

c++ 5일차

C++는 객체 지향 프로그래밍 언어로서, 구조체(structure), 클래스(class), 그리고 객체(object) 개념을 중심으로 프로그램을 구성합니다. 이 세 가지 개념은 데이터와 그 데이터를 조작하는 함수를 묶어 관리하는 데 중요한 역할을 합니다. 이해를 돕기 위해 각각의 개념에 대해 자세히 설명하겠습니다. 구조체 (Structure) 구조체는 서로 다른 타입을 가진 변수들을 하나의 단위로 묶어 새로운 타입을 정의하는 방법입니다. C++에서 구조체는 단순히 데이터의 집합으로 사용될 수 있으며, 기본적으로 모든 멤버가 public입니다. 이는 구조체의 멤버에 외부에서 직접 접근할 수 있음을 의미합니다.

```
struct Person {  
    std::string name;  
    int age;  
};
```

위 예시에서 Person이라는 구조체는 name과 age라는 두 멤버 변수를 가지고 있습니다. 구조체를 사용하면 관련 데이터를 논리적으로 묶어서 관리할 수 있습니다. 클래스 (Class) 클래스는 구조체와 비슷하게 데이터 멤버와 함수 멤버(메서드)를 포함할 수 있는데, 가장 큰 차이점은 클래스의 멤버는 기본적으로 private입니다. 즉, 클래스 외부에서는 직접 접근할 수 없으며, 멤버 함수를 통해서만 접근할 수 있습니다. 이는 데이터 은닉과 캡슐화를 가능하게 하여, 객체의 상태와 행동을 보호하고 관리할 수 있게 합니다.

```
class Animal {  
private:  
    std::string name;  
    int age;  
  
public:  
    void setName(std::string n) { name = n; }  
    std::string getName() { return name; }  
    void setAge(int a) { age = a; }  
    int getAge() { return age; }  
};
```

위의 Animal 클래스는 name과 age라는 두 개의 private 멤버 변수와, 이 변수들의 값을 설정하고 읽을 수 있는 public 멤버 함수를 가집니다. 객체 (Object) 객체는 클래스나 구조체로부터 생성된 인스턴스입니다. 클래스나 구조체를 정의하는 것은 마치 새로운 타입을 정의하는 것과 같으며, 이 타입으로부터 객체를 생성하여 사용합니다. 객체를 생성하면 클래스나 구조체에 정의된 데이터 멤버와 함수 멤버를 사용할 수 있습니다.

```
Animal myDog; // Animal 클래스의 객체인 myDog 생성
myDog.setName("Buddy");
myDog.setAge(5);
std::cout << myDog.getName() << " is " << myDog.getAge() << " years old."
<< std::endl;
```

객체 지향 프로그래밍에서는 클래스를 사용하여 객체를 정의하고, 이 객체들을 통해 프로그램의 기능을 구현합니다. 클래스와 객체를 사용함으로써 코드의 재사용성, 확장성 및 유지보수성이 향상됩니다. C++에서 구조체와 클래스는 비슷한 역할을 하지만, 클래스가 제공하는 보다 풍부한 기능(예: 접근 제어, 상속 등)으로 인해 객체 지향 프로그래밍에 있어서 핵심적인 요소로 작용합니다.