method1(ExceptionTest.java:17)

at ExceptionTest.main(ExceptionTest.java:5)

© 2



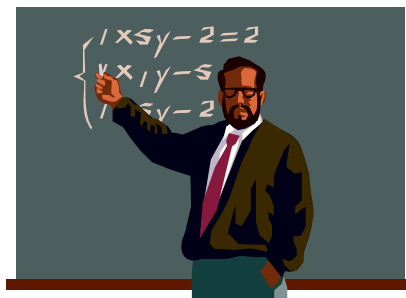
중간 점검 문제

1. `DiskFailureException`을 정의하여 보라. 이 클래스는 매개 변수가 없는 생성자를 가져야 한다. 만약 예외가 매개 변수가 없는 생성자를 이용하여 발생되면 `getMessage()`는 “Disk Failure!”를 반환하여야 한다.
2. 사용자로부터 성적을 입력받아서 평균을 계산하는 프로그램을 작성하여 보자. 만약 사용자가 음수를 입력하면 `NegativeNumberException`을 발생한다. 이 예외를 `catch` 블록으로 잡아서 처리하는 코드도 추가하라.

© 2009 인피니티박스 All rights reserved



Q & A



© 2009 인피니티박스 All rights reserved