

1. 반복문

■ while 반복문

- 주어진 조건식이 참(True)일 동안에만 동일한 명령문을 반복 수행
- 반복문을 빠져나오는 순간은 주어진 조건식이 거짓(False)이 될 때

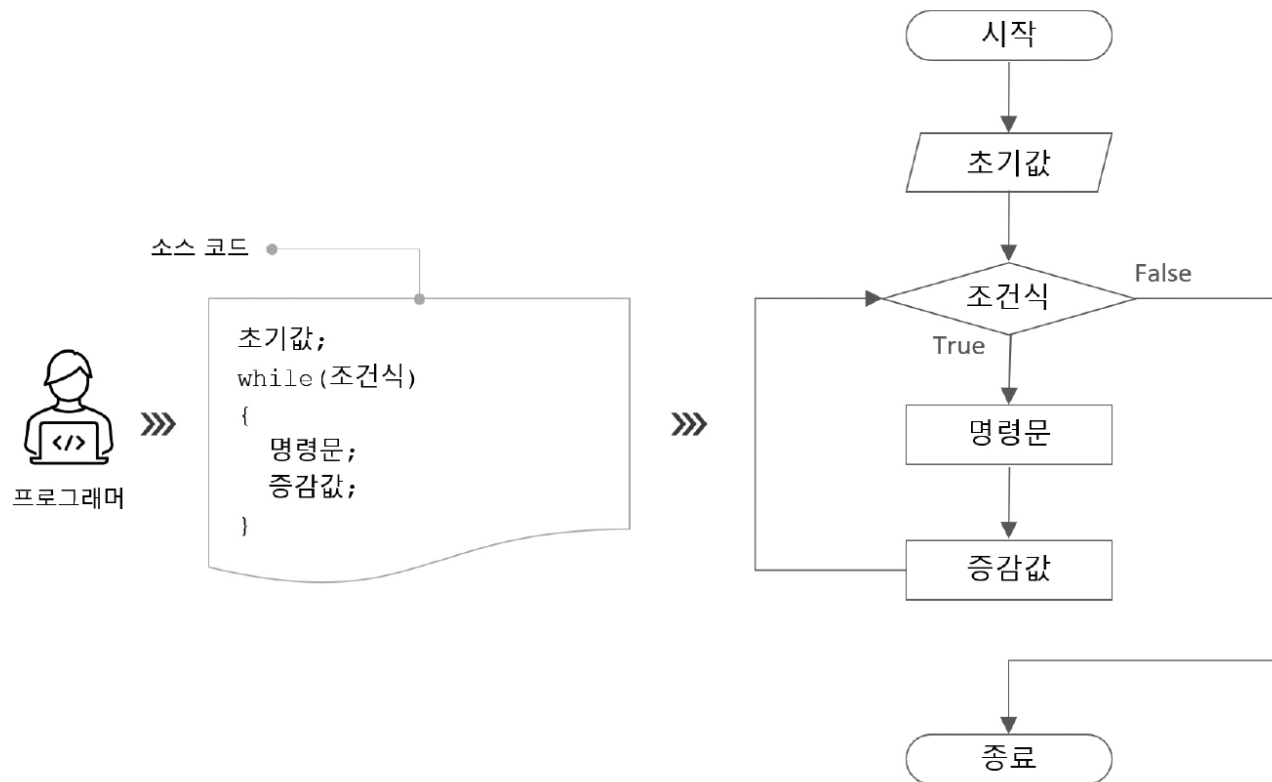


그림 5-1 while 반복문

- **Step 01** | 프로젝트 생성 : 프로젝트명은 'WhileSum'으로 입력합니다. 소스 파일명은 그대로 둡니다.
- **Step 02** | 소스 코드 입력 : while 반복문으로 1부터 100까지 누적 합계를 산출합니다. 그리고 출력하기 위해 다음과 같이 소스 코드를 입력합니다. 단축키 **[F5]**를 눌러 실행 결과를 확인합니다.

```
01 // See https://aka.ms/new-console-template for more information
02 int a;
03
04 Console.Write("정수 입력 : ");
05 a = int.Parse(Console.ReadLine());
06
07 if(a > 0)
08 {
09     Console.WriteLine("양의 정수입니다.");
10 }
```

■ 실행 결과

while 반복문 종료 후 결과값

- > 누적 합계 : $1 + 2 + \dots + 99 + 100 = 5050$
- > 카운트 변수값 : 101

- **Step 01** | 프로젝트 생성 : 프로젝트명은 'WhileInputSum'으로 입력합니다. 소스 파일명은 그대로 둡니다.
- **Step 02** | 소스 코드 입력 : while 반복문으로 1부터 콘솔 창에서 입력한 수까지 누적 합계를 산출합니다. 그리고 출력하기 위해 다음과 같이 소스 코드를 입력합니다. 단축키 [F5]를 눌러 실행 결과를 확인합니다.

```
01 // See https://aka.ms/new-console-template for more information
02 int su, count, sum = 0;
03
04 count = 1;
05
06 Console.Write("누적합을 어디까지 구할까요 : ");
07 su = int.Parse(Console.ReadLine());
08
09 while(count <= su) // 입력한 수까지 반복문 수행
10 {
11     sum += count;
12     count++;
13 }
14 Console.WriteLine(" > 1부터 " + su + "까지 누적 합계 : " + sum);
15 Console.WriteLine(" > 카운트 변수값 : " + count);
```

■ 실행 결과

누적합을 어디까지 구할까요 : 50

> 1부터 50까지 누적 합계 : 1275

> 카운트 변수값 : 51

■ 실행 결과

누적합을 어디까지 구할까요 : -80

> 1부터 50까지 누적 합계 : 0

> 카운트 변수값 : 1

- **Step 01** | 프로젝트 생성 : 프로젝트명은 'WhileIfElseSum'으로 입력합니다. 소스 파일명은 그대로 둡니다.
- **Step 02** | 소스 코드 입력 : if...else 조건문으로 음의 정수는 배제하고 양의 정수만 허용하여 누적 합계를 산출합니다. 그리고 출력하기 위해 다음과 같이 소스 코드를 입력합니다. 단축키 F5를 눌러 실행 결과를 확인합니다.

```
01 // See https://aka.ms/new-console-template for more information
02 int su, count, sum = 0;
03
04 count = 1;
05
06 Console.Write("누적합을 어디까지 구할까요 : ");
07 su = int.Parse(Console.ReadLine());
08
09 if(su < 0)
10 {
11     Console.WriteLine(" > 유효하지 않은 데이터값! ");
12     Console.WriteLine(" > 프로그램 종료! ");
13 }
14 else
15 {
16     while (count <= su) // 입력한 수까지 반복문 수행
17     {
18         sum += count;
19         count++;
20     }
21 }
```



```
20      }  
21      Console.WriteLine(" > 1부터 " + su + "까지 누적 합계 : " + sum);  
22      Console.WriteLine(" > 카운트 변수값 : " + count);  
23  }
```

■ 실행 결과

누적합을 어디까지 구할까요 : 100
> 1부터 50까지 누적 합계 : 5050
> 카운트 변수값 : 101

■ 실행 결과

누적합을 어디까지 구할까요 : -80
> 유효하지 않은 데이터값!
> 프로그램 종료!

1. 반복문

■ do...while 반복문

- 무조건 한 번은 명령문을 수행 후 마지막에 조건식을 판별하여 반복 여부를 판별

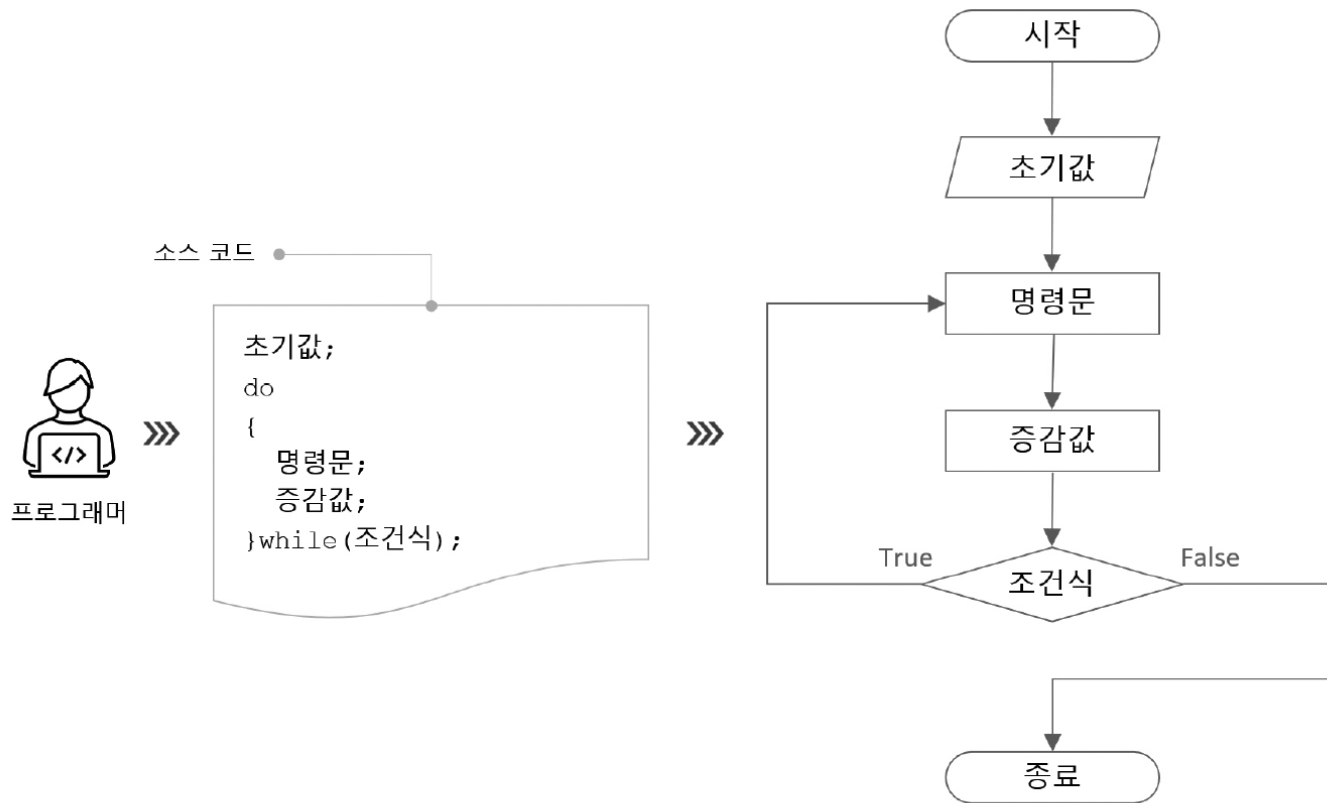


그림 5-2 do...while 반복문

- **Step 01** | 프로젝트 생성 : 프로젝트명은 'DoWhile'로 입력합니다. 소스 파일명은 그대로 둡니다.
- **Step 02** | 소스 코드 입력 : do...while 반복문으로 'exit' 문자열이 입력될 때까지 문자열 입력 명령을 반복 수행합니다. 그리고 출력하기 위해 다음과 같이 소스 코드를 입력합니다. 단축키 **[F5]**를 눌러 실행 결과를 확인합니다.

```
01 // See https://aka.ms/new-console-template for more information
02 int count = 1;
03 string s;
04
05 Console.WriteLine("문자열 exit가 입력되면 종료 ");
06
07 do
08 {
09     Console.Write(" > " + count + "회 문자열 입력 : ");
10     s = Console.ReadLine();
11     count++;
12 }while (s != "exit");
13
14 Console.WriteLine(" > 반복 수행 횟수 : " + (count - 1) + "회 ");
```

■ 실행 결과

문자열 exit가 입력되면 종료

- > 1회 문자열 입력 : space
- > 2회 문자열 입력 : korea
- > 3회 문자열 입력 : exit
- > 반복 수행 횟수 : 3회

1. 반복문

■ for 반복문

- for(초기값; 조건식; 증감값) 형식으로 문법구조가 비교적 간단
- 다른 반복문에 비해 명령문을 반복 수행하는 절차 이해는 까다로움

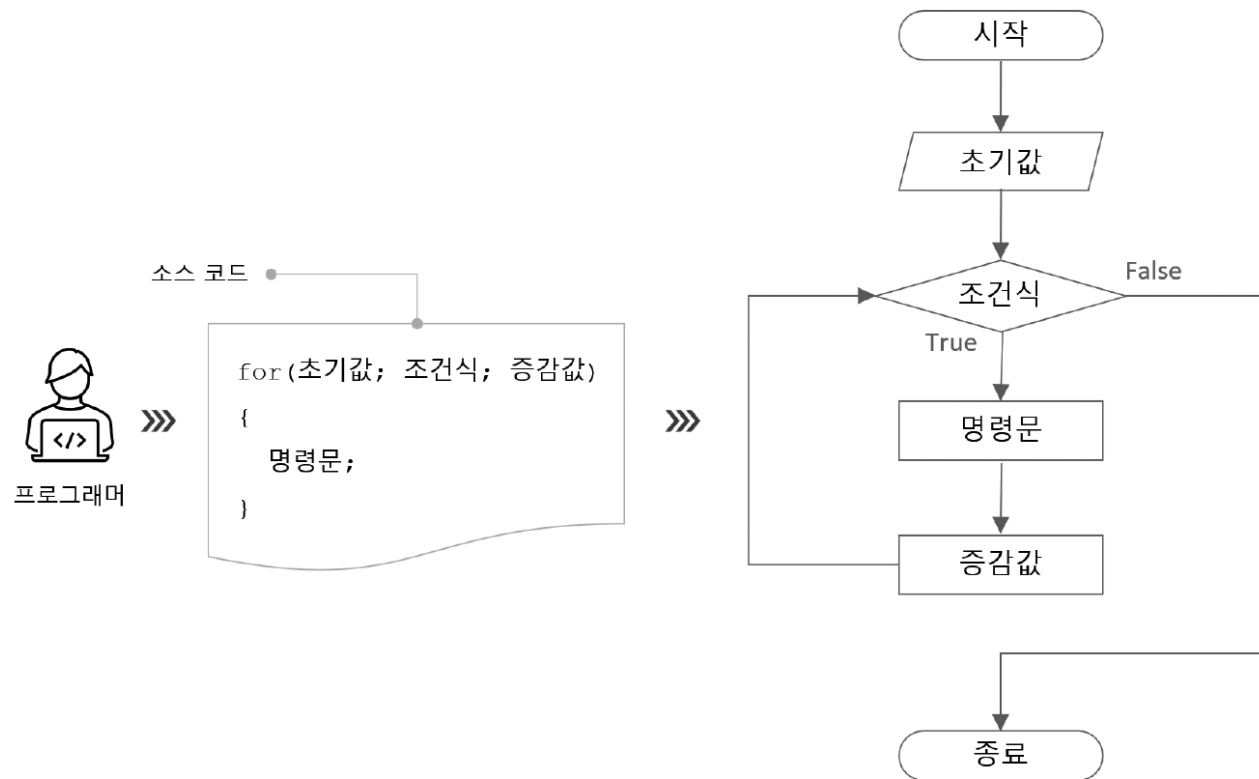


그림 5-3 for 반복문

- **Step 01** | 프로젝트 생성 : 프로젝트명은 'ForSum'으로 입력합니다. 소스 파일명은 그대로 둡니다.
- **Step 02** | 소스 코드 입력 : for 반복문으로 1부터 입력한 수까지 누적되는 합계를 산출합니다. 그리고 출력하기 위해 다음과 같이 소스 코드를 입력합니다. 단축키 **F5**를 눌러 실행 결과를 확인합니다.

//see <https://aka.ms/new-console-template> for more information

```
for (int i = 0; i<10; i++)  
{  
    console.WriteLine("{0}번째 출력합니다.", i);  
}
```


//see <https://aka.ms/new-console-template> for more information

```
for (int i=0; i<3; i++)
{
    Console.WriteLine("i={0}", i);

    for (int j=0; j<3; j++)
    {
        Console.WriteLine("j가 {0} 출력합니다.", j);
    }
}
```

```
01 // See https://aka.ms/new-console-template for more information
02 int count, su, sum = 0;
03
04 Console.Write("누적합을 어디까지 구할까요 : ");
05 su = int.Parse(Console.ReadLine());
06
07 for(count = 1; count <= su; count++)
08 {
09     sum += count; // sum = sum + count;
10 }
11
12 Console.WriteLine(" > 1부터 " + su + "까지 누적 합계 : " + sum);
13 Console.WriteLine(" > 카운트 변수값 : " + count);
```

■ 논리 오류가 발생한 실행 결과

— 실행 결과

누적합을 어디까지 구할까요 : 100

> 1부터 50까지 누적 합계 : 5050

> 카운트 변수값 : 101

— 실행 결과

누적합을 어디까지 구할까요 : -80

> 1부터 50까지 누적 합계 : 0

> 카운트 변수값 : 1

- **Step 01** | 프로젝트 생성 : 프로젝트명은 'ForIFesleSum'으로 입력합니다. 소스 파일명은 그대로 둡니다.
- **Step 02** | 소스 코드 입력 : if...else 조건문으로 음의 정수는 배제하고 양의 정수만 허용하여 누적 합계를 산출합니다. 그리고 출력하기 위해 다음과 같이 소스 코드를 입력합니다. 단축키 F5를 눌러 실행 결과를 확인합니다.

```
01 // See https://aka.ms/new-console-template for more information
02 int count, su, sum = 0;
03
04 Console.Write("누적합을 어디까지 구할까요 : ");
05 su = int.Parse(Console.ReadLine());
06
07 if(su < 0)
08 {
09     Console.WriteLine(" > 유효하지 않은 데이터값! ");
10     Console.WriteLine(" > 프로그램 종료! ");
11 }
12 else
13 {
14     for (count = 1; count <= su; count++)
15     {
16         sum += count; // sum = sum + count;
17     }
18
19     Console.WriteLine(" > 1부터 " + su + "까지 누적 합계 : " + sum);
20     Console.WriteLine(" > 카운트 변수값 : " + count);
21 }
```

■ 논리 오류를 해결한 실행 결과

실행 결과

누적합을 어디까지 구할까요 : -80

- > 유효하지 않은 데이터값!
- > 프로그램 종료!

실행 결과

누적합을 어디까지 구할까요 : 100

- > 1부터 50까지 누적 합계 : 5050
- > 카운트 변수값 : 101

- **Step 01** | 프로젝트 생성 : 프로젝트명은 'ForGugudan'으로 입력합니다. 소스 파일명은 그대로 둡니다.
- **Step 02** | 소스 코드 입력 : for 반복문으로 입력한 숫자의 구구단을 출력하기 위해 다음과 같이 소스 코드를 입력합니다. 단축키 **[F5]**를 눌러 실행 결과를 확인합니다.

```
01 // See https://aka.ms/new-console-template for more information
02 int dan, count, gob;
03
04 Console.Write("출력할 구구단 : ");
05 dan = int.Parse(Console.ReadLine());
06
07 if (dan < 2 || dan > 99)
08 {
09     Console.WriteLine(" > 유효하지 않은 데이터값! ");
10     Console.WriteLine(" > 프로그램 종료! ");
11 }
12 else
13 {
14     for (count = 1; count <= 9; count++)
15     {
16         gob = dan * count;
17         Console.WriteLine(dan + " * " + count + " = " + gob);
18     }
19 }
```


■ 유효하지 않은 데이터값 실행 결과

실행 결과

출력할 구구단 : -5

- > 유효하지 않은 데이터값!
- > 프로그램 종료!

실행 결과

출력할 구구단 : 100

- > 유효하지 않은 데이터값!
- > 프로그램 종료!

■ 유효한 데이터값 실행 결과

— 실행 결과

출력할 구구단 : 8

$$8 * 1 = 8$$

$$8 * 2 = 16$$

$$8 * 3 = 24$$

$$8 * 4 = 32$$

$$8 * 5 = 40$$

$$8 * 6 = 48$$

$$8 * 7 = 56$$

$$8 * 8 = 64$$

$$8 * 9 = 72$$

- **Step 01** | 프로젝트 생성 : 프로젝트명은 'ForNestingGugudan'으로 입력합니다. 소스 파일명은 그대로 둡니다.
- **Step 02** | 소스 코드 입력 : 중첩된 for 반복문으로 구구단을 출력하기 위해 다음과 같이 소스 코드를 입력합니다. 단축키 **[F5]**를 눌러 실행 결과를 확인합니다.

```
01 // See https://aka.ms/new-console-template for more information
02 int dan, count, gob;
03
04 for (dan = 2; dan <= 9; dan++)
05 {
06     for(count = 1; count <= 9; count++)
07     {
08         gob = dan * count;
09         Console.WriteLine(dan + " * " + count + " = " + gob);
10     }
11     Console.WriteLine();
12 }
```

실행 결과

$$2 * 1 = 2$$

$$2 * 2 = 4$$

$$2 * 3 = 6$$

$$2 * 4 = 8$$

$$2 * 5 = 10$$

$$2 * 6 = 12$$

$$2 * 7 = 14$$

$$2 * 8 = 16$$

$$2 * 9 = 18$$

$$3 * 1 = 3$$

$$3 * 2 = 6$$

$$3 * 3 = 9$$

:

- **Step 01** | 프로젝트 생성 : 프로젝트명은 'ForNestingStar'로 입력합니다. 소스 파일명은 그대로 둡니다.
- **Step 02** | 소스 코드 입력 : 중첩된 for 반복문으로 ☆을 5개부터 1개까지 개수가 1개씩 점차 줄어들게 수행합니다. 그리고 출력하기 위해 다음과 같이 소스 코드를 입력합니다. 단축키 **[F5]**를 눌러 실행 결과를 확인합니다.

```
01 // See https://aka.ms/new-console-template for more information
02 int star, count;
03
04 for (count = 1; count <= 5; count++)
05 {
06     for (star = 5; star >= count; star--)
07     {
08         Console.Write("☆");
09     }
10     Console.WriteLine();
11 }
```

■ 실행 결과

☆☆☆☆☆

☆☆☆☆

☆☆☆

☆☆

☆

2. 점프문

■ break 점프문

- 반복문이나 switch...case 조건문을 빠져나올 때 사용
- 중첩된 반복문의 경우 break 문이 속해 있는 반복문 또는 case 문의 위치에서만 빠져나오게 됨

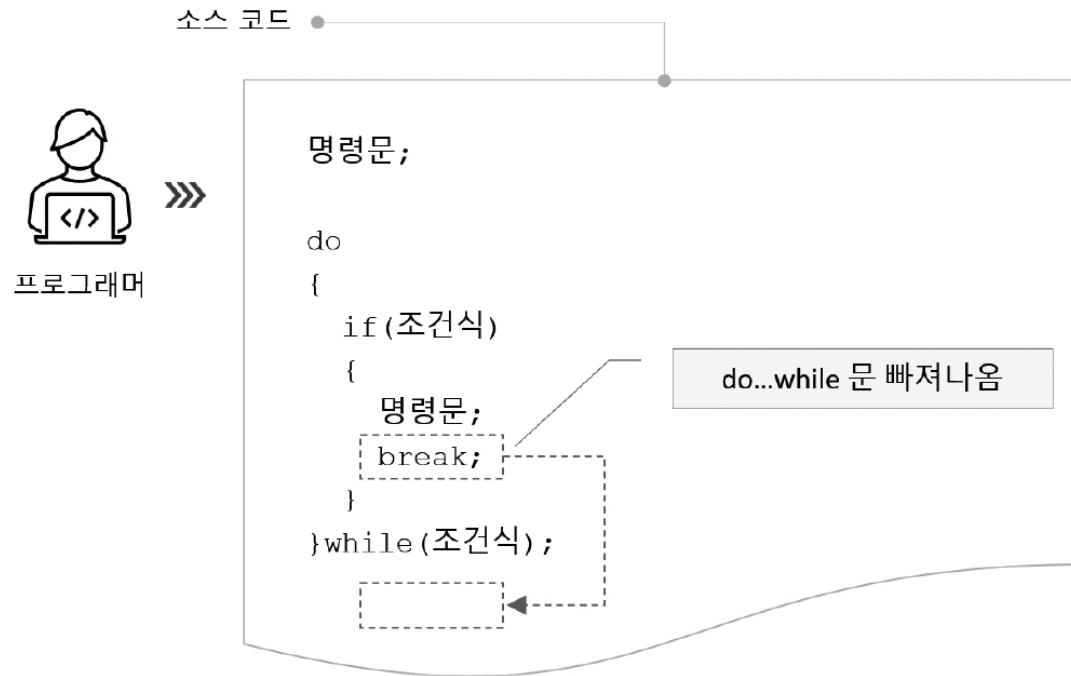


그림 5-4 break 점프문

- **Step 01** | 프로젝트 생성 : 프로젝트명은 'DowhileBreak'로 입력합니다. 소스 파일명은 그대로 둡니다.
- **Step 02** | 소스 코드 입력 : do...while 반복문으로 1부터 100까지 누적 합계를 산출합니다. 그리고 누적합이 500을 넘게 되면 반복문을 빠져나옵니다. 누적 합계를 출력하기 위해 다음과 같이 소스 코드를 입력합니다. 단축키 **[F5]**를 눌러 실행 결과를 확인합니다.

```
01 // See https://aka.ms/new-console-template for more information
02 int count = 1, sum = 0;
03
04 Console.WriteLine("누적합이 500을 넘으면 프로그램 종료 ");
05
06 do
07 {
08     if (sum > 500)
09     {
10         Console.WriteLine("> 누적합 500 초과! ");
11         break;
12     }
13     else
14     {
15         sum += count; // sum = sum + count;
16     }
17     count++;
18 } while (count <= 100); // 또는 while (true) → 무한 반복문 구현 가능
19
20 Console.WriteLine("> 누적 합계 : " + sum);
```

■ 실행 결과

누적합이 500을 넘으면 프로그램 종료

- > 누적합 500초과!
- > 누적 합계 : 528

2. 점프문

■ continue 점프문

- 반복문의 처음 위치로 강제 이동
- 짝수 또는 홀수의 합을 구할 때 조건문의 코드 블록 안에 선언

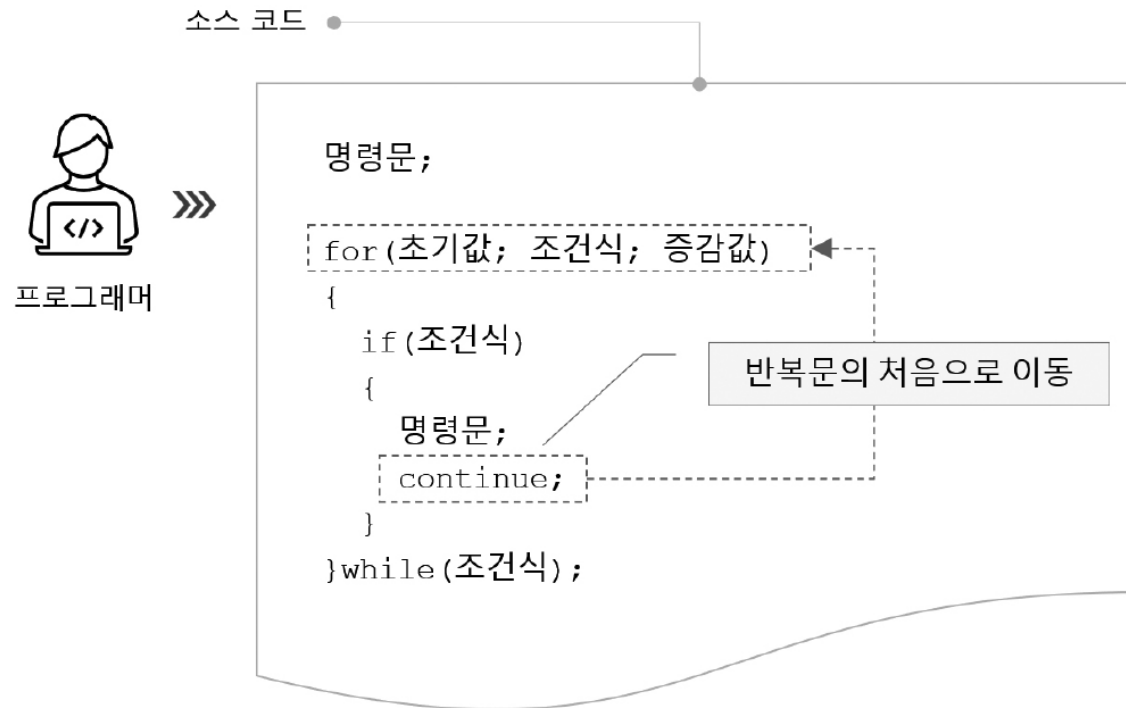


그림 5-5 continue 점프문

- **Step 01** | 프로젝트 생성 : 프로젝트명은 'ForContinue'로 입력합니다. 소스 파일명은 그대로 둡니다.
- **Step 02** | 소스 코드 입력 : for 반복문을 사용하여 1부터 10까지의 숫자 중 홀수만 출력합니다. 홀수는 나머지 연산자(%)와 continue 점프문을 사용하여 판별하기 위해 다음과 같이 소스 코드를 입력합니다. 단축키 **[F5]**를 눌러 실행 결과를 확인합니다.

```
01 // See https://aka.ms/new-console-template for more information
02 int count;
03
04 Console.WriteLine("홀수만 출력하는 프로그램 ");
05 Console.Write(" > ");
06
07 for (count = 1; count <= 10; count++)
08 {
09     if(count % 2 == 0)
10     {
11         continue;
12     }
13     Console.Write(count + " ");
14 }
```

실행 결과

홀수만 출력하는 프로그램

> 1 3 5 7 9

2. 점프문

■ goto 점프문

- 지정된 레이블 위치로 점프할 때 사용
- 레이블 위치를 선정하기 위해 반드시 레이블명을 선언해야 함

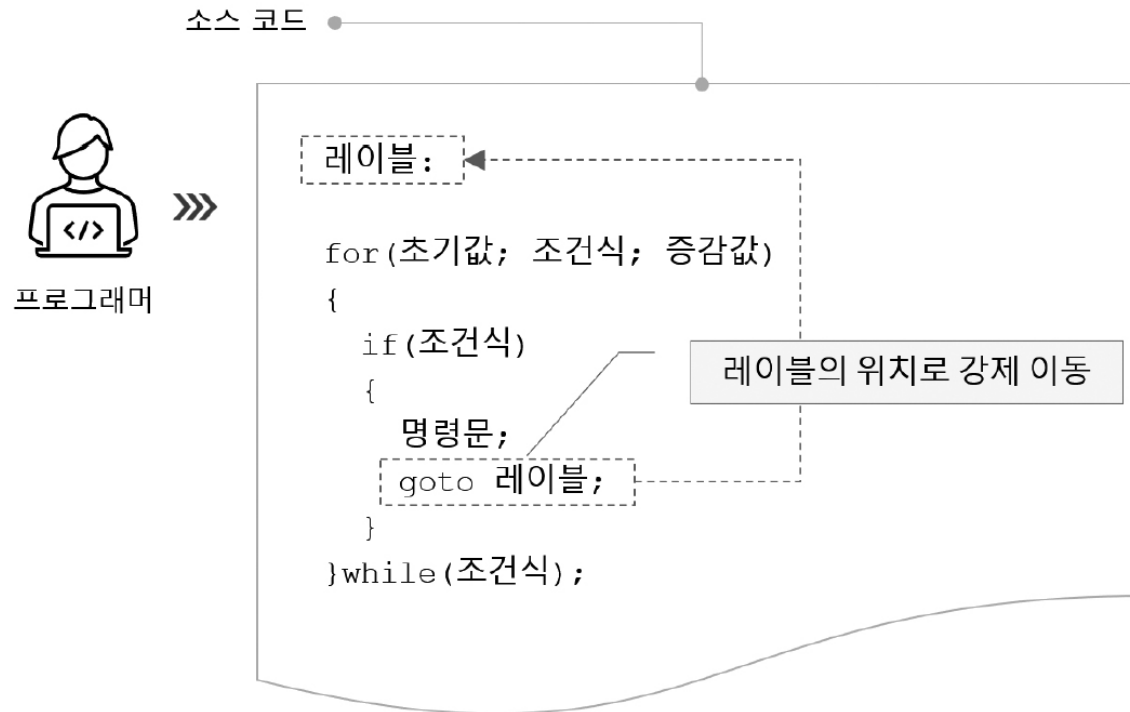


그림 5-6 goto 점프문

- **Step 01** | 프로젝트 생성 : 프로젝트명은 'WhileGoto'로 입력합니다. 소스 파일명은 그대로 둡니다.
- **Step 02** | 소스 코드 입력 : 콘솔 창에서 입력한 수가 음의 정수 데이터값일 경우 goto 점프문을 선언합니다. 그리고 다시 입력하도록 수행하기 위해 다음과 같이 소스 코드를 입력합니다. 단축키 [F5]를 눌러 실행 결과를 확인합니다.

```
01 // See https://aka.ms/new-console-template for more information
02 int su, count, sum = 0;
03
04 count = 1;
05
06 replay: // 레이블 선언
07 Console.Write("누적합을 어디까지 구할까요 : ");
08 su = int.Parse(Console.ReadLine());
09
10 if (su < 0)
11 {
12     Console.WriteLine(" > 유효하지 않은 데이터값! ");
13     Console.WriteLine();
14     goto replay;
15 }
```

```
16 else
17 {
18     while (count <= su)    // 입력한 수까지 반복문 수행
19     {
20         sum += count;
21         count++;
22     }
23     Console.WriteLine(" > 1부터 " + su + "까지 누적 합계 : " + sum);
24     Console.WriteLine(" > 카운트 변수값 : " + count);
25 }
```

■ 실행 결과

누적합을 어디까지 구할까요 : -100

> 유효하지 않은 데이터값!

누적합을 어디까지 구할까요 : 100

> 1부터 100까지 누적 합계 : 5050

> 카운트 변수값 : 101

자신감 뽐뽐!

C# 프로그래밍 Hard Carry



Thank You