

8강

JAVA





예외 이해하기



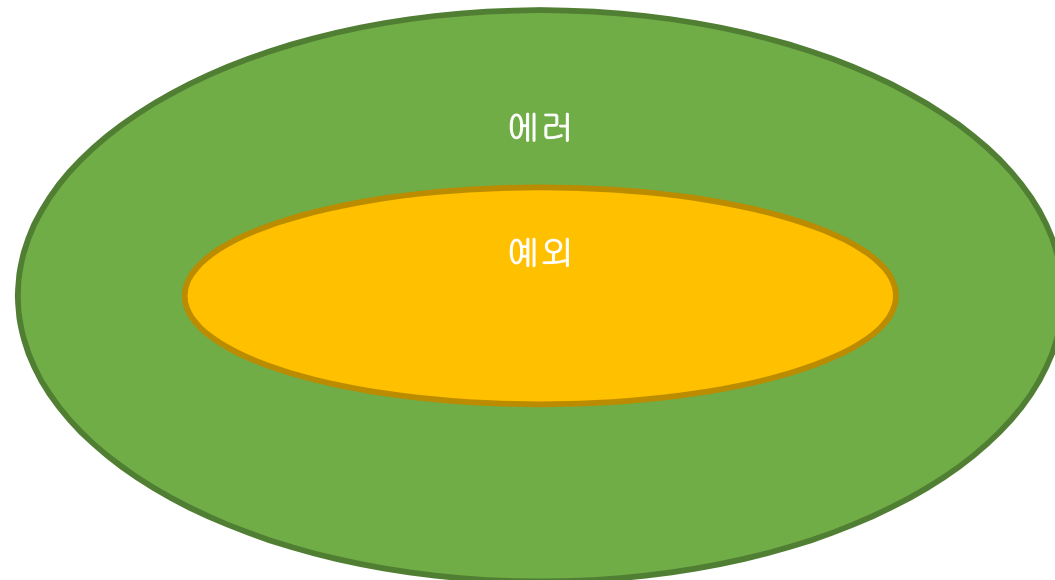
에러 vs 예외

❖ 에러(error)

- 컴퓨터의 연산 처리 결과가 장치의 오동작이나 소프트웨어의 혼란 등으로 목적과 다른 결과가 되는 것

❖ 예외(Exception)

- 컴퓨터 시스템의 동작 도중 예기치 않았던 이상 상태가 발생하여 수행 중인 프로그램이 영향을 받는 것



예외의 종류

❖ 컴파일 예외

- 예외 처리 코드가 없으면 발생
- ClassNotFoundException
- InterruptedException

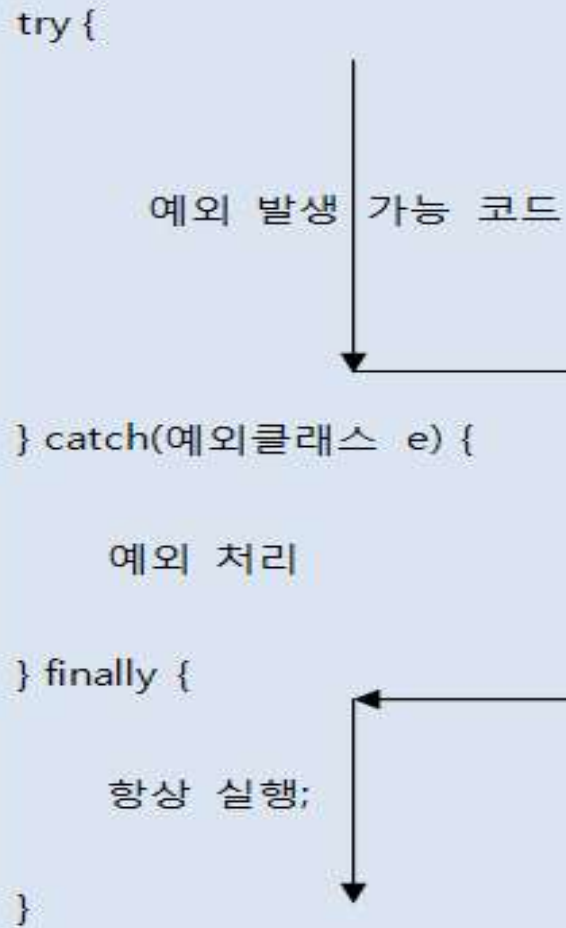
❖ 실행 예외

- 실행 도중 발생
- NullPointerException
- NumberFormatException

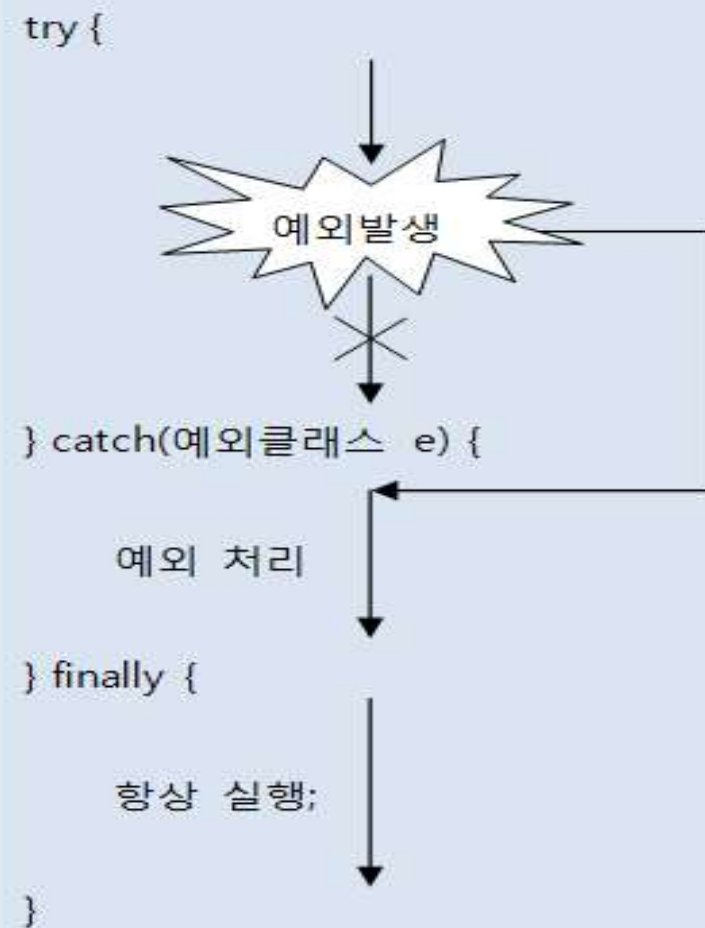
예외처리

예외 코드 처리

정상 실행 되었을 경우

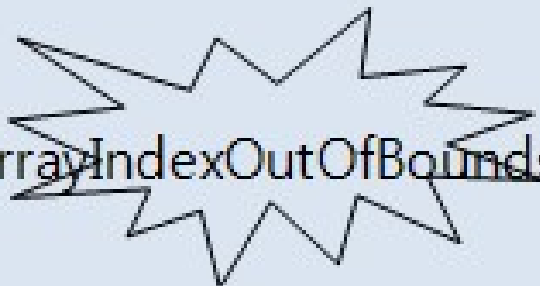


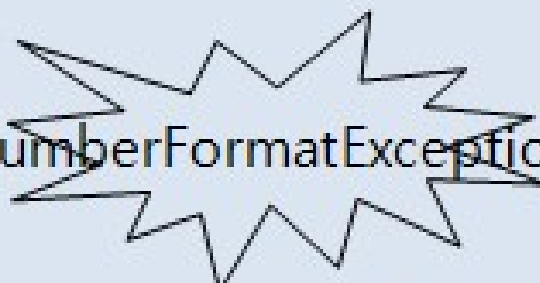
예외가 발생되었을 경우



다중 catch

```
try {
```

 **ArrayIndexOutOfBoundsException** 발생

 **NumberFormatException** 발생

```
} catch(ArrayIndexOutOfBoundsException e) {
```

예외 처리 1

```
} catch(NumberFormatException e) {
```

예외 처리 2

```
}
```

예외 처리 순서

```
try {
```

ArrayIndexOutOfBoundsException 발생

NumberFormatException 발생

```
} catch(Exception e) {
```

예외 처리 1

```
} catch(ArrayIndexOutOfBoundsException e) {
```

예외 처리 2

```
}
```

```
try {
```

ArrayIndexOutOfBoundsException 발생

다른 Exception 발생

```
} catch(ArrayIndexOutOfBoundsException e) {
```

예외 처리 1


```
} catch(Exception e) {
```

예외 처리 2

```
}
```


멀티 catch

```
try {
```

 `ArrayIndexOutOfBoundsException` 또는 `NumberFormatException` 발생

 다른 `Exception` 발생

```
} catch(ArrayIndexOutOfBoundsException | NumberFormatException e) {
```

예외 처리 1

```
} catch(Exception e) {
```

예외 처리 2

```
}
```

throws

```
public void method1() {  
    try {  
        method2();  
    } catch(ClassNotFoundException e) {  
        //예외 처리 코드  
        System.out.println("클래스가 존재하지 않습니다.");  
    }  
}  
  
public void method2() throws ClassNotFoundException {  
    Class clazz = Class.forName("java.lang.String2");  
}
```

호출한 곳에서 예외 처리