**Web개발\_Day08\_노트정리**

**클래스**

ㆍ클래스명과 파일명은 동일해야 한다.

ㆍ하나의 클래스와 다수의 객체

* 같은 클래스로부터 생성되었을지라도, 각 객체의 속성은 서로 다른 값을 유지할 수 있으며, 메소드도 모든 객체가 가져다가 사용할 수 있다.

**객체배열**

ㆍ많은 수의 객체를 다뤄야 할 때 배열로 다루면 편리하다. 객체 역시 배열로 다룰 수 있다.

* 객체 배열 안에 객체가 저장되는 것이 아닌, 객체의 “주소”가 저장된다.

ㆍ객체 배열은 참조변수들을 하나로 묶은 참조 변수 배열이다.

ㆍ클래스명[] 배열명 = new 클래스명[ 배열크기 ];

* 초기값은 null로 자동 초기화 된다.
* 배열크기만큼의 객체 주소들을 저장하게 된다.

**다양위치의 변수 종류**

ㆍ선언 위치에 따른 변수의 종류

Class Variables {

Int i; //인스턴스/객체 변수

Static Int j; //클래스/공유 변수

Void method(){ int k; } //지역변수

}

ㆍ인스턴스/객체 변수 : 클래스 영역에서 선언되며, 클래스의 인스턴스를 생성할때마다 만들어진다.

* 객체A가 i값을 바꿔도 객체B의 i값은 변하지 않는다.
* 객체는 독립적인 저장공간을 가지므로 서로 다른 값을 가질 수 있다.
* 객체마다 고유한 상태를 유지해야하는 속성의 경우 객체변수로 선언하는 편이 좋다.

ㆍ클래스/공유 변수 : 모든 객체가 공유하며 사용할 수 있다.

* Static을 붙이기만 하면 된다.
* 객체마다 독립적인 저장공간을 갖는 객체변수와 달리, 모든 객체에게 공통된 저장공간(변수)를 공유하게 된다.
* 객체A가 j 값을 바꾸면 객체B의 j값도 바뀐다 🡺 “공유”변수의 의미이다.

ㆍ지역변수 : 메소드내에 선언되어 메소드 내에서만 사용가능하다

* 메소드가 종료되면 소멸된다.

ㆍ클래스변수, 객체변수의 특징

* 클래스 변수는 생성없이 사용 가능하나, 객체변수는 객체없이 사용이 불가능하다.
* Static변수 : 클래스명.변수명 으로 사용가능하다.
* 객체변수는 객체가 생성될때마다 생성되므로 객체마다 각기 다른 값을 유지한다. 클래스변수는 모든 객체가 하나의 저장공간을 공유하므로 항상 같은 값을 갖는다.

**메소드**

ㆍ메소드란, 특정 작업을 수행하는 기능/문장들을 하나의 단위로 묶어 둔 것이다.

* 수학에서의 함수와 유사하다
* 어떤 값을 입력하면 작업을 수행하여 그 결과를 반환한다.

ㆍ메소드의 장점

1. 높은 재사용성

* 한번 만들어놓은 메소드는 몇번이고 호출할 수 있다.
* 다른 프로그램에서도 사용할 수 있다.

1. 중복된 코드 제거

* 프로그램 작성시, 중복되는 작업이나 내용을 여러곳에서 똑같이 반복해서 사용하는 일이 많으므로, 이러한 반복되는 문장들을 묶어서 하나의 메소드로 만들면 단 한줄의 메소드 호출만으로 기능을 재현할 수 있다.
* 가독성이 향상된다.

1. 프로그램의 구조화

큰 규모의 프로그램을 작성할때는 작업단위로 나누어 여러 개의 메소드에 담아 프로그램의 구조를 단순화 시킨다. (= 모듈화 시킨다.)

ㆍ메소드의 형태

* 선언부와 구현부
* 자료형 메소드명(매개변수1, 매개변수2, …) { //<- 선언부

…실행할 문장; //<- 구현부

Return (위와동일한)자료형 데이터;

}

* 메소드의 선언부는 나중에 변경사항이 발생하지 않도록 신중하게 결정할 것
* (선언부를 변경한다면 그 메소드를 호출하는 모든 곳을 전부 변경해야 하기 때문)
* 자료형을 void로 설정하면 return이 필요없는 경우이다.

**재귀호출**

ㆍ재귀호출 (recursive) : 메소드 안에서 자기 자신을 다시 호출하는 문법

ㆍ 실무에서는 재귀호출은 비효율적이다. 메모리상 호출마다 메모리공간을 생성하기 때문에 메모리 낭비

* 재귀호출에 드는 비용보다 재귀호출의 간결함이 주는 이득이 충분할 경우에만 사용할 것