



코드의 흐름 제어하기

이것이 C#이다



Contents

❖ 분기문

❖ 반복문

❖ 점프문



5.1 분기문(Branching Statement)

- ❖ 프로그램의 흐름을 조건에 따라 변화시키는 구문
- ❖ if 문
- ❖ switch문



5.1.1 if, else, 그리고 else if

❖ 한 번에 단 하나의 조건 평가

❖ 기본 형식

```
if ( 조건식 )  
{  
    // 참인 경우에  
    // 실행할  
    // 코드  
}
```

❖ 프로그램의 흐름을 여러 갈래로 나누려면

```
if ( (a % 2) == 0 )  
    Console.WriteLine("짝수");  
else  
    Console.WriteLine("홀수");
```

```
int a = - 10;  
  
if ( a < 0 )  
    Console.WriteLine("음수");  
else if ( a > 0 )  
    Console.WriteLine("양수");  
else  
    Console.WriteLine("0");
```

❖ 데모 예제 - IfElse



5.1.2 if 문 중첩해서 사용하기

- ❖ 분기문이나 반복문은 중첩 사용 가능
- ❖ If 문의 중첩 사용 예

```
if ( number > 0 )  
{
```

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

```
        Console.WriteLine("0보다 큰 짝수.");
```

```
    else
```

```
        Console.WriteLine("0보다 큰 홀수.");
```

```
}
```

중첩된 if 문

컴퓨터가 이해할 수 있는 코드는 어느 바보나 다 짤 수 있다.

좋은 프로그래머는 사람이 이해할 수 있는 코드를 짤다.

- 마틴 파울러

- ❖ 데모 예제 - IfIf



5.1.3 switch

❖ 조건식의 다양한 결과를 한 번에 평가 후 프로그램의 흐름을 나눌 때 사용

❖ 사용 형식

■ 조건식 – 정수 형식과 문자열 형식

```
switch (조건식)
```

```
{
```

```
    case 상수1: ●
```

```
        // 실행할 코드
```

```
        break;
```

```
    case 상수2: ●
```

```
        // 실행할 코드
```

```
        break;
```

```
    case 상수N: ●
```

```
        // 실행할 코드
```

```
        break;
```

```
    default: ●
```

```
        break;
```

```
}
```

조건식 == 상수1?

조건식 == 상수2?

조건식 == 상수N?

조건식이 어떤 경우와도 맞지 않은 경우,
default 절은 생략 가능합니다.

❖ 데모 예제 – Switch1



5.1.3 switch (계속)

- ❖ C# 7.0의 변화 – 조건식에 데이터 형식 사용 가능
- ❖ 데이터 형식을 조건을 사용하는 경우
 - case 절에서 데이터 형식 옆에 반드시 식별자 필요

```
object obj = 123 ;
```

C# 컴파일러는 123 리터럴을 평가하여 int 형식임을 유추합니다. 그리고 obj 안에 박싱해넣습니다.

```
switch(obj)  
{
```

```
    case int i:
```

obj에 담겨져 있는 데이터의 형식이 int이므로 프로그램은 이 case절을 따라 분기합니다.

```
        ...  
        break;
```

```
    case float f:
```

```
        ...  
        break;
```

```
    default:
```

```
        ...  
        break;
```

```
}
```

- ❖ 데모 예제 – Switch2



5.2 반복문(Loop Statement)

❖ 루프문

❖ 특정 조건을 만족하는 동안 코드(블록)을 반복 실행

❖ C#에서 지원하는 반복문

- while
- do while
- for
- foreach



5.2.1 while

❖ 형식

■ 조건식 – 논리 형식

```
int a = 10;

while ( a > 0 )
{
    Console.WriteLine ( a );
    a -= 2;
}
```

❖ 데모 예제 – While



5.2.2 do ~ while

❖ 무조건 한 번의 실행을 보장하는 반복문

❖ 사용 형식

```
do
```

```
{
```

```
    반복실행할_코드_블록
```

```
}
```

```
while( 조건식 );
```

이 코드 블록은 최초 한번은 무조건 실행됩니다.

do while 문과 while 문의 또 다른 점은 바로 ;입니다.

❖ 데모 예제 - DoWhile



5.2.3 for

- ❖ while 문보다 더 정교한 반복을 제어하는 반복문
- ❖ 사용 형식

```
for( 초기화식; 조건식; 반복식; )  
    반복실행할_코드;
```

```
for ( int i=0; i<5; i++ )  
    Console.WriteLine(i);
```

- ❖ 데모 예제 – For



5.2.4 중첩 for

❖ 반복문을 중첩할 경우 for문이 가장 적합

❖ 중첩 for의 예

```
for ( int i=0; i<5; i++ )  
    for ( int j=0; j<10; j++ )  
        Console.WriteLine("아싸!");
```

❖ 데모 예제 - ForFor



5.2.5 foreach

❖ 배열이나 컬렉션에 주로 사용

- 배열 – 여러 개의 데이터를 담는 코드 요소
- 컬렉션 – 배열과 비슷하나 데이터 저장 및 액세스 방식이 다름

❖ 배열이나 컬렉션을 순회하며 각 데이터 요소에 접근 가능

❖ 사용 형식

```
foreach(데이터형식 변수명 in 배열_또는_컬렉션)  
    코드_또는_코드블록
```

❖ 데모 예제 – ForEach



5.2.6 for 또는 while을 이용한 무한 반복 코드

❖ For문의 무한 반복

```
for( ; ; )  
    // 반복 실행할 코드 블록
```

❖ While문의 무한 반복

```
while( true )  
    // 반복 실행할 코드 블록
```

❖ 데모 예제 – InfiniteFor

❖ 데모 예제 – InfiniteWhile



5.3 점프문

❖ 실행 흐름을 끊거나 실행 위치를 원하는 곳으로 이동

❖ C#에서 제공하는 점프문

- Break
- Continue
- Goto
- Return
- Throw



5.3.1 break

- ❖ 현재 실행 중인 반복문이나 switch 문의 실행을 중단하고자 할 때 사용
- ❖ 반복문의 사용 예

```
int i = 0;

while ( i >= 0 )
{
    if ( i == 10 )
        break;

    Console.WriteLine( i++ );
}

Console.WriteLine("Prison Break");
```

i를 초기화하는 코드가 실행되고

루프가 실행되다가

i가 10이 되면 while 문에서 탈출합니다.

프로그램의 실행 위치는 while 블록
다음으로 옮겨집니다.

- ❖ 데모 예제 - Break



5.3.2 continue

❖ 한 회 건너 뛰어 반복을 계속 수행하게 하는 기능

❖ 사용 예

```
for ( int i=0; i<5; i++ )  
{  
    if ( i != 3 )  
    {  
        Console.WriteLine( i );  
    }  
}
```

❖ 데모 예제 - Continue



5.3.3 goto

- ❖ 레이블(표지판)이 가리키는 곳으로 바로 건너 뛰는 구문
- ❖ 실행 흐름을 복잡하게 만들어 가독성이 좋지 않다.
- ❖ 사용 형식

```
{  
    Console.WriteLine( " 1 " );  
  
    goto JUMP;  
  
    Console.WriteLine( " 2 " );  
    Console.WriteLine( " 3 " );  
  
    JUMP:  
    Console.WriteLine( " 4 " );  
}
```



Thank You !

이것이 C#이다

