

● while문: 먼저 조건식을 평가해서 조건식이 거짓이면 문장 전체를 벗어나고, 참이면 블록{} 내의 문장을 수행하고 다시 조건식으로 돌아간다. 조건식이 거짓이 될 때까지 이 과정이 계속 반복된다. for문과 while문은 항상 서로 변환이 가능하지만, 초기화나 증감식이 필요하지 않는 경우라면 while문이 더 적합하다.

- 조건식에서 호위형의 증감식이 쓰일 경우: 조건식이 평가된 후에 i의 값이 증감한다.

- while문의 조건식 안에 true값이 저장된 변수를 넣으면 변수가 고정된 값을 유지하므로 무한반복된다. 이런 경우 특정조건을 만족할 때 반복을 멈추게 하는 if문(변수 값을 false로 바꿔주는)이 반복문 안에 꼭 필요하다.

● ● ● ● ●

- while 문 안의 for문을 아무런 실행 문구 없는 문장으로 넣어(이때 {}를 쓰거나 괄호 생략하고 ;만 끝에 써 주어도 됨) 조건식과 증감식을 x번 반복하면서 while문의 실행 시간을 지연시킬 수 있다.

```
class FlowEx24 {
    public static void main(String[] args) {
        int i=11;

        System.out.println("자, 이제 카운트다운을 시작합니다.");

        while(i--!=0) {
            System.out.println(i);

            for(int j=0;j<2_000_000_000;j++) {
                ;
            }
        }

        System.out.println("GAME OVER");
    }
}
```

- 어떤 숫자의 각 자리의 합을 구하는 방식: 어떤 수를 10으로 나머지 연산하면 (% 10) 수의 마지막 자리를 얻을 수 있다. 어떤 수를 10으로 나누면 (/ 10) 마지막 한자리가 제거된다. x /= 10에 의해 한자리씩 줄어들다가 0이 되면, while문을 빠져나가는 조건식 (x!=0)을 사용해서 반복을 멈출 수 있다. (끝에 자리가 0인 것을 제외시키려면 x % 10 != 0이라는 조건을 추가하면 된다)

```

import java.util.*;

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

        System.out.print("숫자를 입력하세요.(예:12345)>");
        Scanner scanner = new Scanner(System.in);
        String tmp = scanner.nextLine(); // 화면을 통해 입력받은 내용을 tmp에 저장
        num = Integer.parseInt(tmp);      // 입력받은 문자열(tmp)을 숫자로 변환

        while(num!=0) {
            // num을 10으로 나눈 나머지(끝 자리 수)를 sum에 더함
            sum += num%10; // sum = sum + num%10;
            System.out.printf("sum=%3d num=%d\n", sum, num);

            num /= 10; // num = num / 10; num을 10으로 나눈 값을 다시 num에 저장(끝자리 지움)
        }

        System.out.println("각 자리수의 합:" + sum);
    }
}

```

- 1부터 몇까지 더하면 누적합계가 x를 넘지 않은 제일 큰 수가 되는지: sum += ++i <= 100으로 조건식을 사용해서 구할 수 있다.

```

class FlowEx26 {
    public static void main(String[] args) {
        int sum = 0;
        int i = 0;

        while((sum += ++i) <= 100) { // i를 1씩 증가시켜서 sum에 계속 더해나간다.
            System.out.printf("%d - %d\n", i, sum);
        }
    } // main의 끝
}

```

● ● ● ● ●

● do-while문: 블럭{}을 먼저 실행한 후에 조건식();을 평가한다. while문은 조건식의 결과에 따라 블럭{}이 한 번도 수행되지 않을 수 있지만, do-while문은 최소한 한 번은 수행될 것을 보장한다. 끝에 ;을 붙여야 한다.

● break문: 자신이 포함된 가장 가까운 반복문을 벗어난다. 주로 if문과 함께 사용되어 특정 조건을 만족하면 반복문을 벗어나도록 한다.

● continue문: 반복문 내에서만 사용될 수 있으며, 반복이 진행되는 도중에 continue문을 만나면 반복문의 끝으로 이동하여 다음 반복문으로 넘어간다. for문의 경우 증감식으로 이동하며, while문과 do-while문의 경우 조건식으로 이동한다. 반복문 전체를 벗어나지 않고 다음 반복을 계속 수행한다. 주로 if문과 함께 사용되어 특정 조건을 만족하는 경우에 continue문 이후에 문장들을 수행하지 않고 다음 반복으로 넘어가서 계속 진행하도록 한다. 전체 반복 중에 특정 조건을 만족하는 경우를 제외하고자 할 때 유용하다.

- 이름 붙은 반복문: 여러 개의 반복문이 중첩된 경우에는 중첩 반복문 앞에 이름을 붙이고 break문과 continue문에 이름을 지정해 줌으로써 하나 이상의 반복문을 벗어나거나 반복을 건너뛸 수 있다.

```
class FlowEx33 {
    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;
            }
            // break;
            // continue Loop1;
            // continue;
            System.out.println(i+"*"+ j +"="+ i*j);
        } // end of for i
        System.out.println();
    } // end of Loop1
}
}
```

```

import java.util.*;

class FlowEx34 {
    public static void main(String[] args) {
        int menu = 0;
        int num = 0;

        Scanner scanner = new Scanner(System.in);

        outer:          // while문 앞에 outer이라는 이름을 붙였다.
        while(true) {
            System.out.println("(1) square");
            System.out.println("(2) square root");
            System.out.println("(3) log");
            System.out.print("원하는 메뉴 (1~3)를 선택하세요. (종료:0)>");

            String tmp = scanner.nextLine(); // 화면에서 입력받은 내용을 tmp에 저장
            menu = Integer.parseInt(tmp);    // 입력받은 문자열(tmp)을 숫자로 변환

            if(menu==0) {
                System.out.println("프로그램을 종료합니다.");
                break;
            } else if (!(1<= menu && menu <= 3)) {
                System.out.println("메뉴를 잘못 선택하셨습니다. (종료는 0)");
                continue;
            }

            for(;;) {
                System.out.print("계산할 값을 입력하세요. (계산 종료:0, 전체 종료:99)>");
                tmp = scanner.nextLine(); // 화면에서 입력받은 내용을 tmp에 저장
                num = Integer.parseInt(tmp); // 입력받은 문자열(tmp)을 숫자로 변환

                if(num==0)
                    break; // 계산 종료. for문을 벗어난다.
                if(num==99)
                    break outer; // 전체 종료. for문과 while문을 모두 벗어난다.

                switch(menu) {
                    case 1:
                        System.out.println("result="+ num*num);
                        break;
                    case 2:
                        System.out.println("result="+ Math.sqrt(num));
                        break;
                    case 3:
                        System.out.println("result="+ Math.log(num));
                        break;
                }
            } // for(;;)
        } // while의 끝
    } // main의 끝
}

```