

JAVA Ch. 4

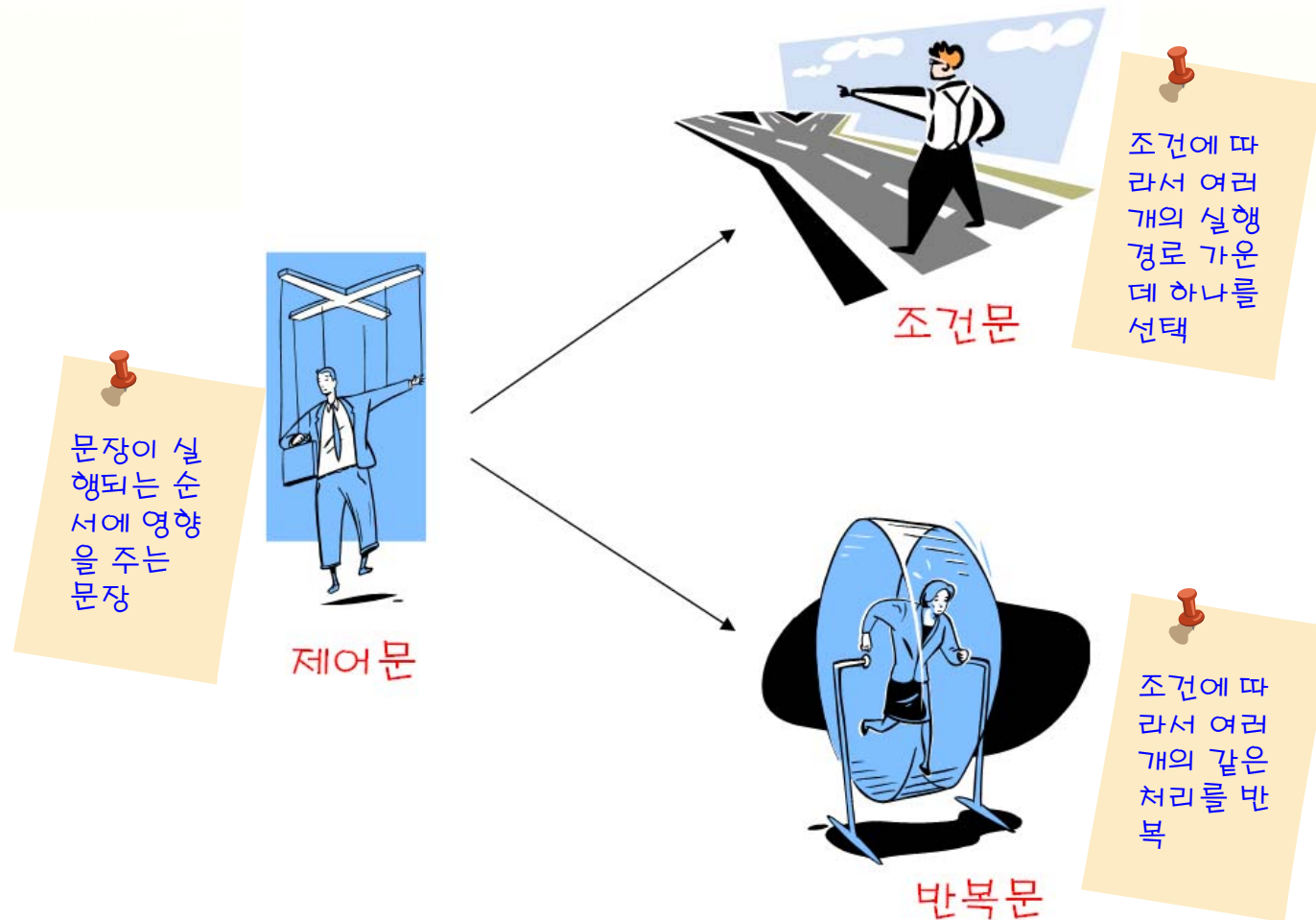
Prof. Youngchul Jung

2012 – Winter class



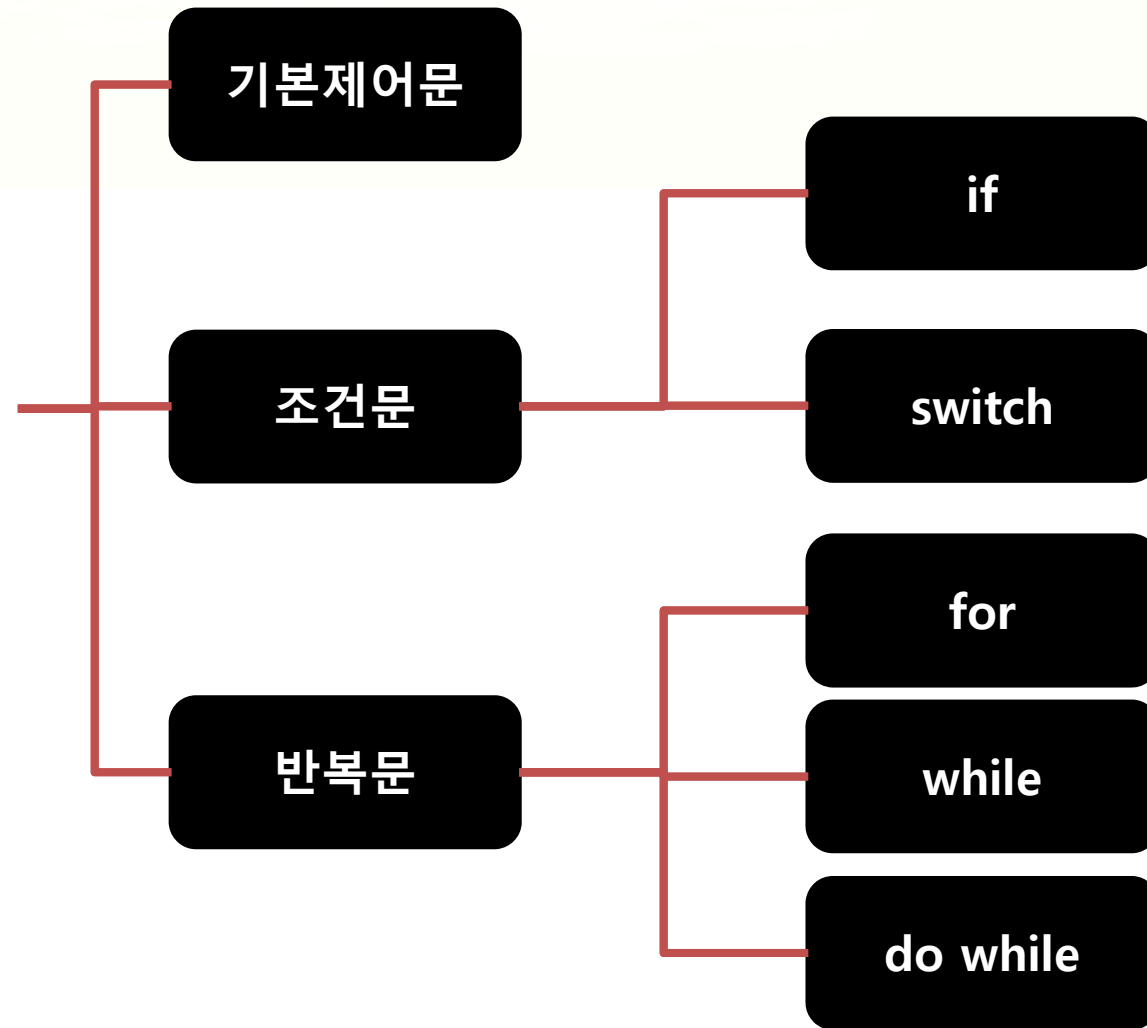
주문식교육의 신실
영진전문대학

흐름제어(Flow control) (1)



흐름제어(Flow control) (2)

흐름제어
(컴퓨터언어)



조건문- if

- if문은 if, if-else, if-else if의 세가지 형태가 있다.
- 조건식의 결과는 반드시 true 또는 false이어야 한다.

```
if(조건식) {  
    // 조건식의 결과가 true일 때 수행될 문장들  
}
```

```
if(조건식) {  
    // 조건식의 결과가 true일 때 수행될 문장들  
} else {  
    // 조건식의 결과가 false일 때 수행될 문장들  
}
```

```
if(조건식1) {  
    // 조건식1의 결과가 true일 때 수행될 문장들  
} else if(조건식2) {  
    // 조건식2의 결과가 true일 때 수행될 문장들  
    // (조건식1의 결과는 false)  
} else if(조건식3) {  
    // 조건식3의 결과가 true일 때 수행될 문장들  
    // (조건식1과 조건식2의 결과는 false)  
} else {  
    // 모든 조건식의 결과가 false일 때 수행될 문장들  
}
```

```
if(score > 60) {  
    System.out.println("합격입니다.");  
}  
  
if(score > 60)  
    System.out.println("합격입니다.");
```

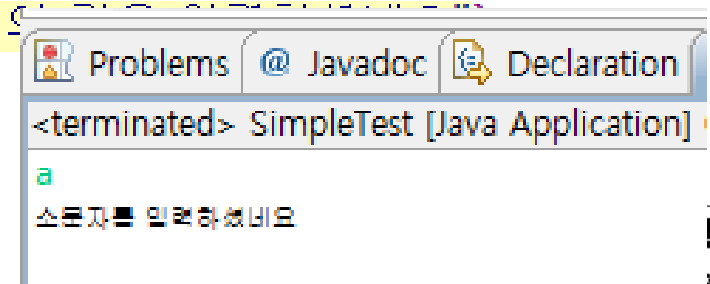
```
if(score > 60) {  
    System.out.println("합격입니다.");  
} else {  
    System.out.println("불합격입니다.");  
}
```

```
if(num > 0) {  
    System.out.println("양수입니다.");  
} else if(num < 0) {  
    System.out.println("음수입니다.");  
} else {  
    System.out.println("영입니다.");  
}
```

```
if(score >= 90) {  
    System.out.println("A등급");  
} else if(score >= 80 && score < 90) { // 80<=score<90  
    System.out.println("B등급");  
} else if(score >= 70 && score < 80) { // 70<=score<80  
    System.out.println("C등급");  
} else { // score < 70  
    System.out.println("F등급");  
}
```

대문자, 소문자를 구별하는 프로그램 예제

```
public class SimpleTest {  
    public static void main(String[] args) {  
        Scanner scan = new Scanner(System.in);  
  
        // 문자열을 키보드로 부터 입력 받는다  
        String varString = scan.next();  
  
        // 문자열의 첫번째 문자를 가지고 온다.  
        char varChar = varString.charAt(0);  
  
        if(varChar >= 'a' && varChar <= 'z')  
            System.out.println("소문자를 입력하셨습니다.");  
        else if(varChar >= 'A' && varChar <= 'Z')  
            System.out.println("대문자를 입력하셨습니다.");  
        else  
            System.out.println("영문자 이외의 문자를 입력하셨습니다.");  
    }  
}
```



0~100사이의 성적을 입력 받아 A~F학점 변환

```
public class SimpleTest {  
  
    public static void main(String[] args) {  
  
        String inputValueStr = JOptionPane.showInputDialog("점수를 입력하세요!");  
        String grade;  
  
        int inputValueInt = Integer.parseInt(inputValueStr);  
  
        if(inputValueInt >= 95)  
            grade = "A+";  
        else if(inputValueInt >= 90)  
            grade = "A";  
        else if(inputValueInt >= 85)  
            grade = "B+";  
        else if(inputValueInt >= 80)  
            grade = "B";  
        else if(inputValueInt >= 75)  
            grade = "C+";  
        else if(inputValueInt >= 70)  
            grade = "C";  
        else if(inputValueInt >= 65)  
            grade = "D+";  
        else if(inputValueInt >= 60)  
            grade = "D";  
        else  
            grade = "F";  
  
        System.out.println(grade);  
    } ? end main ?  
}
```

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  
}
```

switch문

- if문의 조건식과 달리, 조건식의 계산결과가 int범위 이하의 정수, 문자열만 가능
- 조건식의 계산결과와 일치하는 case문으로 이동 후 break문을 만날 때까지 문장들을 수행한다.(break문이 없으면 switch문의 끝까지 진행한다.)
- case문의 값으로 변수를 사용할 수 없다.(리터럴, 상수만 가능)

```
switch (조건식) {  
    case 값1 :  
        // 조건식의 결과가 값1과 같을 경우 수행될 문장들  
        //...  
        break;  
    case 값2 :  
        // 조건식의 결과가 값2와 같을 경우 수행될 문장들  
        //...  
        break;  
    //...  
    default :  
        // 조건식의 결과와 일치하는 case문이 없을 때 수행될 문장들  
        //...  
}
```

```
switch(num) {  
    case 1:  
    case 7:  
        System.out.println("SK");  
        break;  
    case 6:  
    case 8:  
        System.out.println("KTF");  
        break;  
    case 9:  
        System.out.println("LG");  
        break;  
    default:  
        System.out.println("UNKNOWN");  
        break;  
}
```


switch 예제 (1)

```
public class hello {  
  
    public static void main(String[] args) {  
  
        String Club = "삼성라이온즈";  
  
        switch(Club)  
        {  
            case "롯데":  
                System.out.println("정대현 화팅!!");  
                break;  
            case "기아타이거즈":  
                System.out.println("윤석민 화팅!!");  
                break;  
            case "삼성라이온즈":  
                System.out.println("이승엽 화팅!!");  
                break;  
            case "두산":  
                System.out.println("김현수 화팅!!");  
                break;  
            default:  
                System.out.println("영철 화팅!!");  
                break;  
        }  
    } ? end main ?  
} ? end hello ?
```

switch문 예제 (2)

```
public class SimpleTest {  
  
    public static void main(String[] args) {  
  
        String inputValueStr = JOptionPane.showInputDialog("1~5사이 정수 입력!");  
  
        int inputValueInt = Integer.parseInt(inputValueStr);  
  
        switch(inputValueInt)  
        {  
            case 1:  
                System.out.println("1");  
            case 2:  
                System.out.println("2");  
            case 3:  
                System.out.println("3");  
            case 4:  
                System.out.println("4");  
                break;  
            case 5:  
                System.out.println("5");  
            default:  
                System.out.println("default");  
        }  
    }  
} ? end main ?
```

If문으로 작성한 학점 변환기를 switch문으로 변환하기

```
public class SimpleTest {  
    public class SimpleTest {
```

```
        public static void main(String[] args) {
```

```
            String inputValueStr = JOptionPane.showInputDialog("점수를 입력하세요!");  
            String grade;    ;수를 입력하세요!");
```

```
            int inputValueInt = Integer.parseInt(inputValueStr);
```

```
            switch(inputValueInt)
```

```
            {
```

```
                case 100:case 99:case 98:case 97:case 96: case 95:
```

```
                    grade = "A+";
```

```
                    break;
```

```
                case 94:case 93:case 92:case 91:case 90:
```

```
                    grade = "A";
```

```
                    break;
```

```
                case 89:case 88:case 87:case 86:case 85:
```

```
                    grade = "B+";
```

```
                    break;
```

```
                case 84:case 83:case 82:case 81:case 80:
```

```
                    grade = "B";
```

```
                    break;
```

```
                case 79:case 78:case 77:case 76:case 75:
```

```
                    grade = "C+";
```

```
                    break;
```

```
                case 74:case 73:case 72:case 71:case 70:
```

```
                    grade = "C";
```

```
                    break;
```

```
                case 69:case 68:case 67:case 66:case 65:
```

```
                    grade = "D+";
```

```
                    break;
```

```
                case 64:case 63:case 62:case 61:case 60:
```

```
                    grade = "D";
```

```
                    break;
```

```
                default:
```

```
                    grade = "F";
```

```
                    break;
```

```
            } ? end switch inputValueInt ?
```

```
            System.out.println(grade);
```

```
        } ? end main ?
```

```
    }  
}
```

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 switch 문의 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");  
}
```




반복문 (for, while, do-while)

반복문 (for, while, do-while)

- 문장 또는 문장들을 반복해서 수행할 때 사용
- 조건식과 수행할 블록{} 또는 문장으로 구성
- 반복회수가 중요한 경우에 for문을 그 외에는 while 문을 사용한다.
- for문과 while문은 서로 변경 가능하다.
- 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);
```

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

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

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



for문 (1)

- 초기화, 조건식, 증감식 그리고 수행할 블록{} 또는 문장으로 구성

```
for (초기화; 조건식; 증감식) {  
    // 조건식이 true일 때 수행될 문장들을 적는다.  
}
```

[참고] 반복하려는 문장이 단 하나일 때는 중괄호{}를 생략할 수 있다.



예) 1부터 10까지의 정수를 더하기

```
int sum = 0;  
  
for(int i=1; i<=10; i++) {  
    sum += i; // sum = sum + i;  
}
```

i

sum

i	sum
1	
2	
3	
4	
...	
10	



주문식교육의 산실
영진전문대학
컴퓨터정보계열 정영철

for문 (2)

for문 작성 예	설 명
<pre>for(;;) { /* 반복해서 수행할 문장들 */ }</pre>	조건식이 없기 때문에 결과가 true로 간주되어 블록{}안의 문장들을 무한히 반복수행한다.
<pre>for(int i=0;;) { /* 반복해서 수행할 문장들 */ }</pre>	for문에 int형 변수 i를 선언하고 0으로 초기화 했다. 변수 i는 for문 내에 선언되었기 때문에 for문 내에서만 유효하다.
<pre>for(int i=1,j=1;i<10 && i*j<50;i++,j+=2) { /* 반복해서 수행할 문장들 */ }</pre>	쉼표(,)를 이용해서 하나 이상의 변수를 선언하고 초기화 할 수 있다. 단, 같은 타입인 경우만 가능하다. 증감식 역시 쉼표를 이용해서 여러 문장이 수행되게 할 수 있다. 여기서는 매 반복마다 i는 1씩, j는 2씩 증가한다.

중첩 for문

- for문 안에 또 다른 for문을 포함시킬 수 있다.
- for문의 중첩횟수에는 거의 제한이 없다.

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

↓

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

i * j = i*j

```
2 * 1 = 2  
2 * 2 = 4  
2 * 3 = 6  
...  
2 * 9 = 18  
3 * 1 = 3  
3 * 2 = 6  
...  
9 * 8 = 72  
9 * 9 = 81
```

ijk

```
111  
112  
113  
...  
121  
122  
123  
...  
331  
332  
333
```

```
for(int i=1; i<=3; i++) {  
    for(int j=1; j<=3; j++) {  
        for(int k=1; k<=3; k++) {  
            System.out.println(" "+i+j+k);  
        }  
    }  
}
```

→

```
for(int i=1; i<=3; i++)  
    for(int j=1; j<=3; j++)  
        for(int k=1; k<=3; k++)  
            System.out.println(" "+i+j+k);
```


for문 이용 알파벳 출력하기

```
abcdefghijklmnopqrstuvwxyz  
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

```
public class SimpleTest {  
    public static void main(String[] args) {  
        for(char lowStartChar = 'a'; lowStartChar <= 'z'; lowStartChar++)  
            System.out.print(lowStartChar);  
  
        System.out.println();  
  
        for(char upStartChar = 'A'; upStartChar <= 'Z'; upStartChar++)  
            System.out.print(upStartChar);  
    }  
}
```

다음 프로그램의 의미? ㅋㅋㅋ

```
public class SimpleTest {  
    public static void main(String[] args) {  
        for(int i = 1, j = 0 ; i <= 100 && j < 30 ; i++)  
        {  
            if(i%3 == 0)  
            {  
                j += i;  
                System.out.println(i);  
            }  
        }  
    }  
}
```

while 문

- 조건식과 수행할 블록{} 또는 문장으로 구성

```
while (조건식) {  
    // 조건식의 연산결과가 true일 때 수행될 문장들을 적는다.  
}
```

```
int i=10;  
while(i >= 0) {  
    System.out.println(i--);  
}
```



```
for(int i=10;i>=0;i--) {  
    System.out.println(i);  
}
```

while문 예제

```
public class SimpleTest {  
  
    public static void main(String[] args) {  
  
        Random rand = new Random();  
  
        int sum = 0;  
  
        while(sum <= 100)  
        {  
            int generatedValue = rand.nextInt()%10;  
  
            sum += generatedValue;  
  
            System.out.println("Generated value: " + generatedValue);  
            System.out.println("sum: " + sum);  
        }  
    }  
}
```

do-while 문

- while문의 변형. 블록{}을 먼저 수행한 다음에 조건식을 계산한다.
- 블록{}이 최소한 1번 이상 수행될 것을 보장한다.

```
do {  
    // 조건식의 연산결과가 true일 때 수행될 문장들을 적는다.  
} while (조건식);
```


break문

- 자신이 포함된 하나의 반복문 또는 switch문을 빠져 나온다.
- 주로 if문과 함께 사용해서 특정 조건을 만족하면 반복문을 벗어나게 한다.

```
class FlowEx25
{
    public static void main(String[] args)
    {
        int sum = 0;
        int i = 0;

        while(true) {
            if(sum > 100)
                ● break;
            i++;
            sum += i;
        } // end of while

        System.out.println("i=" + i);
        System.out.println("sum=" + sum);
    }
}
```

break문이 수행되면 이 부분은 실행되지 않고 while문을 완전히 벗어난다.

i	sum
0	0
1	1
2	3
3	6
...	...
13	91
14	105

continue문

- 자신이 포함된 반복문의 끝으로 이동한다.(다음 반복으로 넘어간다.)
- continue문 이후의 문장들은 수행되지 않는다.

```
class FlowEx26
{
    public static void main(String[] args)
    {
        for(int i=0; i <= 10; i++) {
            if (i%3==0)
                continue;
            System.out.println(i);
        }
    }
}
```

조건식이 true가 되어 continue문이 수행되면 반복문의 끝으로 이동한다.
break문과 달리 반복문 전체를 벗어나지 않는다.

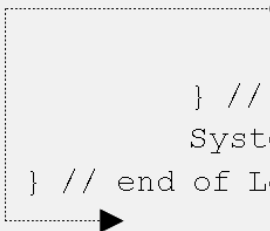
[실행결과]

1
2
4
5
7
8
10

이름 붙은 반복문과 **break**, **continue**

- 반복문 앞에 이름을 붙이고, 그이름을 **break**, **continue**와 같이 사용함으로써 둘 이상의 반복문을 벗어나거나 반복을 건너뛰는 것이 가능하다.

```
class FlowEx27
{
    public static void main(String[] args)
    {
        // for문에 Loop1이라는 이름을 붙였다.
        Loop1 : for(int i=2; i <=9; i++) {
            for(int j=1; j <=9; j++) {
                if(j==5)
                ● break Loop1;
                System.out.println(i+"*"+ j +"="+ i*j);
            } // end of for i
            System.out.println();
        } // end of Loop1
    }
}
```



[실행결과]

```
2*1=2
2*2=4
2*3=6
2*4=8
```