◎ 예외처리

필수예외(checked) : 예외처리 필수 : Exception 상속한 예외 ,
처리방법: try~ catch, throws (예외미루기)
선택예외(unchecked) : 예외처리 선택 : RuntimeException 상속한 예외 ,
처리방법: try~catch

예외를 처리한다는 것은
특정 예외사항을 인식하고
예외사항에 대한 대비코드를 작성함으로 써 프로그램이 비정상 종료를 하지 않게 하는 것
필요하다면 예외 되던지기를 할 수 있다.
(예외를 나눠서 처리해야 하는 경우 , 다시 예외를 일으켜야 하는 경우가 있다.)

예외되던지기 예제 학습

```
//예외 되던지기
//예외의 일부처리하고 다시 예외 일으키기

public class ExceptionTest {

	public static void main(String[] args) {

		ExceptionTest  e = new ExceptionTest();
		try {
			e.method();
		} catch (Exception e1) {
			// TODO Auto-generated catch block
			//e1.printStackTrace();
			System.out.println(" 나머지 부분 처리합니다");
		}

	}


	public void  method() throws Exception {
		try {
		  throw new Exception();
		}catch( Exception e) {
			System.out.println("예외의 일부를 처리합니다");
			throw e;   // 예외 되던지기
		}
	}
}
```

■ Controller에서 예외처리하기

```
@Controller
public class ExceptionCatcher {

	@ResponseStatus(HttpStatus.SERVICE_UNAVAILABLE)
	@ExceptionHandler(Exception.class)
	public String catcher1( Model model, Exception ex ) {

		model.addAttribute("ex", ex);
		return "err";
	}


	@ExceptionHandler(SQLException.class)
	public String catcher2( Model model, SQLException ex ) {
		System.out.println("");
		model.addAttribute("ex", ex);
		return "err";
	}


	@ExceptionHandler(MyException.class)
	public String catcher3(  Model model, MyException ex) {
		System.out.println("");
		model.addAttribute("ex", ex);
		return "err";
	}


	@RequestMapping(value = "/ex1", method = RequestMethod.GET)
	public void method1() throws Exception {
		throw  new   NullPointerException();
	}


	@RequestMapping(value = "/ex2", method = RequestMethod.GET)
	public void method2() throws Exception {
		throw  new   SQLException();
	}


	@RequestMapping(value = "/ex3", method = RequestMethod.GET)
	public void method3() throws MyException {
		throw  new   MyException();

	}


}
```

◎ 모든 controller에서 사용가능한 GlobalCatcher 만들기

```
@ControllerAdvice
public class GlobalCatcher {

	@ResponseStatus(HttpStatus.SERVICE_UNAVAILABLE)
	@ExceptionHandler(Exception.class)
	public String catcher( Model model, Exception ex ) {

		model.addAttribute("ex", ex);
		return "err";
	}

}
```