

컴퓨터 프로그래밍1

실습 7주차

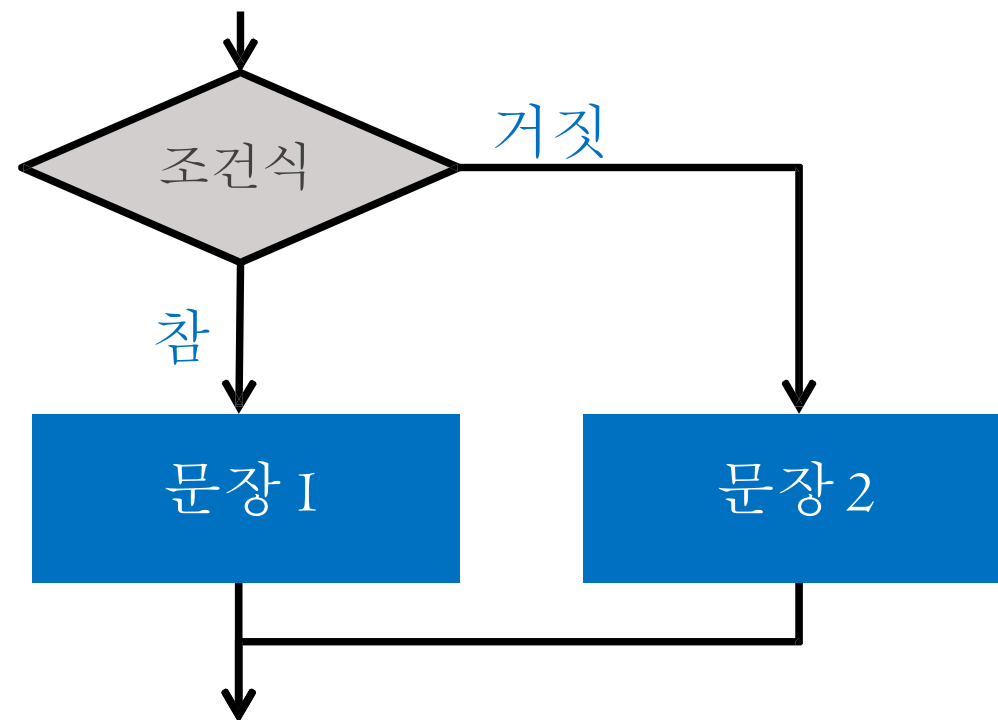
- 조건문, 반복문 -

조건문 (1/8)

if 문

if-else 문은 다음과 같은 형식을 가짐

```
if(조건식)  
    문장1;  
else  
    문장2;
```



조건식이 참으로 계산되면 문장 1이 실행되고, 거짓으로 계산되면 문장 2가 실행

조건식이 참일 때 실행되는 블록을 then 절, 거짓일 때 실행되는 블록은 else 절

조건문 (2/8)

프로그래밍 예제

```
public class Pay {  
    public static void main(String args[]) {  
        final int RATE = 5000;  
        int pay; // 임금  
        int hours; // 시간  
        Scanner input = new Scanner(System.in);  
  
        System.out.print("시간을 입력하시오: "); // 입력 안내 출력  
        hours = input.nextInt();  
  
        if (hours > 8)  
            pay = RATE * 8 + (int) (1.5 * RATE * (hours - 8));  
        else  
            pay = RATE * hours;  
        System.out.printf("임금은 %d입니다.\n", pay);  
    }  
}
```

시간을 입력하시오: 8
임금은 40000입니다.

시간을 입력하시오: 9
임금은 47500입니다.

조건문 (3/8)

중첩 if 문

if-else 문 사이에 다른 if문이 삽입되는 경우를 중첩 if 문이라고 함

```
if( grade >= 80)
    if( grade >= 90)
        System.out.println("당신의 학점은 A입니다.");
    else
        System.out.println("당신의 학점은 B입니다.");
```

else 절은 가장 가까운 if절과 매칭 됨

확실하게 else를 매칭시키려면 {, }로 묶어주면 됨

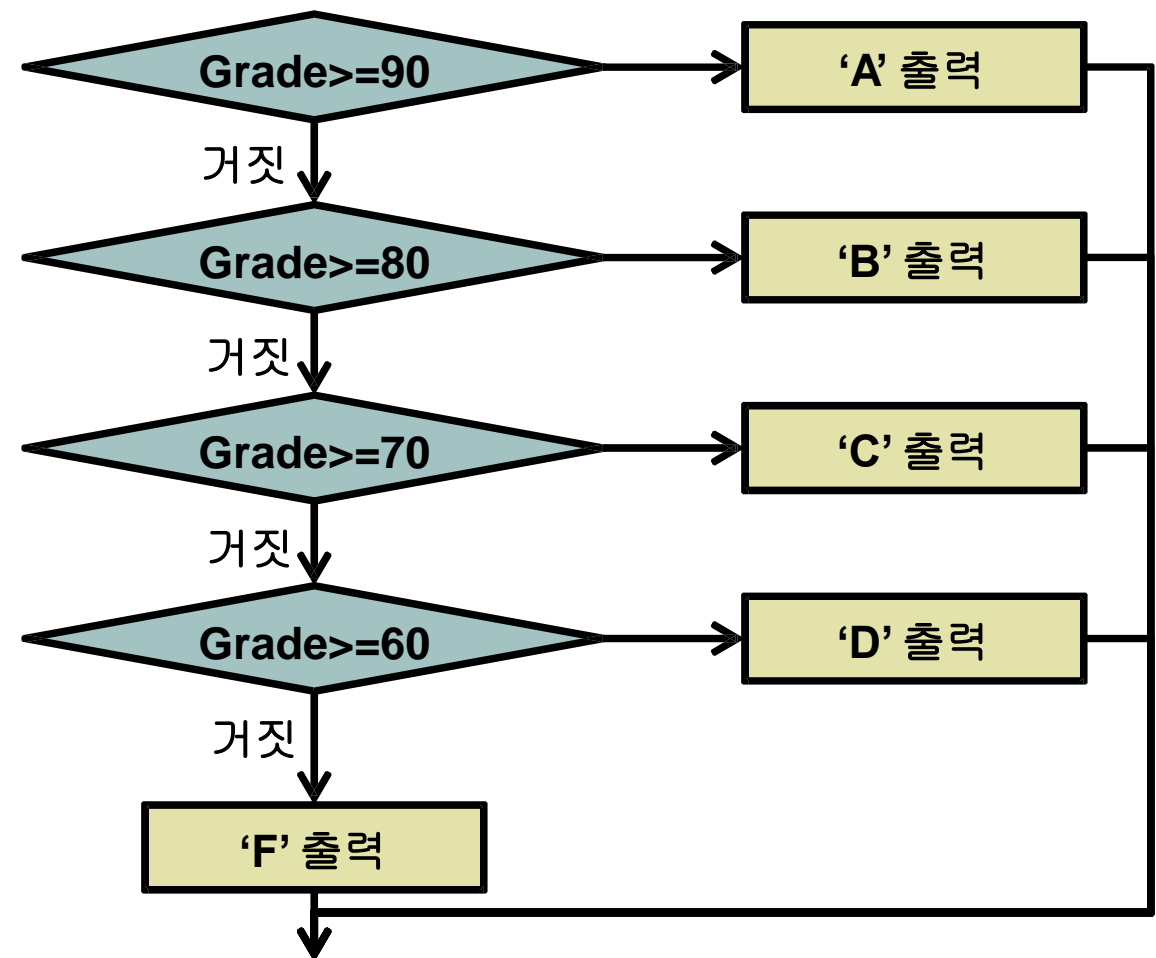
```
if( grade >= 80){
    if( grade >= 90)
        System.out.println("당신의 학점은 A입니다.");
} else
    System.out.println("당신의 학점은 C입니다.");
```

조건문 (4/8)

연속적인 if 문

여러 조건에 의해서 다중으로 분기되는 경우, if 문 다음에 else if 문을 사용하여 해결

```
if( grade >= 90 )  
    System.out.println("A");  
else if( grade >= 80 )  
    System.out.println("B");  
else if( grade >= 70 )  
    System.out.println("C");  
else if( grade >= 60 )  
    System.out.println("D");  
else  
    System.out.println("F");
```



조건문 (5/8)

프로그래밍 예제 (p.108)

```
public class Tax {  
    public static void main(String args[]) {  
        int income; // 과세 표준  
        int tax; // 세금  
        System.out.print("과세 표준 금액을 입력하시오:");  
        Scanner input = new Scanner(System.in);  
        income = input.nextInt();  
        if (income <= 1000)  
            tax = (int) (0.09 * income);  
        else if (income <= 4000)  
            tax = (int) (0.18 * income);  
        else if (income < 8000)  
            tax = (int) (0.27 * income);  
        else  
            tax = (int) (0.36 * income);  
        System.out.println("소득세는 " + tax + "입니다.");  
    }  
}
```

과세 표준 금액을 입력하시오:1000
소득세는 90입니다.

과세 표준 금액을 입력하시오:3000
소득세는 540입니다.

과세 표준 금액을 입력하시오:6000
소득세는 1620입니다.

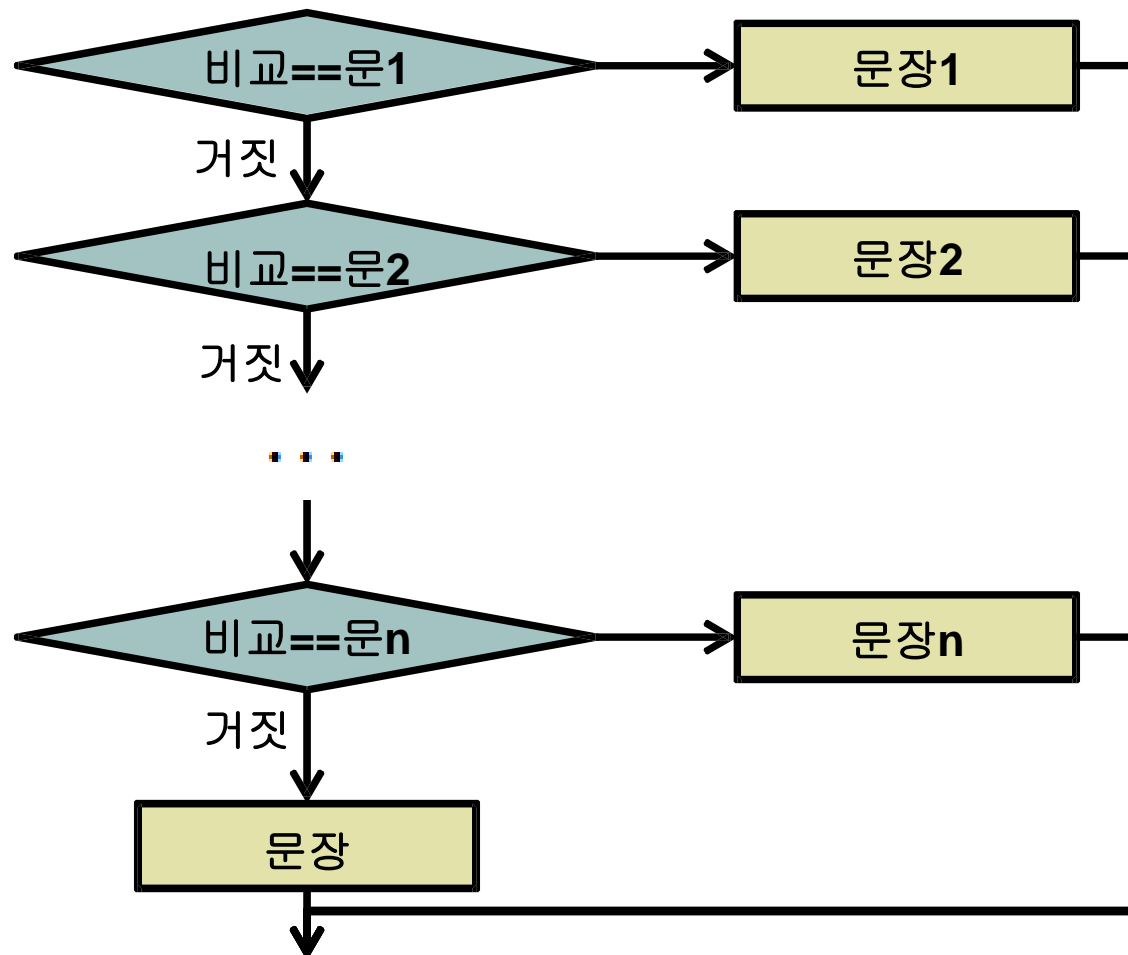
과세 표준 금액을 입력하시오:9000
소득세는 3240입니다.

조건문 (6/8)

switch 문

switch 문은 여러 개의 가능한 실행 경로 중에서 하나를 선택하는데 사용됨

```
switch (비교대상) {  
  case 문1:  
    문장1;  
    break;  
  case 문2:  
    문장2;  
    break;  
  ...  
  case 문n:  
    문장n;  
    break;  
  default:  
    문장;  
}
```



조건문 (7/8)

프로그래밍 예제 (p.110)

```
public class SwitchExample {  
    public static void main(String[] args) {  
        int number;  
  
        Scanner scan = new Scanner(System.in);  
        System.out.print("숫자를 입력하시오: ");  
        number = scan.nextInt();  
        switch (number) {  
            → case 0:  
                System.out.println("없음");  
                break;  
            → case 1:  
                System.out.println("하나");  
                → break;  
            case 2:  
                System.out.println("둘");  
                break;  
            default:  
                System.out.println("많음");  
                break;  
        }  
    }  
}
```

숫자를 입력하시오: 1
하나

조건문(8/8)

프로그래밍 예제 (p.III)

```
public static void main(String[] args) {
    int year = 2013, month, days=0;

    System.out.print("일수를 알고 싶은 달을 입력하시오:");
    Scanner scan = new Scanner(System.in);
    month = scan.nextInt();
    switch (month) {
        case 1: case 3: case 5: case 7: case 8: case 10: case 12:
            days = 31;
            break;
        case 4: case 6: case 9: case 11:
            days = 30;
            break;
        case 2:
            if (((year % 4 == 0) && !(year % 100 == 0)) || (year % 400 == 0)) // 윤년
                days = 29;
            else // 평년
                days = 28;
            break;
        default:
            System.out.println("월이 잘못 입력되었습니다.");
            break;
    }
    System.out.println("월의 날수는 " + days);
}
```

<terminated> Hello Java Application! C:\Program Files\

일수를 알고 싶은 달을 입력하시오:10

월의 날수는 31

실습

실습 – if문 또는 Switch문 사용

사용자로부터 하나의 문자를 입력 받아서 문자가 'R'이면 사각형, 'T'이면 삼각형, 'C'이면 원의 면적을 계산하는 프로그램 작성하기.
(면적을 계산하는데 필요한 숫자는 사용자로부터 입력 받음)

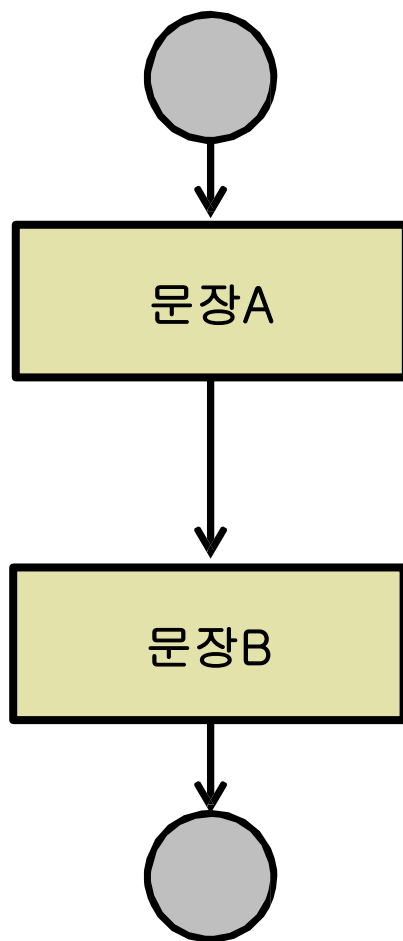
면적을 계산할 도형을 입력하세요.
(사각형: R, 삼각형: T, 원: C)
도형: R
사각형의 면적을 계산합니다.
가로: 10
세로: 20
넓이는 200.00 입니다.

면적을 계산할 도형을 입력하세요.
(사각형: R, 삼각형: T, 원: C)
도형: T
삼각형의 면적을 계산합니다.
밑변: 10
높이: 20
넓이는 100.00 입니다.

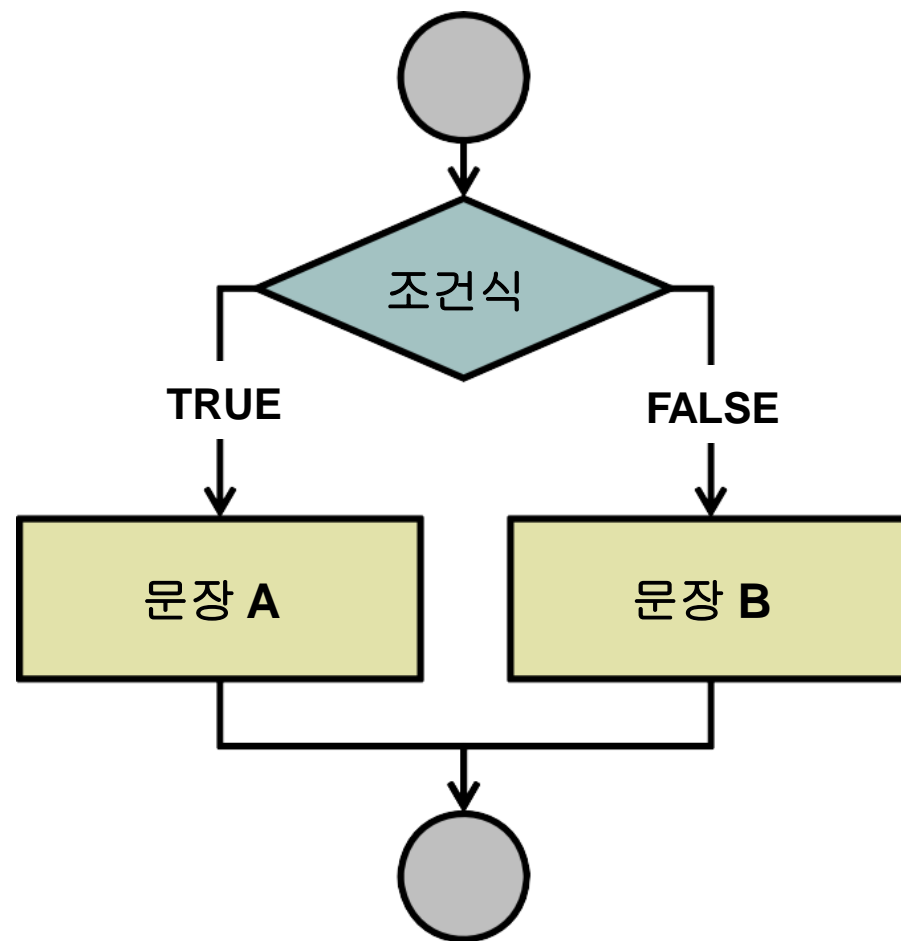
면적을 계산할 도형을 입력하세요.
(사각형: R, 삼각형: T, 원: C)
도형: C
원의 면적을 계산합니다.
반지름: 10
넓이는 314.16 입니다.

반복문 (1/13)

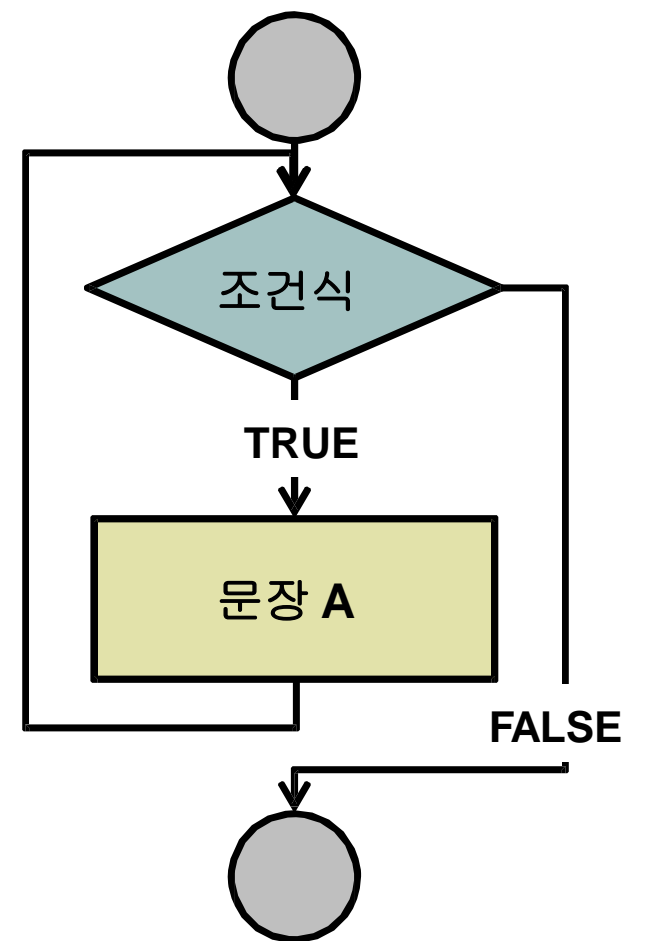
3가지 제어문



순차 구조



선택 구조



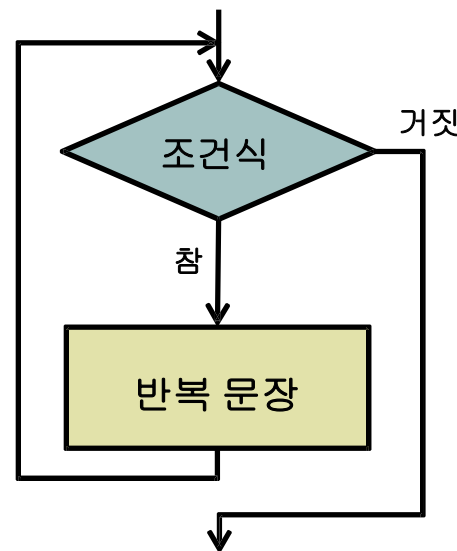
반복 구조

반복문 (2/13)

While 문

- while문은 다음과 같은 형식을 가진다.

```
while (조건식) {  
    반복문장  
}
```



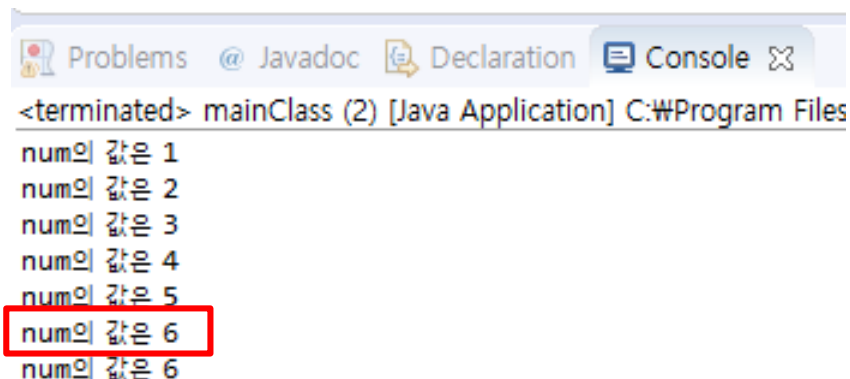
- 조건식이 참이면 문장들의 블록을 반복적으로 실행한다.

반복문 (3/13)

프로그래밍 예제

```
public static void main(String[] args) {  
    int num;  
  
    num = 0;  
  
    while(num < 6){  
        num++;  
        System.out.println("num의 값은 "+num);  
    }  
    System.out.println("num의 값은 "+num);  
}
```

← num이 6보다 작은 경우 반복



Problems @ Javadoc Declaration Console

<terminated> mainClass (2) [Java Application] C:\Program Files

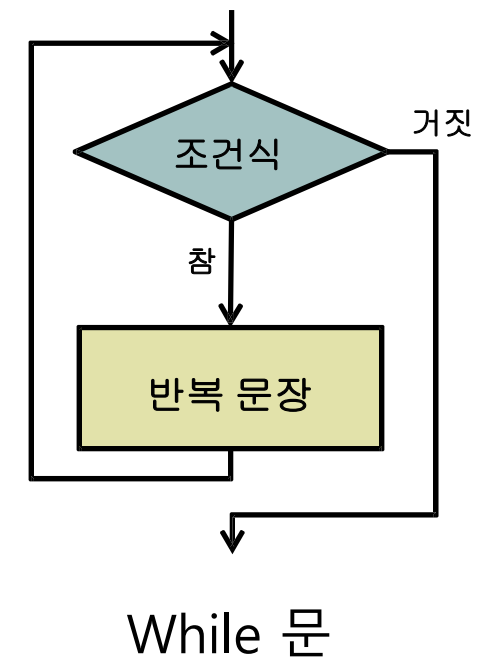
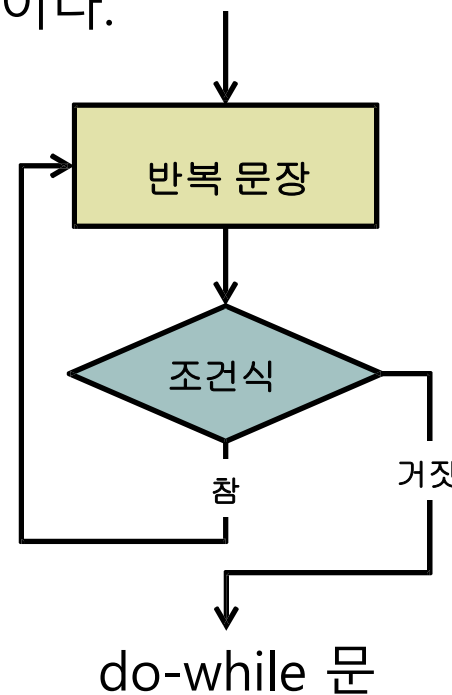
num의 값은 1
num의 값은 2
num의 값은 3
num의 값은 4
num의 값은 5
num의 값은 6
num의 값은 6

반복문 (4/13)

do-while 문

- do-while 문은 while 문과 다르게 반복문장을 먼저 실행하고 조건을 나중에 검사하는 반복 구조이다.

```
do {  
    반복문장;  
} while (조건식);
```



- 즉, do 블록 안에 있는 문장은 적어도 한번은 실행된다.

반복문 (5/13)

프로그래밍 예제

```
public static void main(String[] args) {  
    int answer = 59;  
    int guess;  
    int tries = 0;  
    Scanner in = new Scanner(System.in);
```

```
    do{  
        System.out.println("1 ~ 100 사이 정수를 추측해보세요.");  
        guess = in.nextInt();  
        tries++;  
  
        if(guess > answer){  
            System.out.println("입력한 정수보다 낮습니다.");  
        }else if(guess < answer){  
            System.out.println("입력한 정수보다 높습니다.");  
        }  
    }while(guess != answer);
```

```
    System.out.println("축하합니다. 시도횟수="+tries);  
}
```

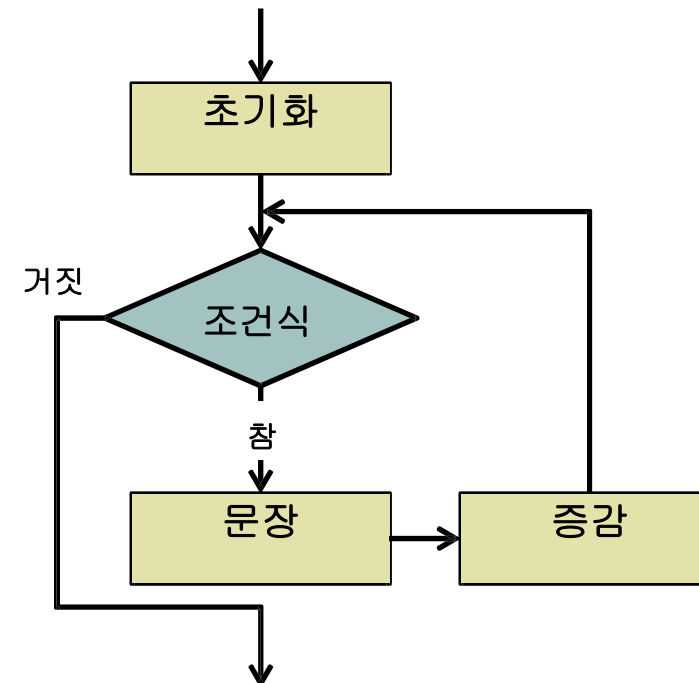
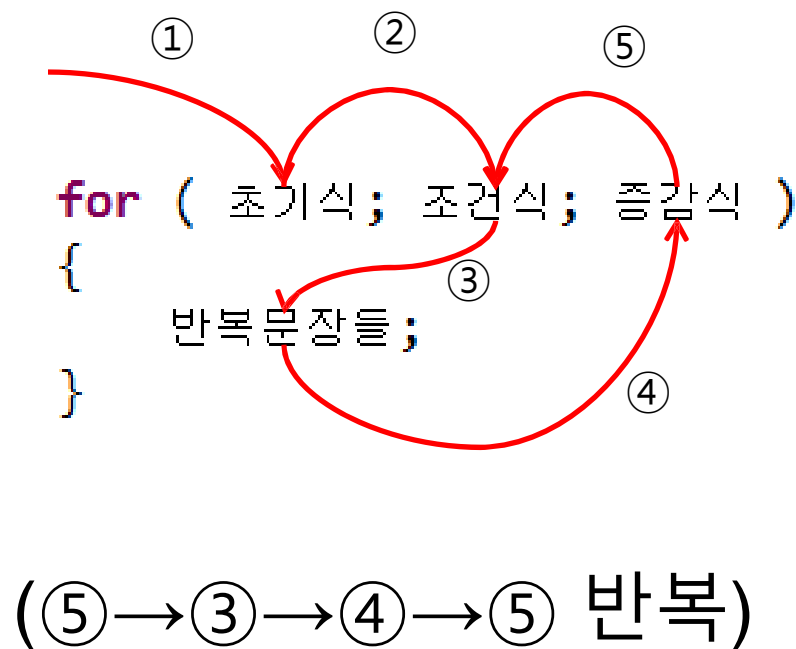
```
Scanner in = new Scanner(System.in);  
1 ~ 100 사이 정수를 추측해보세요.  
72  
입력한 정수보다 낮습니다.  
1 ~ 100 사이 정수를 추측해보세요.  
50  
입력한 정수보다 높습니다.  
1 ~ 100 사이 정수를 추측해보세요.  
59  
축하합니다. 시도횟수=3
```

사용자의 정수를 입력 받아
Answer의 정수와 비교 후
같을 때 까지 반복

반복문 (6/13)

For 문

- for 문은 일정한 횟수만큼 반복할 때 유용하게 사용된다. 흔히 for 루프라고 부른다.
- 초기식 계산 → 조건식 비교 → 반복문 실행 → 증감식 계산 → 조건식 비교 → 반복문 실행(참) → 종료 (거짓)



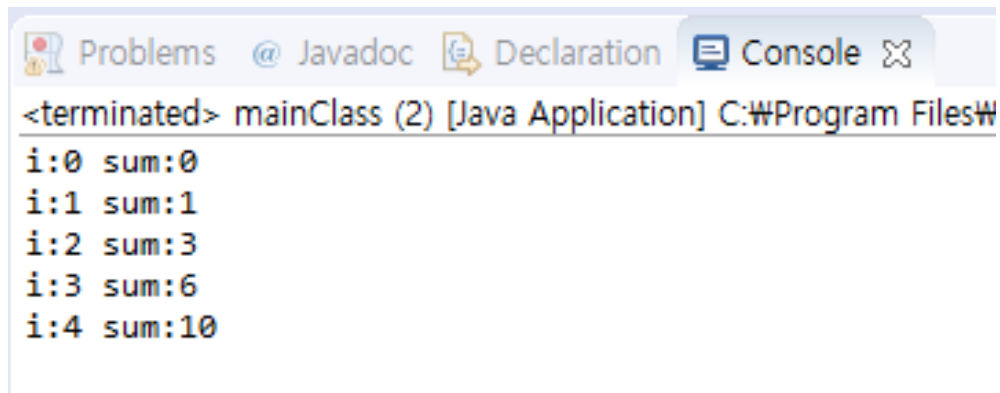
반복문 (7/13)

프로그래밍 예제

```
public static void main(String[] args) {  
    int i;  
    int sum = 0;
```

```
    for(i=0;i<5;i++){  
        sum += i;  
        System.out.println("i:"+i+" sum:"+sum);  
    }
```

i 는 0부터 4까지 증가하며
증가한 i 값을 sum에 합산



```
Problems @ Javadoc Declaration Console X  
<terminated> mainClass (2) [Java Application] C:\Program Files\W  
i:0 sum:0  
i:1 sum:1  
i:2 sum:3  
i:3 sum:6  
i:4 sum:10
```

반복문 (8/13)

중첩 반복문

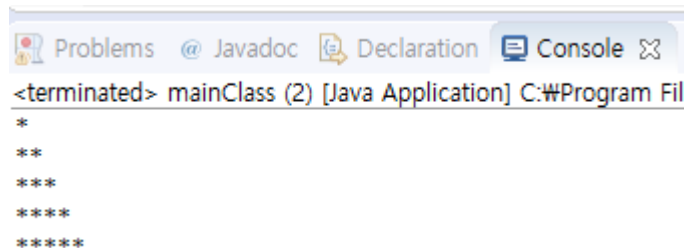
- 반복문은 중첩되어 사용될 수 있음
- 내부에 위치하는 반복문은 외부에 위치하는 반복문이 한번 반복될 때마다 새로 실행됨
- 중첩 반복문에서 주의할 점은 각각의 반복문을 제어하는 변수가 달라야 함
만약 같은 변수를 사용하면 논리적인 오류가 발생할 수 있음

반복문 (9/13)

프로그래밍 예제

```
public static void main(String[] args) {  
    int i;  
    int j;  
  
    for(i=1;i<6;i++){  
        for(j=0;j<i;j++){  
            System.out.print("*");  
        }  
        System.out.print("\n");  
    }  
}
```

각 루프를 독립적으로 생각하면 좋음



```
<terminated> mainClass (2) [Java Application] C:\Program Fil  
*  
**  
***  
****  
*****
```

i=1의 한 루프 안에서 j는 0 ~ 1의 루프
i=2의 한 루프 안에서 j는 0 ~ 2의 루프

i=x의 한 루프 안에서 j는 0 ~ i의 루프

반복문 (10/13)

반복문을 사용하는 경우

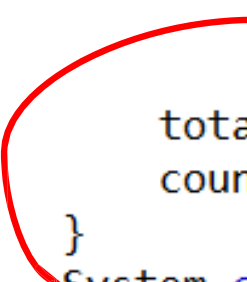
- For문 - 반복 횟수를 아는 경우
- While문 - 조건만 존재하고 정확한 반복횟수를 모르는 경우
- Do-while문 - 반드시 한번은 수행되어야 하는 문장이 있는 경우

반복문 (11/13)

Break 문

```
int total = 0;
int count = 0;
Scanner scan = new Scanner(System.in);
while (true) {
    System.out.print("점수를 입력하시오: ");
    int grade = scan.nextInt();
    if (grade < 0)
        break;

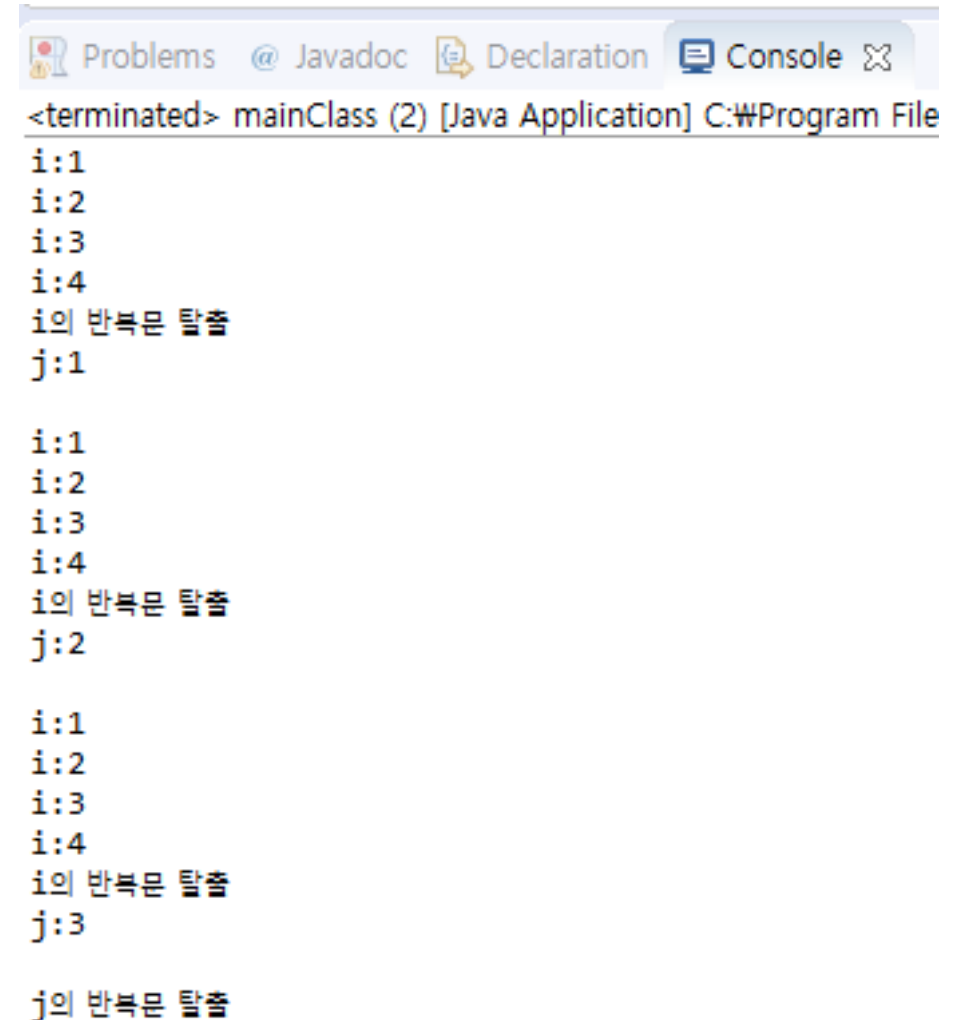
    total += grade;
    count++;
}
System.out.println("평균은 " + total / count);
```



반복문 (12/13)

Break 문

```
public static void main(String[] args) {  
    int i;  
    int j;  
  
    j = 0;  
    while(true){  
        i = 0;  
        while(true){  
            i++;  
            System.out.println("i:"+i);  
            if(i>3)  
                break;  
        }  
        System.out.println("i의 반복문 탈출");  
  
        j++;  
        System.out.println("j:"+j + "\n");  
        if(j>2)  
            break;  
    }  
    System.out.println("j의 반복문 탈출");  
}
```



Problems @ Javadoc Declaration Console

<terminated> mainClass (2) [Java Application] C:\Program File

i:1
i:2
i:3
i:4
i의 반복문 탈출
j:1

i:1
i:2
i:3
i:4
i의 반복문 탈출
j:2

i:1
i:2
i:3
i:4
i의 반복문 탈출
j:3

j의 반복문 탈출

- 반복문에서 break를 만나면 현재 반복하고 있는 한 루프의 밖으로 나감

반복문 (13/13)

Continue 문

```
public static void main(String[] args) {  
    int sum = 0;  
  
    System.out.println("1와 10 사이 짝수의 합");  
  
    for(int i=1;i<11;i++){  
        if(i%2 == 1)  
            continue;  
        sum += i;  
    }  
  
    System.out.println("짝수의 합은 "+sum);  
}
```

i가 홀수 인 경우 continue에 의해 if문 아래에 있는 sum += i를 실행하지 않고 다음으로 넘어간다.

- Continue 문은 Break 문과 반대로 루프 밖으로 나가지 않고, Continue를 만난 시점에서 다음 루프로 이동한다.

실습

프로그래밍 실습

- 구구단을 출력하는 프로그램 작성 (반복문을 중첩하여 구현)
- 1단 ~ 9단 까지의 반복문 (1 ~ 9), 한 단에서 반복문 ($x\text{단} * 1 \sim x\text{단} * 9$)

실행결과

=====								
1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

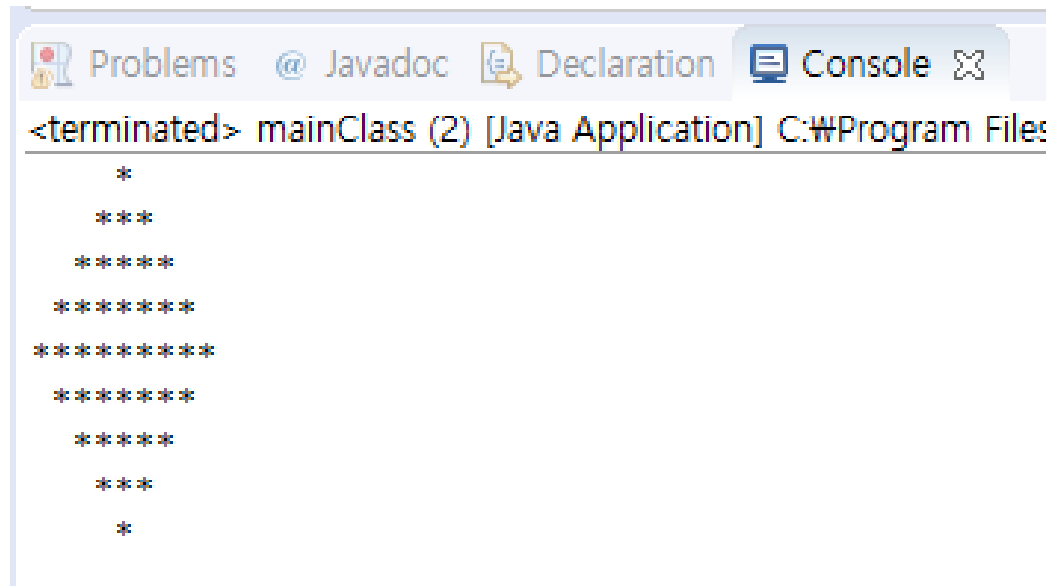
과제

과제

1. 3개의 숫자를 받아서 크기 순으로 정렬하는 프로그램 작성하기 (if-else 사용), p133
2. 사용자로부터 키를 입력받아서 표준 체중을 계산한 후에 사용자의 체중과 비교하여 저체중인지, 표준인지, 과체중인지를 판단하는 프로그램 작성. p.133

표준체중 = $(\text{키} - 100) * 0.9$, 표준 체중의 $\pm 10\%$ 인 경우 정상
그 이상이거나 이하이면 과체중 또는 저체중

3. 반복문을 이용하여 다음과 같은 값을 출력하는 프로그램 작성



```
<terminated> mainClass (2) [Java Application] C:\Program Files\
*
***
*****
*****
*****
*****
*****
*****
***
*
```

Q&A



수고하셨습니다