

6장 예외처리

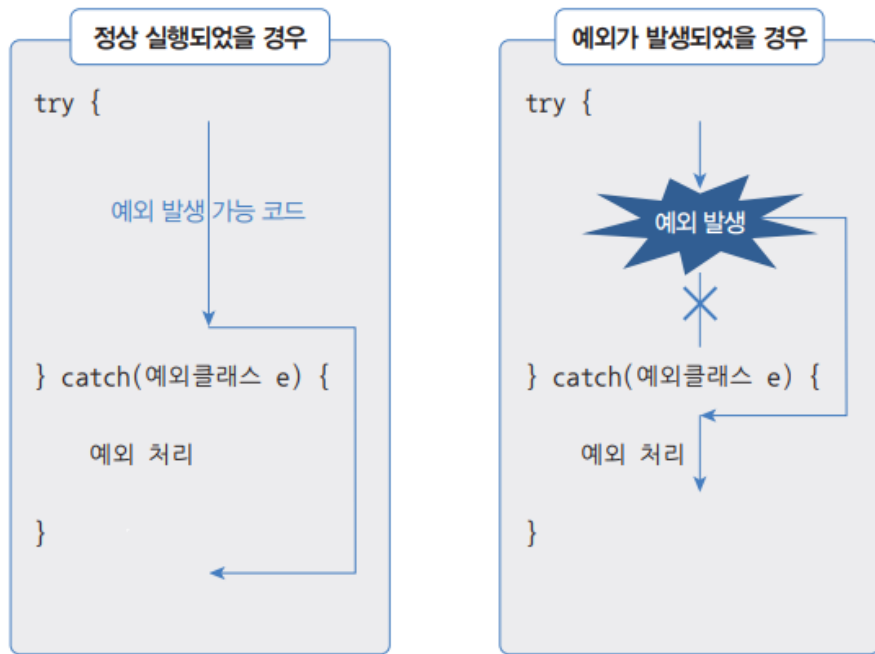
김철학

목차

1. 예외처리
2. 예외 클래스
3. 예외 던지기

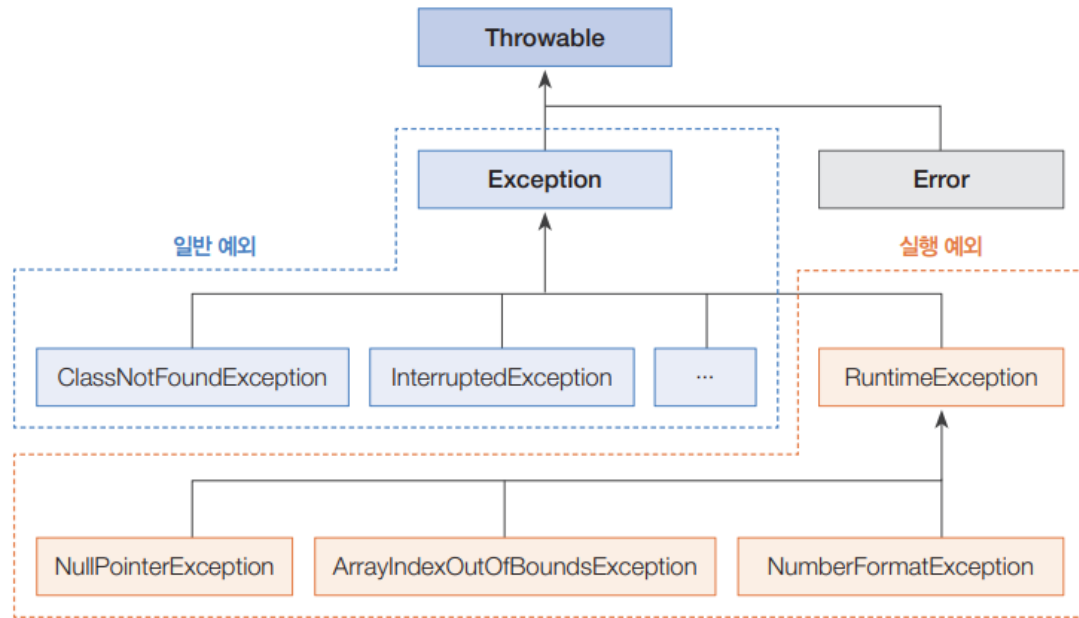
1. 예외 처리

- 예외^{Exception} 는 프로그램 실행 중에 발생하는 에러^{Error}
- 예외 처리는 예외 발생 시 비정상적인 종료를 막고 정상 실행을 유지할 수 있게 처리하는 문법
- 견고하고 안정적인 프로그램을 위해 예외 처리 수행



2. 예외 클래스

- Exception은 일반 예외와 실행 예외 클래스로 나뉨
- 일반 예외는 컴파일 전에 예외 발생 문법을 검사하고 실행 예외는 런타임에 발생하는 예외



3. 예외 던지기

- 예외 던지기는 메소드 내부에서 예외 발생 시 throws 키워드 이용해 메소드를 호출한 곳으로 예외 발생
- throws는 메소드 선언부 끝에 작성하고 떠넘길 예외 클래스를 쉼표로 구분해서 나열
- throw는 사용자가 정의한 예외를 던지는 키워드

```
public void method1() {  
    try {  
        method2(); //method2() 호출  
    } catch(ClassNotFoundException e) {  
        System.out.println("예외 처리: " + e.getMessage());  
    }  
}  
  
public void method2() throws ClassNotFoundException {  
    Class.forName("java.lang.String2");  
}
```



The diagram illustrates the flow of an exception. A blue arrow originates from the `method2()` call inside the `try` block of `method1()`. It points to the `catch` block of `method1()`, which contains the text "호출한 곳에서 예외 처리" (Exception handled at the calling place). Another blue arrow points from the `throws` clause of `method2()` to the same `catch` block, indicating that the exception is being passed back to the caller.