4장 OOP에 익숙해지기

변영철 교수
(ycb@jejunu.ac.kr)

$git pull

1. 인라인 함수와 아웃오브라인 함수

- 인라인(in-line) 함수
  - 클래스 내부에 작성된 멤버 함수
  - 코드가 길어질 경우 복잡하게 보일 수 있음
- 아웃오브라인(out-of-line) 함수
  - 클래스 밖으로 옮긴 멤버 함수
  - 클래스 코드가 작아서 한 눈에 볼 수 있음
  - 프로그램의 가독성이 좋아짐

1. 인라인 함수와 아웃오브라인 함수

- private 키워드 사용
  - 없을 때 보단 private 이라는 것을 분명히 명시
  - 가독성
- 멤버 함수
  - m_iX, m_iY
  - 가독성

2. 생성자 함수와 소멸자 함수

- 생성자 함수
  - 객체가 생성될 때 자동으로 실행되는 멤버 함수
- 소멸자 함수
  - 객체가 제거될 때 자동으로 실행되는 멤버 함수
- 객체 생성자 실행 시점
  - main 함수의 지역 변수(객체)일 경우 : main 함수가 실행될 때 객체 생성 및 생성자 실행
  - 전역 객체의 경우 main 함수 실행 이전
- 객체 소멸자 실행 시점
  - 지역 객체의 경우 함수가 실행될 때, 전역 객체의 경우 프로그램이 종료될 때

3. 멤버는 멤버를 액세스할 수 있다

- C 프로그램의 경우
  - 전역 함수는 전역 변수 혹은 함수를 액세스할 수 있음
- C++의 경우
  - 전역 함수와 전역 변수를 묶은 것 -> 클래스
  - 묶은 결과 전역 함수와 전역 변수 -> 멤버 함수와 멤버 변수
  - 따라서 멤버 함수는 멤버 변수 혹은 함수를 액세스 할 수 있음

```
int CPoint::Add()
{
    return m_iX + m_iY;
}
```

4. 아마 멤버로 정의되어 있을 꺼야

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
	if (CFrameWnd::OnCreate(lpCreateStruct) == -1)
		return -1;

	if (!m_wndToolBar.CreateEx(this, TBSTYLE_FLAT, WS_CHILD |  WS_VISIBLE | CBRS_TOP
		| CBRS_GRIPPER | CBRS_TOOLTIPS | CBRS_FLYBY |  CBRS_SIZE_DYNAMIC) ||
		!m_wndToolBar.LoadToolBar(IDR_MAINFRAME)) {
		TRACE0("도구 모음을 만들지 못했습니다.₩n");
		return -1;      // 만들지 못했습니다.
	}

	if (!m_wndStatusBar.Create(this) ||
		!m_wndStatusBar.SetIndicators(indicators,
		  sizeof(indicators)/sizeof(UINT))) {
		TRACE0("상태 표시줄을 만들지 못했습니다.₩n");
		return -1;      // 만들지 못했습니다.
	}
	// TODO: 도구 모음을 도킹할 수 없게 하려면 이 세 줄을 삭제하십시오.
	m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
	EnableDocking(CBRS_ALIGN_ANY);
	DockControlBar(&m_wndToolBar);

	return 0;
}
```

4. 아마 멤버로 정의되어 있을 꺼야

- EnableDocking 함수, DockControlBar 함수를 보면 무슨 생각?
- m_wndToolBar 멤버 변수
- 아무 생각없이 코드를 보면 객체지향 프로그래밍에 익숙해질 수 없다.
- '코드가 복잡하게 생겼구나'가 아닌 '아마 CMainFrame 클래스의 멤버로 이런 것들이 정의되어 있을 꺼야' 라는 생각

5. 멤버로 정의하라.

```
int CPoint::Add()
{
    iSum = m_iX + m_iY;
    return iSum;
}

int CPoint::Add()
{
        Assign(7, 8);

        iSum = m_iX + m_iY;
        PrintResult();

        return iSum;
}
```

습관적으로 생각하기

(1) 전역변수는 아무데서나 막 접근(액세스)할 수 있다(C언어).

(2) 클래스는 객체 만들라고 있는 것

(3) 멤버(함수)는 멤버(변수나 함수)를 액세스할 수 있다.

(4) 멤버로 정의되어 있겠지.

(5) 멤버로 정의하라.

(6) 클래스를 보지 말고 객체를 보라. 항상 객체 위주(객체지향)로 생각하라.

(7) 객체가 언제 어디에서 만들어지는지를 먼저 찾아보라.