

2부 자바 기본 다루기

- 10장 메소드

최문환



10장 메소드

1. 사용자 정의 메소드 만들기
2. 메소드에 데이터를 전달해주는 방법
3. 결과 값을 되돌리는 방법

1. 사용자 정의 메소드 만들기

◆ 사용자 정의 메소드를 만드는 형식

```
자료형 메소드이름 (전달인자리스트) //메소드의 머릿부
{                                     //메소드의 몸체 시작을 알리는 중괄호
    변수 선언;
    문장;                             //메소드의 몸체
    return (결과 값);
}                                     //메소드의 몸체 끝을 알리는 중괄호
```

메소드는 메소드의 머릿부와 메소드의 몸체로 이루어 진다.
메소드의 몸체에는 메소드의 시작부분에는 열린 중괄호({)를
메소드의 끝부분에는 닫힌 중괄호(})를 사용
다양한 명령들을 기술

1. 사용자 정의 메소드 만들기

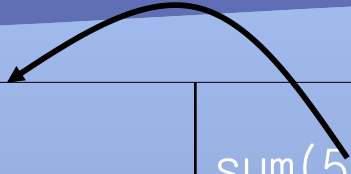
<예제> Hello World 출력하는 메소드 정의하기

```
01:public class MethodEx01 {  
02:    static void hello_func(){  
03:        System.out.println("Hello World!");  
04:    }  
05:    public static void main(String[] args) {  
06:        //사용자 정의 함수 호출  
07:        hello_func();  
08:    }  
09:}
```

2. 메소드에 데이터를 전달해주는 방법

```
static void sum(int n){  
    :  
}
```

```
sum(5);  
sum(10);
```



형식 매개변수(예 : 매개변수 n)

- 메소드를 호출할 때 넘어온 값을 담아두기 위한 기억 장소
- 메소드를 정의하는 측에서는 어떤 값이 전달될지 알 수 없다.
- 형식(자료형)만 맞으면 유효하다.
- 형식매개변수가 있는 메소드를 호출해서 사용하려는 쪽에서는 반드시 특정한 값을 넘겨주어야 한다.

실 매개변수

- 메소드를 호출할 때 기술한 5나 10처럼 실제로 프로그램에서 사용하는 데이터

<tip> void형 메소드

메소드의 리턴값이 없는 경우 “리턴이 없다”라는 것을 적극적으로 표현하기 위해서 void라는 특별한 자료형이 준비되어 있다.

<예제> 주어진 값까지의 합을 구하기

```
public class MethodEx02 {  
    //n은 메소드를 호출할 때 넘어온 값을 담아두기 위한 기억  
    static void sum(int n) {           //이를 형식매개변수라 함  
        int i;           //제어변수 선언  
        int tot=0;       //합을 누적할 변수 선언  
  
        for(i=1; i<=n; i++) //1부터 메소드로 전달된 값까지의 자연수를 구함  
            tot += i;       //합을 누적  
  
        System.out.println("1 ~ "+ n + " = " + tot); //구해진 합을 출력  
    }  
    public static void main(String[] args) {  
        sum(5);   //메소드를 호출할 때 실매개변수로 5를 넘겨줌  
        sum(10);  //메소드를 호출할 때 실매개변수로 10을 넘겨줌  
    }  
}
```

3. 결과 값을 되돌리는 방법

```
return (값);
```

메소드가 호출되면 메소드 몸체 부분의 내용이 수행된 후에 결과값을 가지고 되돌아 가게 할 경우가 있습니다.

이 때 사용하는 것이 return 문입니다

<예제> 절대값을 구하는 메소드 만들기

```
public class MethodEx03 {  
    //정수값을 하나 전달받음  
    static int abs(int data)    {  
        if(data<0)    //전달받은 정수값이 음수라면  
            data = -data; //-연산자를 이용하여 양수로 변경  
        return data;    //결과값을 되돌립니다.  
    }  
  
    public static void main(String[] args) {  
        int num=Integer.parseInt(args[0]);  
        System.out.println(" absolute data => "+abs(num));  
    }  
}
```

<문제>

1. name_func이란 이름의 자신의 이름을 출력하는 메소드 정의하시오.
2. 9단을 매개변수로 받아 해당 구구단을 출력하는 메소드 만드시오.