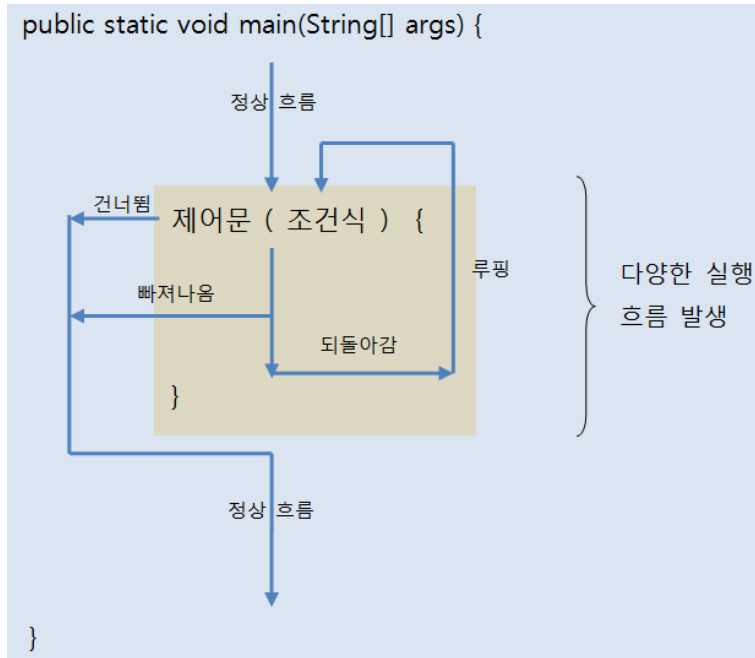


04장 조건문과 반복문

4.1 코드 실행 흐름 제어

- 정상적인 코드 실행 흐름
 - main() 메소드의 시작인 중괄호 { 에서 끝 중괄호 } 까지 위->아래 방향으로 실행
- 제어문의 역할
 - 코드 실행 흐름을 개발자가 원하는 방향으로 변경할 수 있도록 도와줌



- 제어문의 종류
 - 조건문: if문, switch문
 - 반복문: for문, while문, do-while문
 - break문, continue문
- 제어문의 중첩
 - 제어문의 중괄호 내부에 다른 제어문 작성 가능 -> 다양한 흐름 발생 가능

4.2 조건문(if문, switch문)

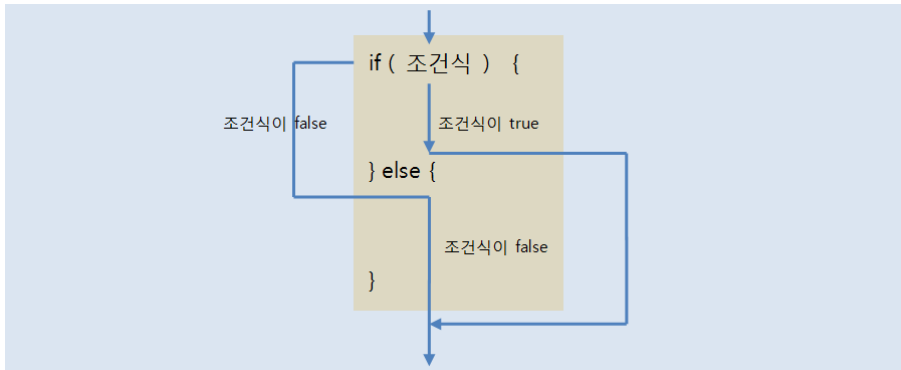
4.2.1 if문

- 조건식 결과 따라 중괄호 { } 블록을 실행할지 여부 결정할 때 사용
- 조건식
 - true 또는 false값을 산출할 수 있는 연산식
 - boolean 변수
 - 조건식이 true이면 블록 실행하고 false 이면 블록 실행하지 않음
- 기본형

```
if ( 조건식 ) {  
    실행문;  
    실행문;  
    ...  
}
```

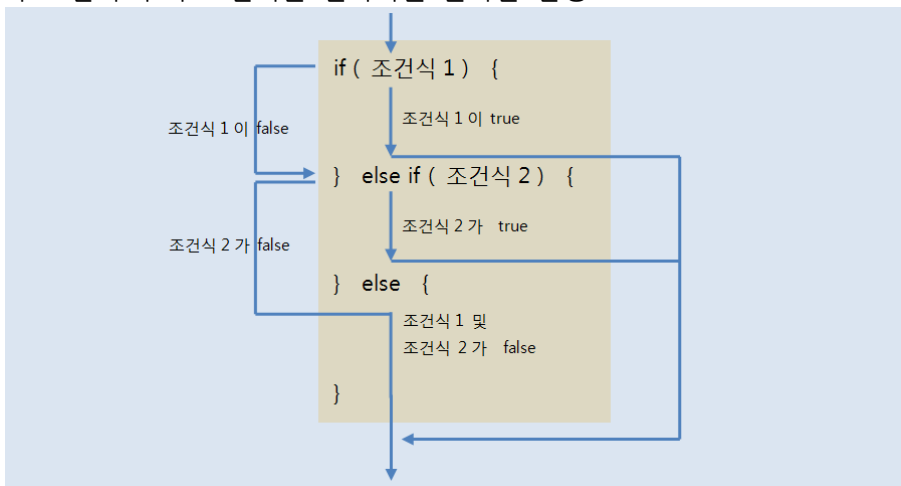
4.2.2 if-else문

- 조건식 결과 따라 실행 블록 선택



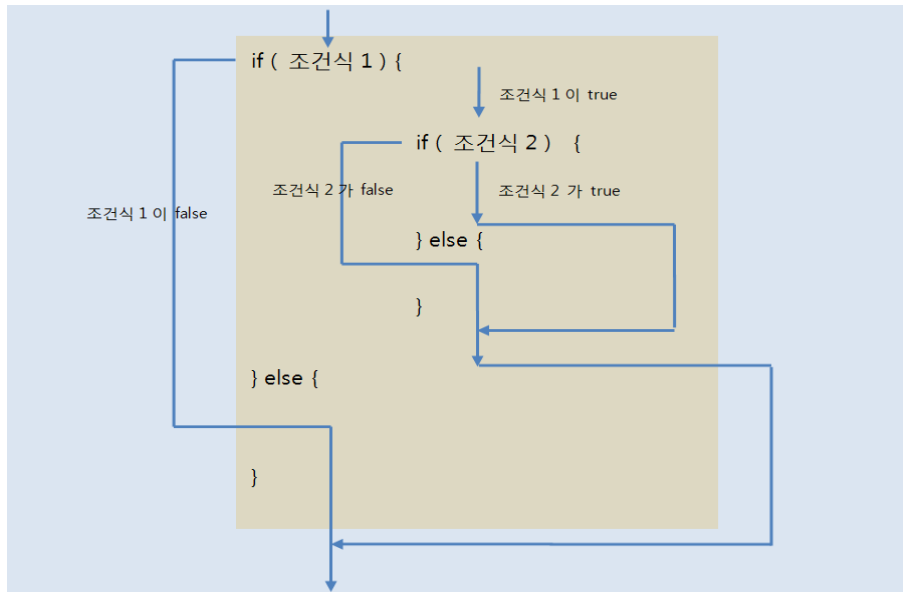
4.2.3 if-else if-else문

- 복수의 조건식 두어 조건식을 만족하는 블록만 실행



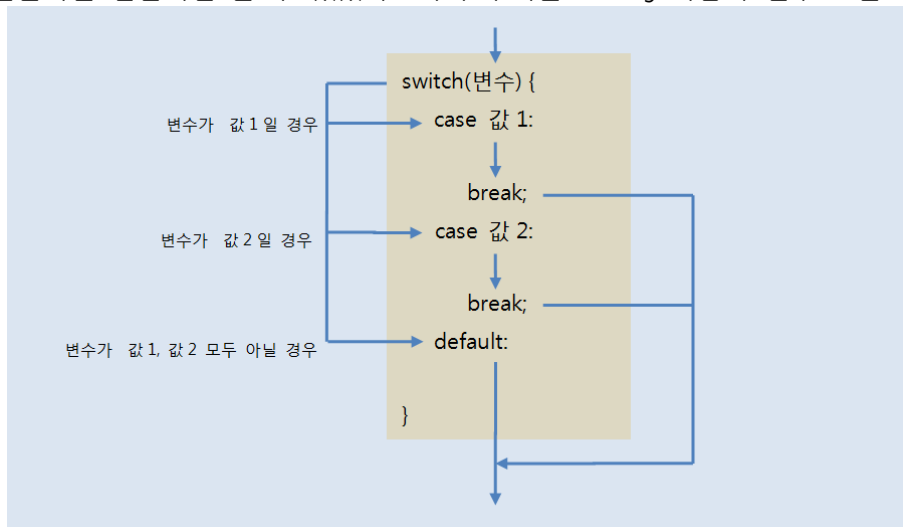
4.2.4 중첩 if문

- 코드 실행 흐름을 이해하는 것이 가장 중요



4.2.5 switch문

- 변수나 연산식의 값에 따라 실행문 선택할 때 사용
- 자바6까지는 switch문의 괄호에는 정수 타입(byte, char, short, int, long) 변수나 정수값을 산출하는 연산식만 올 수 있었다. 자바7부터는 String 타입의 변수도 올 수 있다.

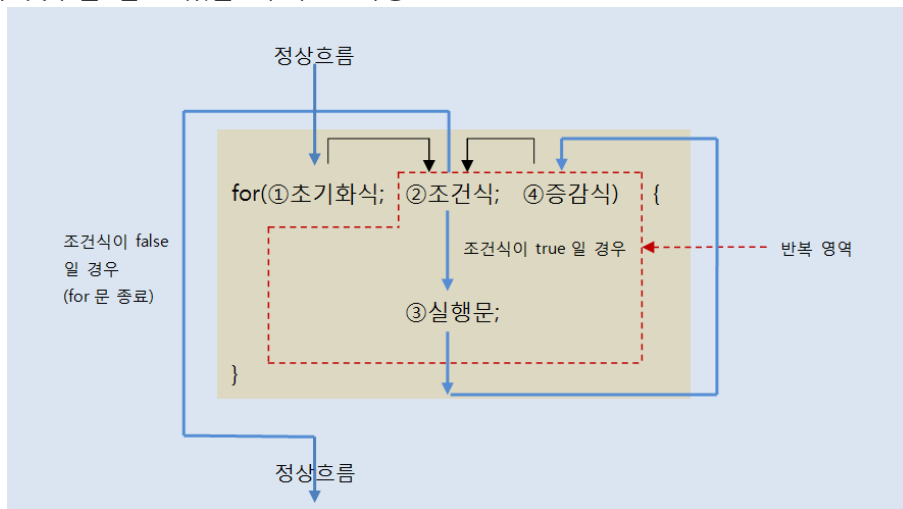


4.3 반복문(for문, while문, do-while문)

- 중괄호 블록 내용을 반복적으로 실행할 때 사용
- 종류: for문, while문, do-while문

4.3.1 for문

- 반복 횟수를 알고 있을 때 주로 사용



- 예)

```
int sum = 0;
for (int i=1; i<=100; i++) {
    sum = sum + i;
}
```

[과제] 구구단 구하기

- 구구단(2~9단)을 열방향(세로방향)으로 출력하는 프로그램을 작성하라.

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

[Gugudan.java]

```
package verify;

public class Gugudan {

    public static void main(String[] args) {

        // 단 title출력
        for (int dan = 2; dan <= 9; dan++) {
            System.out.print "[" + dan + "단" + "\t");
        }
        System.out.println();

        // ...

    }
}
```

[과제] 팩토리얼

- 키보드를 이용해서 입력한 정수의 팩토리얼을 구하는 프로그램을 작성하라. ex) $3!=6$, $5!=120$

[Factorial.java]

```
package verify;

import java.util.Scanner;

public class Factorial {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        int n, f = 1;
        System.out.print("정수를 입력 하세요?");
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();

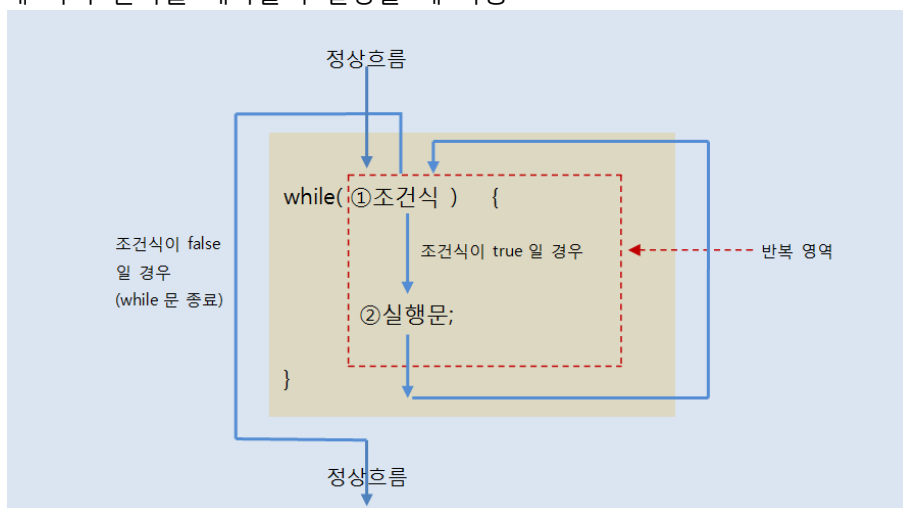
        ...

    }

}
```

4.3.2 while문

- 조건에 따라 반복을 계속할지 결정할 때 사용

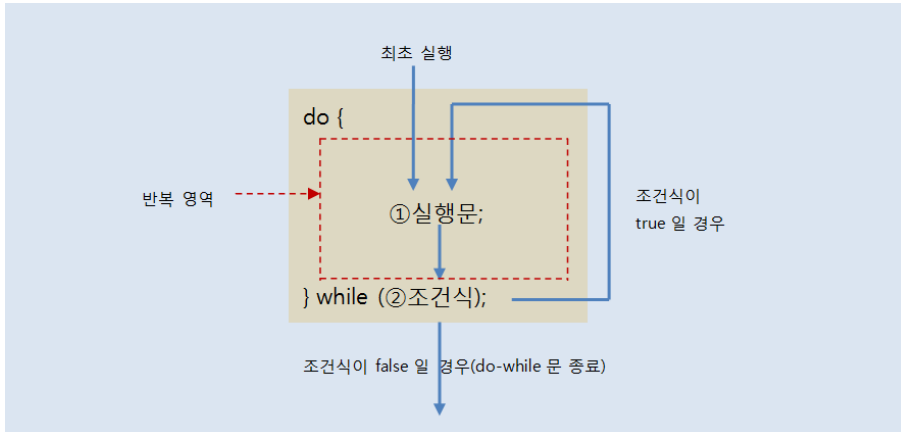


- 예)

```
int i = 1;
while (i<=10) {
    System.out.println(i);
    i++;
}
```

4.3.3 do-while문

- 조건 따라 반복 계속할지 결정할 때 사용하는 것은 while문과 동일
- 무조건 중괄호 { } 블록을 한 번 실행한 후, 조건 검사해 반복 결정



[과제] 숫자 맞추기

- 프로그램이 임의로 생성한 1부터 100사이의 숫자(ran)를 맞춰보세요. 단 입력된 값(inNum)이 ran보다 크면 "작은 수를 입력하세요", ran보다 작으면 "큰수를 입력하세요", ran과 같으면 "몇번에 맞쳤습니다"라고 출력하고 종료합니다.

[DoWhileExample1.java]

```
package verify;

import java.util.Scanner;

public class DoWhileExample1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        //1~100 사이의 수
        int ran = (int)(Math.random()*100) + 1;
        int cnt = 0, inNum =0;

        // ...

        System.out.println(cnt+"번에 맞쳤습니다.");

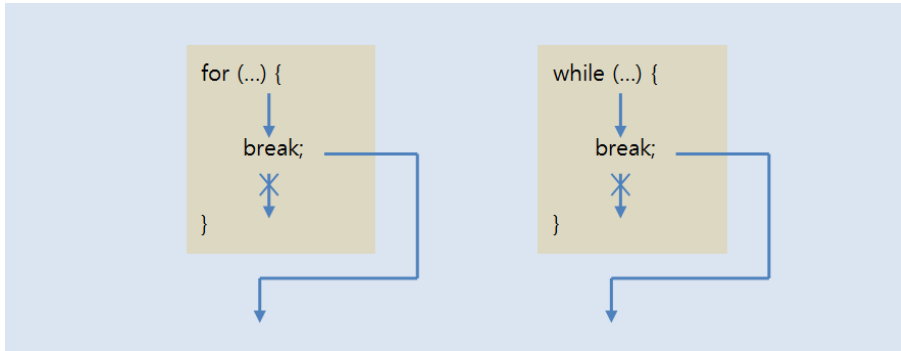
        sc.close();

    }

}
```

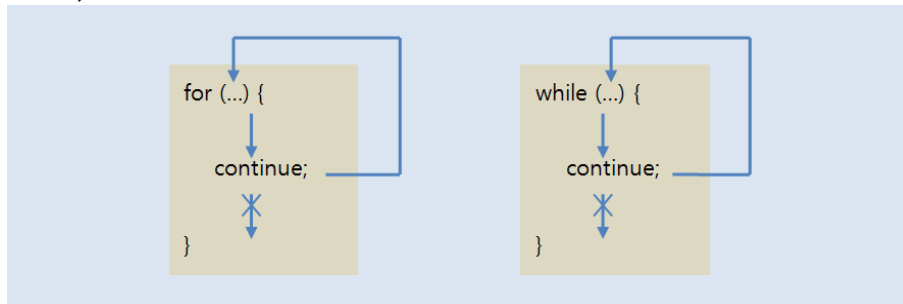
4.3.4 break문

- for문, while문, do-while문 종료 (반복 취소)
- Switch문 종료
- 대개 if문과 같이 사용
 - if문 조건식에 따라 for문과 while문 종료할 때 사용



4.3.5 continue문

- for문, while문, do-while문에서 사용
- for문: 증감식으로 이동
 - while문, do-while문: 조건식으로 이동



[과제] 확인 문제

1. 조건문과 반복문의 종류를 괄호() 속에 넣어 보세요.

- 조건문: (), ()
- 반복문: (), (), ()

2. 조건문과 반복문을 설명한 것 중 틀린 것은 무엇입니까?

- (1) if문은 조건식의 결과에 따라 실행 흐름을 달리할 수 있다.
- (2) switch문에서 사용할 수 있는 변수의 타입은 int, double이 될 수 있다.
- (3) for문은 카운터 변수로 지정한 횟수만큼 반복시킬 때 사용할 수 있다.
- (4) break문은 switch문, for문, while문을 종료할 때 사용할 수 있다.

3. for문을 이용해서 1부터 100까지의 정수 중에서 3의 배수의 총합을 구하는 코드를 작성해보세요.

[Exercise03.java]

```
package verify;

public class Exercise03 {
    public static void main(String[] args) {
        // 작성 위치
    }

    // 실행 결과
    // 3의 배수의 합: 1683
```

4. while문과 Math.random() 메소드를 이용해서 두 개의 주사위를 던졌을 때 나오는 눈을 (눈1, 눈2) 형태로 출력하고, 눈의 합이 5가 아니면 계속 주사위를 던지고, 눈의 합이 5이면 실행을 멈추는 코드를 작성해 보세요. 눈의 합이 5가 되는 조합은 (1,4), (4,1), (2,3), (3,2)입니다.

[Exercise04.java]

```
package verify;

public class Exercise04 {
    public static void main(String[] args) {
        // 작성 위치
    }

    // 실행 결과
    // (6, 4)
    // (3, 6)
    // (6, 4)
    // (3, 2)
```

5. 중첩 for문을 이용하여 방정식 $4x + 5y = 60$ 의 모든 해를 구해서 (x,y) 형태로 출력해 보세요. 단, x와 y는 10 이하의 자연수입니다.

[Exercise05.java]

```
package verify;

public class Exercise05 {
    public static void main(String[] args) {
        // 작성 위치
    }

    // 실행 결과
    // (5, 8)
    // (10, 4)
```


6. for문을 이용해서 실행 결과와 같은 삼각형을 출력하는 코드를 작성해 보세요.

[Exercise06.java]

```
package verify;

public class Exercise06 {
    public static void main(String[] args) {
        // 작성 위치
    }
}

// 실행 결과
// *
// **
// ***
// ****
// *****
```

7. while문과 Scanner를 이용해서 키보드로부터 입력된 데이터로 예금, 출금, 조회, 종료 기능을 제공하는 코드를 작성해 보세요.

[Exercise07.java]

```
package verify;

import java.util.Scanner;

public class Exercise07 {
    public static void main(String[] args) {
        boolean run = true;

        int balance = 0;

        Scanner scanner = new Scanner(System.in);

        while (run) {
            System.out.println("-----");
            System.out.println("1.예금 | 2.출금 | 3.잔고 | 4.종료");
            System.out.println("-----");
            System.out.print("선택> ");

            // 작성 위치

        }

        System.out.println("프로그램 종료");
    }
}

// 실행 결과
// -----
// 1.예금 | 2.출금 | 3.잔고 | 4.종료
// -----
// 선택> 1
// 예금액> 10000
```

```
//
// -----
// 1.예금 | 2.출금 | 3.잔고 | 4.종료
// -----
// 선택> 2
// 출금액> 2000
//
// -----
// 1.예금 | 2.출금 | 3.잔고 | 4.종료
// -----
// 선택> 3
// 잔고> 8000
//
// -----
// 1.예금 | 2.출금 | 3.잔고 | 4.종료
// -----
// 선택 4
//
// 프로그램 종료
```