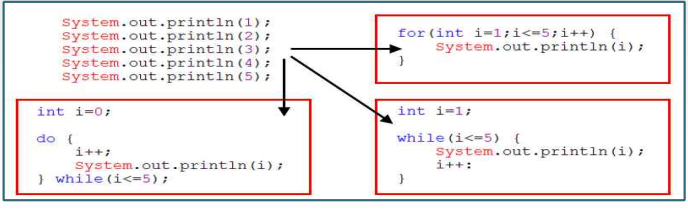
제5장
반복문(for, while, do~while)

1. 반복문 이란?

- 반복문은 어떤 조건으로, 문장 또는 문장들을 반복적으로 실행할 때, 사용된다.
- 반복문도 역시 조건식과 동일하게 조건문과 실행될 블록{ }으로 구성된다.
- **자바에서는 for문, while문, do~while문 세 가지가 있다.**
- 보통, 현업에서는 for문과 while문이 잘 사용되어지며, **do~while문은 while문의 변형된 형태로 실행될 블록{ }이 최소한 한번은 수행될 것을 보장**할 때, 사용한다.

* 중요 : 일반적 반복(for문), 무한 루프용(while문)



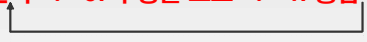
2. for문(single loop)

- 기본 형태

```
for(초기화; 조건식; 증감식) {
    //조건식이 true이면 실행될 코드 작성
}
```

- 실행 순서(중요함)

1. 초기화 → 2. 조건식 → 3. 수행될 코드 → 4. 증감식



ex) 1에서 5까지의 정수 더하기

```
int total = 0;

for(int i=1; i<=5; i++)  {
    total += i;
}
System.out.println("1~5까지의 합 : " + total);
```

2. for문 - 예제

```
int sum = 0;
int i = 0;

for(i=1; i<=100; i++) {
    sum += i; //복합대입연산자
    //if(sum >= 2200)
    //  break;
}
System.out.println("1~" + (i-1) + " 합 : " + sum);
```

3. 중첩 for문(double loop)

- 기본 형태

```
for(초기화; 조건식; 증감식) {
 for(초기화; 조건식; 증감식) {
        // 수행할 코드 작성
    }
 }
```

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

- 실행 순서(중요함)

1. 외부 초기화 → 2. 외부 조건식 → 3. 내부 초기화 → 4. 내부 조건 ->
5. 내부 수행할 코드 → 6. 내부 증감식 → 7. 외부 증감식

1, 2, 3, 4, 5, 6, 4, 5, 6(반복), 7, 2, 3, 4, 5, 6(반복) ……이렇게 수행한다.

4. while문

- 기본 형태

```
while(조건식) {
    // 조건식의 연산결과가 true일 때 실행코드
}
```

- 조건이 맞지 않으면, 1번도 수행 안될 수 있다.
- 통상, break문과 더불어 무한 루프 돌릴 때, 사용한다.

```
int i=10;

while(i < 10) {
    System.out.println(i--);
}
```

```
int i=1;

while (true) {
    //if(i>=11)
    //  break;
    System.out.println(i);
    i++;
}
```

5. 중첩 while문

- 기본 형태

```
while(조건식) {
        while(조건식){
        // 조건식의 연산결과가 true일 때 실행코드
        }
}
```

- for문이나 while문은 중첩의 제한은 없다.
- 하지만, 중첩while문은 사용빈도가 현저히 낮다.
- 뒤에 배울 배열과 궁합이 잘 맞는 것은 while문보다 for문이 더 낫다.
- while문은 무한 루프에 자주 사용하다.

6. do~while문

- 기본 형태

  do{

  //실행될 코드

  }while(조건문)

- while문에서 파생된 것으로, 블록{ }을 먼저 수행한 다음 조건식을 참조한다.
- 최소한 1번 이상 수행될 것을 보장한다.

```
Scanner scanner = new Scanner(System.in);
System.out.println("메시지를 입력하세요");
System.out.println("프로그램을 종료하려면 q를 입력하세요.");
String inputString = null;
//do-while문은 최소 한번은 무조건 실행한다.이유는 조건이 뒤에 있기 때문이다.
//(while과의 차이점)
do {
     System.out.print(">");
     inputString = scanner.nextLine();
     System.out.println(inputString);
  //q를 입력하면 조건이 false이므로 루프 빠져나간다.
} while(!inputString.equals("q"));
```

7. break문

- 하나의 반복문 또는 switch문을 빠져 나올 때, 사용한다.
- 주로, if문과 함께 사용하여 특정 조건을 만족하면 반복문을 벗어나는 용도로 사용함.

```
int i=1;
//if문이 없다면, 계속 무한루프를 돌것이다.
while(true) {
    if(i>=11) {
        break;
    }
    System.out.println(i);
    i++;
}
```

8. continue문

- 자신이 포함된 반복문의 끝으로 이동한다.(즉, 다음 반복문 실행함)
- continue문 이후의 문장은 실행되지 아니한다.
- break문과 달리 반복문을 빠져나가지 않는다.

```
System.out.println("2의 배수를 출력하는 프로그램입니다.");
for(int i=1; i<=20; i++) {
    //2의배수
    if(i%2 != 0) {
        continue;
    }
    System.out.println(i);
}
```

감사합니다.