4. 데이터를 가공하는 연산자

4.1 C#에서 제공하는 연산자

\*연산자들 중 대부분은 각각 특정 형식에 대해 사용가능

-산술연산자, 증감연산자, 관계연산자, 조건연산자, Null조건부 연산자, 논리연산자, 비트연산자, 할당연산자 ,

4.2 산술연산자

- 수치 형식의 데이터를 다루는 연산자: 정수형식과 부동소수점 형식, decimal형식

+,-,\*,/(몫),%(나머지)

4.3 증가연산자와 감소연산자

- 증가연산자 ++, 감소연산자 –

-연산자의 위치에 따라

\*전위증감연산자(변수의 값을 변경한 후 해당 구문 실행)

\*후위증감연산자(해당 구문을 실행한 후 변수의 값 변경)

Int a=10;

Console.WriteLine(a++) //11이 아닌 10을 출력후 a는 11로 증가

Console.WriteLine(++a) //11을 출력

4.4 문자열 결합 연산자

\* +연산자

Int result = 123+456; //산술연산자

String result = “123” + “456”; // 문자열 결합 연산자

4.5 논리연산자

4.6

4.7 조건 연산자

\*조건 연산자의 형식 타입 변수명 = a == 10 ? “참이면” : ”거짓이면”

4.8 널 조건부 연산자

\* C# 6.0에서 도입함

\*객체의 멤버에 접근하기전에 널인지 검사

- 객체?. 반환멤버

객체?[배열(컬렉션)의 인덱스]

널조건부 연산자가 없으면 == 이용

Class Foo

{ public int member; }

Foo foo = null;

Int? bar;

If(foo == null)

Bar = null;

Else

Bar = foo.member;

?. 널 조건부 연산자 있을 때

Class Foo

{ public int member; }

Foo foo = null;

Int?bar;

Bar = foo?.member; // foo객체가 null이면 null을 반환 null이 아니면 member필드에 접근하게 해줌

널이 아니면 실행.

4.9 비트 연산자

- 데이터 형식의 크기는 주로 바이트 단위 사용

-더 작은 단위로 데이터를 가공해야 할 경우

- 1바이트 = 8비트

- 비트 연산자의 종류

<< : 왼쪽 시프트 연산자

>> : 오른쪽 시프트 연산자

& : 논리곱(AND)연산자

| : 논리합(OR)연산자

^ : 배타적 논리합(XOR)연산자

~ : 보수(NOT)연산자

4.9.1 시프트 연산자

-비트를 왼쪽이나 오른쪽으로 이동

-왼쪽 시프트 연산

4.9.2 비트 논리 연산자

- 각 비트에 대해 수행하는 논리 연산

- Bool형식과 정수 계열 형식

- 각 비트에 대해 1은 True, 0은 False

-논리곱연산자 & : 참 & 참 = 1, 참&거짓 = 0

-논리합 연산자 : 둘중 1개라도 1이면 1

-배타 논리합 연산자: ^ = 둘중 1개만 1일 때 1

- 보수 연산자 ~ : 1과 0 이 반대

int a = 9;

int b = 10;

Console.WriteLine($"{a} & {b} :{ a & b }");

Console.WriteLine($"{a} | {b} :{ a | b }");

Console.WriteLine($"{a} ^ {b} :{ a ^ b }");

int c = 255;

Console.WriteLine("~{0}(0x{0:X8}) :{1}(0x{1:X8})",c,~c);

4.10 할당 연산자

-변수 또는 상수에 피연산자 데이터를할당하는 기능

-할당연산자의 종류

=, +=, -+, \*=, /=, %=, &=, |=, ^=, >>=, <<=

4.11 Null 병합 연산자 ??

-필요한 변수/객체의 null 검사를 간결하게 만들어주는 역할

- 형식 : OP1 ?? OP2

- OP1이 Null인지 여부에 따라

int? num = null;

Console.WriteLine($"{ num ?? 0}");

num = 99;

Console.WriteLine($"{ num ?? 0}");

string str = null;

Console.WriteLine($"{str ?? "Default"}");

str = "Specific";

Console.WriteLine($"{str ?? "Default"}");

4.12 연산자 우선순위