



## 10장. 예외처리

이것이 자바다(<http://cafe.naver.com/thisjava>)

# Contents

- ❖ 1절. 예외와 예외 클래스
- ❖ 2절. 실행 예외
- ❖ 3절. 예외 처리 코드
- ❖ 4절. 예외 종류에 따른 처리 코드
- ❖ 5절. 자동 리소스 닫기
- ❖ 6절. 예외 처리 떠넘기기
- ❖ 7절. 사용자 정의 예외와 예외 발생



# 1절. 예외와 예외 클래스

## ❖ 오류의 종류

### ■ 에러(Error)

- 하드웨어나 소프트웨어의 잘못된 동작 또는 고장으로 인한 오류
- 에러가 발생되면 프로그램 종료
- 정상 실행 상태로 돌아갈 수 없음

### ■ 예외(Exception)

- 사용자의 잘못된 조작 또는 개발자의 잘못된 코딩으로 인한 오류
- 예외가 발생되면 프로그램 종료
- 예외 처리 추가하면 정상 실행 상태로 돌아갈 수 있음



# 1절. 예외와 예외 클래스

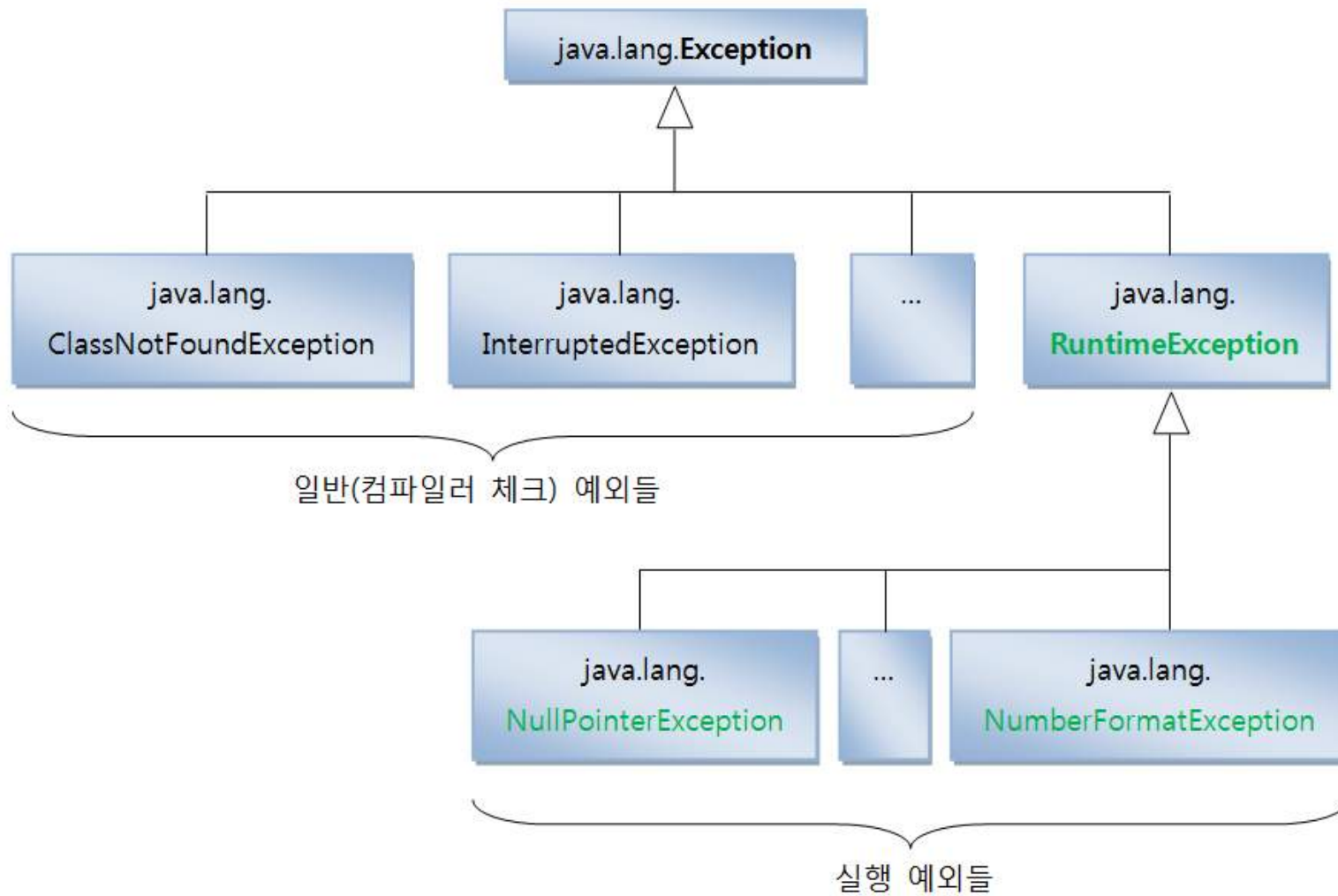
## ❖ 예외의 종류

- 일반(컴파일 체크) 예외(Exception)
  - 예외 처리 코드 없으면 컴파일 오류 발생
- 실행 예외(RuntimeException)
  - 예외 처리 코드를 생략하더라도 컴파일이 되는 예외
  - 경험 따라 예외 처리 코드 작성 필요



# 1절. 예외와 예외 클래스

## ❖ 예외 클래스



# 예제

```
public class TryCatchFinallyExample {  
    public static void main(String[] args) {  
  
        Class clazz = Class.forName("java.lang.String2");  
  
    }  
}
```

## forName

```
public static Class<?> forName(String className)  
    throws ClassNotFoundException
```

Returns the Class object associated with the class or interface with the given string name. Invoking this method is equivalent to:

```
Class.forName(className, true, currentLoader)
```

where currentLoader denotes the defining class loader of the current class.

For example, the following code fragment returns the runtime Class descriptor for the class named java.lang.Thread:

```
Class t = Class.forName("java.lang.Thread")
```

A call to forName("X") causes the class named X to be initialized.

### Parameters:

className - the fully qualified name of the desired class.

### Returns:

the Class object for the class with the specified name.



## 2절. 실행 예외(RuntimeException)

### ❖ NullPointerException

- 객체 참조가 없는 상태

- null 값 갖는 참조변수로 객체 접근 연산자인 도트(.) 사용했을 때 발생

```
String data = null;  
System.out.println(data.toString());
```

### ❖ ArrayIndexOutOfBoundsException (p.424~425)

- 배열에서 인덱스 범위 초과하여 사용할 경우 발생

```
public static void main(String[] args) {  
    String data1 = args[0];  
    String data2 = args[1];  
  
    System.out.println("args[0]: " + data1);  
    System.out.println("args[1]: " + data2);  
}
```



# 예제

```
public class NullPointerExceptionExample {  
    public static void main(String[] args) {  
        String data = null;  
        System.out.println(data.toString());  
    }  
}
```

Exception in thread "main" [java.lang.NullPointerException](#)  
at sec02\_runtime\_exception.NullPointerExceptionExample.main([NullPointerExceptionExample.java:6](#))





# 예제

```
public class ArrayIndexOutOfBoundsExceptionExample {  
    public static void main(String[] args) {  
        String[] input = {"a"};  
  
        String data1 = input[0];  
        System.out.println("input[0]: " + data1);  
        String data2 = input[1];  
        System.out.println("input[1]: " + data2);  
    }  
}
```

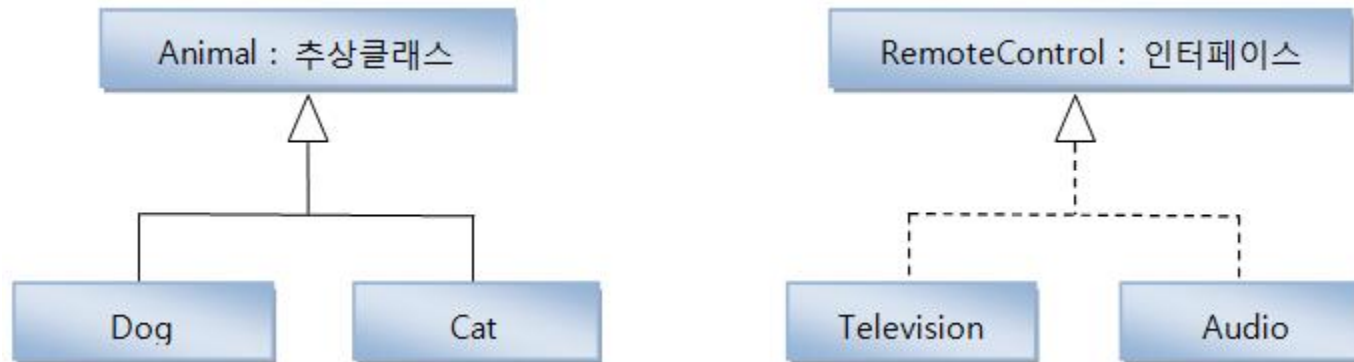
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 1  
at  
sec02\_runtime\_exception.ArrayIndexOutOfBoundsExceptionExample.main(ArrayIndexOutOfBoundsExceptionExample.java:19)



## 2절. 실행 예외(RuntimeException)

### ❖ ClassCastException

- 타입 변환이 되지 않을 경우 발생



#### ■ 정상 코드

```
Animal animal = new Dog();  
Dog dog = (Dog) animal;
```

```
RemoteControl rc = new Television();  
Television tv = (Television) rc;
```

#### ■ 예외 발생 코드

```
Animal animal = new Dog();  
Cat cat = (Cat) animal;
```

```
RemoteControl rc = new Television();  
Audio audio = (Audio) rc;
```



# 예제

```
public class ClassCastExceptionExample {
    public static void main(String[] args) {
        Dog dog = new Dog();
        changeDog(dog);

        Cat cat = new Cat();
        changeDog(cat);
    }

    public static void changeDog(Animal animal) {
        //if(animal instanceof Dog) {
        Dog dog = (Dog) animal;           //ClassCastException 발생 가능
        //}
    }
}

class Animal {}
class Dog extends Animal {}
class Cat extends Animal {}
```

Exception in thread "main" [java.lang.ClassCastException](#): sec02\_runtime\_exception.Cat cannot be cast to sec02\_runtime\_exception.Dog  
at sec02\_runtime\_exception.ClassCastExceptionExample.changeDog([ClassCastExceptionExample.java:14](#))  
at sec02\_runtime\_exception.ClassCastExceptionExample.main([ClassCastExceptionExample.java:9](#))



# 예제

```
public class ClassCastExceptionExample {
    public static void main(String[] args) {
        Dog dog = new Dog();
        changeDog(dog);

        Cat cat = new Cat();
        changeDog(cat);
    }

    public static void changeDog(Animal animal) {

        Dog dog = (Dog) animal;
        //ClassCastException 발생

    }
}

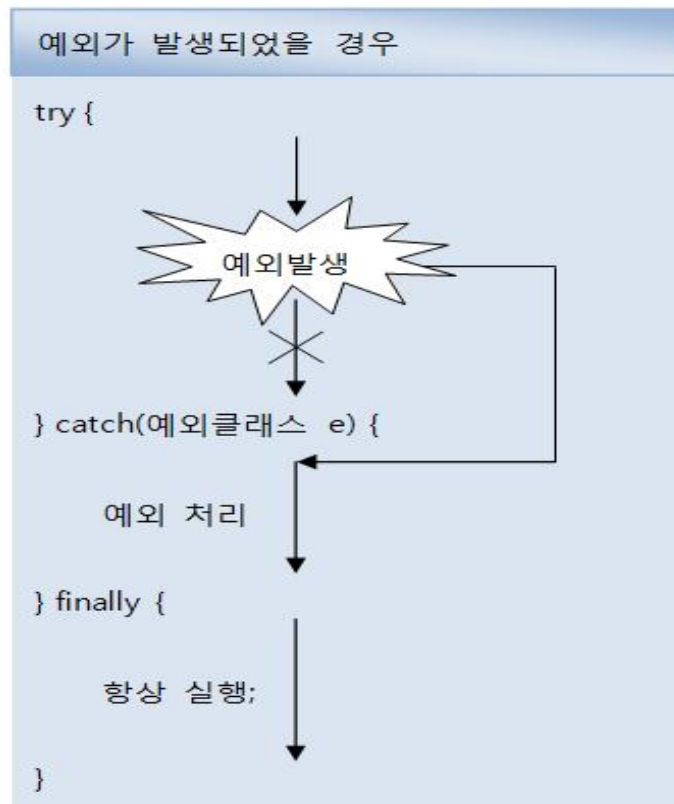
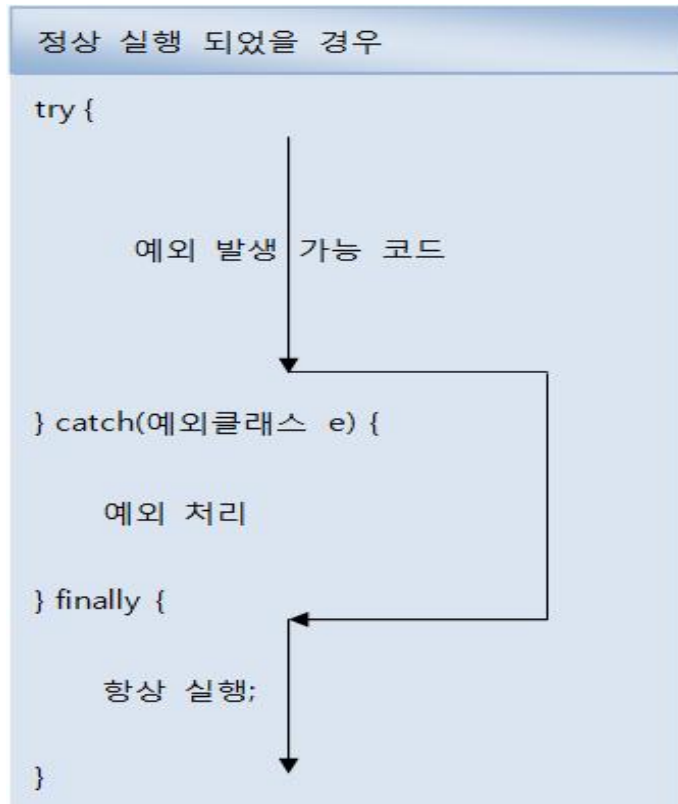
class Animal {}
class Dog extends Animal {}
class Cat extends Animal {}
```

Exception in thread "main" [java.lang.ClassCastException: ch10.exam1\\_runtime\\_exception\\_sol.Cat cannot be cast to ch10.exam1\\_runtime\\_exception\\_sol.Dog](#)  
at [ch10.exam1\\_runtime\\_exception\\_sol.ClassCastExceptionExample.changeDog\(ClassCastExceptionExample.java:14\)](#)  
at [ch10.exam1\\_runtime\\_exception\\_sol.ClassCastExceptionExample.main\(ClassCastExceptionExample.java:9\)](#)

### 3절. 예외 처리 코드(try-catch-finally)

#### ❖ 예외 처리 코드 (p.429~432)

- 예외 발생시 프로그램 종료 막고, 정상 실행 유지할 수 있도록 처리
  - 일반 예외: 반드시 작성해야 컴파일 가능
  - 실행 예외: 컴파일러가 체크해주지 않으며 개발자 경험 의해 작성
- try - catch - finally 블록 이용해 예외 처리 코드 작성



# 예제(일반예외처리)

```
public class TryCatchFinallyExample {  
    public static void main(String[] args) {  
        try {  
            Class clazz = Class.forName("java.lang.String2");  
        } catch(ClassNotFoundException e) {  
            System.out.println("클래스가 존재 하지 않습니다.");  
        }  
    }  
}
```

## forName

```
public static Class<?> forName(String className)  
    throws ClassNotFoundException
```

Returns the Class object associated with the class or interface with the given string name. Invoking this method is equivalent to:

```
Class.forName(className, true, currentLoader)
```

where currentLoader denotes the defining class loader of the current class.

For example, the following code fragment returns the runtime Class descriptor for the class named java.lang.Thread:

```
Class t = Class.forName("java.lang.Thread")
```

A call to forName("X") causes the class named X to be initialized.

### Parameters:

className - the fully qualified name of the desired class.

### Returns:

the Class object for the class with the specified name.

### Throws:

LinkageError - if the linkage fails

ExceptionInInitializerError - if the initialization provoked by this method fails

ClassNotFoundException - if the class cannot be located



```
public class NullPointerExceptionExample {
    public static void main(String[] args) {

        String data = null;
        try
        {
            System.out.println(data.toString());
        }
        catch(NullPointerException e)
        {
            System.out.println("catch : NullPointerException ");
        }
    }
}
```



```
public class ArrayIndexOutOfBoundsExceptionExample {
    public static void main(String[] args) {
        String[] input = {"a"};
        try
        {
            String data1 = input[0];
            System.out.println("input[0]: " + data1);
            String data2 = input[1];
            System.out.println("input[1]: " + data2);
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("catch : ArrayIndexOutOfBoundsException ");
        }
    }
}
```





# 예제(RuntimeException)

```
public class TryCatchFinallyExceptionExample {  
    public static void main(String[] args) {  
        String data1="100";  
        String data2="a100";  
  
        try {  
            int value1 = Integer.parseInt(data1);  
            int value2 = Integer.parseInt(data2);  
            int result = value1 + value2;  
            System.out.println(data1 + "+" + data2 + "=" + result);  
        } catch(NumberFormatException e) {  
            System.out.println("숫자로 변환할 수 없습니다.");  
        } finally {  
            System.out.println("다시 실행하세요.");  
        }  
    }  
}
```



# 예제(RuntimeException)

```
public class ClassCastExceptionExample {
    public static void main(String[] args) {
        Dog dog = new Dog();
        changeDog(dog);

        Cat cat = new Cat();
        changeDog(cat);
    }

    public static void changeDog(Animal animal) {
        if (animal instanceof Dog) {
            Dog dog = (Dog) animal;
            //ClassCastException 발생 가능
        }
    }
}

class Animal {}
class Dog extends Animal {}
class Cat extends Animal {}
```

# 실습

- ❖ String str = “this is a java”
- ❖ String subStr=null;
- 1. substring() 메소드를 이용하여 프로그램을 작성하세요
- 2. beginIndex를 -1로 설정 exception 을 발생시켜 보세요.

## substring

```
public String substring(int beginIndex)
```

Returns a new string that is a **substring** of this string. The **substring** begins with the character at the specified index and extends to the end of this string.

Examples:

```
"unhappy".substring(2) returns "happy"  
"Harbison".substring(3) returns "bison"  
"emptiness".substring(9) returns "" (an empty string)
```

Parameters:

beginIndex - the beginning index, inclusive.

Returns:

the specified **substring**.

Throws:

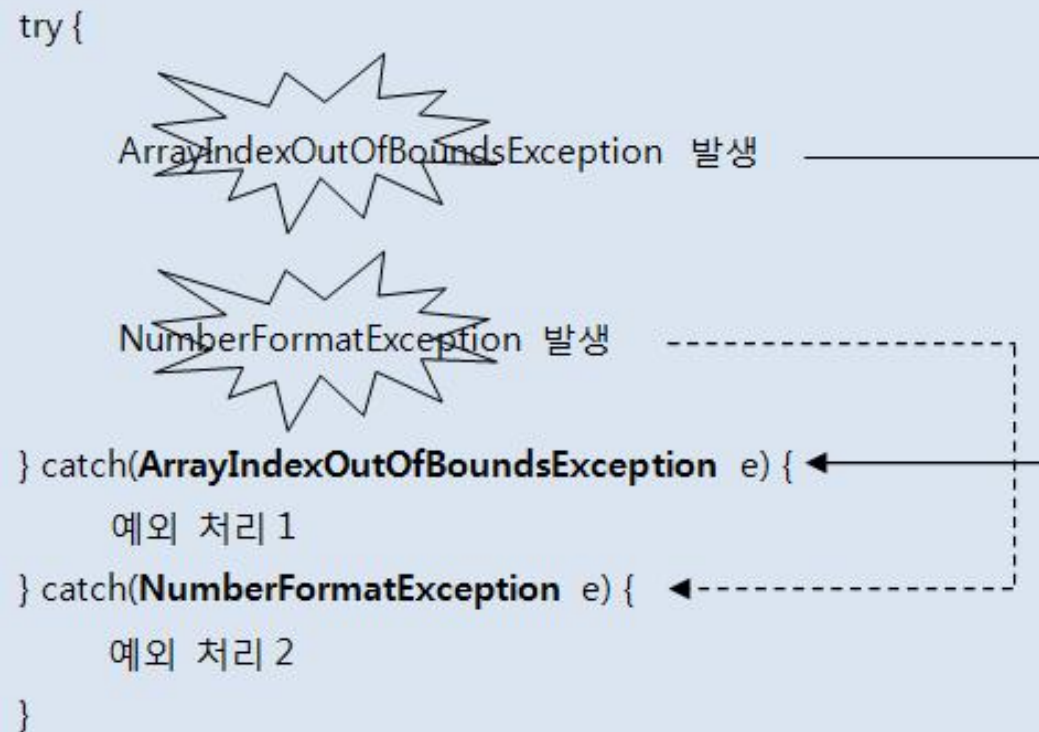
IndexOutOfBoundsException - if beginIndex is negative or larger than the length of this String object.



## 4절. 예외 종류에 따른 처리 코드

### ❖ 다중 catch

- 예외 별로 예외 처리 코드 다르게 구현



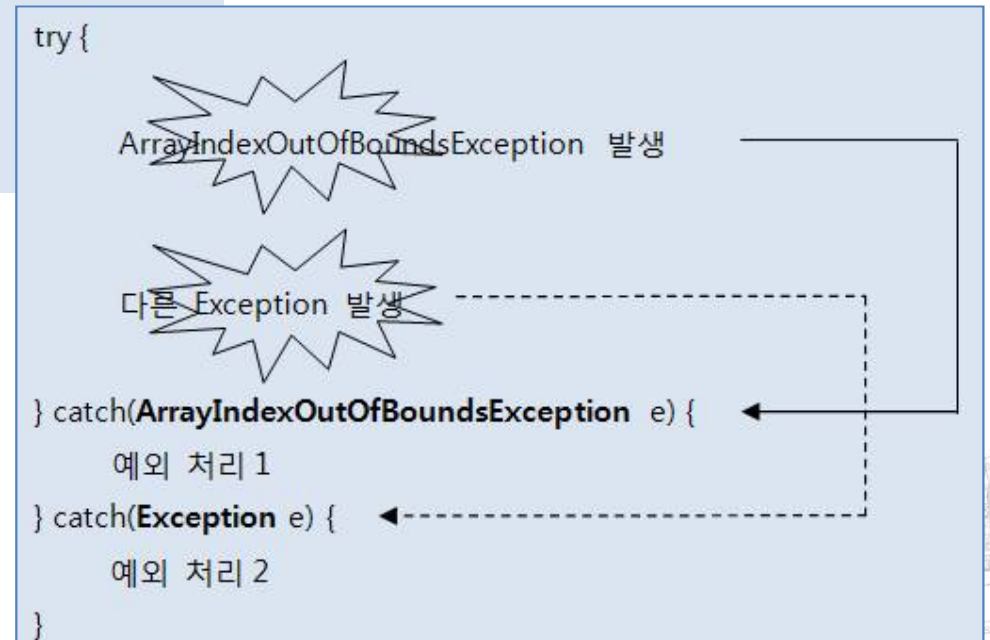
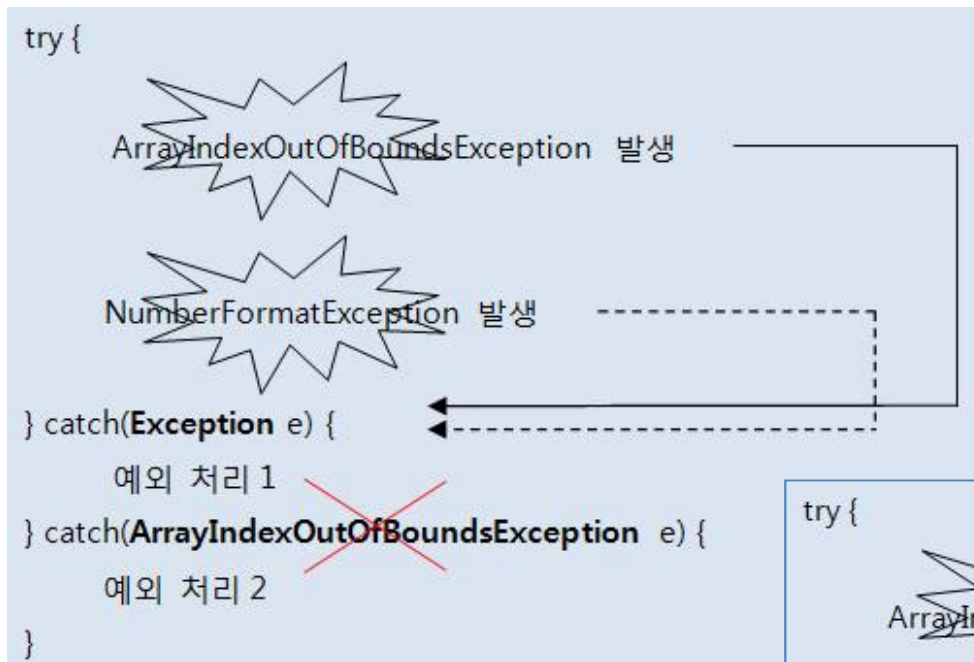
# 예제

```
public class ExecClass {  
    public static void main(String[] args) {  
        String[] input = {"a"};  
        try {  
            String data1 = input[0];  
            String data2 = input[1];  
            int value1 = Integer.parseInt(data1);  
            int value2 = Integer.parseInt(data2);  
            int result = value1 + value2;  
            System.out.println(data1 + "+" + data2 + "=" + result);  
        } catch (ArrayIndexOutOfBoundsException e) {  
            System.out.println(" 인덱스가 올바르지 않습니다..");  
        } catch (NumberFormatException e) {  
            System.out.println("숫자로 변환할 수 없습니다.");  
        } finally {  
            System.out.println("다시 실행하세요.");  
        }  
    }  
}
```



## 4절. 예외 종류에 따른 처리 코드

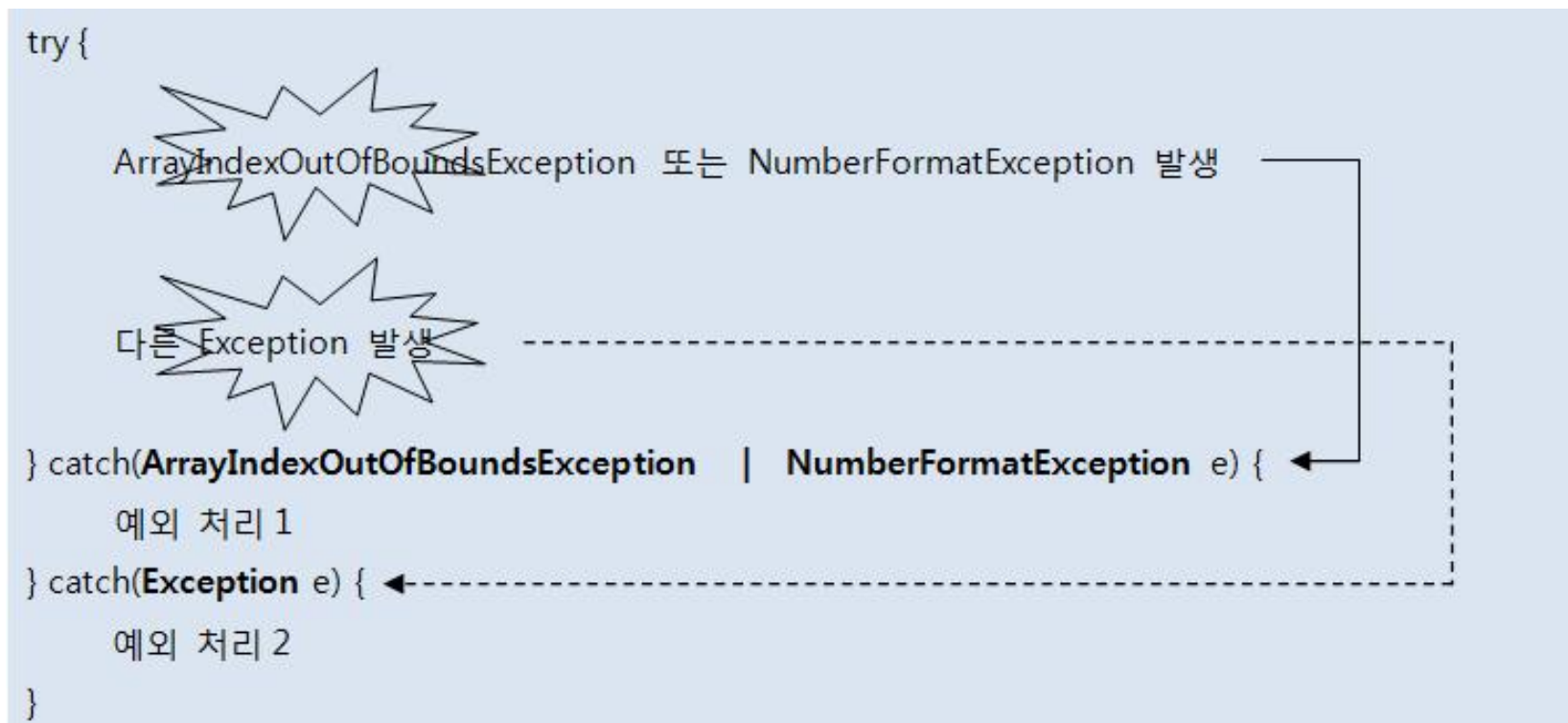
### ❖ catch 순서 – 하위 클래스가 위에 위치해야



## 4절. 예외 종류에 따른 처리 코드

### ❖ 멀티(multi) catch (p.436~437)

- 자바 7부터는 하나의 catch 블록에서 여러 개의 예외 처리 가능
  - 동일하게 처리하고 싶은 예외를 |로 연결



# 예제

```
public class MultiCatchExample {  
    public static void main(String[] args) {  
        String[] input={"a"};  
        try {  
            String data1 = input[0];  
            String data2 = input[1];  
            int value1 = Integer.parseInt(data1);  
            int value2 = Integer.parseInt(data2);  
            int result = value1 + value2;  
            System.out.println(data1 + "+" + data2 + "=" + result);  
        } catch(ArrayIndexOutOfBoundsException | NumberFormatException e) {  
            System.out.println("인덱스 or 숫자형식에 문제가 있습니다..");  
        } catch(Exception e) {  
            System.out.println("알수 없는 예외 발생");  
        } finally {  
            System.out.println("다시 실행하세요.");  
        }  
    }  
}
```





## 6절. 예외 떠 넘기기

### ❖ throws

#### ■ 메소드 선언부 끝에 작성

```
리턴타입 메소드명(매개변수,...) throws 예외클래스 1, 예외클래스 2, ... {  
}
```

#### ■ 메소드에서 처리하지 않은 예외를 호출한 곳으로 떠 넘기는 역할

```
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

# 예제

```
public class ThrowsExample {
    public static void main(String[] args) {
        try {
            findClass();
        } catch (ClassNotFoundException e) {
            System.out.println("클래스가 존재하지 않습니다.");
        }
    }

    public static void findClass() throws ClassNotFoundException {
        Class clazz = Class.forName("java.lang.String2");
    }
}
```

```
public class ThrowsExampleBefore {

    public static void main(String[] args) {
        findClass();
    }

    public static void findClass() {
        try {
            Class clazz = Class.forName("java.lang.String2");
        } catch (ClassNotFoundException e) {
            System.out.println("클래스가 존재하지 않습니다.");
        }
    }
}
```



# 실습

- ❖ String str = “this is a java”
- ❖ String subStr=null;
- ❖ 아래에 대해서 해당 메소드가 아닌 상위메소드에서 예외처리를 하게 만드세요
  1. substring() 메소드를 이용하여 프로그램을 작성하세요
  2. beginIndex를 -1로 설정 exception 을 발생시켜 보세요.





# Thank You !

이것이 자바다(<http://cafe.naver.com/thisjava>)