

4. Java 메소드(method)

1) 메소드(method); 작업을 수행하기 위한 명령문의 집합

- 절차지향 방식의 비효율을 보완하기 위한 객체 지향 방식(함수)
- 로직을 만들어 두고, 데이터가 들어가면 메소드가 알아서 결과값을 반환

접근제한자 [static] 리턴type 메소드명([매개변수1, 매개변수2, ...]) {//return이 없을 경우 void
처리할 프로세스들

[return 리턴 값]; //하나의 변수만 가능. 2개 이상의 값을 리턴하고 싶으면 배열로 묶어서 전달

}

- 객체 지향 프로그래밍으로 발전(∴데이터多→코드 양多)

※객체: 동일한 성질의 데이터와 메소드를 한 곳에 모아두고 필요한 곳에서 언제든지 이용할 수 있게 만들어 놓은 덩어리

5. 클래스(class)

1) 클래스 제작

Package com.ch.ex;

(1) 패키지명

public class ExClass {

(2) 클래스명

private 자료형 인스턴스 변수(=속성=필드)명;

(3) 데이터※

public ExClass() { }

(4) 생성자 함수※

public method() {

(5) 메소드

}

(6) getter & setter

}

(3) 데이터(인스턴스 변수=멤버 변수, 필드)

- 생성자나 setter를 이용해 초기화 하지 않으면 객체: null, 숫자: 0, boolean: false로 초기화

(4) 생성자 함수: 클래스명과 동일한 리턴타입이 없는 메소드를 생성자라 함.

- 처음 클래스형 객체를 만들 때 호출 됨.
- 모든 클래스는 반드시 하나 이상의 생성자 필요.
- 만약 하나도 없으면 JVM이 디폴트 생성자를 만들어 줌(new 연산자로 호출되는 메소드)

※객체지향 프로그래밍 특징

- ① 캡슐화(Encapsulation)와 데이터 은닉; 접근제어자(access modifier) 키워드 사용
- ② 다형성; 같은 모습이지만 다른 기능 cf.오버로딩
- ③ 상속