**3.4 Type Conversion (데이터 형식 변환)**

* **크기(표현 범위)가 서로 다른 정수 형식 사이의 변환**

작은->큰 (O)

큰->작은 (X, overflow가능성)

|  |  |  |
| --- | --- | --- |
| 형식 | 크기 | 담을 수 있는 값의 범위 |
| sbyte | 1바이트 | -128~127 |
| byte | 1바이트 | 0~255 (2^8) |
| short | 2바이트 | -32,768~32,767 |
| ushort | 2바이트 | 0~65,535 |
| int | 4바이트 | -2,147,483,648~2,147,483,647 |
| uint | 4바이트 | 0~4,294,967,295 |
| long | 8바이트 | -922,337,203,685,477,508~922,337,203,685,477,507 |
| ulong | 8바이트 | 0~18,446,744,073,709,551,615 |
| char | 2바이트 | 문자(유니코드) |

[예제]

sbyte a = 127;

int b = a; //작은 -> 큰

int x = 128;

sbyte y = (sbyte) x; //overflow -128

* **크기(표현 범위)가 서로 다른 부동 소수점 형식 사이의 변환**

No overflow but not accurate

정밀성에 손상: 형식 변화시, 10진수로 복원 -> 이진수로 변환

이진수로 표현하는 소수는 완전하지 않음

|  |  |  |
| --- | --- | --- |
| 형식 | 크기 | 담을 수 있는 값의 범위 |
| float | 4바이트 | -3.402823e38~3.402823e38 |
| double | 8바이트 | -1.79769313486232e308~1.79769313486232e308 |

[예제]

float a = 69.6875f;

double b = a;

float x = 0.1f;

double y = (double) x; // y = 0.100000001490116 (0.1 is a non-terminating value in binary)

* **부호 있는(Signed) 정수 형식과 부호 없는(Unsigned) 정수 형식 사이의 변환**

|  |  |  |
| --- | --- | --- |
| int | 4바이트 | -2,147,483,648~2,147,483,647 |
| uint | 4바이트 | 0~4,294,967,295 |

[예제]

int a = 500;

uint b = (uint) a;

int x = -30;

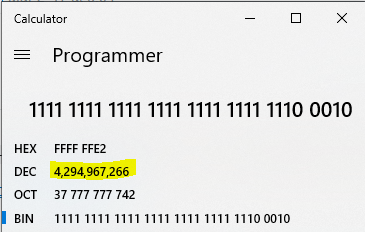
uint y = (uint) x; // **underflow**: y = 4294967266

Two’s complement (flip bit + 1)

30 dec = 0000 0000 0000 0000 0000 0000 0001 1110 bin

flip: 1111 1111 1111 1111 1111 1111 1110 0001

+ 1 = 1111 1111 1111 1111 1111 1111 1110 0010



* **부동 소수점 형식과 정수 형식 사이의 변환 (float -> Integer)**

소수점 이하 버림.

0.9 => 0

1.1 => 1

\*\*데이터 손실, 정밀도 떨어짐

* **문자열과 숫자 사이의 변환 (“12345” -> 12345)**

string -> num: Parse(), Convert.ToInt32 (“12345”) //12345

ToInt16, 32, 64

num -> string: .ToString()

**3.5 const & enum (상수와 열거 형식)**

* **Assign constant선언**

const int MAX\_INT = 2157483647;

* **enum**

같은 범주에 속하는 여러 개의 상수를 선언할 때

[예제]

enum VisaType { VISITOR, STUDENT, CLOSEDWORK, OPENWORK, PGWP }

0 1 2 3 4 (기본형 int)

enum VisaType { **VISITOR=10**, STUDENT, CLOSEDWO RK, **OPENWORK=30**, PGWP }

10 11 12 30 31

**3.6 Nullable**

* **Nullable선언**

datatype**?** var\_name;

int? num = null;

* **HasValue (bool, 변수가 값을 갖고 있는지 없는지), Value(변수의 값)**

|  |  |
| --- | --- |
| int**?** num = null;  num.**HasValue** // False  num.**Value** // null | num = 10;  num.**HasValue** // True  num.**Value** // 10 |

**3.7 var 형식 (Weakly typed)**

* **선언과 동시 초기화 assign + initialize**
* **지역 변수로만 local variable (let처럼)**
* **.GetType()**

var a = 20;

var b = 3.14;

var c = “Hyunju”;

var d = new int[] { 1, 2, 3};

int num = 10;

num.GetType() // System.Int32

a.GetType() // System.Int32

b.GetType() // System.Double

c.GetType() // System.String

d.GetType() // System.Int32[]

**3.8 Common Type System (CTS) 공용 형식 시스템 - GetType()**

.NET Framework의 형식 체계의 표준 준수: .NET 언어들 간의 호환성 (C, C++, C#, VB..)

System.Int32

System.String

.NET아기텍처에서 제공하는 CTS의 데이터형식에 기반한 것