

Chap03. 선택문

JAVA

```
public class SumTest {  
    public static void main(String a1[])  
    {  
        int a, b, sum;  
        a = Integer.parseInt(a1[0]);  
        b = Integer.parseInt(a1[1]);  
        sum = a + b ; // 두 수를 더하는 부분입니다  
        System.out.println("두수의 합은 " + sum + "입니다");  
    }  
}
```

선택 논리를 대표하는 if문에 관해서 학습합니다.

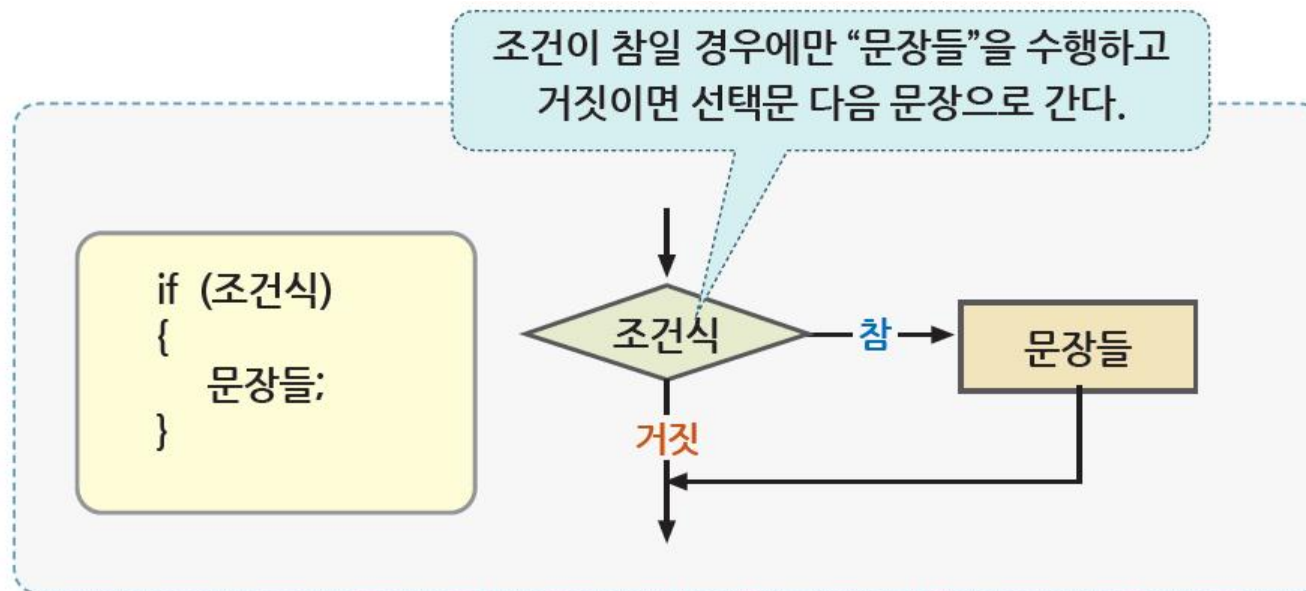
단순 if문, 이중 if문, 다중 if문, 내포된 if문에 관해 학습합니다.

선택문과 반복문에 사용되는 조건문에 대해 학습합니다.

다중 선택문인 switch문에 관해 학습합니다.

if문 : 단순 if문(if)

- 우리가 작성하는 프로그램은 순서, 선택, 반복의 논리로 구성
- 대표적인 선택문 if
- 단순 if문(if)



if문 : 단순 if문(if)

```
if ( num % 2 == 0 )
```

```
{
```

```
    System.out.print("입력된 수 : " + num);
```

```
    System.out.println("은(는) 짝수입니다");
```

```
}
```

조건을 나타내는 문장이 한 문장보다 많으면 반드시 블록("{}")으로 묶어야 한다.

```
if ( num < 0 )
```

조건을 나타내는 문장에는 ";"을 붙이면 안 된다.

```
    System.out.println("음수입니다");
```

문장이 한 문장일 경우 "{}"로 묶지 않아도 된다.

if문 : 단순 if문(if)

● 예제

```
import java.util.Scanner;
public class SimpleIFTest1 {
    public static void main(String args[])
    {
        Scanner stdin = new Scanner(System.in);
        System.out.print("한 개의 숫자를 입력 : ");
        int count = stdin.nextInt();
        if (count < 0)
            System.out.println(count + "은(는) 음수입니다 ");
    }
}
```

if문 : 단순 if문(if)

● 예제

```
import java.util.Scanner;
public class SimpleIFTest2 {
    public static void main(String args[])
    {
        Scanner stdin = new Scanner(System.in);
        System.out.print("한 개의 숫자를 입력 : ");
        int count = stdin.nextInt();
        if (count < 0);
            System.out.println(count + "은(는) 음수입니다 ");
    }
}
```

if문 : 단순 if문(if)

● 예제

```
import java.util.Scanner;
public class SimpleIFTest3 {
    public static void main(String args[])
    {
        Scanner stdin = new Scanner(System.in);
        System.out.print("성적을 입력하세요 : ");
        int grade = stdin.nextInt();
        if (grade >= 90)
        {
            System.out.println("축하합니다 ");
            System.out.println("A학점을 취득하셨습니다 ");
        }
        System.out.println("감사합니다");
    }
}
```

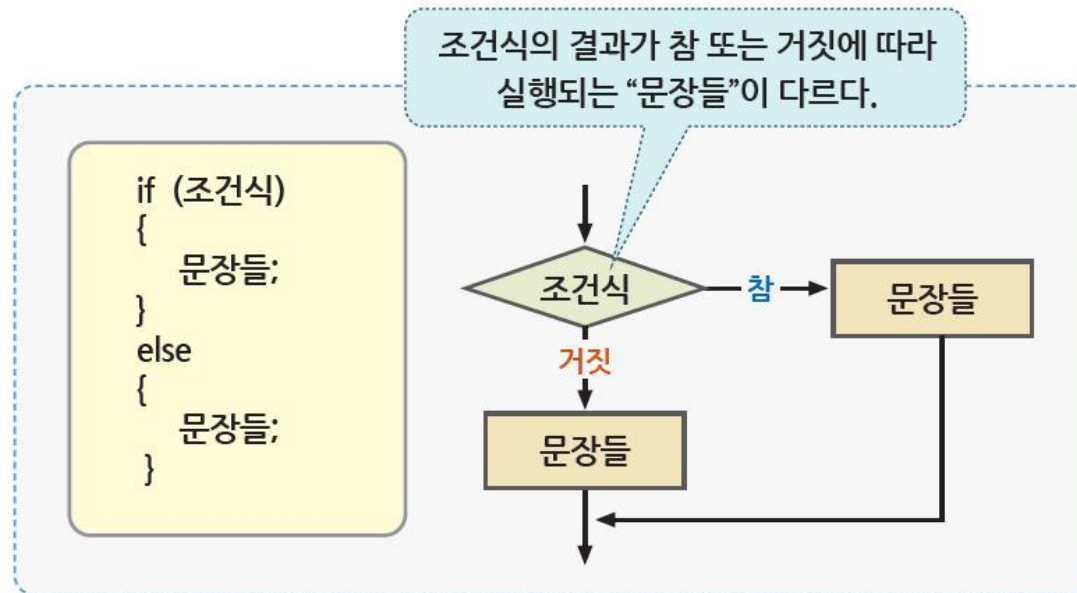
if문 : 단순 if문(if)

● 예제

```
import java.util.Scanner;
public class SimpleIFTest4 {
    public static void main(String args[])
    {
        Scanner stdin = new Scanner(System.in);
        System.out.print("성적을 입력하세요 : ");
        int grade = stdin.nextInt();
        if (grade >= 90)
            System.out.println("축하합니다 ");
            System.out.println("A학점을 취득하셨습니다
");
        System.out.println("감사합니다");
    }
}
```


if문 : 이중 if문(if-else)

- 조건식의 결과에 따라 특정 작업을 수행해야 하는 경우 사용



if문 : 이중 if문(if-else)

```
if ( score >= 60 )
    System.out.println("학점 취득 성공");
else
    System.out.println("학점 취득 실패");
System.out.println("학점 취득 여부와 상관없이 무조건 수행됨");

if ( num % 2 == 0 )
{
    System.out.println("입력된 수 : " + num);
    System.out.println("짝수입니다");
}
else
{
    System.out.println("입력된 수 : " + num);
    System.out.println("홀수입니다");
}
```

← 참일 경우 수행(한 문장)

← 거짓일 경우 수행(한 문장)

← 선택문과 상관없는 문장

← 여러 문장일 경우 묶는다.

← 여러 문장일 경우 묶는다.

if문 : 이중 if문(if-else)

● 예제

```
import java.util.Scanner;
public class DoubleIFTest1 {
    public static void main(String args[])
    {
        Scanner stdin = new Scanner(System.in);
        System.out.print("성적을 입력하세요 : ");
        int grade = stdin.nextInt();
        if (grade >= 90)
            System.out.println("A학점 취득 성공");
        else
            System.out.println("A학점 취득 실패"); */

        System.out.println("감사합니다");
    }
}
```

if문 : 이중 if문(if-else)

```
if (grade >= 90)
    System.out.println("A학점 취득 성공");
else
    System.out.println("A학점 취득 실패");
```

위의 문장을 3항 연산자를 이용하면

```
System.out.println(grade >= 90 ? "A학점 취득 성공" : "A학점 취득 실패");
```

한 문장으로 표현할 수 있습니다.

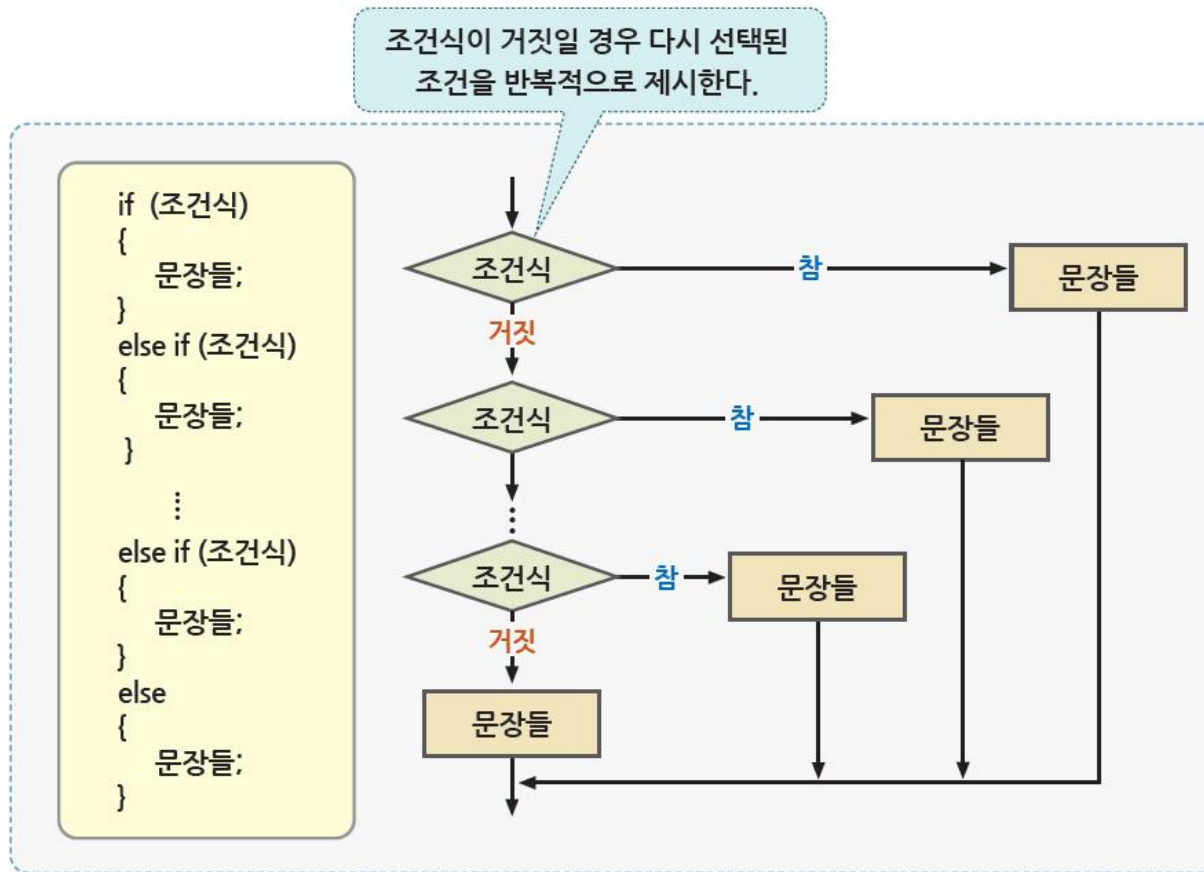
if문 : 이중 if문(if-else)

● 예제

```
import java.util.Scanner;
public class DoubleIFTest2 {
    public static void main(String args[])
    {
        Scanner stdin = new Scanner(System.in);
        System.out.print("숫자를 입력 : ");
        int count = stdin.nextInt();
        if (count < 0 )
        {
            System.out.print("입력된 수 " + count);
            System.out.println("은(는) 음수입니다");
        }
        else
        {
            System.out.print("입력된 수 " + count);
            System.out.println("은(는) 음수가 아닙니다");
        }
    }
}
```


if문 : 다중 if문(if-else if)

- 다중 if문은 조건이 거짓일 경우 다시 조건을 제시하는 선택문



if문 : 다중 if문(if-else if)

● 예제

```
import java.util.Scanner;
public class MultiIFTest1 {
    public static void main(String args[])
    {
        Scanner stdin = new Scanner(System.in);
        System.out.print("성적을 입력하세요 : ");
        int grade = stdin.nextInt();
        if (grade >= 90)
            System.out.println("A학점 취득");
        else if (grade >= 80)
            System.out.println("B학점 취득");
        else if (grade >= 70)
            System.out.println("C학점 취득");
        else if (grade >= 60)
            System.out.println("D학점 취득");
        else
            System.out.println("학점 취득 실패");
    }
}
```

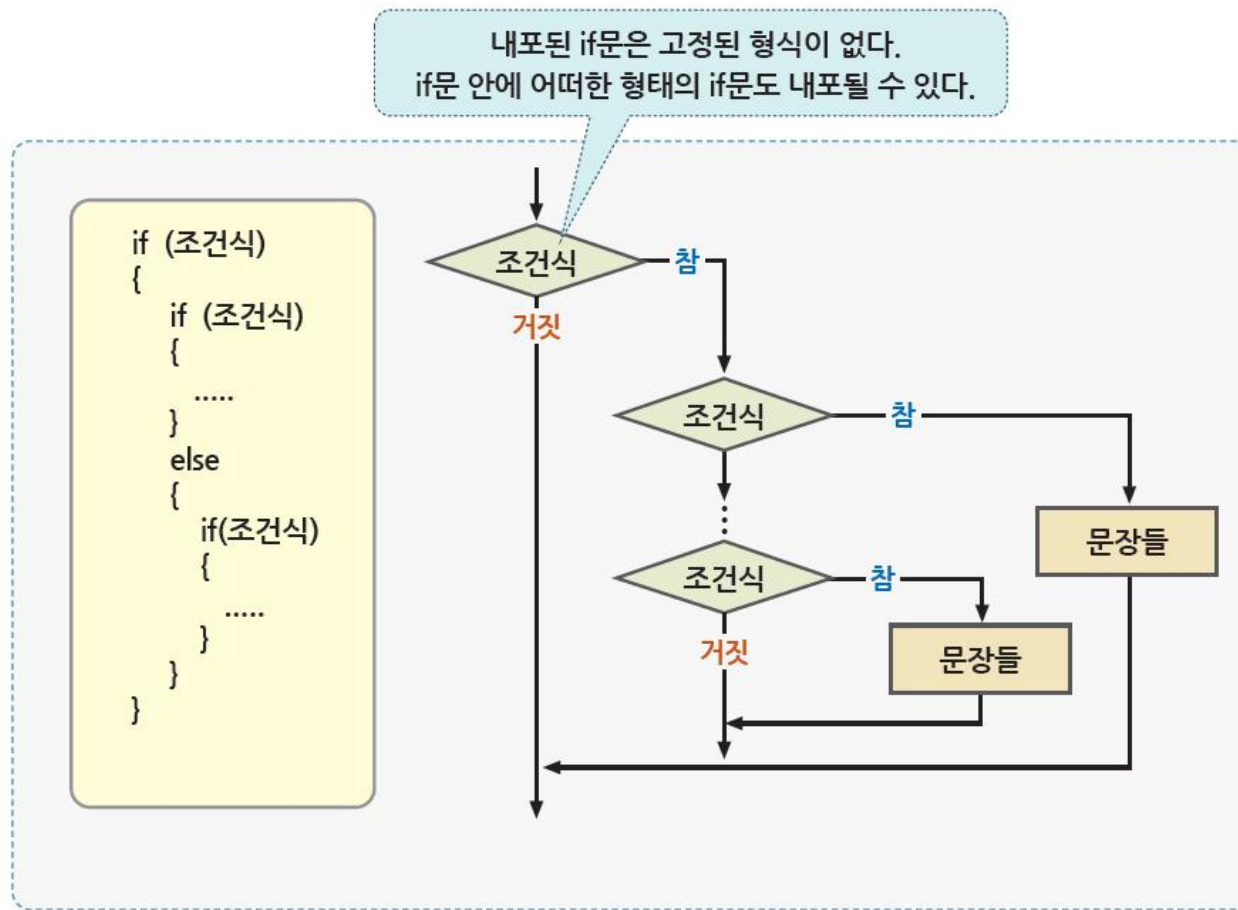
if문 : 다중 if문(if-else if)

● 예제

```
import java.util.Scanner;
public class MultiIFTest2 {
    public static void main(String args[])
    {
        Scanner stdin = new Scanner(System.in);
        System.out.print("숫자를 입력 : ");
        int count = stdin.nextInt();
        if (count < 0 )
        {
            System.out.print("입력된 수 " + count);
            System.out.println("은(는) 음수입니다");
        }
        else if (count == 0)
            System.out.print("입력된 수는 0 입니다");
        else
        {
            System.out.print("입력된 수 " + count);
            System.out.println("은(는) 양수입니다");
        }
    }
}
```

if문 : 내포된 if문

- 선택문안에 선택문이 내포될 수 있다



if문 : 내포된 if문

```
if (score >= 90) <--
    if (score >= 95) <--
        System.out.println("A+ 학점입니다");
    else <--
        System.out.println("A- 학점입니다");
else <--
    if (score >= 60) <--
        if (score >= 80) <--
            System.out.println("B 학점입니다");
        else <--
            System.out.println("C 또는 D 학점입니다");
    else <--
        System.out.println("학점취득 실패");
```

내포된 관계에서 else는 가장 가까운 if의 else입니다.
내포된 관계의 if문을 사용할 때는 들여쓰기를 명확하게 하는 것이 좋습니다.

```
if(score > 60)
    if(score > 70)
        if(score > 80)
            if(score > 90) <--
                System.out.println("A학점");
    else <--
        System.out.println("학점 취득 실패");
```

들여쓰기의 잘못된 예.
첫 if문의 else처럼 보이지만
마지막 if문의 else

if문 : 내포된 if문

● 예제

```
import java.util.Scanner;
public class NestedIFTest1 {
    public static void main(String args[])
    {
        Scanner stdin = new Scanner(System.in);
        System.out.print("성적을 입력하세요 : ");
        int score = stdin.nextInt();
        if (score >= 80)
        {
            System.out.println("우수 학점군에 속합니다");
            if (score >= 90)
                System.out.println("A 학점입니다");
            else
                System.out.println("B 학점입니다");
        }
    }
}
```

if문 : 내포된 if문

● 예제

```
else
{
    if (score >= 60)
    {
        System.out.println("보통학점군에 속합니다");
        if (score >= 70)
            System.out.println("C 학점입니다");
        else
            System.out.println("D 학점입니다");
    }
    else
        System.out.println("학점취득 실패");
}
}
```

조건식의 추출

● 개발하려는 프로그램의 핵심은 조건식

- 일반적인 문제에서 조건식을 명확하게 추출하는 것이 프로그램의 핵심
- 조건식은 3장에서 학습한 관계 연산자와 논리 연산자로 구성

일반적인 문제	추출된 조건식
입력된 수가 양수인 경우	<code>(input > 0)</code>
두 과목 성적이 모두 80점 이상인 경우	<code>(score1 >= 80 && score2 >= 80)</code>
두 과목 성적 중에서 하나 이상이 80점 이상인 경우	<code>(score1 >= 80 score2 >= 80)</code>
두 과목 성적의 합이 150인 경우	<code>(score1 + score2 == 150)</code>
입력된 두 수가 모두 짝수인 경우	<code>(input1 % 2 == 0 && input2 % 2 == 0)</code>
입력된 점수가 60보다 크고 100보다 작거나 같은 경우	<code>(score > 60 && score <= 100)</code> (<code>60 < score <= 100</code>) ← 허용되지 않음
직급이 7 또는 8급이고, 나이가 30대(30-39)인 경우	<code>((grade == 7 grade == 8) && (30 <= age && age <= 39))</code>

조건식 : 조건식의 추출

드모르간의 법칙

- $a \ \&\& \ b$ 는 $!(\ !a \ || \ !b)$ 과 같습니다.
- $a \ || \ b$ 는 $!(\ !a \ \&\& \ !b)$ 와 같습니다.

표 4-2 조건식과 드모르간 법칙

조건식	동일한 조건식(드모르간 법칙 적용)
$(score1 \geq 80 \ \ score2 \geq 80)$	$!(\ score1 < 80 \ \&\& \ score2 < 80)$
$(score > 60 \ \&\& \ score \leq 100)$	$!(\ score \leq 60 \ \ score > 100)$
$(input1 \% 2 == 0 \ \&\& \ input2 \% 2 == 0)$	$!(input1 \% 2 != 0 \ \ input2 \% 2 != 0)$

조건식 : 조건식의 추출

● 예제

```
import java.util.Scanner;
public class ComConditionTest1 {
    public static void main(String args[])
    {
        Scanner stdin = new Scanner(System.in);
        System.out.print("월을 입력하세요 : ");
        int month = stdin.nextInt();
        if (3 <= month && month <= 5 )
            System.out.println("봄 입니다");
        else if (6 <= month && month <= 8 )
            System.out.println("여름 입니다");
        else if (9 <= month && month <= 11 )
            System.out.println("가을 입니다");
        else if (1 == month || month == 2 || month == 12 )
            System.out.println("겨울 입니다");
        else
            System.out.println("해당되는 계절이 없습니다");
    }
}
```


조건식 : 단락 평가 연산자

- 논리 연산자와 비트 논리 연산자가 다르게 동작한다

```
int a=10, b=20;  
System.out.println((a >= 20) & (b >= 20)); ←----- false 출력  
System.out.println((a >= 20) && (b >= 20)); ←----- false 출력
```

- 결과는 같지만, 실제 실행은 다르다
- 논리 연산자는 단락 평가 연산자로서 한 쪽을 평가하여 다른 한 쪽을 평가할 필요가 없는 경우 바로 결과를 반환하지만, 비트 논리 연산자는 그 경우에도 남은 부분을 수행한다

조건식 : 단락 평가 연산자

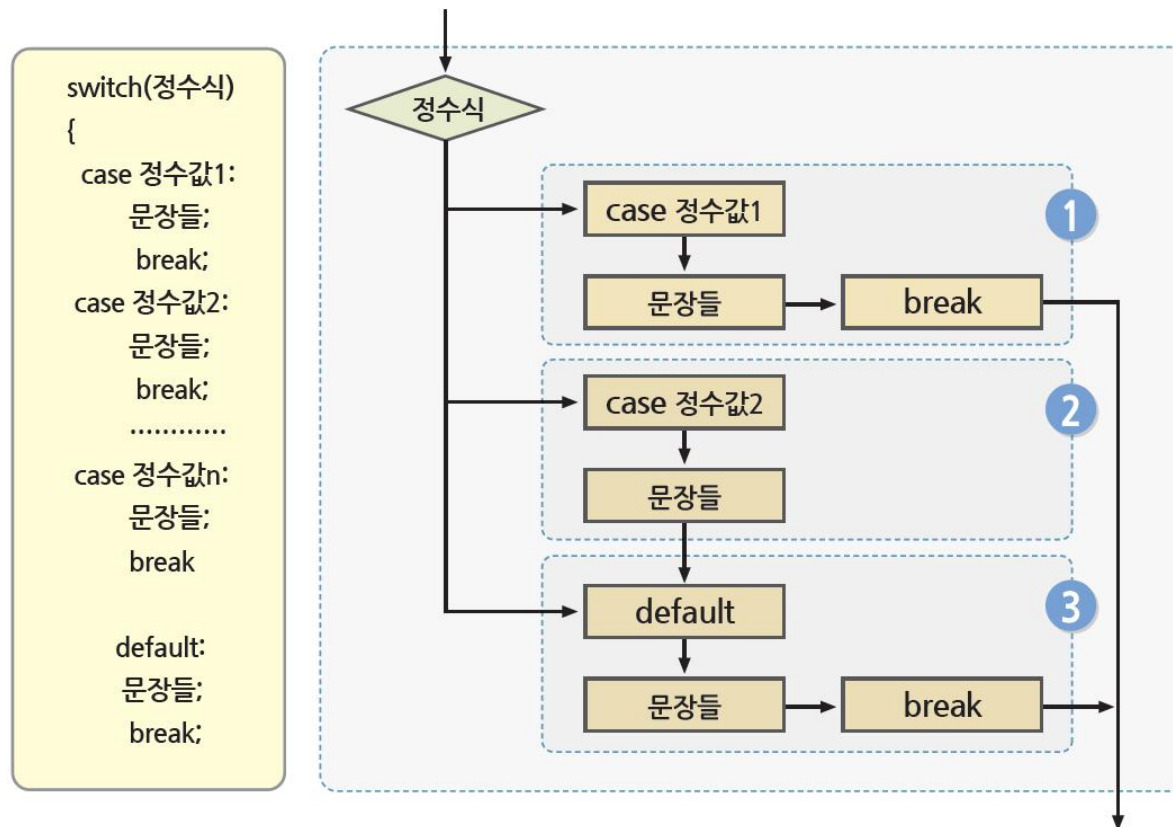
```
int a=10, b=20;
if (a >=20 & ++b >= 20) <----- 두 번째 피연산자도 평가되어 b값이 21로 증가
    System.out.println("true");
else
    System.out.println("false"); <----- false 출력
System.out.println(b); <----- 21이 출력
```

```
int a=10, b=20;
if (a >=20 && ++b >= 20) <----- 첫 번째 피연산자가 false이므로 두 번째 피연산자를 평가하지 않는다.
    System.out.println("true");
else
    System.out.println("false"); <----- false 출력
System.out.println(b); <----- 20이 출력
```

Switch문

● 다중 선택 기능을 제공하기 위한 switch문

- 정수값을 가지는 정수식에 따라 선택
- case 절의 정수값은 반드시 상숫값. 변수 사용 불가



● 예제

```
import java.util.Scanner;
public class SwitchTest1 {
    public static void main(String args[])
    {
        Scanner stdin = new Scanner(System.in);
        System.out.print("월을 입력하세요 : ");
        int month = stdin.nextInt();
        String MtoS;
        switch (month)
        {
            case 12:
            case 1:
            case 2:
                MtoS = "겨울입니다.";
                break;

            case 3:
            case 4:
            case 5:
                MtoS = "봄입니다.";
                break;

            case 6:
            case 7:
            case 8:
                MtoS = "여름입니다.";
                break;
```

● 예제

```
        MtoS = "여름입니다.";
        break;
    case 9:
        System.out.print("멋진 9월과 ");
    case 10:
        System.out.print("아름다운 10월과 ");
    case 11:
        System.out.print("낙엽의 11월은 ");
        MtoS = "가을입니다.";
        break;
    default:
        MtoS = "1~12월을 벗어난 달입니다.";
        break;
    }
    System.out.println(MtoS);
}
```


● 예제

```
import java.util.Scanner;
public class SwitchTest2 {
    public static void main(String args[])
    {
        Scanner stdin = new Scanner(System.in);
        System.out.print("숫자를 입력하세요 : ");
        int number = stdin.nextInt();
        String MtoS;
        switch (number % 3)
        {
            case 0:
                System.out.println("가위를 선택하셨습니다");
                break;
            case 1:
                System.out.println("바위를 선택하셨습니다");
                break;
            case 2:
                System.out.println("보를 선택하셨습니다");
                break;
        }
    }
}
```