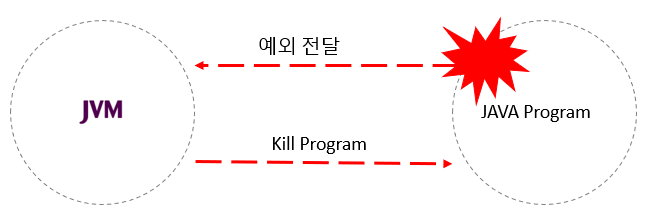
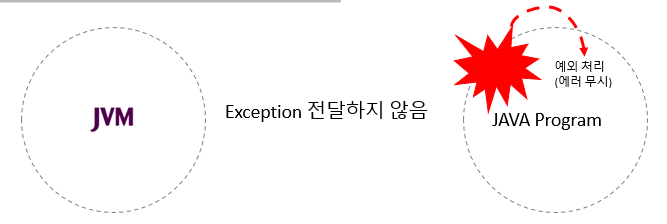
**예외처리**

* 프로그램이 동작하는 과정 중 잘못된 입력 값이 전달되거나, 읽어야 하는 파일이 존재하지 않거나, 배열의 인덱스를 잘못 전달했거나, Null 인스턴스에 접근을 하려했거나 등등의 이유로 **오류**가 발생하는데 이것을 “**예외**” 라고 함

예외가 발생한 경우 Java 프로그램은 JVM에 의해서 강제 종료됨.



예외 처리는 예외가 발생하더라도 **프로그램이 종료되지 않도록 하는 방법**이면서, **예외가 발생하지 않도록 하는 방어코딩**의 방법



**Java에서 가장 많이 발생하는 예외 목록**

1. NullPointerException

Null Instance에 접근하려 함

2. ArrayIndexOutOfBoundsException

List의 인덱스 범위를 벗어남

3. IndexOutOfBoundsException

배열의 인덱스 범위를 벗어남

4. NumberFormatException

숫자가 아닌 문자를 숫자로 변환하려 함

상기의 예외들은 if를 활용해 예외를 방지할 수 있음

1. NullPointerException

Null Instance에 접근하려 함

**-> Instance가 Null인지 먼저 확인**

2. ArrayIndexOutOfBoundsException

List의 인덱스 범위를 벗어남

**-> 찾으려는 인덱스가 존재하는지 먼저 확인**

3. IndexOutOfBoundsException

배열의 인덱스 범위를 벗어남

**-> 찾으려는 인덱스가 존재하는지 먼저 확인**

4. NumberFormatException

숫자가 아닌 문자를 숫자로 변환하려 함

**-> 문자가 숫자형태인지 확인**

if 로 예외 처리할 수 없거나 애매할 경우 try ~ catch ~ finally를 사용

exception class를 상속받았으면 반드시 try catch문을 사용

runtiome exception 상속 받았으면 if로 예외처리