

1. 예외(Exception) 처리

1.1 예외 처리의 정의

- 프로그램에서 문제가 발생할 만한 곳을 예상하여 사전에 "문제가 발생하면 이렇게 해라"라고 프로그래밍 하는 것을 예외 처리라고 한다.

1.2 try ~ catch 예외처리 문법 사용예

```
try
{
    try블럭 ; 익셉션이 발생할 가능성이 있는 명령문들(문제가 발생할 수 있는 로직을 기술)
}
catch(익셉션타입 익셉션변수) - 예외 처리를 여러 개 지정할 수 있다.
{
    그 익셉션을 처리하는 명령문(try블록안에서 문제가 발생했을 때 대처방안 기술);
}
finally - try와 catch와 상관없이 무조건 실행
{
    익셉션 발생 여부와 상관없이 맨 마지막에 실행할 명령문;
}
```

- Java는 예외를 객체로 취급하여 예외가 발생하면 예외 객체가 자동 생성된다.

1.3 throws 예외처리 문법

- try ~ catch 예외 처리 방법은 예외가 발생했을 때 자체적으로 catch문을 이용해서 해결했지만 throws의 경우에는 나를 호출한(실행시키는) 쪽으로 예외를 던져버리는 방식이다.

1.4 일반적인 예외들

- `ArrayIndexOutOfBoundsException` > 배열을 사용시 존재하지 않는 index값을 호출하면 발생.
- `NullPointerException` > 존재하지 않는 객체를 가리킬 때 발생.
- `NumberFormatException` > 숫자로 변경할 수 없는 문자열을 변경하려 할 때 발생.

DB 관련 Exception

- `ClassNotFoundException` : 드라이브 이름을 찾지 못했을 때 발생.
- `SQLException` : db url, id, pw가 올바르지 않을 때 발생.