

메소드 사용 하는 이유 ?

1. 가독성이 좋다.
2. 유지보수 용이하다?
3. 캡슐화 (여러곳에서 중복된 코드를 작성 할 필요가 없다)

메소드 (method, 함수)

로직만 만들어 놓고, 그때 그때 데이터를 주면

메소드가 알아서 결과 값을 반환하는 방식.

객체

동일한 성질의 데이터와 메소드를 한곳에 모아두고

필요한 곳에서 언제든지 이용 할 수 있게 만든 덩어리

메소드 >>>

메소드란? 어떤 작업을 수행하기 위한 명령문의 집합

어떤 값을 입력 받아서 처리하고, 그 결과를 돌려줌

(입력 받는 값이 없을 수도 있고 결과를 돌려주지 않을 수도 있다)

메소드의 장점과 작성지침 ; 반복적으로 수행되는 여러 문장을

메소드로 작성한다.

```

package com.lec.ex;

public class Ex998 {
    public static void main(String[] args) {
        Ex997 king = new Ex997(); // 메소드의 클래스
        king.Strkingsugnbín(null, null);
        System.out.println(king.Strkingsugnbín(null, null));
    }
}

package com.lec.ex;

public class Ex997 {
    public void StrCombine(String a, String b) {
        // 메소드 StrCombine 을 호출 했을때
        // (같이 넣을 타입, 변수명)
        // void는 실행한다.
        String result = a + b;
        System.out.println(result);
    }
    public String Strkingsugnbín(String c, String d) {
        String result = c + d ; // return 리턴형 메소드
        return result;
    }
}

```

Ex997 : 메소드 클래스

1) Public void Strkingsugnbín (변수) (문자형, a ,b 변수 선언{

문자형 타입 result 변수 = a+ b ; 넣어주기

출력 (result)

2) public String Strkingsugnbín (변수) (문자형 , c, d 변수선언)

문자형 result(변수) = c + d 넣어주기

Return result ; 리턴형

출력할 패키지 Ex998 생성

메인함수

Ex998 타입에 king (변수) = new (새로운)

Ex998 > 클래스를 함축해서 king(변수)에 넣어준거다.

King.Strkingsungbin(null,null);

King에 Strkingsugnbín을 호출 했다.

하지만 아직 아무런 값이 없기 때문에

null로 표시가 됨.

1. 메소드 프로그램의 한계

- 메소드를 이용하면 로직의 재사용이 가능함 >개발을 효율적으로 할 수 있었음
- 절차지향언어에서 중복된 로직을 메소드의 사용으로 인해 효율성을 높일 수 있다.
- 하지만, 메소드만 가지고는 많은 양의 로직을 처리하기에 한계 발생
- 한 문서 내에 메소드의 수가 많아질 경우에 추후 유지보수 할 때 어려움 발생.

클래스의 기초적인 코딩방법

(cf) 캡슐화(Encapsulation) : 객체가 포함한

속성과 메서드는 객체간의 관계에 있어서

감추거나 권한에 따라 접근이 가능하게 처리하는 것을 말한다. 여기에 사용되는

Keyword로 접근제어자 (access modifier)가 있다.

클래스 제작

```
Package com.ch.ex;
```

```
Public class ExClass {
```

```
Private 자료형 인스턴스변수 (= 속성 = 필드) 명;
```

```
Public Exclass(){ }
```

```
Public method() {
```

```
... 
```

```
}
```

```
}
```

① 패키지명

② 클래스명

③ 데이터(인스턴스 변수=멤버변수, 필드) :

이 데이터는 생성자나 setter를 이용해서 초기화 하지 않는다면 객체는 null, 숫자는 0, boolean은 false로

초기화 되어 들어간다.

④ 생성자함수 : 클래스명과 똑같이 리턴타입이 없는

메소드를 생성자라 하며 처음 클래스형 객체를 만들때

호출된다. 모든 클래스는 반드시 하나 이상의 생성자 가 있어야 한다. 만약 하나도 없으면 JVM이 디폴트 생성자를 만들어 준다 (new 연산자로 호출되는 메서드)

생성자의 이해

생성자는 >

매개변수 있는 생성자 / 매개변수 없는 생성자

생성자가 없을때는 디폴트 생성자가 컴파일러 단계에서 자동생성.

한 개 이상의 생성자가 있을 경우

디폴트 생성자는 자동생성 되지 않음.

클래스는 속성(데이터), 메소드, 생성자, setter & getter

로 구성 되어 있음.