제3장
C# 프로그래밍에 익숙해지기

변 영 철

1. Console2 프로젝트 (C++)

```
#include <stdio.h>

class XXX
{
private:
    int iX;
    int iY;

public:
    void Assign(int x, int y) {
        iX = x;
        iY = y;
    }

    int Add() {
        return iX + iY;
    }
};

XXX gildong;

void main()
{
    gildong.Assign(2, 3);

    int iResult = gildong.Add();

    printf("두 개의 값을 더한 결과 : %d\n", iResult);
}
```

2. C++ 코드를 C# 코드로

1) C# 프로젝트 만든 후 C++ 코드 복사
2) #include 문 제거
3) 멤버 별로 접근 지정자 각각 선언
   - 귀찮을까?
4) 클래스 선언 끝의 세미콜론(;) 제거
5) gildong 객체를 main 함수 지역 변수로 옮기기
6) main 함수를 **클래스 안으로** 옮기기
   - 이름 바꾸기: main → Main
   - 정적(static) 멤버 함수로 선언하기
   - 정적 멤버 함수는? (닷넷 프레임워크에서 호출)
7) 객체생성 코드 수정 (레퍼런스란?)
8) 문자열 출력 함수 수정

3. Main 멤버 함수 분리, 왜?

- 추상화와 코드 분리
  - 서로 관련된 코드끼리만 클래스로 묶음.
  - 관련이 있는지 없는지는 멤버 변수를 중심으로 판단
  - Main 함수는 어떤 프로그램을 작성하더라도 항상 작성해야 하는 필수 함수로서, 나머지 코드와 직접적인 관련이 없음.
  - 따라서 Main 함수를 다른 클래스로 분리

4. C# 코딩 익숙해지기

- 멤버는 멤버를 액세스할 수 있다.
- 아마 멤버로 정의되어 있을 거야.
- 멤버로 정의하라.

4. C# 코딩 익숙해지기

- Main 함수가 클래스 안에 들어간 이유는?
  - 전역 함수를 클래스 멤버 함수로 만들어서 모든 함수가 클래스 멤버 함수가 되도록
  - 진정한 객체지향 프로그램
- Main 함수가 static(정적) 멤버 함수이어야 하는 이유는?
  - 객체를 만들지 않아도 닷넷 프레임워크가 호출할 수 있도록

5. 네임 스페이스(name space)

- 네가 만든 클래스와 내가 만든 클래스 명이 같으면(클래스 이름 충돌)
  - 네가 클래스 이름을 바꿔라?
- 해결방법은?
  - 네임 스페이스
  - 따라서, 네임 스페이스를 사용하는 이유는?

    [답] 이름 충돌 방지를 위하여...
- using 문
  - using 문을 쓰는 이유는? 반복적으로 네임 스페이스를 붙이는 것이 귀찮아서

6. 가장 간단한 C# 프로그램

- My2 C# 프로젝트