

제 17 강

메서드

교재: p94~99

목차

1. 메서드

1. 메서드란
2. 메서드의 사용목적
3. 메서드의 사용

1. 메서드의 정의

정의

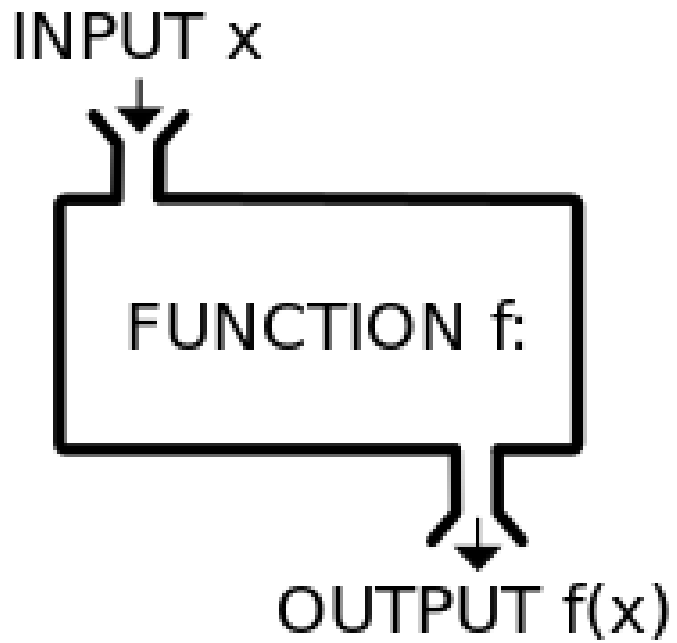
여러 명령들의 나열된 묶음
입력 값을 받아서 내부에서 처리 후 결과를 출력 또는 반환하는 역할

특징

메서드는 멤버 함수(member function)라고도 불리며,
클래스 내부에서 생성

1. 메서드의 정의

구조



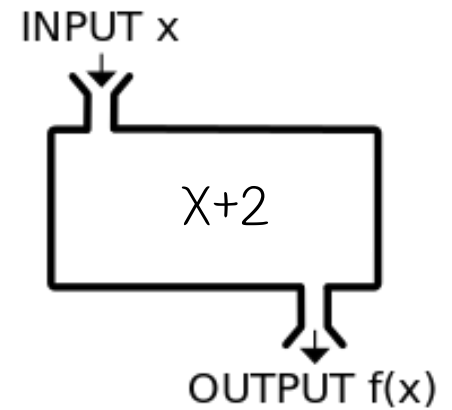
예시

$$f(x) = x + 2$$

입력값: x

출력값: $x + 2$

$$f(3) = ?$$



2. 메서드의 사용목적

Ex) 빵집 알바 썰...



문을 열어
바닥 청소 해
포스기 켜
빵을 진열해



2. 메서드의 사용목적

Ex) 빵집 알바 썰... 한달 뒤...



우리 약속하자
앞으로 "오픈해"
하면 이 것들 하는거야

1. 문을 열어
2. 바닥 청소 해
3. 포스기 켜
4. 빵을 진열해



2. 메서드의 사용목적

“여러 명령들을 **하나의 이름**을 호출(사용)함으로써,
한 번에 명령을 내리고자 사용하는 것이 바로 **메서드**다!”

3. 메서드의 사용

구조 <구현부>

```
반환타입 메서드명(타입 변수명, 타입 변수명) {  
    명령1;  
    명령2;  
    return 값;  
}
```

```
static void open() {  
    System.out.println("문을 열어");  
    System.out.println("바닥 청소 해");  
    System.out.println("포스기 켜");  
    System.out.println("빵을 진열해");  
}
```


3. 메서드의 사용

구조 <호출부>

메서드명();

```
public static void main(String[] args) {  
    Method1.open();  
}
```

3. 메서드의 사용

구조 <구현부>

```
반환타입 메서드명 (타입 변수명, 타입 변수명)
{
    명령1;
    명령2;
    return 값;
}
```

구조 <호출부>

```
메서드명(값1, 값2..);
```

매개변수

함수의 호출부에서 구현부로 전달하고 싶은 값이 있을 때 전달할 값을 저장할 공간

<실습> Method1.java

함수를 호출할 때 100이라는 값을 구현부에 전달하고 함수의 구현부에서 전달받은 값을 출력해보자!

3. 메서드의 사용

구조 <구현부>

```
반환타입 메서드명(타입 변수명, 타입 변수명)
{
    명령1;
    명령2;
    return 값;
}
```

구조 <호출부>

```
메서드명(값1, 값2..);
```

반환타입과 return 값

return 값?

: 함수에 구현된 모든 명령 실행 후,
함수가 갖게 될 결과 값

Return type(반환 타입)?

: 반환하는 값의 자료형

3. 메서드의 사용

반환타입과 return 값

return 값?

: 함수에 구현된 모든 명령 실행 후,
함수가 갖게 될 결과 값

Return type(반환 타입)?

: 반환하는 값의 자료형

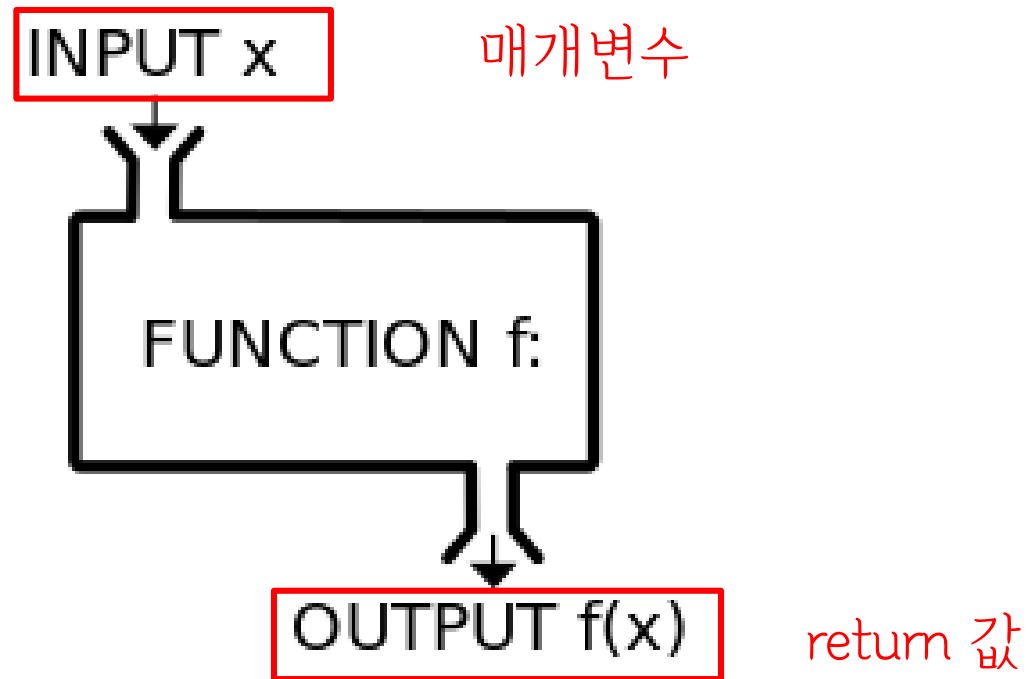
<실습> Method2.java

함수를 출력하면 100이라는 값이 출력되도록 구현하기

```
public class Method1 {  
    public static void main(String[] args) {  
  
        System.out.println(Method1.f());  
    }  
  
    static int f() {  
        return 100;  
    }  
}
```

3. 메서드의 사용

구조



3. 메서드의 사용(p95)

<실습> Exam-35.java

사각형의 넓이를 구하는 메서드를 구현하기

```
3 public class Method2 {  
4     public static void main(String[] args) {  
5         System.out.println("사각형의 넓이:"+Method2.calculator(5, 10));  
6     }  
7     static int calculator(int a,int b) {  
8         System.out.println("<계산을 시작합니다.>");  
9         int area=a*b;  
10        return area;  
11    }  
12  
13 }
```

<계산을 시작합니다.>
사각형의 넓이:50