

6장 함수

1. 함수(Method)정의

*파라미터=매개변수

- 특정 기능을 제공하기 위한 프로그램의 모듈
- 함수는 만들어져 있는 것을 사용하기도 하고, 직접 만들기도 함

```
함수타입 함수명 (파라미터 리스트 )  
{  
  
    실행문;  
  
}
```

```
int add(int a, int b)  
{  
  
    int c = 0;  
    c = a + b;  
    return c; /반 환 값  
}
```

1. 함수 정의 - 함수타입

- 함수 타입은 함수가 반환하는 값의 타입

	String 반환 함수 타입은 String
<pre>int intFunc() { return 1; }</pre>	<pre>String func() { String str= "hellö" ; return str; }</pre>

반환하는 값이 없을 경우 함수 타입은 void

```
void print()
{
    System.out.println();
}
```

1. 함수명과 파라미터(매개변수)

함수명은 변수명 규칙과 동일.

- 파라미터(매개변수)는 항상 있는 것 아니고, 필요한 경우에만, 필요한 개수만큼.
- 파라미터(매개변수)를 지정할 때는 변수 선언과 동일하게 타입과 이름을 명시.
- 함수 호출 시 넣어주는 값을 저장하기 위한 변수

예제)

```
void print()
{
    System.out.println("hello");
}
```

예제)

```
void print2(String str)
{
    System.out.println(str);
}
```

예제)

```
int sum(int kor, int eng, int math)
{
    return kor + eng + math;
}
```

2. 함수호출- 함수를 부름

- 함수는 호출하지 않으면 실행되지 않음.
- 함수 정의코드는 main()밖, 클래스 안에
- 함수 호출은 main()이나 다른 함수 안에서 할 수 있음
- add()함수가 호출될 때, argument(전달인자) 값이 복사되어 전달

2. 함수호출- 함수를 부름

```
public class Hello {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int c = 0;  
        c = add( 3, 5);  
        System.out.println(c);  
    }  
  
    static int add(int a, int b)  
    {  
        int c = 0;  
        c = a+ b;  
        return c;  
    }  
}
```

- a, b, c 변수는 모두 지역변수. 해당 함수, 블록 내에서만 유효.

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    int c = 0;  
    int x = 10, y = 20, z = 0;  
    z = add( x, y);  
    System.out.println(z);  
  
}
```

argument(전달인자)가 필요하면 넣어서 호출.

```
static int add(int a, int b)  
{  
    int c = 0;  
    c = a + b;  
    return c;  
}
```

```
public class Hello {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int a = 10, b = 5;  
        System.out.println("호출 전 --");  
        System.out.println(a);  
        System.out.println(b);  
  
        addOne(a, b);  
  
        System.out.println("호출 후 --");  
        System.out.println(a);  
        System.out.println(b);  
    }  
  
    static void addOne(int x, int y)  
    {  
        x++;  
        y++;  
        System.out.println("addOne함수");  
        System.out.println(x);  
        System.out.println(y);  
    }  
}
```


예제

- 세 개의 정수를 파라미터로 받아 이들의 합을 반환하는 함수.
- 위 함수를 호출하는 메인함수
- 함수 타입은 무엇에 의해 결정되나?
 - 파라미터 타입
 - 파라미터 개수
 - 리턴 값의 타입
 - 리턴 값의 개수

3. 메소드의 종료

```
public class Test {  
    int add(int a, int b)  
    {  
        int c;  
        c = a+b;  
        return c;  
    }  
  
    void print()  
    {  
        System.out.println("before return");  
        return;  
        System.out.println("after return");  
    }  
  
    int retInt()  
    {  
        int x = 10;  
        return x;  
    }  
  
    void noRet(int x)  
    {  
        System.out.println("x=" + x);  
    }  
}
```

*모든 함수를 호출하는 적당한 main()를 작성하라

4. 메소드의 다양한 형태

```
public class Test {  
    ➤ public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int result_int = 0;  
        float result_float = 0;  
        int a = 3, b = 5;  
        float x = 3.2f, y = 8.5f;  
        String result_string = null;  
        String str1 = new String("aaa");//str1 = "aaa";와 동일  
        String str2 = new String("bbb");//str2 = "bbb";와 동일.  
  
        result_int = add_int(a, b);  
        System.out.println(result_int);  
  
        result_float = add_float(x, y);  
        System.out.println(result_float);  
  
        result_string = add_string(str1, str2);  
        System.out.println(result_string);  
    }  
    ➤ static int add_int(int a, int b)  
    {  
        int c = 0;  
        c = a+b;  
        return c;  
    }  
    ➤ static float add_float(float a, float b)  
    {  
        float c = 0;  
        c = a+b;  
        return c;  
    }  
    ➤ static String add_string(String str, String str2)  
    {  
        String str3 = null;  
        str3 = str + str2;  
        return str3;  
    }  
}
```

4. 메소드의 다양한 형태 - 배열 파라미터

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    int[] arr1 = { 1, 2, 3, 4, 5};  
    int max = 0;  
    int sum = 0;  
    float avg = 0.0f;  
  
    max = makeMax(arr1);  
    System.out.println(max);  
  
    sum = makeSum(arr1);  
    System.out.println(sum);  
  
    avg = makeAvg(arr1);  
    System.out.println(avg);  
  
    int[] arr2 = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
  
    max = makeMax(arr2);  
    System.out.println(max);  
  
    sum = makeSum(arr2);  
    System.out.println(sum);  
  
    avg = makeAvg(arr2);  
    System.out.println(avg);  
}
```

4. 배열 파라미터

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    int[] arr1 = { 1, 2, 3, 4, 5};  
    int max = 0;  
    int sum = 0;  
    float avg = 0.0f;  
  
    max = makeMax(arr1);  
    System.out.println(max);  
  
    sum = makeSum(arr1);  
    System.out.println(sum);  
  
    avg = makeAvg(arr1);  
    System.out.println(avg);  
  
    int[] arr2 = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
  
    max = makeMax(arr2);  
    System.out.println(max);  
  
    sum = makeSum(arr2);  
    System.out.println(sum);  
  
    avg = makeAvg(arr2);  
    System.out.println(avg);  
}
```

```
static int makeMax(int[] arr)  
{  
    int max = 0;  
    for (int i = 0; i < arr.length; i++)  
    {  
        if ( max < arr[i])  
        {  
            max = arr[i];  
        }  
    }  
    return max ;  
}  
  
static float makeAvg(int[] arr)  
{  
    float avg = 0;  
    avg = (float)makeSum(arr) / arr.length;  
    return avg;  
}  
  
static int makeSum(int[] arr)  
{  
    int sum = 0;  
    for (int i = 0; i < arr.length; i++)  
    {  
        sum = sum + arr[i];  
    }  
    return sum;  
}
```

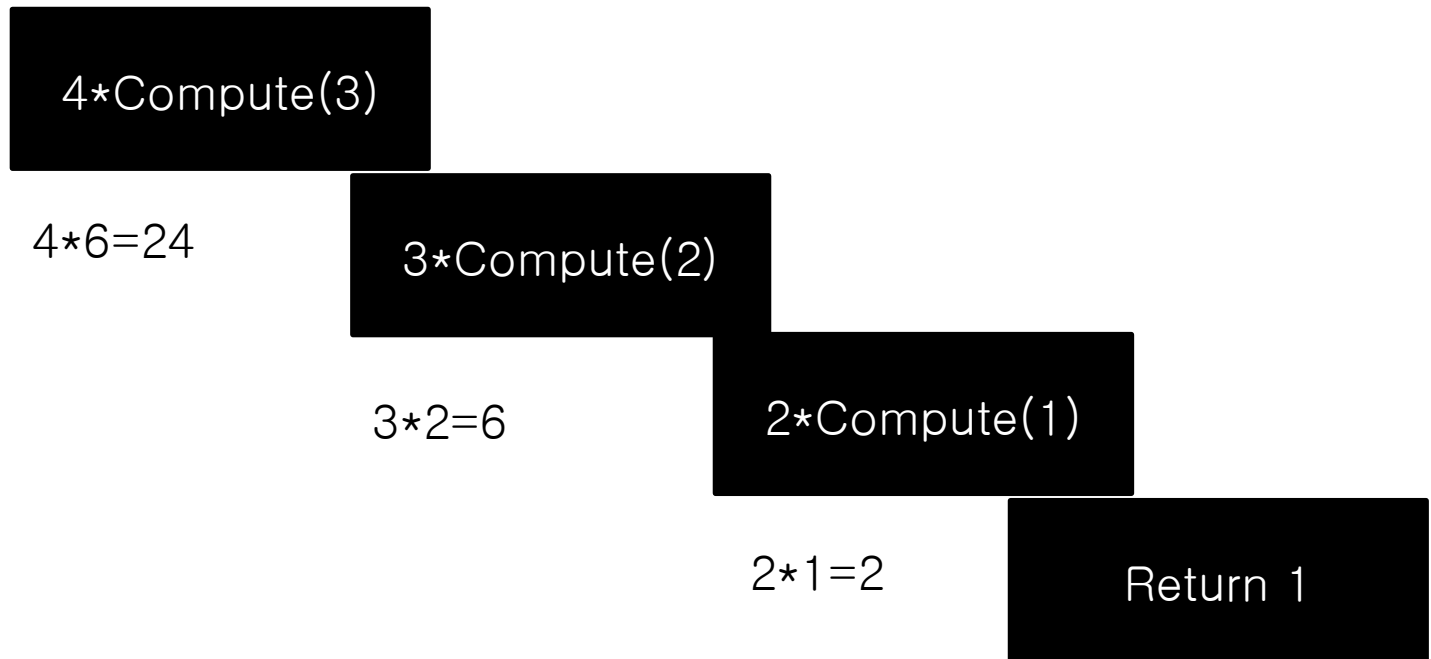
4. 재귀 메소드

- 메소드 내에서 자신을 호출하는 메소드.
- 반복 동작을 구현하기 편하나 동시에 많은 메소드가 호출되어 스택 메모리의 사용량이 크게 늘고 부하 줄 수 있음. 반복문으로 대체.

```
public class Hello {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int result = compute(4);  
        System.out.println(result);  
    }  
  
    static int compute(int x)  
    {  
        int ret = 0;  
  
        if ( x == 1 )  
        {  
            ret = 1;  
        }  
        else  
        {  
            ret = x * ( compute(x-1));  
        }  
  
        return ret;  
    }  
}
```

4.메소드의 다양한 형태 - 재귀메소드

Compute(4)



예 제

1. sum()함수를 완성하시오.

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    int x, y;  
    int[] a1 = { 1, 2, 3};  
    int[] a2 = {4, 5, 6};  
  
    x = sum(a1); // 배열의 요소들의 합을 구하는 sum함수  
    y = sum(a2);  
  
    System.out.println(x);  
    System.out.println(y);  
  
}
```


2. 두 개의 2차원 배열에 임의의 값(1-9)을 부여한 후, 그 2차원 배열(행렬)의 합과 곱을 구하는 프로그램을 작성하라

```
Problems Declaration Console
<terminated> Hello [Java Application] C:\Program Files\Java\jdk-9.0.4\bin
4      6      2      9
3      8      4      1      배열
-----
2      7      1      4
7      1      2      5      배열
-----
6      13     3      13
10     9      6      6      ← 합
-----
8      42     2      36
21     8      8      5      ← 곱
```

3. 합과 곱을 구하는 메소드(함수)를 각각 정의하여 그것을 활용한 프로그램을 작성하라. 배열의 내용을 출력하는 함수도 추가하라.

4. 랜덤한 숫자(1-10)를 리턴해주는 함수(함수명 `rando`)를 작성하고, `main()`에서 호출하라.

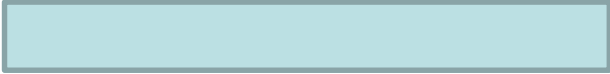
Hint) `Math.random()`

5. 4에서 만든 함수를 활용하여, 1과 9사이의 중복되지 않은 숫자로 이루어진 3자리 숫자를 만들어내는 프로그램을 작성하시오.

함수 정의를 하시오

```
package uu;

public class Three {

    public static void main(String[] args) {
        int i = 5, j=2,k;
        k = sub(i, j);
        System.out.println("k="+k+"i"+" "+j);
    }
    
    {
        return i-j;
    }
}
```

함수 정의하시오

```
public class Method {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int result;  
        result=pow(3);  
        System.out.println("3x3=" + result);  
    }  
  
    //함수 정의  
  
}
```

함수 호출

```
public class Method {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int age=0;  
        //age값을 28로 할당될 수 있도록 함수를 호출한다  
  
    }  
  
    static int setAge(int a)  
    {  
        return a+20;  
    }  
  
}
```

```

public static void main(String[] args) {
// TODO Auto-generated method stub
int[] star = new int[10];

System.out.println("-----");
for(int i = 0; i < 10; i++)
{
    System.out.print(star[i]+"Wt");
}
System.out.println();

for(int i = 0; i < 10; i++)
{
    star[i] = i+1;

System.out.println("-----");
for(int i = 0; i < 10; i++)
{
    System.out.print(star[i]+"Wt");
}
System.out.println();

```

```

for(int i = 0; i < 10; i++)
{
    star[i] = star[i] + 1;
}

System.out.println("-----");
for(int i = 0; i < 10; i++)
{
    System.out.print(star[i]+"Wt");
}
System.out.println();

} //main

```

반복되는 부분을 찾아,
함수로 분리하시오

함수를 정의하시오

```
public class Method {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int i = 6, j = 9;  
        int result = mul(i, j);  
        System.out.println(mul);  
    } //main  
  
    //함수 정의  
  
}
```


함수를 정의하시오

```
public class Method {  
  
    public static void main(String[] args) {  
  
        float result = add(3.14f, 3.14f);  
  
    } //main  
  
    //함수 정의  
  
}
```

함수정의와 호출을 하시오

```
public class Method {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int[] array = new int[3];  
        for ( int i = 0; i < 3; i++)  
        {  
            array[i] = i+1;  
        }  
  
        //array값을 변경하는 함수를 호출하라.  
        //배열의 값을 출력해주는 함수를 호출하시오.  
  
    }//main  
  
    //array값을 변경하는 함수  
    static void changeArray(int[] arr)  
    {  
        for(int i = 0; i < arr.length; i++)  
        {  
            arr[i] = arr[i] + 2;  
        }  
  
        //배열의 값을 출력해주는 함수 호출하시오  
    }  
  
    //배열의 값을 출력해주는 함수 정의.  
  
}
```

출력함수 2가지를 정의하시오

```
public class Method {
```

```
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        String[] student = {"박진수", "장원우", "강한솔", "최진욱"};  
  
        print(student);  
  
        printJjang(student, "조장");  
        printJjang(student, "짱!");  
    }
```

```
    //1) 출력 함수를 정의하시오
```

```
    //2) 출력 함수를 정의하시오
```

```
    // 박진수 조장  
    // 장원우 조장  
    // 강한솔 조장  
    // 최진욱 조장  
    //위와 같이 출력되도록 하시오
```

```
}
```

```
public class Method {
```

```
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String[] fruit = {"banana", "orange", "peach", "strawberry", "cherry"};

        String storeName=new String("싱싱과일가게");

        //fruit의 배열의 값인 과일 이름의 철자가 가장 긴 과일을 찾는 함수를 호출하라

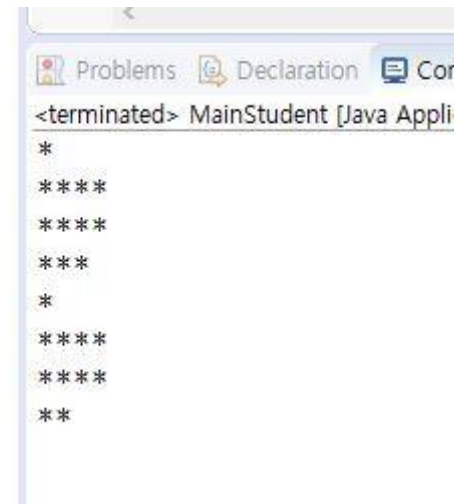
        //아래 빈칸을 채우시오
        //출력은
        //싱싱과일가게에서 가장 긴 이름을 가진 과일은 strawberry
        System.out.println(storeName + "에서 가장 긴 이름을 가진 과일은 "+      );

    }
    /*가장 긴 철자를 가지고 있는 과일이름 찾는 함수*/
    static String maxLengthFruit(String[] s)
    {
        String result=null;//과일이름 저장변수
        int max=0;//철자의 길이 저장변수
        for ( int i = 0; i < s.length; i++)
        {
            if ( s[i].length()> max)//철자의 길이를 돌려주는 함수 호출.
            {
                max = s[i].length();
                result=s[i];
            }
        }
        return result;
    }
}
```

11. 배열 answer에 담긴 데이터를 읽고 각 숫자의 개수를 세어서 개수만큼 '*'를 찍어서 그래프를 그리는 프로그램.

11-1. 출력하는 함수로 분리하라,

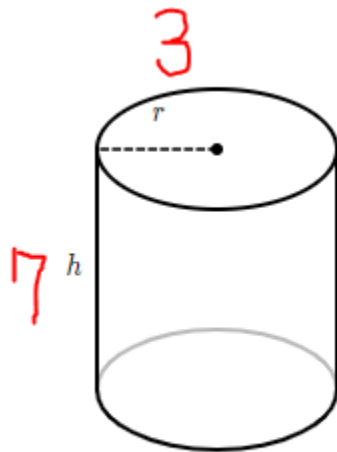
11-2. *, #, @ 등 입력 받는 문자로 출력할 수 있도록 한다



```
<terminated> MainStudent [Java Appli
*
*****
*****
***
*
*****
*****
**
```

- 1학년 1반 친구들 10명의 몸무게, 키, 신발 사이즈를 입력 받은 후, 몸무게 중 가장 큰 값을 출력하고, 키 값중 가장 큰 키값을 출력하고, 신발사이즈중 가장 큰 사이즈값을 출력하시오.

- 원기둥의 부피는 아래 식과 같이 구한다.
원기둥의 부피를 출력하라. 반지름과 높이를 사용자에게 입력 받는 함수로 분리해서 작성하라



$$V = \pi r^2 h$$

```
Problems @ Javadoc
<terminated> haft [Java Appli
원기둥의 부피를 구하시오
반지름은?
5
높이는?
6
부피는 471.0입니다.
```

- 극장예약시스템

- 배열을 이용하여 간단한 극장예약시스템을 작성하여 보자. 좌석은 10개 이고, 사용자가 예약을 하려고 하면 먼저 좌석 배치표를 보여준다. 예약이 끝난 좌석은 1로, 예약이 안 된 좌석은 0으로 나타낸다. -1이 입력될때, 예약시스템이 종료된다. (최대한 함수로 분리할 수 있는 기능들은 분리한다)