

능동적 사고 방식의

java

강사 박주병

Park Ju Byeong

Park Ju Byeong



Part04 반복문

01 for

02 while

03 do while, 중첩 반복문

04 실습 문제



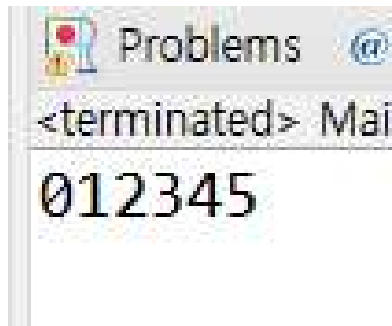
01

for

실습문제1

1. For 반복문을 이용하여 아래와 같이 출력해보자(normal)
 - System.out.print() 를 사용하면 줄이 넘어가지 않는다.

```
for(int i =0;i<=5;i++)  
    System.out.print(i);
```



해설

1.반복문 진입시 최초 한번만 실행된다.

int 타입의 변수를 선언하며 값은 0으로 초기화 하였다.

2.조건식이 true일 경우 for문 내부로 들어간다.

```
for(int i =0;i<=5;i++)  
    System.out.print(i);
```

4.i의 값을 1증가 시킨다.

3.조건식이 true일 경우 for문 내부로 들어간다.

2. 숫자 1 부터 10까지의 합계를 출력하시오(normal)

```
int sum = 0;
```

```
1부터0까지의 합:0  
1부터1까지의 합:1  
1부터2까지의 합:3  
1부터3까지의 합:6  
1부터4까지의 합:10  
1부터5까지의 합:15  
1부터6까지의 합:21  
1부터7까지의 합:28  
1부터8까지의 합:36  
1부터9까지의 합:45  
1부터10까지의 합:55
```

해설

```
int sum = 0;
```

합계를 저장해놓을 변수
(반복문 내에서 변수를 생성시 값을 누적할수 없다)

```
for(int i=0; i<=10;i++)  
{  
    sum += i;  
    System.out.println("1부터"+i+"까지의 합:"+sum);  
}
```

반복문이 실행될때마다 sum 변수에 i의 값을 누적시킨다.

`+=` 복합연산자는 `sum=sum+i` 와 동일하다.

3. 0부터 10까지 증가 하는 값과 10부터 0까지 감소하는 값을 출력하시오(hard)

- 중첩 반복문은 필요 없음

```
<terminate
0 10
1 9
2 8
3 7
4 6
5 5
6 4
7 3
8 2
9 1
10 0
```


해설

```
int max = 10;  
  
for(int i=0; i<=max; i++)  
{  
    System.out.println(i + " " + (max-i));  
}
```

두번째 값은 10부터 1씩 감소한다.
즉 i변수는 1씩 증가 되므로 i변수 만큼 빼주면 된다.

4. 아래의 그림과 같이 삼각형 모양의 별을 출력하시오(hard)

- 중첩 반복문을 사용해야 한다.
- System.out.println(); 을 사용하면 줄을 넘길수 있다.
- System.out.print('*'); 사용시 줄을 넘기지 않고 출력한다.

```
<terminated> m
*
**
***
****
*****
```

해설

```
for(int i = 0 ; i < 5 ; i++)  
{  
    for(int j = 0 ; j <= i ; j++)  
    {  
        System.out.print("*");  
    }  
    System.out.println();  
}
```

세로가 몇줄인지를 결정한다.

가로로 별을 몇번 출력할지 결정하는데
한줄 넘어갈때 마다 반복 횟수가 추가가
된다.

한줄에 대해 별을 모두 출력하면
다음줄로 줄넘김한다.

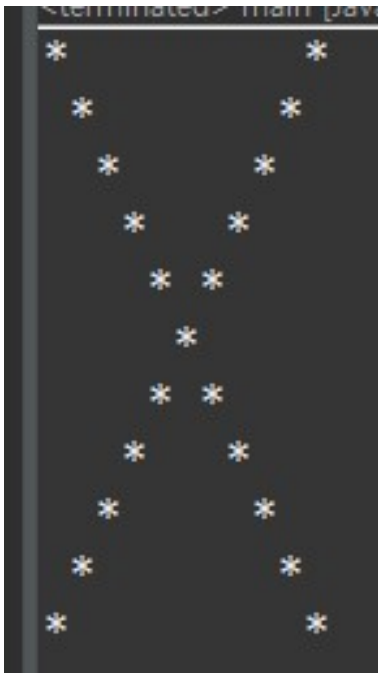
5. 아래의 그림과 같이 구구단을 출력하시오(9단까지)(hard)

```
<terminated> main [Jav
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
```

```
for(int i = 2 ;i<9; i++)
{
    for(int j = 1 ;j<=9; j++)
    {
        System.out.println(i+" x " +j +" = "+ i*j);
    }
}
```

중첩 반복문을 이용해 두개의 변수를 곱한다.

6. 아래의 모양을 출력하시오(expert)



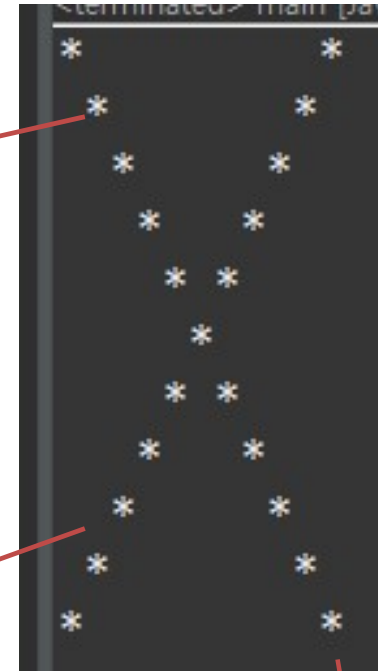
해설

*이 출력되어야 하는 위치를 좌표로 이해하면 쉽다. 왼쪽 가장위부터 0,0 좌표로 보고 특정 좌표에서만 *을 출력하고 나머지는 공백 1개를 출력 하면 된다.

```
final int MAX = 10;

for(int i = 0 ;i<=MAX; i++)
{
    for(int j = 0 ;j<=MAX; j++)
    {
        if(i==j || MAX-i==j)
            System.out.print("*");
        else
            System.out.print(" ");
    }
    System.out.println();
}
```

1,1 좌표이다.



해당 사선의 좌표는 (10,1) (9,2) (8,3)...
즉 줄넘김이 한번 일어 날때마다 x좌표가
줄어든다.

해당 사선의 좌표는 (1,1) (2,2) (3,3)...
즉 두개의 좌표가 동일하다.



—• 02

while

실습문제2

1. While 문을 이용해 1부터 10까지 출력하시오(normal)

```
<terminated> Print
1
2
3
4
5
6
7
8
9
10
```

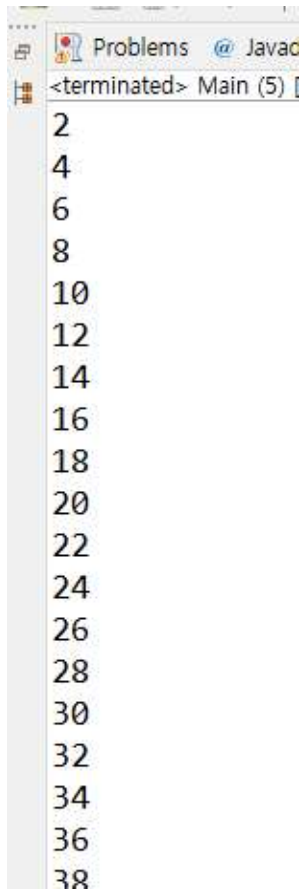
```
int i=1;

while(i<=10)
{
    System.out.println(i);
    i++;
}
```

→ i가 10보다 작으면 계속 실행된다.

→ 내부에서 반복할수록 i의값을 증가 시키므로 언젠간 종료된다.

2. while문을 이용해 1부터 100 숫자중 짝수만 출력하시오(normal)



```
<terminated> Main (5) [
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
```

```
int cnt=1;
while(cnt<=100)
{
    if(cnt %2 ==0)
        System.out.println(cnt);

    cnt++;
}
```

→ 2로 나누어떨어지면 짝수이다.

3. $1+2+3+4...$ 합계를 누적하여 몇까지 올라가야 누적합계가 150이상 되는지를 출력하시오 (hard)

```
<terminated> main [Java Application] C:\Users\W
누적 합계:28   현재 숫자 : 7
누적 합계:36   현재 숫자 : 8
누적 합계:45   현재 숫자 : 9
누적 합계:55   현재 숫자 : 10
누적 합계:66   현재 숫자 : 11
누적 합계:78   현재 숫자 : 12
누적 합계:91   현재 숫자 : 13
누적 합계:105  현재 숫자 : 14
누적 합계:120  현재 숫자 : 15
누적 합계:136  현재 숫자 : 16
누적 합계:153  현재 숫자 : 17
```

```
int sum = 0;
int i=0;
while(sum<150)
{
    sum +=++i;
    System.out.println("누적 합계:"+sum+ "   현재 숫자 : "+i);
}
```

반복문 밖에서 만들어야
누적이 가능하다.

몇번 반복이 아닌 특정조건이
만족될때까지이다.

4. 사용자로부터 숫자를 입력 받아 while을 이용해 각 자리의 합을 구하시오(hard)

예) 12345 -> 1+2+3+4+5 =15

```
숫자를 입력하세요:  
12345  
현재 합계 : 5  
현재 합계 : 9  
현재 합계 : 12  
현재 합계 : 14  
현재 합계 : 15  
각 자리의 합계:15
```

```
Scanner scan = new Scanner(System.in);  
  
System.out.println("숫자를 입력하세요:");  
int number = scan.nextInt();  
  
int sum = 0;  
while(number > 0){  
    // 현재 수의 합계는 현재 1의 자리와 숫자를 더하고 10으로 나눈다.  
    sum += number % 10;  
  
    System.out.println("현재 합계 : " + sum);  
  
    // 10으로 나누어 2의 자리를 찾는다.  
    number /= 10;  
}  
  
System.out.println("각 자리의 합계:" + sum);
```

해설

```
Scanner scan = new Scanner(System.in);

System.out.println("숫자를 입력하세요:");
int number = scan.nextInt();

int sum = 0;
while(number != 0)
{
    //나머지연산을 하여 1의 자리의 숫자를 가져와 더한다.
    sum += number%10;

    System.out.println("현재 합계 : " + sum);

    //10으로 나누어 1의 자리를 없앤다.
    number /= 10;
}

System.out.println("각 자리의 합계:" + sum);
```

반복될수록 number가 줄어든다.

— 04

실습문제

실습문제3

1. 10부터 1까지 숫자를 반복해서 출력하시오(normal)



```
Problems Javadoc S
<terminated> Main (5) [Java App
10
9
8
7
6
5
4
3
2
1
```

```
for(int i=10;i>=1 ;i-- )
{
    System.out.println(i);
}
```

감소식도 가능하다.

2. do while문을 이용하여 숫자 맞추기 게임을 만들어보자 (normal)

- 컴퓨터는 1~100사이의 랜덤한 숫자를 저장하고 유저는 숫자를 입력받아 값을 맞추도록하자
- 정답을 맞출시 몇번만에 성공한것인지도 출력해보자

```
<terminated> main [Java Application] C:\Users\Wze  
15  
15 보다 작습니다.  
1~100 사이의 숫자를 입력하세요:  
7  
7 보다 작습니다.  
1~100 사이의 숫자를 입력하세요:  
3  
3 보다 작습니다.  
1~100 사이의 숫자를 입력하세요:  
2  
정답입니다. 6 번 시도하였습니다.
```

```
int count=0;  
Scanner scanner = new Scanner(System.in); //입력을 받기 위한 스캐너 객체를  
  
int computer =(int)(Math.random()*100)+1;  
int user;  
do  
{  
    //시도 횟수를 올린다.  
    count++;  
  
    System.out.println("1~100 사이의 숫자를 입력하세요:");  
    user = scanner.nextInt(); //키보드로부터 값을 입력 받아 저장한다.  
    if(computer > user)  
        System.out.println(user+" 보다 큼니다.");  
    else if(computer < user)  
        System.out.println(user+" 보다 작습니다.");  
    else  
    {  
        System.out.println("정답입니다. " + count+" 번 시도하였습니다.");  
    }  
}  
  
}while(computer != user);
```

탐색 알고리즘

1. 순차탐색 : 모든 데이터를 전부 확인하면서 찾아낸다.
2. 이진탐색 : 정렬된 데이터에서 절반씩 줄여나가면서 찾아낸다.
3. 해시 탐색 : 데이터와 그 위치를 연결 지어 보관하고
해시알고리즘을 이용해 탐색

빅오표기법과 시간복잡도

1. 순차탐색 : 데이터가 100개면 최악의 경우 100번만에 찾는다. $O(N)$
2. 이진탐색 : 데이터가 100개면 최악의 경우 7번만에 찾는다. $O(\log n)$
3. 해시 탐색 : 데이터가 100개면 해시충돌이 없다면 1번, 충돌 한다면 최악의 경우 100번만에 찾는다. $O(1), O(n)$

해설

```
int count=0;
Scanner scanner = new Scanner(System.in); //입력을 받기 위한 스캐너 객체를

int computer =(int)(Math.random()*100)+1;
int user;
do
{
    //시도 횟수를 올린다.
    count++;

    System.out.println("1~100 사이의 숫자를 입력하세요:");
    user = scanner.nextInt(); //키보드로부터 값을 입력 받아 저장한다.
    if(computer > user)
        System.out.println(user+" 보다 큼니다.");
    else if(computer < user)
        System.out.println(user+" 보다 작습니다.");
    else
    {
        System.out.println("정답입니다. " + count+" 번 시도하였습니다.");
    }
}while(computer != user);
```

틀리면 반복적으로 입력을 받아야 함으로
반복문 내에서 값을 입력 받는다.

실습문제4

3. 1~100 사이의 소수를 출력하여보자(hard)

- 소수: 1과 자기자신으로만 나누어 떨어지는수
- 중첩반복문을 사용해야 한다.



```
<terminated> main [Java Application] C:\Users\wzest1\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot  
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53
```

해설

```
int isPrimeNumber;
for (int i = 1 ; i <= 100 ; i++)
{
    for(isPrimeNumber=2;isPrimeNumber<i;isPrimeNumber++)
        if(i%isPrimeNumber == 0) // 1과 자기자신외에 수가 나누어 떨어진다면 더 볼것 없이 소수가 아니다.
            break;

    if(i==isPrimeNumber)
        System.out.print(i+" ");
}
}
```

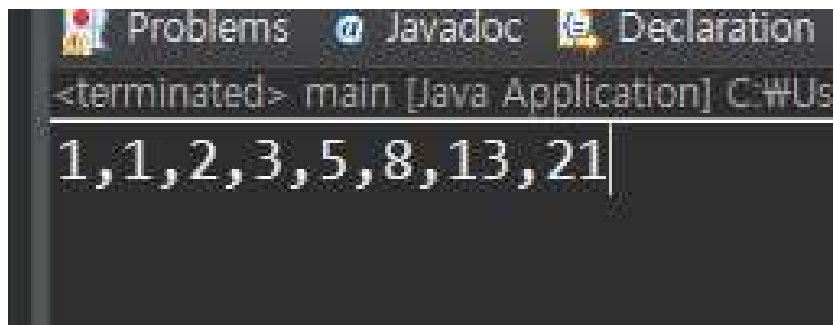
1은 모두 나누어떨어지니 고려하지 않는다.

소수를 찾는것이 아닌 소수가 아니라고 판명나면 break를 통해 빠져 나가고 break에 걸리지 않고 자기자신까지 반복되었다면 소수인것이다.

4. 1과 1부터 시작 하는 피보나치수열로 12번째 숫자가 무엇인지 출력하여보자(expert)

피보나치 수열 : 앞의 두 수를 더하여 다음의 수를 만들어 내는 수열이다.

ex) 1, 1, 2, 3, 5, 8, 13



```
Problems Javadoc Declaration
<terminated> main [Java Application] C:\#Us
1,1,2,3,5,8,13,21
```

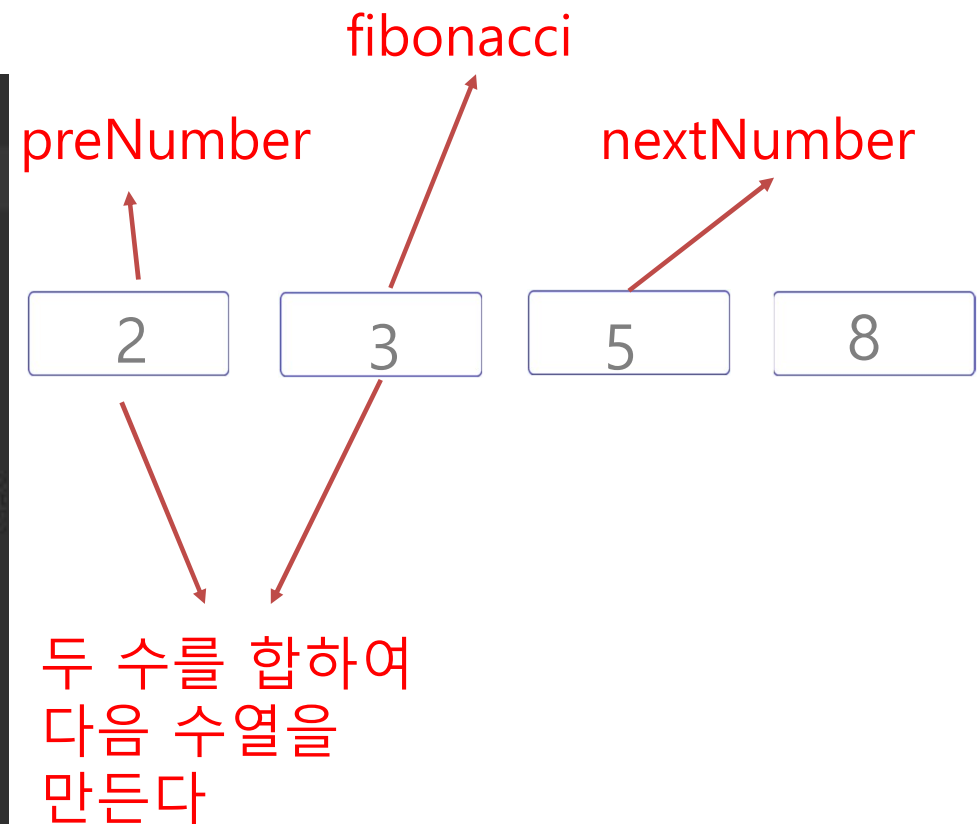
해설

```
int preNumber=0;
int fibonacci=1;
int nextNumber=1;

for(int i =0;i<12;i++)
{
    nextNumber = preNumber+fibonacci;

    System.out.print(fibonacci+" ");

    preNumber=fibonacci;
    fibonacci = nextNumber;
}
```



다음수를 구했다면 한칸씩 당긴다.

5. 아래의 그림과 같이 구구단을 출력하시오(expert)

- 일정한 간격만큼 띄우고 싶으면 \t 를 문자열에 포함시키면 된다.

ex) "2*1=2 \t 3*1=3"

2*1=2	3*1=3	4*1=4
2*2=4	3*2=6	4*2=8
2*3=6	3*3=9	4*3=12
5*1=5	6*1=6	7*1=7
5*2=10	6*2=12	7*2=14
5*3=15	6*3=18	7*3=21
8*1=8	9*1=9	
8*2=16	9*2=18	
8*3=24	9*3=27	

해설

```
for (int i = 1 ; i <= 9 ; i++)
{
    for (int j = 1; j <=3; j++)
    {
        int x=(j+1)+(i-1)/3*3;
        int y= i%3==0 ? 3:i%3;

        if(x>=10)
            break;

        System.out.print(x+"*" + y+ "=" + x*y+"\t");
    }
    System.out.println();
    if(i%3==0)
        System.out.println();
}
```

수식이 복잡함으로
print문에서 바로 적지 않고
따로 분리하였다.

10단은 출력하지 않는다.

3줄 다음엔 2줄을 띄운다.

2*1=2	3*1=3	4*1=4
2*2=4	3*2=6	4*2=8
2*3=6	3*3=9	4*3=12
5*1=5	6*1=6	7*1=7
5*2=10	6*2=12	7*2=14
5*3=15	6*3=18	7*3=21
8*1=8	9*1=9	
8*2=16	9*2=18	
8*3=24	9*3=27	



THANK YOU



강사 박주병