17. 05. 01

예외처리

동일한 예외처리가 여러 클래스에 분산되어 있어서 try~catch문을 여러 번 반복해야할때 분산된 예외 처리를 최종적으로 호출되는 곳에서만 처리할 때, 각 분산된 메소드에서 예외 떠넘기기를 해서 한번에 처리할 수 있다.(말졸라못하네)

리턴타입 메소드() thorws 예외클래스1, 예외클래스2{}

* 예외가 발생할만한 코드(try catch를 한꺼번에 처리하기 위해 메소드명 옆에 thorws로 떠넘기기 처리)

사용자 정의 예외와 예외발생

프로그램을 개발 시, 자바 표준 API에서 제공하는 예외 클래스에서 더 추가적인 예외 클래스를 만들어야할 때가 있다.

이를 사용자 정의 예외 클래스라고 한다.

ex)s

class XXXException extends Exception{ // 상속받은 후 사용자 정의  
 public XXXException(){};

public XXXException(message){

super(message);

}

}

예외 발생 : 내장된 예외는 throw new Exception(); 식으로 명시를 하지 않더라도 내부적으로 발생함 but 사용자 정의 예외는 명시해야함 throw new XXXException()

정의 🡺 예외발생시 예외객체 던지기 🡺 try catch 처리

ex)

public void metho() throws XXXException{

throws new XXXException();

// 명시적 호출 처리

}

호출하는 곳에서 예외처리(try-catch)

ex)

try{

method();

}catch(XXXException xe){ // 사용자 정의 예외 객체로 받아야함

예외가 처리할 내용

}

예외 클래스 메시지 전송

예외가 발생했을 때,

* if나 강제적으로..
* throw new XXXException(“넘기는 메시지”);
* 정의된 곳에서 super(msg);
* Exception

message = msg; // 할당;

getMessage() 에서 넘겨지는 메시지를 호출

숙제

1단계

PointException : 정수가 특정 점수 이하로 되었을 때, 예외처리로 “불합격입니다.”

Exam : getPoint(int) : 예외처리 메소드

* 입력받은 점수가 60 이하일 때, 예외 객체 호출

2단계

PointException

* 추가 메소드 : showResult() : @@점 획득 결과는 재수강

Exam : 과목명, 시험점수 🡺 getPoint() 과목과 Random 활용

API!

목표

자바프로그램에서 API의 역할을 안다.

프로그래밍 시, API를 활용할 줄 안다.

사용을 많이 하는 클래스들에 대한 활용 방법을 읽힌다.

* String, Object, StringBuffer, Arrays, Wrapper, Date, Calender

자바 API document

API(Application Programming Interface) : 자바에서 라이브러리라고 부르기도 하는데, 프로그램 개발에 자주 사용되는 클래스 및 인터페이스 모음

<JDK설치 경로> /jre/lib/rt.jar 라는 압축파일에 저장

API document는

<http://docs.oracle.com/javase/8/docs/api>

API 세 개의 프레임으로 나뉘어짐

1. 패키지 전체목록
2. 패키지에 속하는 클래스와 인터페이스
3. 중앙 프레임 – 상세설명

상세 내용 중 메소드

* return 값 메소드명 parameter로 입력할 값
* return 값에 대한 상세 설명

주요 패키지들

java.lang 패키지 : import 없이 바로 사용 가능한 default 패키지

* Object : 자바 클래스(api 클래스, 사용자 정의 클래스)의 최상위 클래스
* System :
  + .in : 표준 입력장치(키보드)로부터 data를 입력받을 때 사용
  + .out : 표준 출력장치(모니터)로 출력하기 위해 사용
  + JVM을 종료시킬 때 사용
  + 쓰레기 수집기(사용하지 않는 메모리 수집) 실행 요청할 때 사용

Class : 클래스를 메모리 로딩할 때 사용

* Class.forName(“지정된 클래스”) 지정된 클래스를 메모리에 올림

String : 문자열을 지정하고 여러가지 정보를 얻을 때 사용

StringBuffer, StringBuilder : 문자열을 저장하고 내부 문자열을 조작할 때 사용

Math : 수학 함수

Wrapper (Integer, Double) : 기본 type의 data를 갖는 객체를 만들 때 사용

method()를 통해서 활용

jsp

form의 속성

method = “get/post”

<form method=”post”>

get 방식 : url 기반으로 query String 형식으로 data 전송

속성값을 정하지 않으면 default가 get방식으로 url에 data전송이 보인다.

query string : data를 전송시, key1=value1&key2=value2

<name>

<name>

<name>

get 방식

http://localhost:6080/jspExp/a01\_basic/a10\_reload.jsp?favFruit=fff

뒷 부분 보면 get 방식이다!!!

url에서 fff 대신 다른 걸 넣어도 data를 전송할 수 있다. 즉, input tag에 data를 입력하지 않아도 된다.

BUT 한계가 있다.

1. data 전송 용량의 한계
2. 보안에 취약

post 방식

url이 아니라 data 영역을 이용해서 data를 전송하기에 웹 브라우저나 서버 상관없이 전송할 수 있는 parameter 길이 제한이 없다.

* query string으로 data는 전달하지만,

form에서 data 전송

request

* getParameterValues(“멸티 name”) 🡺 String [ ];
* < name = “name01”/>
* < name = “name01”/>
* < name = “name01”/>