

{ JAVA }

선린인터넷

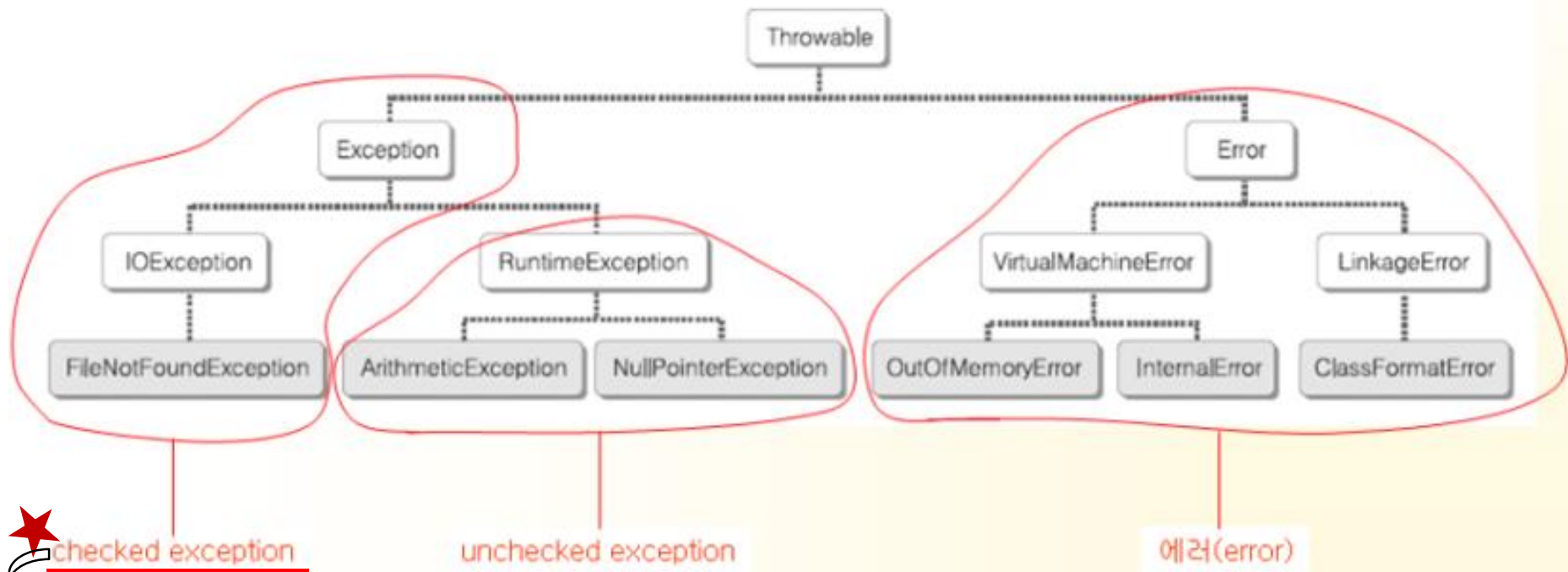
웹 운영 과

예외 처리

Exception Handling

❖ java.lang. Exception 클래스 : Throwable클래스를 상속받음.

- 자바에서 예외 처리를 위해 제공되는 클래스



파일 입출력, 메모리 입출력, 데이터베이스 입출력, 네트워크 입출력 등

학습정리

❖ 예외 Exception 처리방법

- `try ~ catch (~ finally)` 구문

1. 예외 **처리** 코드

- 예외를 잡아주기

- `throws` 구문

2. 예외 처리 **떠넘기기**

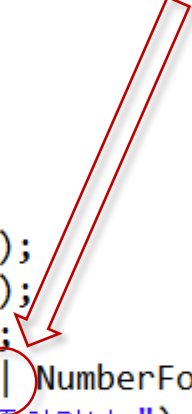
- 예외를 던지기 (책임 미루기)

멀티 try ~ catch 문

❖ 멀티 try ~ catch (멀티 catch)

하나의 catch 블록에서 여러 개의 예외를 처리할 수 있도록 함.
catch 괄호() 안에 동일하게 처리하고 싶은 예외를 **!**로 연결함

```
1 public class MultiCatchExam {  
2     public static void main(String[] args) {  
3         try{  
4             String data1 = args[0];  
5             String data2 = args[1];  
6             int value1 = Integer.parseInt(data1);  
7             int value2 = Integer.parseInt(data2);  
8             System.out.println( value1+value2 );  
9         } catch(ArrayIndexOutOfBoundsException | NumberFormatException e) {  
10            System.out.print("실행 매개변수 수가 부족하거나 ");  
11            System.out.println("숫자로 변환할 수 없습니다.");  
12        }  
13    }  
14 }
```



예외 던지기 Throws

❖ 예외를 던지기!

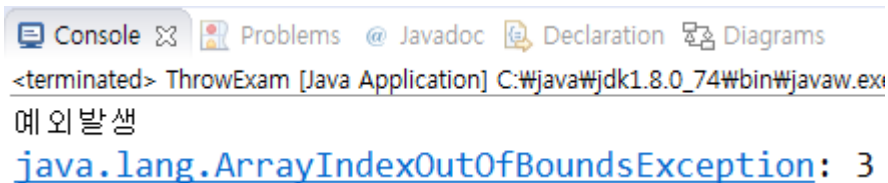
- 위험한 메소드를 호출할 때 컴파일러에게 try~catch로 그 사실을 알려야 하지만, 일단 미뤄 둘 수 있다

```
class ThrowsExam {  
    public static void arrayMethod( ) throws ArrayIndexOutOfBoundsException {
```

throws의 사용

```
        String[ ] name = new String[3];  
        name[3] = "김선린";  
    }
```

```
    public static void main(String[] args) {  
        try { arrayMethod( ); }  
        catch ( ArrayIndexOutOfBoundsException ae ) {  
            System.out.println("예외발생");  
            System.out.println(ae.toString( ));  
        }  
    }  
}
```



Console Problems Javadoc Declaration Diagrams
<terminated> ThrowExam [Java Application] C:\java\jdk1.8.0_74\bin\javaw.exe
예외발생
[java.lang.ArrayIndexOutOfBoundsException: 3](#)

arrayMethod()
main()

main()에게
ArrayIndexOutOfBoundsException
를 던진다!

JVM : 예외 내용을 콘솔에 출력
하는 것으로 처리 - **좋지 않음!**

main()도 예외처리를 던져버리면 어떻게 될까?

```
1 public class ThrowExam {  
2     public static void arrayMethod() throws ArrayIndexOutOfBoundsException {  
3         String[] name = new String[3];  
4         name[3] = "김선린";  
5     }  
6     public static void main(String[] args) throws ArrayIndexOutOfBoundsException {  
7         arrayMethod();  
8     }  
9 }
```

Console Problems Javadoc Declaration Diagrams

<terminated> ThrowExam [Java Application] C:\java\jdk1.8.0_74\bin\javaw.exe (2017. 9. 27. 오전 9:51:22)

Exception in thread "main" [java.lang.ArrayIndexOutOfBoundsException: 3](#)
at ThrowExam.arrayMethod(ThrowExam.java:4)
at ThrowExam.main(ThrowExam.java:7)

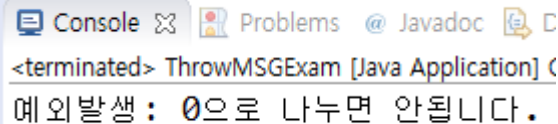
예외 메시지
직접 만들기

❖ 예외 메시지 직접 만들기

- JAVA가 제공하는 예외메시지를 사용하지 않고 직접 만들어 사용할 수 있음

```
throw new Exception("사용자가 만든 오류 메시지");
```

```
1 public class ThrowMSGExam {  
2     public static void main(String[] args) {  
3         int a=100, b=0;  
4         int result;  
5  
6         try{  
7             if(b==0)  
8                 throw new Exception("0으로 나누면 안됩니다.");  
9         } catch(Exception e) {  
10            System.out.print("예외발생: ");  
11            System.out.println( e.getMessage() );  
12        }  
13    }  
14 }
```



Console Problems Javadoc

<terminated> ThrowMSGExam [Java Application] (

예외발생: 0으로 나누면 안됩니다.

(참고)

사용자 정의 예외


Exception Handling

(참고) 사용자 정의 예외

❖ 발생시키려는 예외클래스를 만든다 : 2개의 생성자를 포함하는 게 일반적

- checkedException 예외: **Exception**을 상속
- uncheckedException 예외: **RuntimeException**을 상속

```
public class XXXException extends [Exception | RuntimeException]{  
    public XXXException( ){ }  
    public XXXException(String message){ super(message); }  
}
```



❖ **throw**를 이용해서 예외를 발생시킨다

```
throw new XXXException();  
throw new XXXException( "메세지" );
```

Exception Handling

(참고) 사용자 정의 예외 - 예제

파일명: CustomExam.java

```
1 class MyException extends Exception { //사용자 정의 예외클래스 정의
2     MyException(){
3     }
4     MyException(String message) { // 예외메세지가 있을 때...
5         super( message );
6     }
7 }
8 public class CustomExam {
9     public static void main(String[] args) {
10         try {
11             throw new MyException("홍길동 예외 객체"); //예외 발생
12         } catch(MyException e) {
13             System.out.print("예외 발생: ");
14             System.out.println(e.getMessage()); // 메시지 출력
15         }
16     }
17 }
```

Console Problems Java
<terminated> CustomExam [Java Applic
예외 발생: 홍길동 예외 객체

[실습 예제]

(실습예제)

❖ 사칙연산을 처리하는 간단한 계산기를 작성해보자. **ExceptionEx5.java**

- 단, 다음 3가지 예외상황을 처리하여 적당한 메시지를 발생시켜 보자.

| 사용자 예외상황 | 실행 예시 |
|-------------------|-------------------------------------------------------------------------------------------|
| ① 0으로 나누는 것 | 첫번째 수를 입력하세요 : 2 계산할 연산자를 입력하세요 : / 두번째 수를 입력하세요 : 0 발생 오류 ==> 0으로 나누면 안됩니다. |
| ② 0을 곱하는 것 | 첫번째 수를 입력하세요 : 0 계산할 연산자를 입력하세요 : * 두번째 수를 입력하세요 : 1 발생 오류 ==> 0으로 곱하면 어차피 0입니다. |
| ③ 처리결과가 음수로 나오는 것 | 첫번째 수를 입력하세요 : 1 계산할 연산자를 입력하세요 : - 두번째 수를 입력하세요 : 2 발생 오류 ==> 결과가 음수네요. πππ |

Exception Handling

(실습예제: Hint) – 아래를 참고하여, 완성해봅시다.

```

1 import java.util.Scanner;
2 public class ExceptionEx5 {
3     public static void main(String[] args) throws Exception {
4         Scanner sc = new Scanner(System.in);
5         int num1, num2; // 정수형 변수 2개
6         String op; // 연산자
7         int result = 0; // 계산 및 표현의 편의를 위해 int형으로 합니다.
8
9         System.out.print("첫번째 수를 입력하세요: ");
10        num1 = sc.nextInt();
11        System.out.print("계산할 연산자를 입력하세요: ");
12        op = sc.next(); // +, -, *, / : 4개 중에서만 입력된다고 가정합니다.
13        System.out.print("두번째 수를 입력하세요: ");
14        num2 = sc.nextInt();
15
16        switch(op) {
17            case "+": result = num1 + num2; break;
18            case "-": result = num1 - num2; break;
19            case "*": result = num1 * num2; break;
20            case "/": result = num1 / num2; break;
21        }
22        System.out.println(num1 + op + num2 + "=" + result);
23    }
24 }

```

ExceptionEx5.java

| 사용자 예외상황 | 실행예시 |
|-------------------|-------------------------------------------------------------------------------------------|
| ① 0으로 나누는 것 | 첫번째 수를 입력하세요 : 2 계산할 연산자를 입력하세요 : / 두번째 수를 입력하세요 : 0 발생 오류 ==> 0으로 나누면 안됩니다. |
| ② 0을 곱하는 것 | 첫번째 수를 입력하세요 : 0 계산할 연산자를 입력하세요 : * 두번째 수를 입력하세요 : 1 발생 오류 ==> 0으로 곱하면 여차피 0입니다. |
| ③ 처리결과가 음수로 나오는 것 | 첫번째 수를 입력하세요 : 1 계산할 연산자를 입력하세요 : - 두번째 수를 입력하세요 : 2 발생 오류 ==> 결과가 음수네요.ㅠㅠ |

적절한 예외 처리를 통하여
위 실행결과와 같이 되도록
프로그래밍해봅시다! :-)

2학기 중간고사

화 이팅 ...

