

# 예외처리

1. 직접 처리
2. 예외 떠넘기기(알리기)

# 프로그램에서 오류의 종류

발행 시점에 따라서

1. 컴파일 에러

2. 런타임 에러

# 프로그램에서 오류의 종류

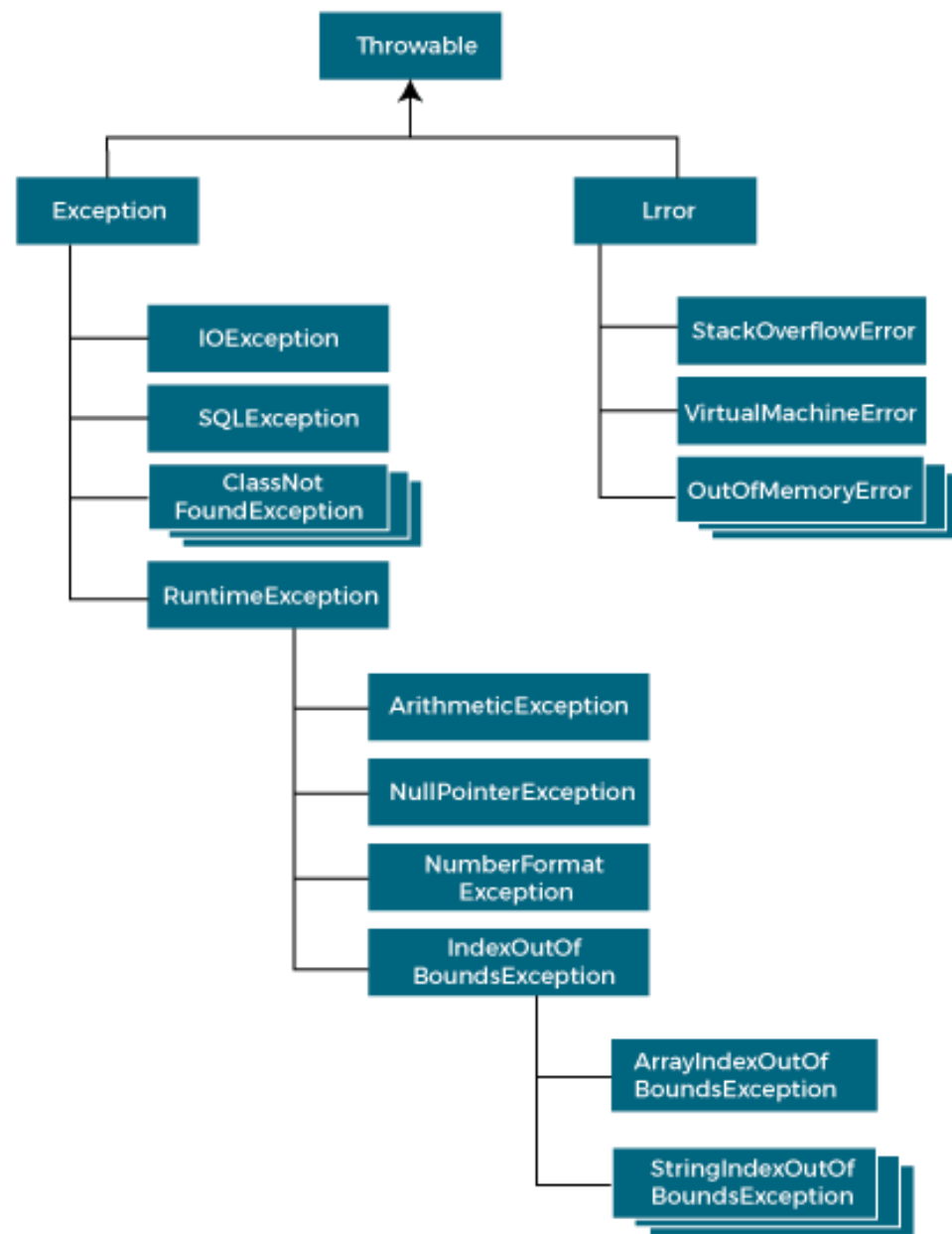
발행 시점에 따라서

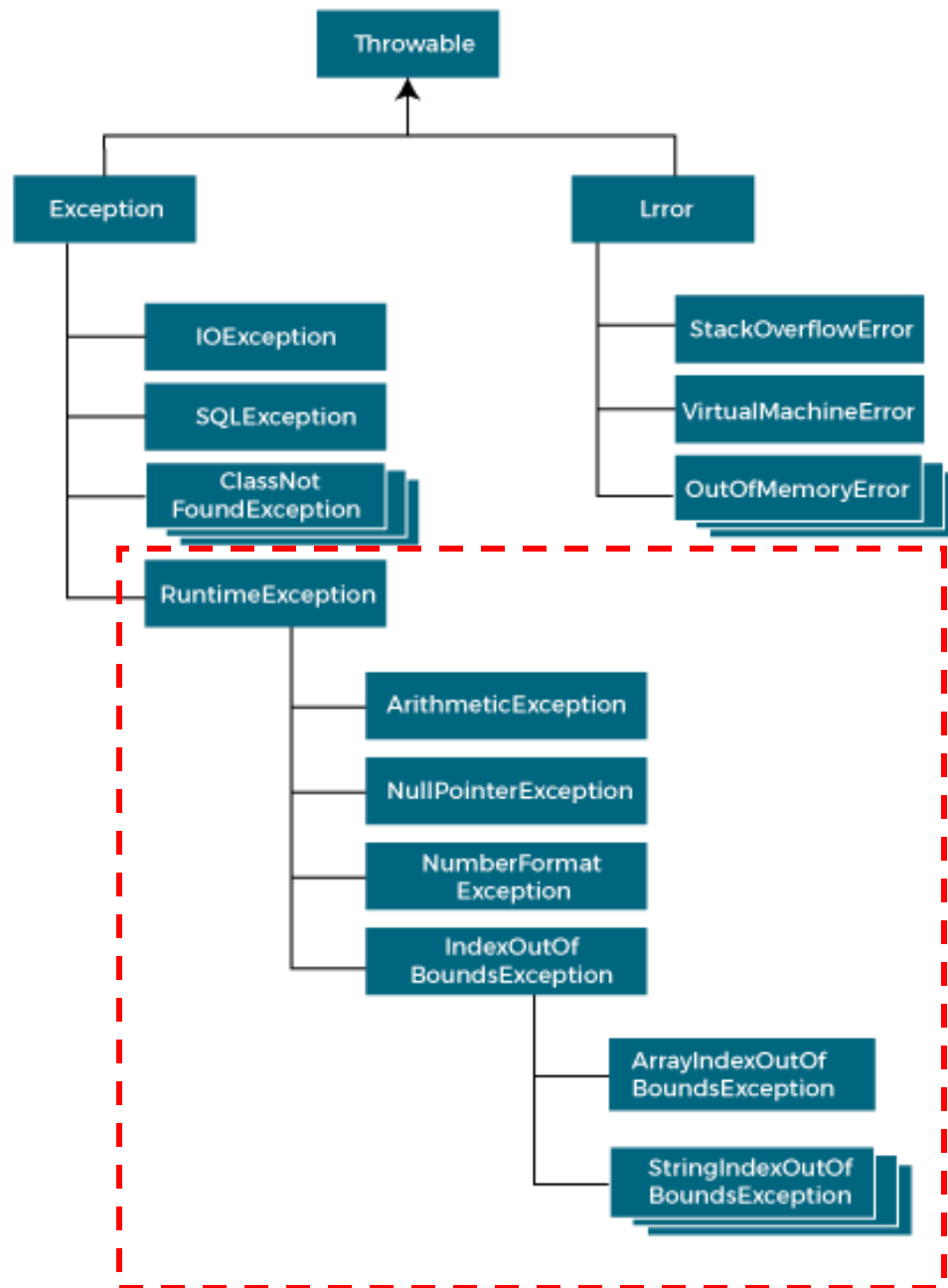
1. 컴파일 에러

2. 런타임 에러

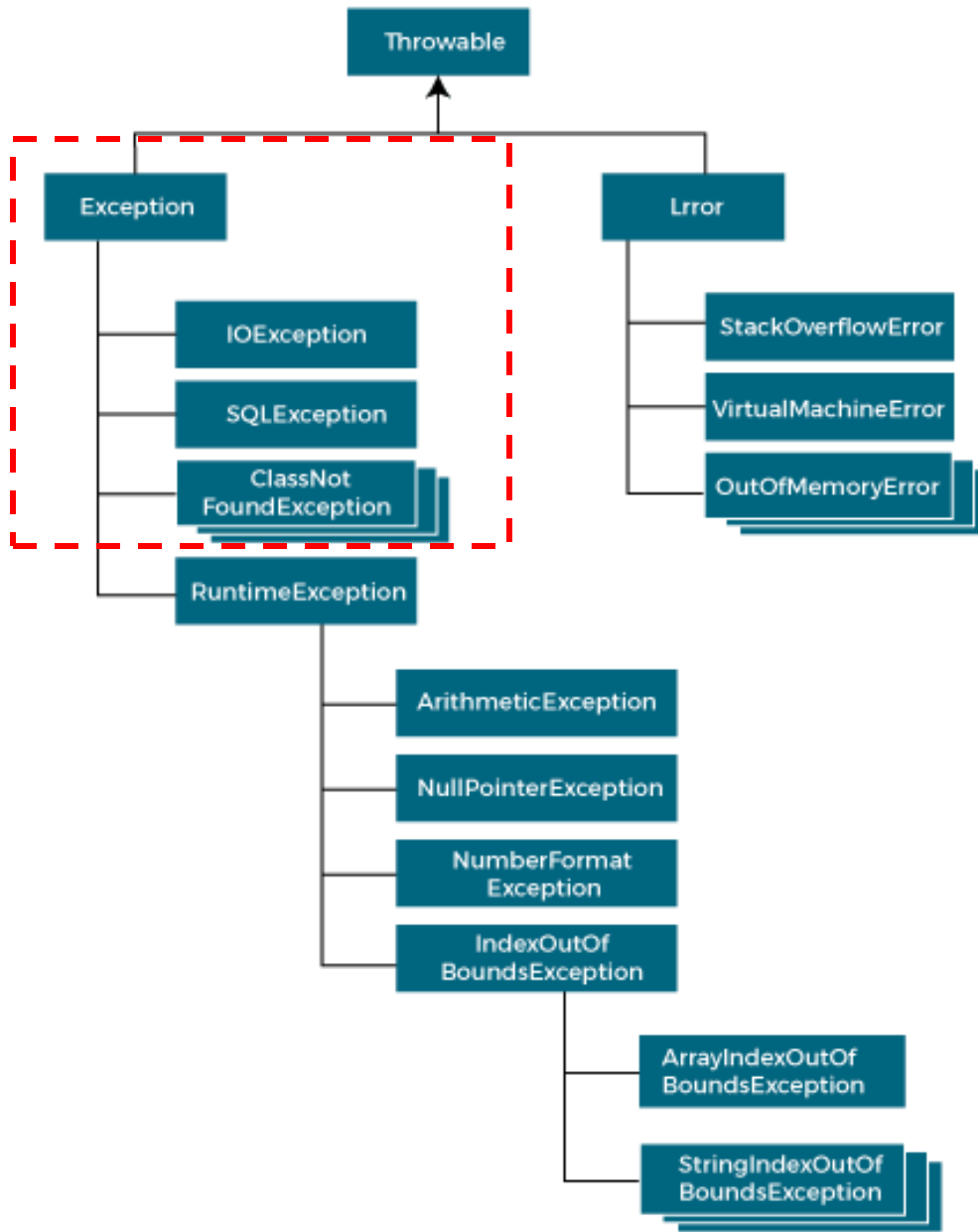
- Error : 수습할 수 없는 심각한 문제. 코드로 처리 불가
- Exception : 수습될 수 있는 덜 심각한 것.

코드로 예외 처리 가능.





Unchecked Exception



Checked Exception

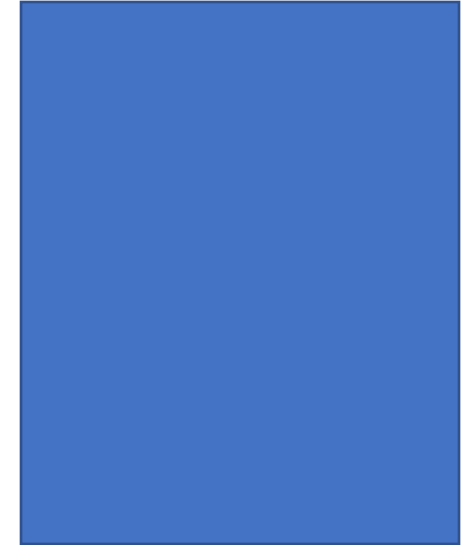
# 직접처리 : Try-catch

```
try{  
    1 // 예외가 발생할 가능성이 있는 문장들을 넣는다.  
} catch(Exception e1) {  
    2 // Exception1 이 발생했을 경우, 이를 처리하기 위한 문장을 적는다.  
} catch(Exception e2) {  
    // Exception2 이 발생했을 경우, 이를 처리하기 위한 문장을 적는다.  
}
```

# 직접처리 : Try-catch

```
try {  
    System.out.println(0/0); ←  
} catch (ArithmeticException ae) {  
    ae.printStackTrace();  
} catch (Exception e) {  
  
}
```

주소 0X100 에  
ArithmeticException 객체 생  
성

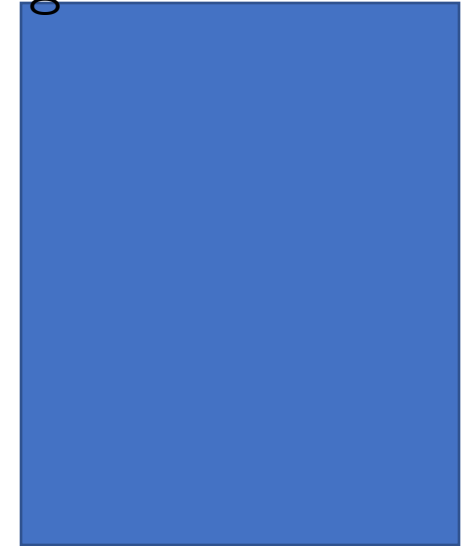




# 직접처리 : Try-catch

```
try {  
    System.out.println(0/0);  
} catch (ArithmeticException ae) {  
    ae.printStackTrace();  
} catch (Exception e) {  
  
}
```

주소 0X100 에  
ArithmeticException 객체 생  
성



# 직접처리 : Try-catch

```
try {  
    System.out.println(0/0);  
} catch (ArithmeticException ae) {  
    ae.printStackTrace();  
} catch (Exception e) {  
  
}
```

참조변수 ae의 유효범위  
(ae)

주소 0X100 에  
ArithmeticException 객체 생  
성

PrintStackTrace()  
GetMessage()

# 예외 떠넘기기 (알리기)

예외 선언하기 : throws 키워드 + Exception 나열하기

```
void method() throws Exception1, Exception2, ... Exception3 {  
    // ...  
}
```

예외를 선언함으로써 메서드를 호출하는 쪽에 알리는 것.

# 예외 떠넘기기 (알리기)

Java.io.FileInputStream - FileInputStream(File) 메서드 명세서

## **FileInputStream**

```
public FileInputStream(File file)
    throws FileNotFoundException
```

Creates a `FileInputStream` by opening a connection to an actual file, the file named by the `File` object `file` in the file system. A new `FileDescriptor` object is created to represent this file connection.

First, if there is a security manager, its `checkRead` method is called with the path represented by the `file` argument as its argument.

If the named file does not exist, is a directory rather than a regular file, or for some other reason cannot be opened for reading then a `FileNotFoundException` is thrown.

### **Parameters:**

`file` - the file to be opened for reading.

### **Throws:**

`FileNotFoundException` - if the file does not exist, is a directory rather than a regular file, or for some other reason cannot be opened for reading.

`SecurityException` - if a security manager exists and its `checkRead` method denies read access to the file.

### **See Also:**

`File.getPath()`, `SecurityManager.checkRead(java.lang.String)`

# 예외 떠넘기기 (알리기)

Java.io.FileInputStream - FileInputStream(File) 메서드 명세서

## FileInputStream

```
public FileInputStream(File file)
    throws FileNotFoundException
```

Creates a `FileInputStream` by opening a connection to an actual file, the file named by the `File` object `file` in the file system. A new `FileDescriptor` object is created to represent this file connection.

First, if there is a security manager, its `checkRead` method is called with the path represented by the `file` argument as its argument.

If the named file does not exist, is a directory rather than a regular file, or for some other reason cannot be opened for reading then a `FileNotFoundException` is thrown.

### Parameters:

`file` - the file to be opened for reading.

### Throws:

`FileNotFoundException` - if the file does not exist, is a directory rather than a regular file, or for some other reason cannot be opened for reading.

`SecurityException` - if a security manager exists and its `checkRead` method denies read access to the file.

### See Also:

`File.getPath()`, `SecurityManager.checkRead(java.lang.String)`

- 선언된 예외(changed exception)이 있다면 반드시 예외처리를 해줘야 한다.

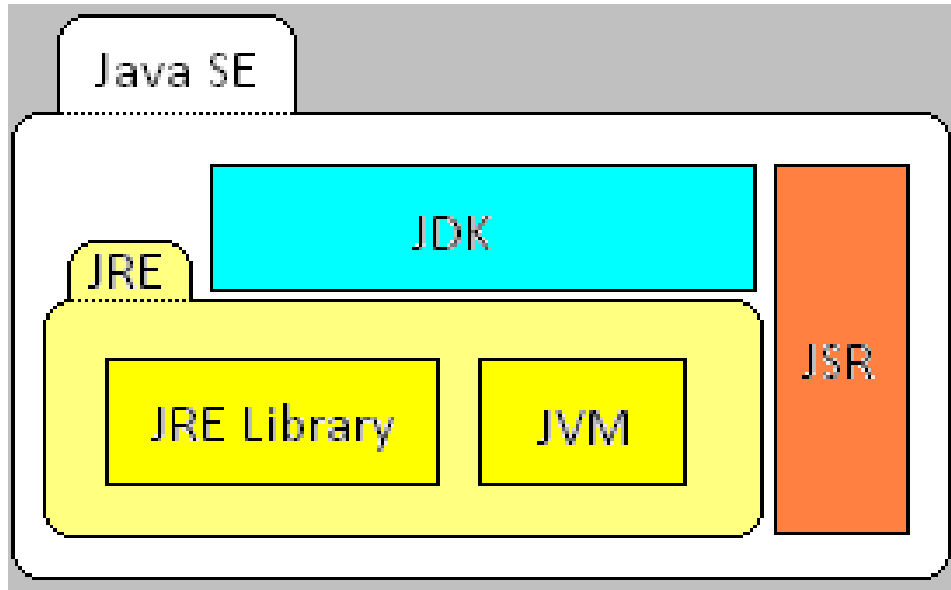
- Unchecked exception이라 해도 예외 선언을 통해 개발자는 발생할 수 있는 에러에 대해 인지하고 처리해줄 수 있음

```
File file = new File( pathname: "path");
FileInputStream fis = new FileInputStream(file);
```

Unhandled exception: java.io.FileNotFoundException

Add exception to method signature ↗ ↖ ↺ More actions... ↗ ↖ ↺

# Java SE & Java EE 차이



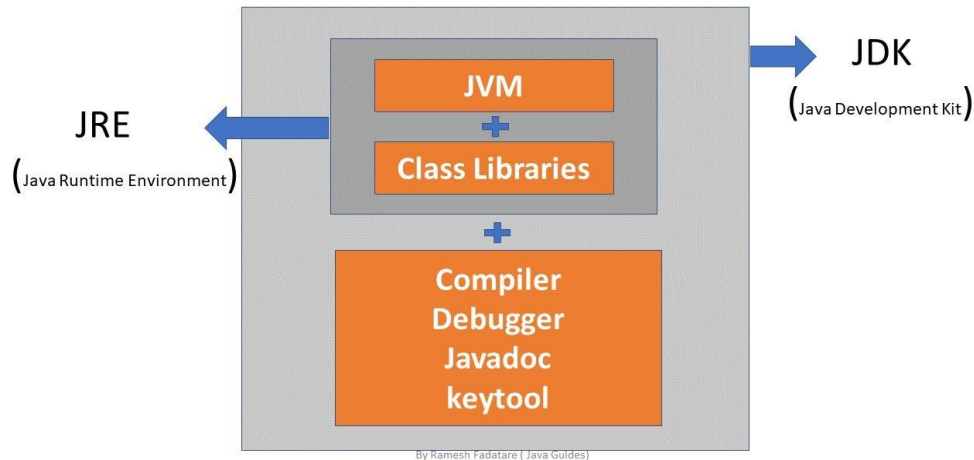
Java SE (Java Standard Edition)

- Java의 표준 문법을 의미한다.
- 자바 언어의 핵심 기능을 제공한다.

# Java SE & Java EE 차이

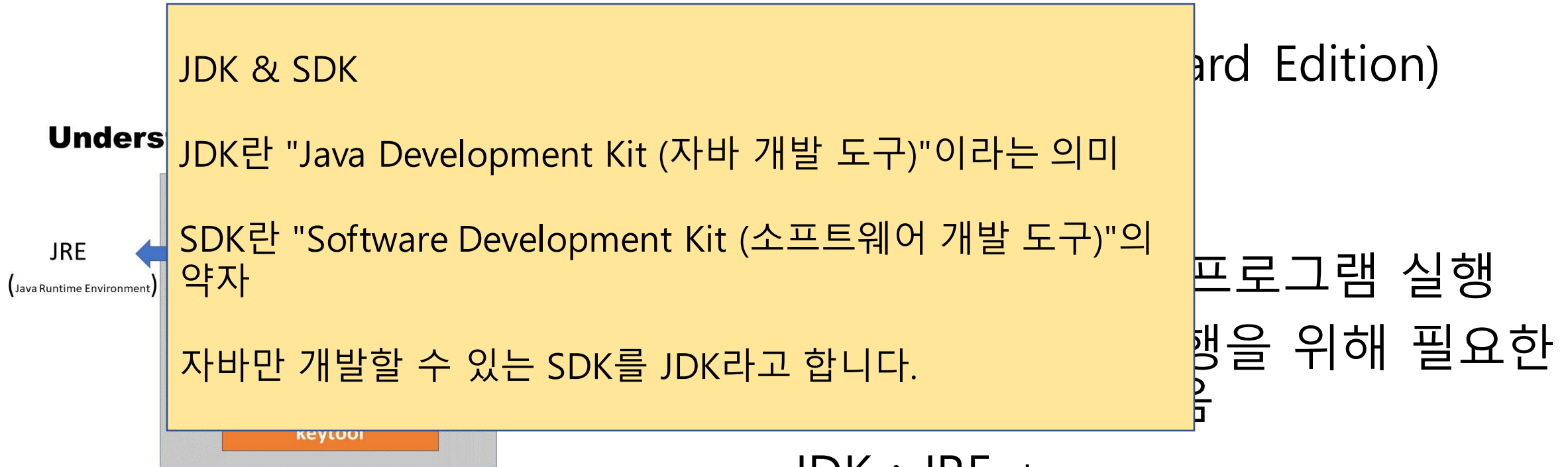
## Java SE (Java Standard Edition)

### Understand JDK, JRE and JVM



- JDK > JRE > JVM
- JVM : 가상머신 / 프로그램 실행
- JRE : 프로그램 실행을 위해 필요한 라이브러리들의 묶음
- JDK : JRE +  
자바 어플리케이션 개발 도구

# Java SE & Java EE 차이



ard Edition)

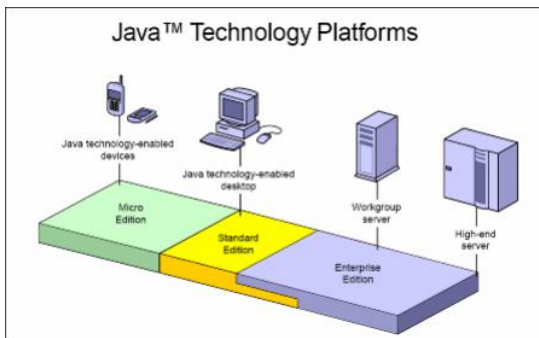
프로그램 실행  
행을 위해 필요한  
용

- JDK : JRE +  
자바 어플리케이션 개발 도구



# Java SE & Java EE 차이

## Java EE (Java Enterprise Edition)



- Java EE 플랫폼은 Java SE 플랫폼을 기반으로 그 위에 탑재된다.
- 웹 프로그래밍에 필요한 기능을 다수 포함
  - JSP, Servlet, JDBC, JNDI, JTA, EJB 등
- 대규모, 다계층, 확장성, 신뢰성, 보안 네트워킹 API, 환경 등