**Web개발\_Day14\_노트정리**

**인터페이스, interface**

ㆍ인터페이스. 추상화를 구현할 수 있게 java Class에서 제공하는 기법이다.

* Interface라는 클래스를 생성하고, implement를 이용하여 class 뒤에 추가한다.
* 콤마(,)로 구분하며, 다중 상속이 가능하다.

ㆍ인터페이스의 장점

1. 개발시간을 단축할 수 있다.

* 한쪽에서는 인터페이스를 만들고 다른 팀은 그 내용을 구현하게 되므로, 동시작업으로 개발시간을 현저히 줄일 수 있다.

1. 표준화가 가능하다

* 프로젝트에 사용된 기본틀을 인터페이스로 작업하고 개발자들에게 인터페이스를 구현하여 프로그램을 작성하도록 함으로 보다 일관성있게 정형화된 프로그램을 개발할 수 있다.

1. 관계가 없는 클래스들도 관계를 만들어줄 수 있다.

* 서로 상속관계가 없고 같은 부모 클래스를 가지고 있지 않은 아무런 관계없는 클래스에게 하나의 인터페이스를 공통적으로 구현하도록 함으로 관계를 맺어 줄 수 있다.

1. 독립적인 프로그래밍이 가능하다.

* 인터페이스를 이용하면 클래스 선언과 구현을 분리시킬 수 있다. 따라서, 실제 구현에 독립적인 프로그램을 작성하는 것이 가능해진다.

**추상화, abstract**

ㆍ인터페이스의 역할도 하면서 구현체도 가지고 있는 클래스. (추상클래스는 인터페이스로 대체하는 것이 좋다)

* 인터페이스처럼 선언만 하는것도 가능하며, 부모클래스처럼 구현하는 것도 가능하다.

ㆍ인터페이스는 ‘강제성’이라는 기능이 있지만, 추상클래스는 강제성은 없다.

* 결과적으로 추상클래스는 ‘가능’하다~ 라는 개념이며, 선언만 하는 경우 인터페이스가 더욱 추천된다.

ㆍ추상메소드 : 인터페이스의 메소드와 같은 역할을 하는 메소드이다. “구현부”가 없다.

* 메소드 자료형 앞에 abstract가 붙는다.
* 이러한 추상메소드가 포함된 클래스를 추상클래스라고 부른다.
* Class앞에 abstract 키워드가 붙는다.

**예외처리, Exception**

ㆍ예외처리 : 오류, 에러를 사전에 방지하는 문법이다.

ㆍ프로그램을 실행 중 어떤 원인에 의해서 오작동하거나, 비정상적으로 종료되는 경우가 발생한다. 이것을 “오류” 또는 “예외”라고 부른다.

ㆍ오류종류가 나뉘는데, 발생 시점에 따라 “컴파일 에러” 와 “런타임 에러”. 2가지로 분류된다.

1. 컴파일 에러 : 컴파일 할 때 발생하여 프로그램이 시작조차 되지 않는 에러이다.
2. 런타임 에러 : 프로그램이 시작되고 사용하다가 실행도중에 발생하는 에러이다.

ㆍ자주 발생하지 않는 “논리적 에러”가 있다.

* 실행은 되나, 의도와 다르게 동작하는 경우를 의미한다.

ㆍtry~catch구문

* Try{

오류가 발생할 가능성이 있는 구문

}

Catch(발생할 가능성이 있는 에러타입 에러변수명){

예외 발생시 할 처리행동 구문

}

Finally{

예외가 발생하든 말든 실행할 구문

}

* 예외가 발생하지 않으면 catch를 지나지 않는다.
* Finally는 생략가능하다.
* 모든 catch문의 에러가 해당된다면, 윗 줄에 적힌 순서대로 실행된다.

**예외발생시키기, throw, throws**

ㆍ예외를 직접 발생시킨다.

ㆍ예외를 사용하려면 “RuntimeException” 를 extends해줘야한다.

ㆍthrow, 예외발생

* 해당 키워드가 실행되면, 바로 에러를 “발생”시킨다.
* Throw new 에러명(사용자 정의 가능)

ㆍthrows, 예외 던지기

* 소갲가 “특정 에러명”을 가진 에러를 “발생”만 시킨다면, throws는 에러를 호출한 지점으로 던진다.
* 에러가 “발생”하면그 발생한 에러를 throws가 메소드를 통해, 호출한 곳으로 던지는 개념이다.
* 메소드명 throws 에러명

ㆍ프로그래밍시 예외를 하는 위치도 대단히 중요하다. 프로그램의 수행여부를 결정하기도 하고 또한, 트랜잭션 처리와도 밀접한 관계가 있다.

**트랜잭션, transactoin**

ㆍ하나의 **작업 “단위”** 를 의미한다.

ㆍ예시로, ‘상품발송’ 트랜잭션에는 아래의 작업들이 있다.

* 포장, 영수증 발행, 택배 발송
* 상품 발송이라는 하나의 트랜잭션은 3가지 작업중 하나라도 실패할 경우, 발송이 이루어지면 안된다.
* 한가지의 작업중, 하나라도 에러가 발생할 경우, 그 작업에서 throws에러를 해준다면, 해당 트랜잭션은 즉시 종료하고, catch구문이 실행되는 것이 좋은 예이다.