

Day-18

Entities, Declaration and Definition

Entities

- C++ 프로그램은 언어별 키워드(for, if, new, true 등), 식별자(변수, 함수, 구조 네임스페이스 등의 기호), 연산자 시퀀스로 정의된 표현식, 리터럴(상수 값 토큰)로 구성된 집합
- 엔티티는 값, 객체, 참조, 함수, 열거형, 타입, 클래스 멤버, 템플릿 또는 네임스페이스임
 - 식별자 및 사용자 정의 연산자는 엔티티를 참조하는데 사용되는 이름
 - 엔티티는 표현식의 결과도 캡처함
 - 전처리기 매크로는 C++ 엔티티가 아님

Declaration and Definition(선언, 정의)

- Declaration/Prototype(선언, 프로토타입)
 - 선언(또는 프로토타입)은 유형과 속성을 설명하는 식별자가 있는 엔티티를 소개함
 - 선언은 컴파일러와 링커가 해당 식별자에 대한 참조(사용)를 받아들이는데 필요한 것
 - 엔티티는 여러번 선언할 수 있음
 - 모든 선언은 동일함
- Definition/Implementation(정의/구현)
 - 엔티티 정의는 선언을 구현하는 것
 - 엔티티의 속성과 동작을 정의함

Declaration/Definition(선언, 정의) 함수 예제

```

void f(int a, char* b); // 함수 선언

void f(int a, char* ){ // 함수 정의(구현)
    // 코드들          // "b"는 사용하지 않는 경우 생략 가능함
}
void f(int a, char* b); // 함수 선언
// 선언을 여러개

f(3, "abc");           // 사용법

```

Declaration/Definition(선언, 정의) struct (구조체) 예제

구체적인 구현이 없는 선언은 불완전한 유형임(무효)

```

struct A; // 구조체A 선언
struct A; // 구조체A 선언2, 선언은 여러개 가능
struct B { // 구조체B 선언 및 정의
    int b;
// A x; // compile error, 불완전한 유형
    A* y; // ok, 불완전한 유형에 대한 포인터
};
struct A { // 구조체 정의(구현)
    char c;
}

```