**05. 코드의 흐름 제어하기**

수업목표:

-분기문

-반복문

-점프문

5.1 분기문(Branching Statement)

-if문

-switch문

5.1.1 if,else 그리고 else if

-한번에 단 하나의 조건 평가

-기본 형식: if(조건)

실행할 문장

5.1.2 if문 중첩해서 사용하기

-분기문이나 반복문은 중첩 사용가능

5.1.3~4 switch

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

-사용형식

조건식 - 정수형식과 문자열 형식

-C# 7.0의 변화 – 조건식에 데이터 형식사용가능

-----------------------------------

Object obj = 123;

Switch(obj):

{ Case int i:

…

Break;

Case float f:

…

Break;

Default:

…

Break;

}

----------------------------

5.2 반복문(Loop Statement)

-루프문(반복함)

While, do while, for, foreach

5.2.1 while

조건식 – 논리형식

Int a = 10;

While(a>0) // 괄호안이 참일 때 반복함. A가 0보다 작으면 종료

5.2.2 do~while

-무조건 한 번의 실행을 보장하는 반복문 ( 무조건 한번은 실행한다)

----------------

do

{ 반복할 코드}

While (조건식);

------------------

5.2.3 for

-while보다 더 정교

----------------

For( 초기화식 ; 조건식 ; 반복식)

반복 실행할 코드

------------------

5.2.4 중첩 for

-반복문을 중첩할 경우 for문이 가장 적합함

5.2.5 foreach

-배열이나 컬렉션에 주로 사용

Foreach(데이터형식 변수명 in 배열or컬렉션)

코드

int[] arr = new int[] { 0, 1, 2, 3, 4 };

foreach(int a in arr)

{

Console.WriteLine(a);

}

---------

5.2.6 for or while 무한반복

For( ; ; )

While( true )

5.3.1 break

-반복문, switch문의 실행 중단

5.3.2 continue

-한 회 **건너 뛰어서** 반복을 계속 수행하게 하는 기능

5.3.3 goto

-레이블(표지판)이 가리키는 곳으로 바로 건너 뛰는 구문

-실행 흐름을 복잡하게 만들어 가독성이 (엄청) 좋지 않음

Goto JUMP;

~~

JUMP:

~~