JAVA기초입문과정 — CHAPTER10 예외처리

Contents

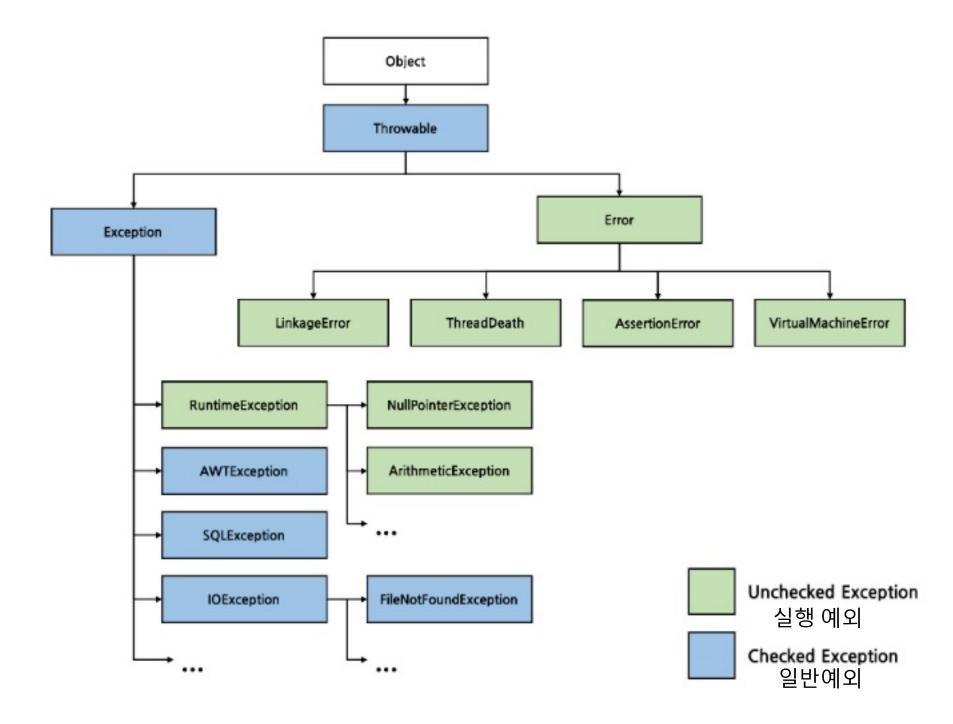




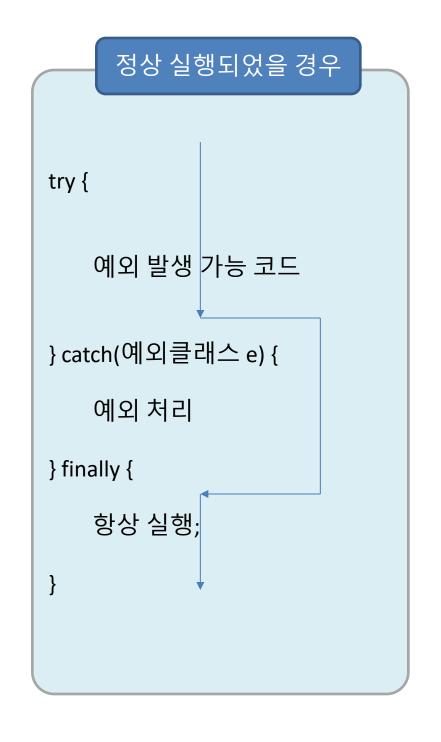
예외처리

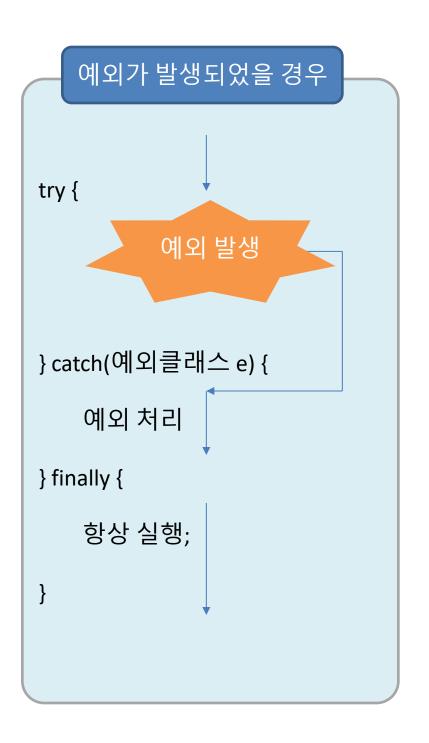
1. 예외와 예외 클래스

- 일반 예외(Exception): 컴파일러가 <u>예외 처리 코드 여부를 검사</u>하는 예외를 말한다.
- 실행 예외(Runtime Exception): 컴파일러가 <u>예외 처리 코드 여부를 검사하지 않는</u> 예외를 말한다.



2. 예외 처리 코드

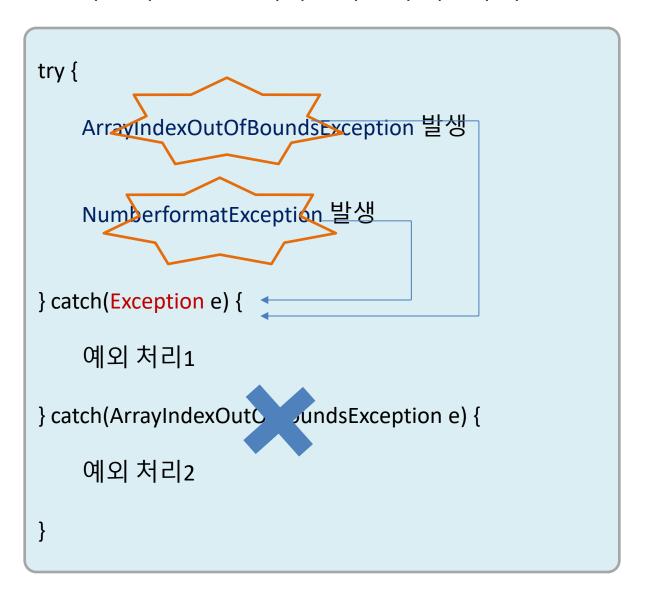




예외처리

3. 예외 종류에 따른 처리

- 예외가 발생하면 catch블록은 위에서 부터 차례대로 검사 대상
- 상위 클래스 catch블록이 먼저 검사 대상이 되면 안됨



```
try {
    ArrayIndexOutOfBoundsException 발생
    NumberformatException 발생
} catch(ArrayIndexOutOfBoundsException e) {
    예외 처리1
} catch(Exception e) { 
    예외 처리2
```

4. 예외 떠넘기기

• 메소드를 호출한 곳으로 예외를 떠넘기고 싶을때

```
리턴 타입 메소드명(매개변수, ...) throws 예외 클래스1, 예외 클래스2, ... {
}
```

```
public class ThrowsExample1 {{
    public static void main(String[] args) {
        findClass();
    } catch(ClassNotFoundException e) {
        System.out.println("예외 처리: " + e.toString());
    }
}

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

5. 사용자 정의 예외

• 표준 라이브러리에 존재하지 않는 예외 클래스를 정의해서 사용해야 할때

```
public class XXXException extends [Exception | RuntimeException] {
    public XXXException() {
        기본 생성자
    }

public XXXException(String message) {
        super(message);
    }
```

```
public class InsufficientException extends Exception {
   public InsufficientException() {
        public InsufficientException(String message) {
            super(message);
        }
}
```

Thank You!