

# 예외 처리 코드 (try-catch-finally)

## 예외 처리 코드(try-catch-finally)

---

### ❖ 예외 처리 코드

- 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

## 예외 처리 코드(try-catch-finally)

---

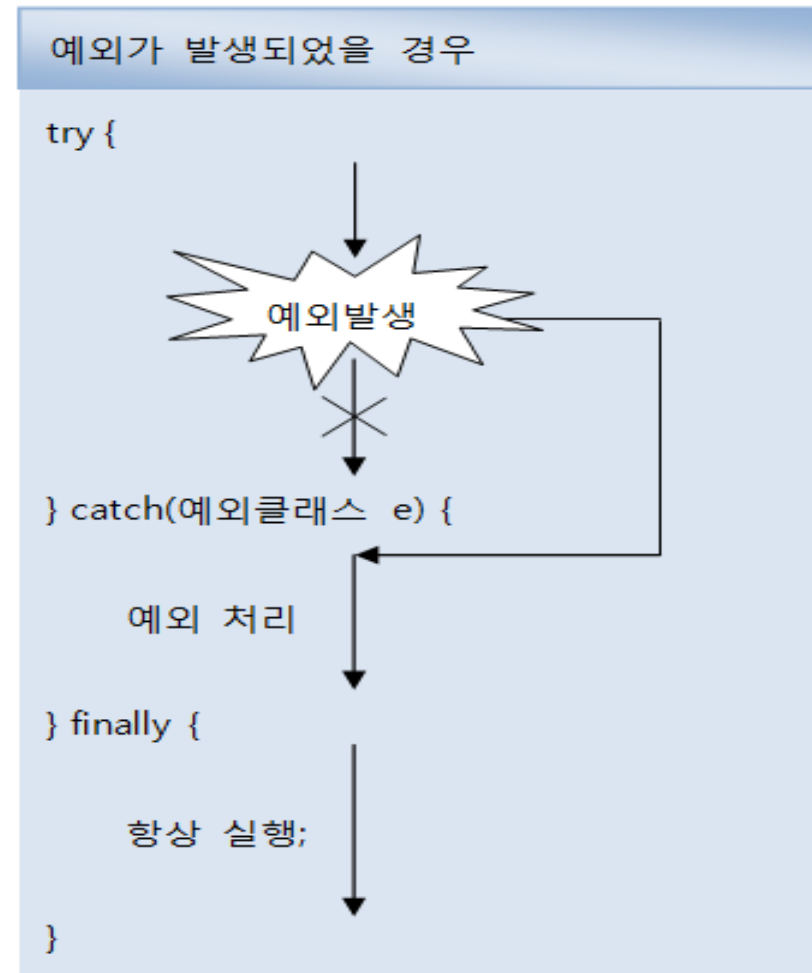
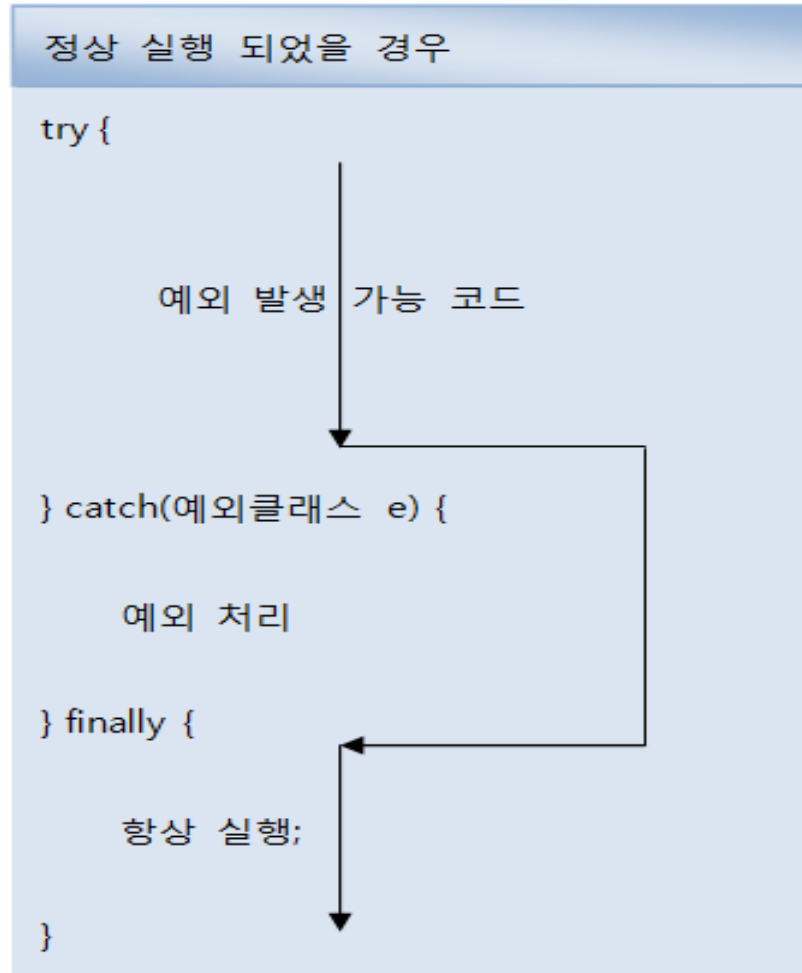
### ❖ 일반 예외 처리: TryCatchFinallyExample.java

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

## 예외 처리 코드(try-catch-finally)

### ❖ 예외 처리 코드

- o try - catch - finally 블록 이용해 예외 처리 코드 작성



## 예외 처리 코드(try-catch-finally)

### ❖ 실행 예외 처리: TryCatchFinallyRuntimeExceptionExample.java

```
public class TryCatchFinallyRuntimeExceptionExample {
    public static void main(String[] args) {
        String data1 = null;
        String data2 = null;
        try {
            data1 = args[0];
            data2 = args[1];
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("실행 매개값의 수가 부족합니다.");
            System.out.println("[실행 방법]");
            System.out.println(
                "java TryCatchFinallyRuntimeExceptionExample num1 num2");
            return;
        }
    }
}
```

## 예외 처리 코드(try-catch-finally)

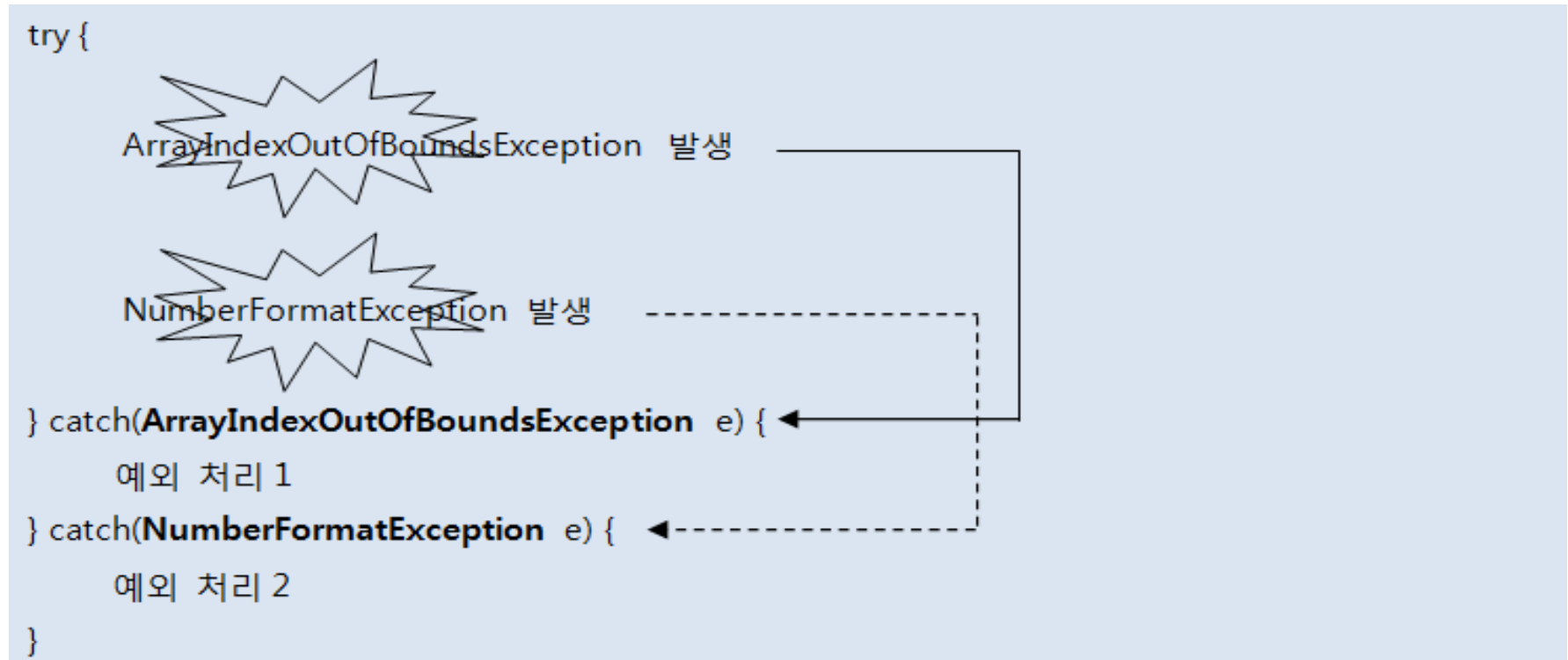
### ❖ 실행 예외 처리: TryCatchFinallyRuntimeExceptionExample.java

```
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.");
}
}
```

## 예외 처리 코드(try-catch-finally)

### ❖ 예외 종류에 따른 처리 코드 → 다중 catch

- 예외 별로 예외 처리 코드 다르게 구현



## 예외 처리 코드(try-catch-finally)

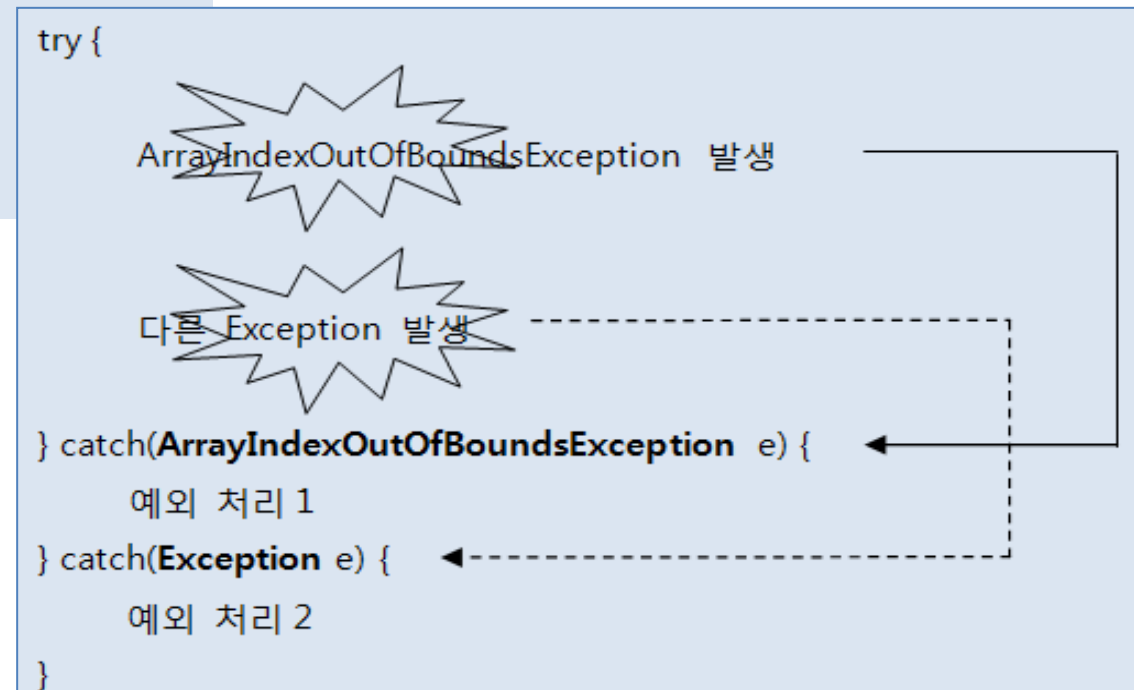
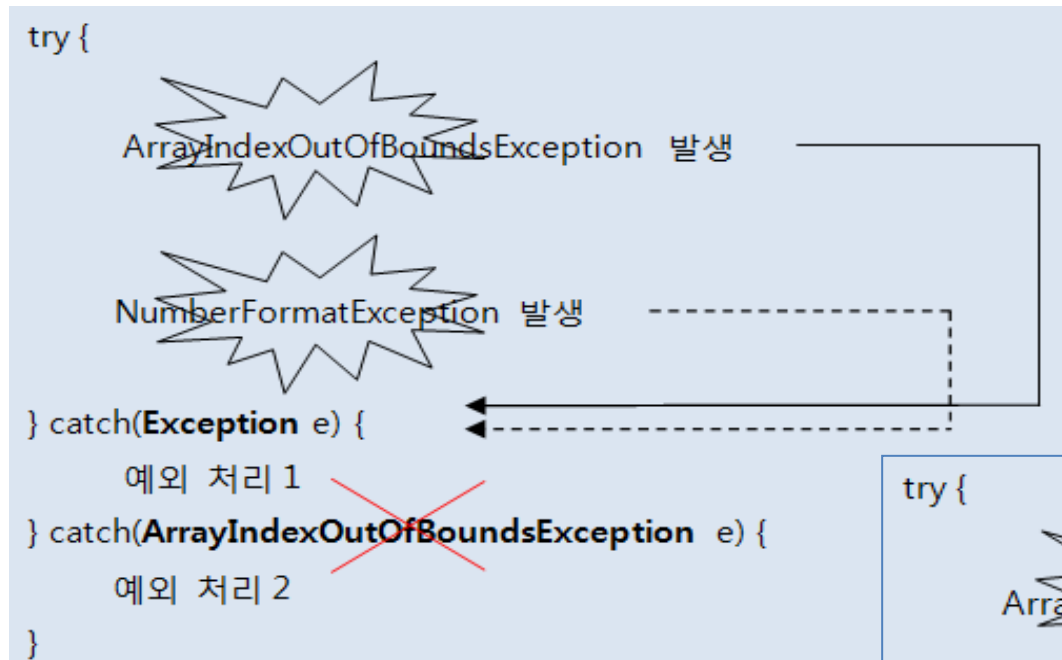
### ❖ 다중 catch: CatchByExceptionKindExample.java

```
public class CatchByExceptionKindExample {
    public static void main(String[] args) {
        try {
            String data1 = args[0];
            String data2 = args[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("실행 매개값의 수가 부족합니다.");
            System.out.println("[실행 방법]");
            System.out.println("java CatchByExceptionKindExample num1 num2");
        } catch (NumberFormatException e) {
            System.out.println("숫자로 변환할 수 없습니다.");
        } finally {
            System.out.println("다시 실행하세요.");
        }
    }
}
```



## 예외 처리 코드(try-catch-finally)

### ❖ catch 순서 – 상위 클래스가 위에 위치해야



## 예외 처리 코드(try-catch-finally)

### ❖ catch 블록의 순서: CatchOrderExample.java

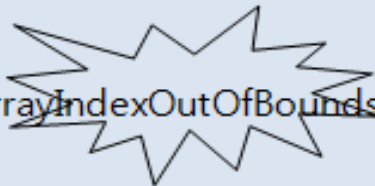
```
public class CatchOrderExample {
    public static void main(String[] args) {
        try {
            String data1 = args[0];
            String data2 = args[1];
            int value1 = Integer.parseInt(data1);
            int value2 = Integer.parseInt(data2);
            int result = value1 + value2;
            System.out.println(data1 + "+" + data2 + "=" + result);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("실행 매개값의 수가 부족합니다.");
        } catch (Exception e) {
            System.out.println("실행에 문제가 있습니다.");
        } finally {
            System.out.println("다시 실행하세요.");
        }
    }
}
```

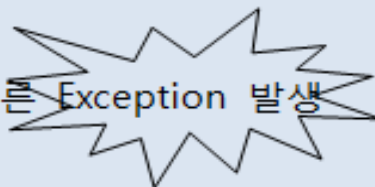
## 예외 처리 코드(try-catch-finally)

### ❖ 멀티(multi) catch

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

```
try {
```

 `ArrayIndexOutOfBoundsException` 또는 `NumberFormatException` 발생

 다른 `Exception` 발생

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

예외 처리 1

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

예외 처리 2

```
}
```

## 예외 처리 코드(try-catch-finally)

### ❖ 멀티 catch: MultiCatchExample.java

```
import java.io.IOException;
import java.net.Socket;
import java.net.UnknownHostException;

public class MultiCatchExample {
    public static void main(String[] args) {
        try {
            String data1 = args[0];
            String data2 = args[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(
                "실행 매개값의 수가 부족하거나 숫자로 변환할 수 없습니다..");
        } catch(Exception e) {
            System.out.println("알수 없는 예외 발생");
        } finally {
            System.out.println("다시 실행하세요.");
        }
    }
}
```