

CPP Base

CHAPTER 3

**클래스와 객체

*객체의 캡슐화

- 객체는 캡슐화를 통해 외부의 접근을 통제하여 자신의 내부 요소들을 보호함
- 하지만 모든 요소를 외부의 접근으로부터 완전히 차단하면 정보 교환과 통신에 제한이 되기 때문에 일부 요소는 노출시킨다.

*C++ 객체

- C++ 객체는 멤버 변수와 멤버 함수로 구성이 된다.
- 멤버 변수는 개체의 상태를 나타내며, 멤버 함수는 행동을 구현한 코드다.

*C++ 클래스

- 클래스 선언 및 구현 예시 (1.cpp 코드 참조)
- 객체가 생성되면 클래스 크기의 메모리가 할당 된다.
- 객체 멤버 접근을 위해서는 이름뒤에 .을 찍고 멤버를 쓴다. (객체이름.멤버)

*생성자

- 생성자는 특별한 멤버 함수를 통해 객체를 초기화 한다.
- 여러개의 생성자를 작성할 수 있으나. 하나만 실행 된다.
- 여러 생성자 예시 (2.cpp 코드 참조)
- 위임 생성자 예시 (3.cpp 코드 참조)
- 멤버 변수 초기화
 - 1)생성자 코드에서는 멤버 변수 초기화가 가능
 - 2) 생성자 구현부에서 멤버 변수와 초기값을 지정 Point::Point(int a, int b) : x(a), y(a) 형태
 - 3) C++11부터 선언문에서 직접 초기화도 가능

-기본 생성자란 어떠한 생성자도 없을 때 컴파일러가 자동으로 생성해주는 생성자다.

*소멸자

- 소멸자는 객체가 사라질 때 할당 받은 메모리를 운영체제에 돌려주거나 열어 놓은 파일을 저장하고 닫거나, 연결된 네트워크를 해제하는등 객체가 사라지기전 필요한 조치를 하도록 한다.
- 소멸자는 클래스 이름 앞에 ~를 붙이며 어떠한 리턴값도 존재하지 않는다
- 소멸자는 오직 한 개만 존재하며 매개변수가 없고, 생성된 순서와 반대순으로 소멸한다.
- 소멸자가 선언되어있지 않으면 기본 소멸자가 자동 생성된다.
- 소멸자 예시 (4.cpp 코드 참조.)

*접근 지정자

- private 멤버 : 클래스 내의 멤버 함수들에게만 접근이 허용
- public 멤버 : 클래스 내외를 막론하고 프로그램의 모든 함수들에게 접근 허용
- protected 멤버 : 해당 클래스를 상속 받은 파생 클래스의 멤버 함수에게만 접근이 허용 됨.
- 디폴트 접근 지정은 private
- 멤버 변수는 private로 지정하는게 바람직하다.
- 생성자는 public으로 해야 클래스 외부에서 객체 생성이 가능하다. 그렇지 않으면 컴파일 오류 발생.

*인라인 함수

- 함수 호출 오버헤드로 인한 프로그램의 실행 속도 저하를 막기 위해 도입됨.
- 장점 : inline 키워드의 경우 멤버 변수의 값을 알아내거나 멤버 변수에 값을 쓰는 getter나 setter와 같이 작은 함수들을 인라인 함수로 선언하면 C++프로그램의 실행 속도를 향상시킬 수 있다.
- 단점 : 인라인 함수를 호출하는 곳에 인라인 함수의 코드를 단순 삽입하므로, 호출하는 곳이 여러군데 있으면 그만큼 전체 크기가 늘어나게 된다.
- 제약 사항
 - 1)inline 선언은 요청일뿐 강제사항이 아니기 때문에 컴파일러는 함수의 크기나 효율을 따져서 불필요하면 inline선언을 무시한다.
 - 2)컴파일러에 따라 재귀함수, static 변수, 반복문, switch문, goto문 등을 가진 함수는 인라인 함수로 허용하지 않는다.

-생성자를 포함하여 클래스의 모든 멤버 함수가 인라인으로 선언될 수 있다.

-멤버 함수의 크기가 작으면 클래스의 선언부에 직접 구현해도 되는데 컴파일러는 클래스의 선언부에 구현된 멤버 함수들에 대해서는 inline이 없어도 인라인 함수도 자동 처리한다.

*C++ 구조체 선언

- C++은 C언어와의 호환성을 위해 구조체를 지원한다. 또한 C++ 구조체는 표준 C구조체의 기능을 확장하여 클래스와 동일한 구조를 지닌다.
- 구조체의 디폴트 지정 접근은 클래스와 다르게 Public이다.
- C++ 구조체의 객체 생성은 클래스의 객체 선언 방식과 같이 구조체 타입 뒤에 객체 이름을 지정하면 된다.

ex)

```
struct StructName{;  
StructName stObj; (0)  
struct StructName stObj; (X)
```

*바람직한 C++ 프로그램 작성법

- C++ 프로그램의 소스코드는 다음과 같은 원칙하에 분리하여 작성한다
 - 1) 클래스마다 선언부는 헤더파일에 구현부는 cpp파일에 분리하여 작성한다.
 - 2) main() 등 함수나 전역 변수는 한 개 이상의 cpp 파일에 나누어 작성한다.