

제 어 문

강사 : 강병준

조건문 – if, switch

if (조건식) { 문장들 }

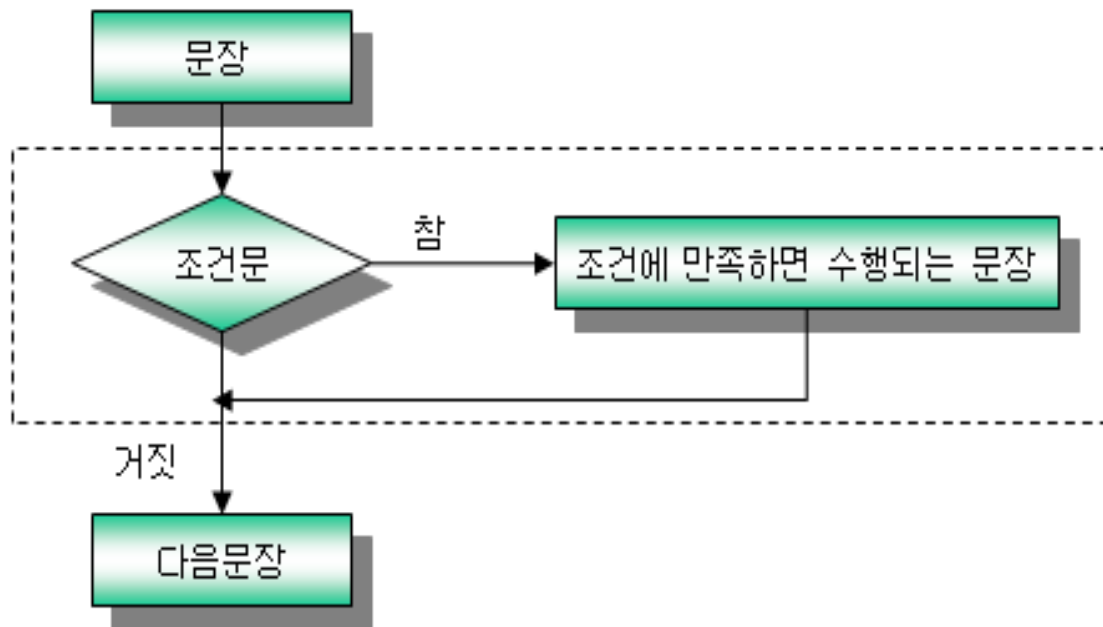
- 조건문은 조건식과 실행될 하나의 문장 또는 블록{}으로 구성
- Java에서 조건문은 if문과 switch문 두 가지 뿐이다.
- if문이 주로 사용되며, 경우의 수가 많은 경우 switch문을 사용할 것을 고려한다.
- 모든 switch문은 if문으로 변경이 가능하지만, if문은 switch문으로 변경할 수 없는 경우가 많다.

```
if(num==1) {  
    System.out.println("SK");  
} else if(num==6) {  
    System.out.println("KTF");  
} else if(num==9) {  
    System.out.println("LG");  
} else {  
    System.out.println("UNKNOWN");  
}
```

```
switch(num) {  
    case 1:  
        System.out.println("SK");  
        break;  
    case 6:  
        System.out.println("KTF");  
        break;  
    case 9:  
        System.out.println("LG");  
        break;  
    default:  
        System.out.println("UNKNOWN");  
}
```

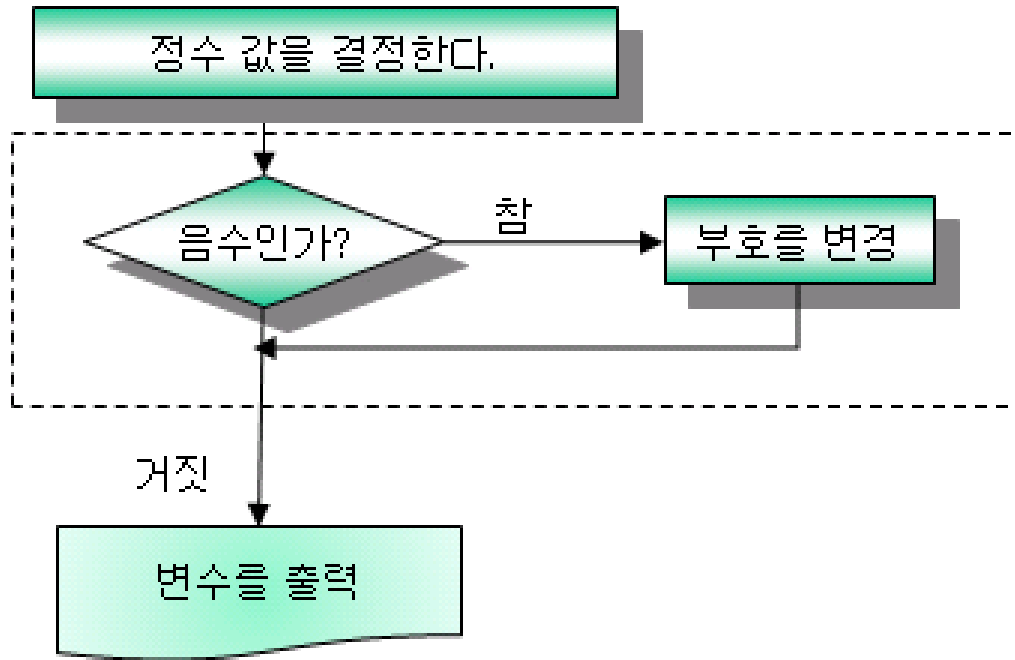
if 문

```
if(조건문){  
    조건에 만족할 경우 실행되는 문장;  
}
```



절대 값 구하기

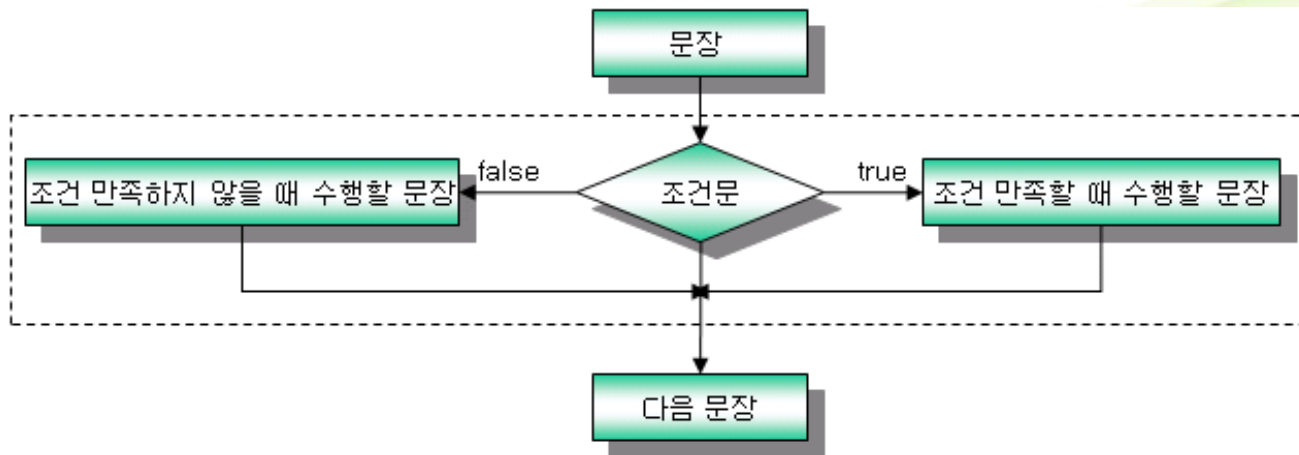
- ◆ 절대 값에서는 부호가 무시되고 크기만 언급된다.
- ◆ 즉, -5에 대한 절대 값은 5이고 7에 대한 절대 값은 7이다.



if~else 문

if~else 문은 참일 때와 거짓일 때 각각 다른 문장을 수행하도록 지정

```
if(조건){  
    조건에 만족할 경우 실행되는 문장;  
}  
else{  
    조건에 만족하지 않을 경우 실행되는 문장;  
}
```



분기문 - 단일 if문 예제

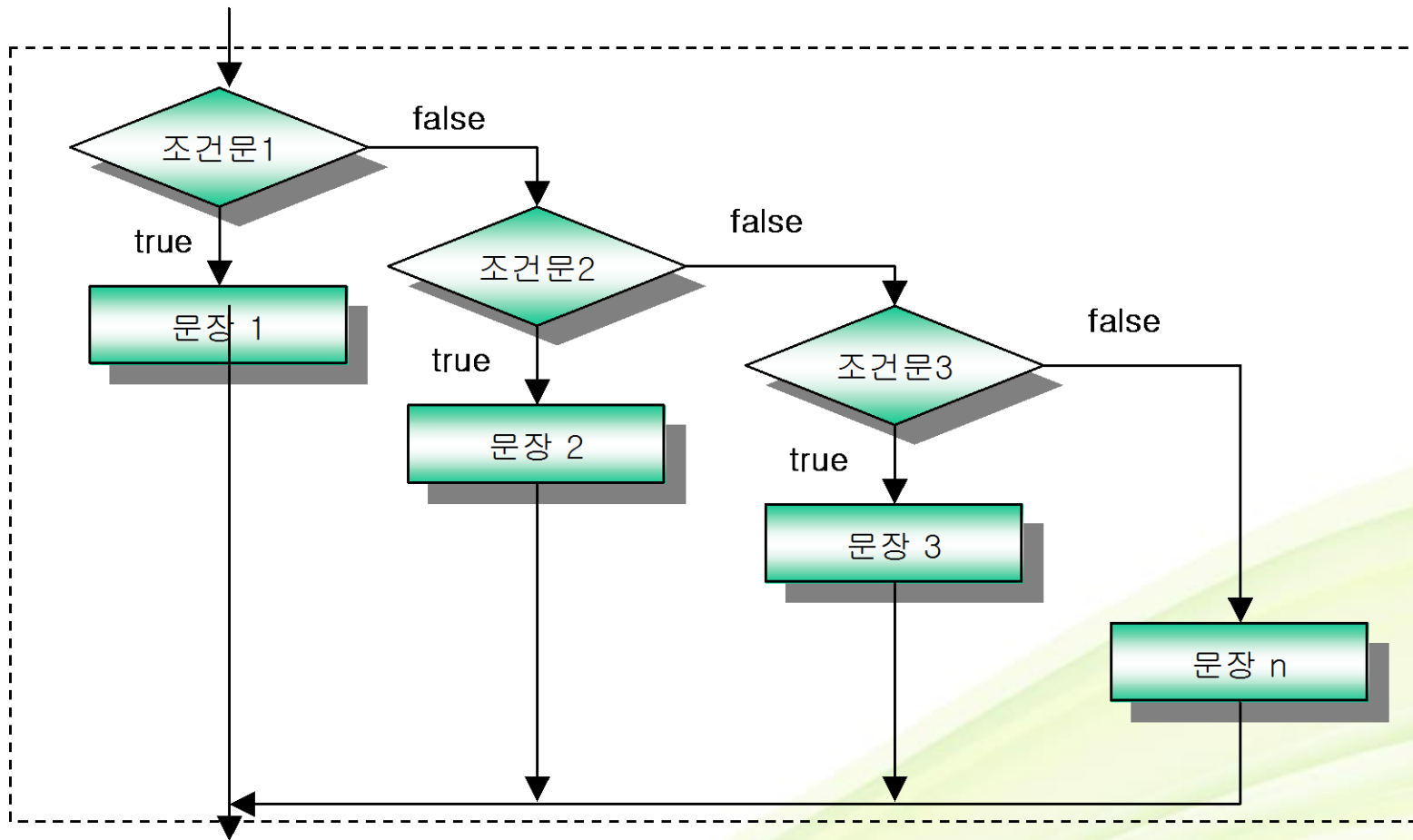
```
class IfTest1 {  
    public static void main(String[] arg){  
        int a=5;  
        if(a<5)  
            System.out.println("a는 5보다 작다");  
        else  
            System.out.println("a는 5보다 작지 않다.");  
        }  
    }
```

if~else~if 문

둘이 아닌 셋 이상에서 하나를 선택해야 할 경우에는 if~else 문을 중첩해서 사용해야 한다.

```
if(조건1){
    조건1에 만족할 때 처리할 문장;
}
else if(조건2){
    조건1에 만족하지 않지만 조건2에 만족할 때 처리할 문장;
}
. . .
else if(조건n){
    조건1부터 조건n-1에 만족하지 않지만 조건n에 만족할 때 처리할 문장;
}
else{
    위에서 언급한 모든 조건에 대해서 만족하지 않을 때 처리할 문장;
}
다음 문장;
```

if~else~if 문



분기문 - 다중 if문

```
class IfTest2 {  
    public static void main(String[] arg){  
        int a=5,b=6;  
        int c;  
        if(a<5){  
            c=a+b;  
            System.out.println(c);  
        }else{  
            c=a-b;  
            System.out.println(c);  
        }  
    }  
}
```

분기문 - 다중 if 문

//다중 if else 문

```
class IfTest3{
    public static void main(String[] arg) {
        int a;
        a=Integer.parseInt(arg[0]);
        if(a>=95)
            System.out.println("당신의 점수는 "+a+" 이고 학점은 A++입니다.");
        else if(a>=90)
            System.out.println("당신의 점수는 "+a+" 이고 학점은 A입니다.");
        else if(a>=85)
            System.out.println("당신의 점수는 "+a+" 이고 학점은 B++입니다.");
        else if(a>=80)
            System.out.println("당신의 점수는 "+a+" 이고 학점은 B입니다.");
        else if(a>=75)
            System.out.println("당신의 점수는 "+a+" 이고 학점은 C++입니다.");
        else if(a>=70)
            System.out.println("당신의 점수는 "+a+" 이고 학점은 C입니다.");
        else if(a>=65)
            System.out.println("당신의 점수는 "+a+" 이고 학점은 D++입니다.");
        else if(a>=60)
            System.out.println("당신의 점수는 "+a+" 이고 학점은 D입니다.");
        else
            System.out.println("당신의 점수는 "+a+" 이고 학점은 F입니다.");
    }
}
```

중첩 if문

- if문 안에 또 다른 if문을 중첩해서 넣을 수 있다.
- if문의 중첩횟수에는 거의 제한이 없다.

```
if (조건식1) {
```

```
    // 조건식1의 연산결과가 true일 때 수행될 문장들을 적는다.
```

```
    if (조건식2) {
```

```
        // 조건식1과 조건식2가 모두 true일 때 수행될 문장들
```

```
    } else {
```

```
        // 조건식1이 true이고, 조건식2가 false일 때 수행되는 문장들
```

```
    }
```

```
} else {
```

```
    // 조건식1이 false일 때 수행되는 문장들
```

```
}
```

```

public class If6 {
    public static void main(String[] args) {
        int score = Integer.parseInt(args[0]);
        String grade;
        if (score >= 90) {grade = "A";
            if (score > 97) {grade += "+";
                } else if (score < 94) {grade += "-";
                } else {grade += "0";}
        } else if (score >= 80) {grade = "B";
            if (score > 87) {grade += "+";
                } else if (score < 84) {grade += "-";
                } else {grade += "0";}
        } else if (score >= 70) {grade = "C";
            if (score > 77) {grade += "+";
                } else if (score < 74) {grade += "-";
                } else {grade += "0";}
        } else if (score >= 60) {grade = "D";
            if (score > 67) {grade += "+";
                } else if (score < 64) {grade += "-";
                } else {grade += "0";}
        } else {grade = "F";}
        System.out.println("점수는 "+score+" 이고 학점은 "+grade+" 입니다");
    }
}

```

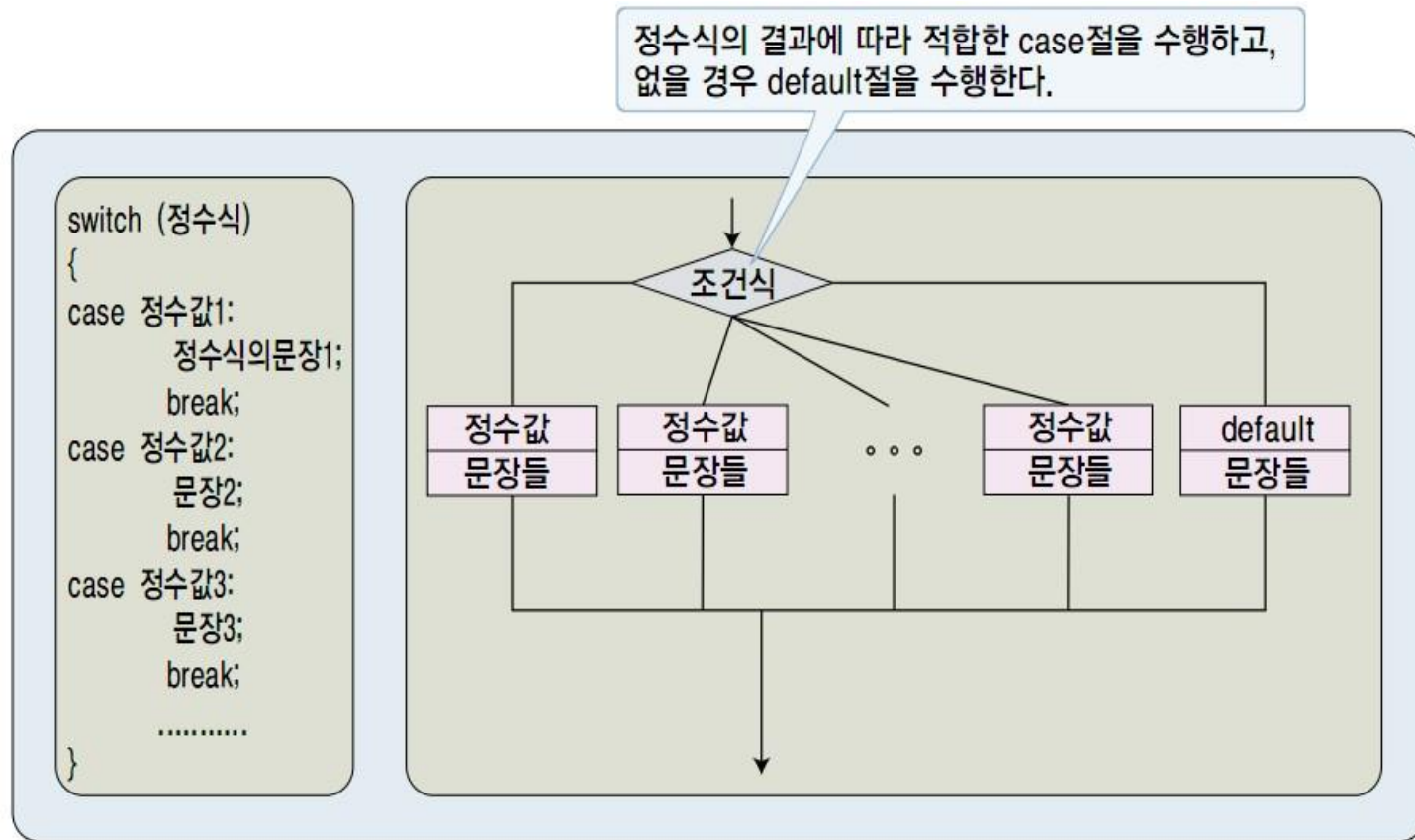
실습문제

입력 받은 값을 다음과 같이 비교 판단하는 프로그램을 작성한다.

- 1) 10 미만의 값입니다
- 2) 10이상 100 미만의 값 입니다
- 3) 100이상 1000 미만의 값입니다
- 4) 1000 이상의 값입니다

선택문(분기문)

- 형식 : switch 문



분기문 - switch문

▶ switch-case문

☞ 형식

```
switch(변수) {  
    case 값 1:  
        실행문 1;  
        break;  
    case 값 2:  
        실행문 2;  
        break;  
    case 값 3:  
        실행문 3;  
        break;  
    default:  
        실행문 4;  
        break;  
}
```

예제)

```
switch(Integer.parseInt(args[0]))  
{  
    case 1:  
        System.out.println("1인데요");  
        break;  
    case 2:  
        System.out.println("2인데요");  
        break;  
    default:  
        System.out.println("값이넘었  
네요");  
        break;  
}
```

분기문 - switch문

```
class SwithTest1{  
    public static void main(String[] arg){  
        int money=Integer.parseInt(arg[0]);  
  
        switch(money){  
            case 500:  
                System.out.println("버스를 타시오");  
                break;  
            case 5000:  
                System.out.println("일반택시를 타시오");  
                break;  
            case 50000:  
                System.out.println("모범택시를 타시오");  
                break;  
            default:  
                System.out.println("걸어 가시오");  
        }  
    }  
}
```


중첩 switch문

- switch문 안에 또 다른 switch문을 중첩해서 넣을 수 있다.
- switch문의 중첩횟수에는 거의 제한이 없다.

```
switch(num) {  
    case 1:  
    case 7:  
        System.out.println("SK");  
        switch(num) {  
            case 1:  
                System.out.println("1");  
                break;  
            case 7:  
                System.out.println("7");  
                break;  
        }  
        break;  
    case 6:  
        System.out.println("KTF");  
        break;  
    case 9:  
        System.out.println("LG");  
        break;  
    default:  
        System.out.println("UNKNOWN");  
}
```

```
switch(num) {  
    case 1:  
    case 7:  
        System.out.println("SK");  
        if(num==1) {  
            System.out.println("1");  
        } else if(num==7) {  
            System.out.println("7");  
        }  
        break;  
    case 6:  
        System.out.println("KTF");  
        break;  
    case 9:  
        System.out.println("LG");  
        break;  
    default:  
        System.out.println("UNKNOWN");  
}
```

if문과 switch문의 비교

- if문이 주로 사용되며, 경우의 수가 많은 경우 switch문을 사용할 것을 고려한다.
- 모든 switch문은 if문으로 변경이 가능하지만, if문은 switch문으로 변경할 수 없는 경우가 많다.
- if문 보다 switch문이 더 간결하고 효율적이다.

```
if(num==1) {  
    System.out.println("SK");  
} else if(num==6) {  
    System.out.println("KTF");  
} else if(num==9) {  
    System.out.println("LG");  
} else {  
    System.out.println("UNKNOWN");  
}
```

```
switch(num) {  
    case 1:  
        System.out.println("SK");  
        break;  
    case 6:  
        System.out.println("KTF");  
        break;  
    case 9:  
        System.out.println("LG");  
        break;  
    default:  
        System.out.println("UNKNOWN");  
}
```

실습예제

다음프로그램은 사용자로부터 월을 입력 받아 해당하는 계절을 출력하는 프로그램을 작성하시오

case 절에서 처리할 내용이 없을 경우에는 그대로 비워둔다

- 1) 12, 1, 2 : 겨울입니다
- 2) 3, 4, 5 : 봄입니다
- 3) 6, 7, 8 : 여름입니다
- 4) 9, 10, 11 : 가을입니다

if 문으로 변경하시오

반복문 – for, while, do-while

- 문장 또는 문장들을 반복해서 수행할 때 사용
- 조건식과 수행할 블록{} 또는 문장으로 구성
- 반복회수가 중요한 경우에 for문을 그 외에는 while문을 사용한다.
- for문과 while문은 서로 변경가능하다.
- do-while문은 while문의 변형으로 블록{}이 최소한 한번은 수행될 것을 보장한다.

```
System.out.println(1);  
System.out.println(2);  
System.out.println(3);  
System.out.println(4);  
System.out.println(5);
```

```
int i=0;  
  
do {  
    i++;  
    System.out.println(i);  
} while(i<=5);
```

```
for(int i=1;i<=5;i++) {  
    System.out.println(i);  
}
```

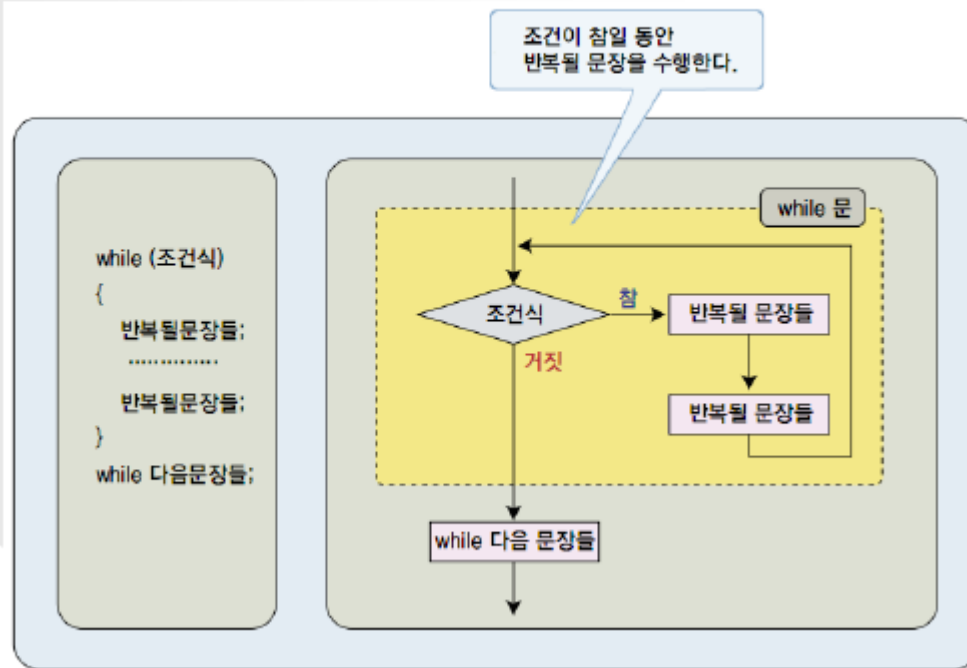
```
int i=1;  
  
while(i<=5) {  
    System.out.println(i);  
    i++;  
}
```

반복문 - while문

- 조건을 잘못 주면 무한반복이 될 수도 있다.
- 항상 조건의 값은 boolean자료 형인 true, false이다.
- 실행 문이 한 줄일 때에는 { }를 생략해도 된다.

● while문

- 특정 조건이 만족하는 동안 지정된 영역을 반복할 수 있는 기능



예제

```
a=6;
while( a > 5)
{
    System.out.println("안녕");
    a--;
}
```

반복문 - while문

//1 부터 10까지의 합을 구한다.

```
class WhileTest1{
    public static void main(String[] arg){
        int sum=0;
        int i=1;

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

반복문 – while문 문제

//실행결과가 이런 식으로 나오는 while문을 작성하시오

i가 1일때 sum은 1이다

i가 2일때 sum은 3이다

i가 3일때 sum은 6이다

i가 4일때 sum은 10이다

i가 5일때 sum은 15이다

i가 6일때 sum은 21이다

i가 7일때 sum은 28이다

i가 8일때 sum은 36이다

i가 9일때 sum은 45이다

i가 10일때 sum은 55이다

반복문 - while문

```
class WhileTest2 {  
    public static void main(String[] arg){  
        int sum=0;  
        int l =Integer.parseInt(arg[0]);  
  
        while(i<=10){  
            sum += i;  
            System.out.println("i가 "+i+"일때 sum은 "+sum+"이다");  
            i++;  
        }  
    }  
}
```


입출력 프로그램 참조

```
public class Exam_01 {  
  
    public static void main(String[] ar) throws  
        java.io.IOException{  
  
        int x = System.in.read();  
  
        System.out.println("x = " + x);  
  
        System.out.println("기본 출력 스트림...");  
    }  
}
```

중첩 while문

- while문 안에 또 다른 while문을 포함시킬 수 있다.
- while문의 중첩횟수에는 거의 제한이 없다.

```
public class While6 {  
    public static void main(String[] args) {  
        System.out.println("구구단");  
        System.out.println("=====");  
        int i = 1, j = 2;  
        while(i <= 9) {  
            while(j <= 9) {  
                System.out.printf("%d * %d = %d\t", j, i, (i*j));  
                j++;  
            }  
            i++;  
            j = 2;  
            System.out.println();  
        }  
    }  
}
```

반복문 - do-while문

▶ do-while문

while문가 거의 비슷하지만 일단 먼저 한번은 실행 문을 실행하고 비교를 한다.

☞ 형식

```
do {  
    실행문;  
}  
while(조건)
```

예제

```
a=6;  
do  
{  
    System.out.println("안녕");  
    a--;  
}while( a > 5)
```

-항상 조건의 값은 boolean자료 형인 true, false이다.

```
import java.io.IOException;
```

```
public class doWhileOperator {  
    public static void main(String[] args) throws IOException {  
        int a;  
        do {  
            System.out.println("숫자를 입력하시오");  
            a = System.in.read() - '0';  
            System.in.read();  
            System.in.read();  
        } while(a%2==1);  
        System.out.println("a = " + a + " 0/고 짝수");  
    }  
}
```

```
import java.util.Scanner;
public class ScannerTest {
    /**
     * Scanner 테스트
     */
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        //next
        String nextString = sc.next();
        System.out.println("nextString = " + nextString);

        //nextInt
        int nextIntNum = sc.nextInt();
        System.out.println("nextIntNum = " + nextIntNum);
    }
}
```

Math.random()

- Math클래스에 정의된 난수(亂數) 발생 함수
- 0.0과 1.0 사이의 double값을 반환한다. ($0.0 \leq \text{Math.random()} < 1.0$)

예) 1~10범위의 임의의 정수를 얻는 식 만들기

1. 각 변수에 10을 곱한다.

```
0.0 * 10 <= Math.random() * 10 < 1.0 * 10  
0.0 <= Math.random() * 10 < 10.0
```

2. 각 변수를 int형으로 변환한다.

```
(int)0.0 <= (int)(Math.random() * 10) < (int)10.0  
0 <= (int)(Math.random() * 10) < 10
```

3. 각 변수에 1을 더한다.

```
0 + 1 <= (int)(Math.random() * 10) + 1 < 10 + 1  
1 <= (int)(Math.random() * 10) + 1 < 11
```

int score = (int)(Math.random() * 10) + 1;

난수 테스트

```
import java.util.Scanner;
public class FlowTest21 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int number = (int)(Math.random()*100)+1;
        int inNumber = 0;
        do{
            System.out.println("숫자를 입력하세요...");
            inNumber = sc.nextInt();
            if(inNumber == number){
                System.out.println("맞혔습니다.");
                break;
            } else if(inNumber < number){
                System.out.println("숫자가 너무 작아요.");
            } else {
                System.out.println("숫자가 너무 커요.");
            }
        } while(true);
    }
}
```

반복문 - for 문

▶ for 문

반복횟수를 정해주어서 정해진 횟수만큼 반복을 하는 문입니다. 아주 유용하게 잘 쓰이는 문입니다.

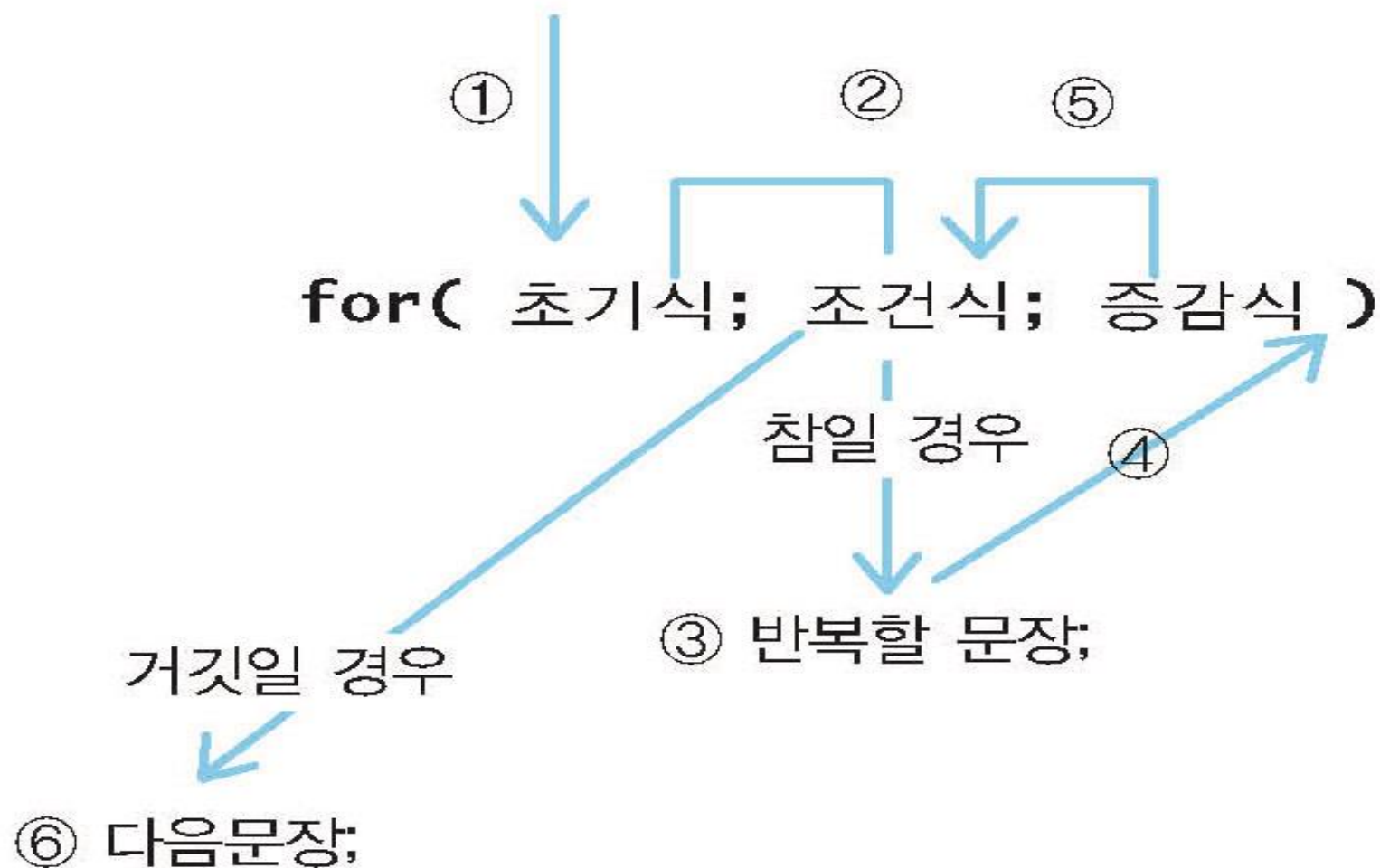
☞ 형식

```
for(반복변수 초기화; 반복변수 조건검사; 반복변수의 값 변화)
{
    실행문
}
```

☞ 예제

```
for(int a=0; a<4; a++)
{
    System.out.println("현재 a의 갯수는 " + a);
}
```


for 문



for 문 기본 다루기

a. 0 3 6 9

◆ for(int i=0; i<=10; i+=3)

◆ b. 1 6 11 16

◆ for(int i=1; i<=20; i+=5)

c. 2 4 8 16 32 64

◆ for(int i=2; i<=100; i*=2)

d. 5 4 3 2 1

◆ for(int i=5; i>=1; i--)

반복문 - for 문

//1부터 100까지의 합을 구하시오

```
class ForTest1 {  
    public static void main(String[] arg){  
        int sum=0;  
        for(int i=1; i<=100; i++){  
            sum +=i;  
            System.out.println("i가 "+i+"일때 sum은 "+sum+"이다");  
        }  
    }  
}
```

반복문 - for 문

//1부터 100까지의 홀수의 합을 구하시오

```
class ForTest2 {  
    public static void main(String[] arg){  
        int sum=0;  
        for(int i=1; i<=100; i+=2){  
            sum +=i;  
            System.out.println("i가 "+i+"일때 sum은 "+sum+"이다");  
        }  
    }  
}
```

반복문 - for문 문제-(구구단)

이런 결과가 나오도록 프로그램을 작성 하세요(for문을 이용)

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

반복문 - for문

```
class gugudan1{  
    public static void main(String[] a){  
        int result;  
  
        for(int i=1; i<10; i++){  
            System.out.println("");  
  
            for(int j=1; j<10; j++){  
                result=i*j;  
                System.out.print("  "+j+"*"+i+"="+result);  
            }  
        }  
        System.out.println("");  
    }  
}
```

반복문 – for문 문제-(구구단)

//2단 만 출력

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

//5단 만 출력

5*1=5 5*2=10 5*3=15 5*4=20 5*5=25 5*6=30 5*7=35 5*8=40 5*9=45

//9단 만 출력

9*1=9 9*2=18 9*3=27 9*4=36 9*5=45 9*6=54 9*7=63 9*8=72 9*9=81

```
import java.io.*;
```

```
public class Ex06 {  
    public static void main(String[] ar) throws IOException {  
        BufferedReader in = new BufferedReader(new  
            InputStreamReader(System.in));  
        System.out.print("키보드 하나를 누르세요 : ");  
        //int a = System.in.read();  
        char a = (char)System.in.read();  
        System.out.println("a = '" + a + "'");  
        //System.in.read();// \r  
        //System.in.read();// \n  
        in.readLine();  
        System.out.print("수를 하나 누르세요 : ");  
        int b = System.in.read() - '0';  
        System.out.println("b = " + b);  
    }  
}
```


반복문 - for문

```
class gugudan2{  
    public static void main(String[] a){  
        int result;  
        int az=Integer.parseInt(a[0]);  
  
        for(int j=1; j<10; j++){  
            result=az*j;  
            System.out.print(" "+az+"*"+j+"="+result);  
        }  
    }  
}
```

반복문 – for문

```
class ForTest{  
    public static void main(String[] arg){  
        for(int i=1; i<=5; i++){  
            for(int j=1; j<=i; j++){  
                System.out.print("*");  
            }  
            System.out.println("");  
        }  
    }  
}
```

결과값

*

**

반복문 – for문 (결과값들)

▶ 이런 결과값들이 나오도록 프로그램을 작성하시요
(for문을 이용)

*	*****	*	*****
**	*****	**	*****
***	***	***	***
****	**	****	**
*****	*	*****	*

반복문 – for문

```
class ForTest5 {  
    public static void main(String[] arg){  
        for(int i=1; i<=5; i++){  
            for(int j=5; j>=i; j--){  
                System.out.print("*");  
            }  
            System.out.println("");  
        }  
    }  
}
```

결과값

**
*

반복문 – for문 문제

```
class ForTest6 {  
    public static void main(String[] arg){  
        for(int i=1; i<=5; i++){  
            for(int j=1; j<=5-i; j++){  
                System.out.print(" ");  
                for(int z=1; z<=i; z++){  
                    System.out.print("*");  
                }  
                System.out.println("");  
            }  
        }  
    }  
}
```

결과값

```
*  
**  
***  
****  
*****
```

반복문 – for문 문제

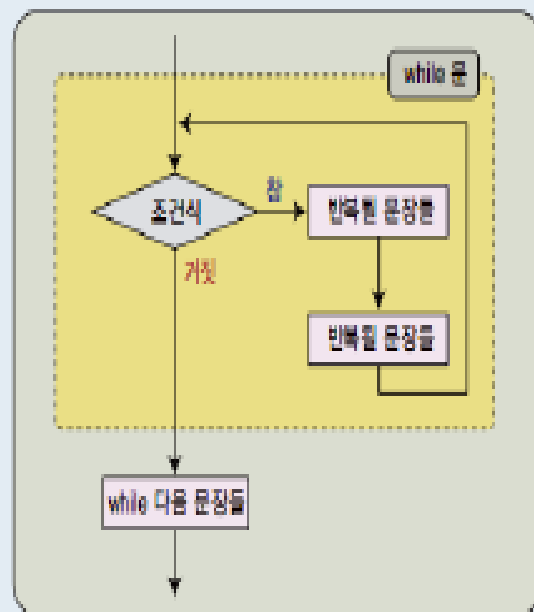
```
class ForTest7 {  
    public static void main(String[] arg){  
        for(int i=1; i<=5; i++){  
            for(int j=1; j<=i; j++){  
                System.out.print(" ");  
                for(int z=1; z<=6-i; z++){  
                    System.out.print("*");  
                }  
                System.out.println("");  
            }  
        }  
    }  
}
```

결과값

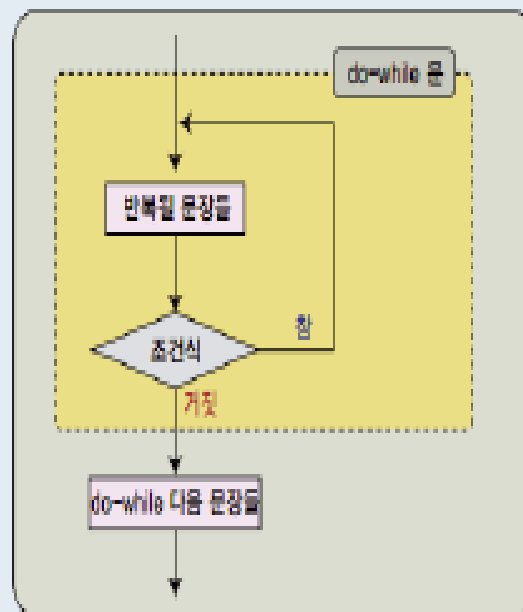
**

*

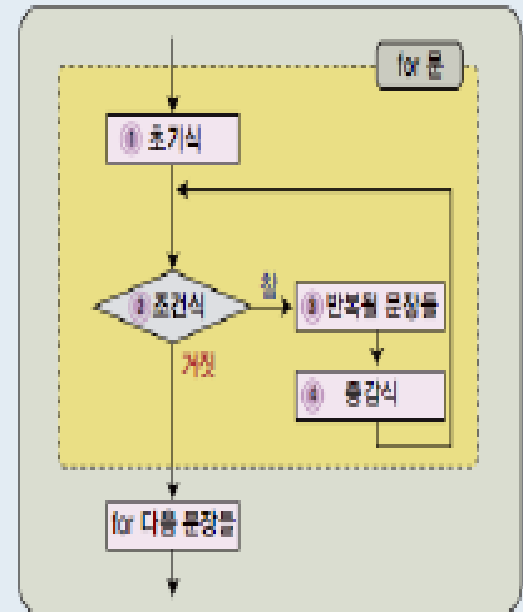
● 3개의 반복문의 구조



while 문



do-while 문



for 문

제어문 – break문

▶ break문(1)

break문은 반복 문을 빠져 나오게 하는 문이고 반복 문 내의 어떤 곳에 위치할 수도 있다.

☞ 예제

```
Int a=6;  
while( a > 5) {  
    System.out.println("안녕");  
    a--;  
    break; // 반복문을 빠져나가게 한다.  
}
```


제어문 – break문

```
class BreakTest {  
    public static void main(String[] arg){  
        int sum=0;  
        for(int i=1; i<=100; i++){  
            if(i==51)  
                break;  
            sum +=i;  
            System.out.println("i가 "+i+"일 때 sum은 "+sum+"이다");  
        }  
    }  
}
```

제어문 – break문

▶ break문(2)

break문은 반복 문을 한번에 하나밖에 빠져 나올 수 없다.

```
int a=1;
while(true){
    while(true){
        System.out.println("무한 반복을 합니다.");
        if(a>=4)
            break; // 반복 문을 빠져나가게 한다.
        a++;
    }
    System.out.println("여기로 빠져 나오겠죠.");
    break; //반복 문을 빠져 나온다..
}
```

제어문 – continue문

▶ continue문

반복 문을 빠져 나가지 않고 반복문의 조건부로 실행위치를 옮긴다. **continue**이후에 실행 문들은 실행이 되지 않는다.

☞ 예제

```
int a=1;
for(int x=1; x<=5;x++){
    System.out.println("나는 반복한다.");
    a++;
    if(a>3)
        continue;
    System.out.println("나두 반복되고 싶어");
}
```

제 어 문 – continue 문 예 제

```
class ConTest{
    public static void main(String[] arg){
        int a=1;
        for(int x=1; x<=5;x++){
            System.out.println("나는 반복한다.");
            a++;
            if(a>3)
                continue;
            System.out.println("나두 반복되고 싶어");
        }
    }
}
```

제 어 문 – label문

```
class LabelTest {  
    public static void main(String[] arg){  
        outer:for(int i=0;i<10;i++){  
            for(int j=0;j<10;j++){  
                if(j==3)  
                    break outer;  
                System.out.println("j="+j);  
            }  
            System.out.println("i="+i);  
        }  
    }  
}
```

연습문제

1. 1부터 20까지 정수 중에서 2또는 3의 배수가 아닌 수의 합
2. $1+(1+2)+(1+2+3)+\dots+(1+2+3+4+5+\dots+9+10)$ 의 결과
3. 주사위를 던져서 두수의 합이 6이되는 경우를 출력
 $1 + 5 = 6 \quad 2+4=6 \quad \text{-----} \quad 5 + 1 = 6$

힌트

1. $i\%2 \neq 0 \ \&\& \ i\%3 \neq 0$
2. $\text{sum} += i; \quad \text{totalSum} += \text{sum};$
3.

```
for(int i = 1; i<=6;i++) {  
    for(int j=1;j<=6;j++) {  
        if (i+j==6)  
    }  
}
```