Lab Study Report

배홍섭

목차

- DLL과 SLL의 차이점
- Private 과 public의 차이점 (+protected)
- UE5에서 C++ 빌드 과정
- rigging with landmarks (mediapipe)

DLL과 SLL의 차이점

DLL과 SLL의 차이점

DLL (Dynamic Link Library)

- 프로그램 실행 시 필요시만 외부 DLL 파일에서 함수를 참조
- 프로그램 실행 시 프로그램 로딩 시간이 단축
- 함수 업그레이드 시 해당 DLL만 수정 배포한다
- 소스 외부 유출 방지 효과
- 실행 파일 만들때 필요한 파일: *.h,*.lib (*.dll 참조 용)
- 프로그램 실행할 할 때 필요한 파일: *.dll (배포할 때 *.dll 필요)
- dll 제작 시 lib도 같이 생성됨

SLL (Static Link Library)

- 필요한 함수를 프로그램 코드에 붙여 프로그램 자체에서 참조
- 프로그램 실행 후 빠른 처리시간
- 프로그램 실행 파일만 있으면 실행(하나의 파일만 있으면 됨)
- 소스 외부 유출 방지 효과
- 실행 파일 만들때 필요한 파일: *.h,*.lib (별도의 *.dll 필요 없음)

참조 : https://it4all.tistory.com/16

실제 실행 예제

private 과 public의 차이점 (+protected)

private 과 public의 차이점 (+protected)

부모 클래스의 속성	상속 속성	상속 시 속성
private	private	private
private	protected	private
private	**public**	**private**
protected	private	private
protected	protected	protected
protected	**public**	**protected**
public	private	private
public	protected	protected
public	**public**	**public**

Base 클래스속성	클래스내에서 접근	객체에서 접근	상속받은 파생 클래스내에서 접근	상속받은 객체에서 접근
private	가능	불가능	불가능	불가능
protected	가능	불가능	가능	불가능
public	가능	가능	가능	가능

private < protected < public

private란 상속도 불가능. Only 현재 클래스에서만 접근 가능 ex) 아빠가 사용한 속옷
protected 란 상속을 했을 때만 사용 가능. ex) 아빠가 물려준 차 but 친구는 아빠한테 상속받지 않았으니 쓸 수 없음.
public이란 상속과 관계없이 언제든지 사용 가능 ex) 아버지가 가진 책. 아들이 빌릴 수도, 친구가 빌릴 수도 있음

실제 실행 예제

```
4   #include <iostream>
5
6   using namespace std;
7
8   class A {
9   public:
10      int num1;
11
12      A() : num1(5), num2(6), num3(7) {}
13  protected:
14      int num2;
15  private:
16      int num3;
17
18  };
19
20
21
22  int main(void) {
23      A a;
24
25      cout << a.num1 << endl; //컴파일 OK!!
26      cout << a.num2 << endl; //컴파일 Error!!
27      cout << a.num3 << endl; //컴파일 Error!!
28
29  }
30
```

0 % ⊗ 2 ⚠ 0

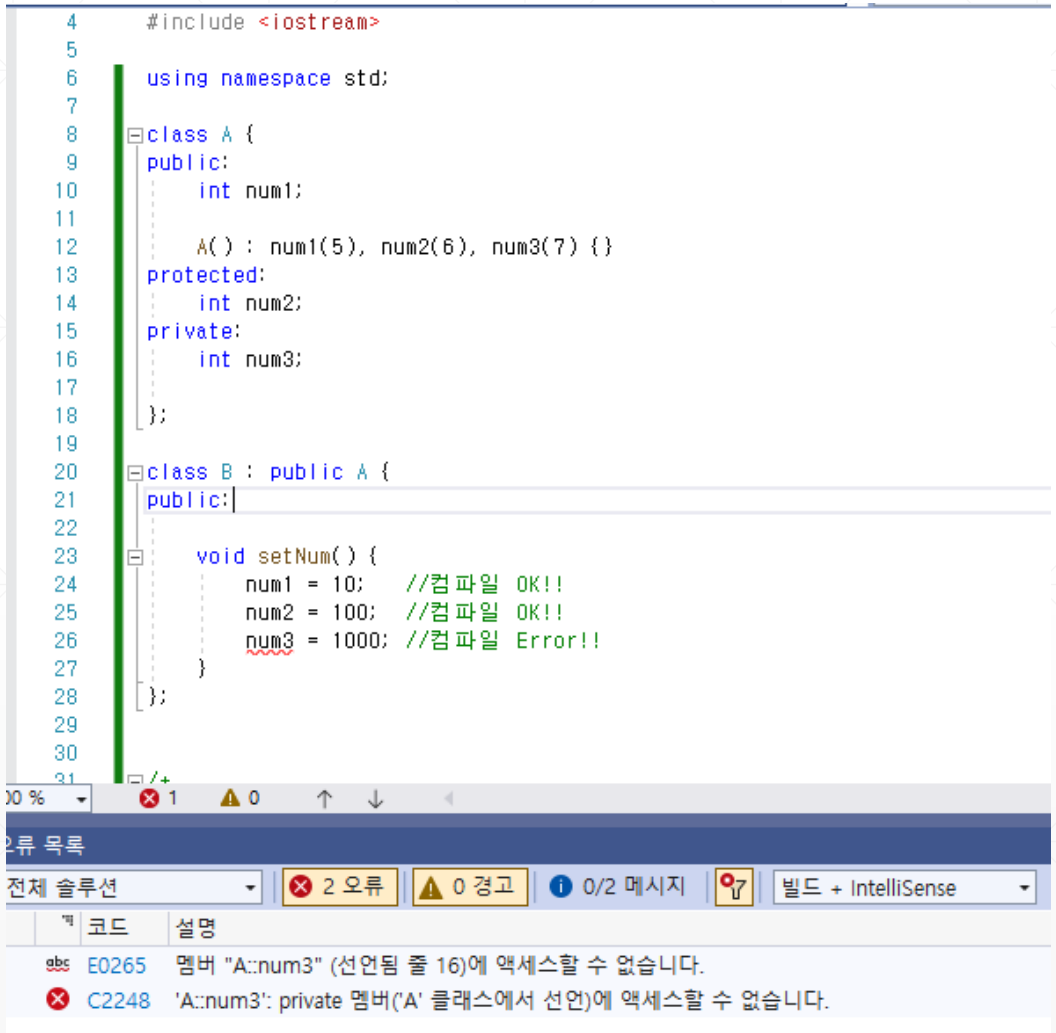
류 목록

전체 솔루션 | ⊗ 4 오류 | ⚠ 0 경고 | ⓘ 0/4 메시지 | 빌드 + IntelliSense

	코드	설명
abc	E0265	멤버 "A::num2" (선언됨 줄 14)에 액세스할 수 없습니다.
abc	E0265	멤버 "A::num3" (선언됨 줄 16)에 액세스할 수 없습니다.
⊗	C2248	'A::num2': protected 멤버('A' 클래스에서 선언)에 액세스할 수 없습니다.
⊗	C2248	'A::num3': private 멤버('A' 클래스에서 선언)에 액세스할 수 없습니다.

protect 와 private 변수에 대해서는 다른 객체에서 접근이 불가능함을 볼 수 있다.

실제 실행 예제



```
4   #include <iostream>
5
6   using namespace std;
7
8   class A {
9   public:
10      int num1;
11
12      A() : num1(5), num2(6), num3(7) {}
13  protected:
14      int num2;
15  private:
16      int num3;
17
18  };
19
20  class B : public A {
21  public:
22
23      void setNum() {
24          num1 = 10;   //컴파일 OK!!
25          num2 = 100;  //컴파일 OK!!
26          num3 = 1000; //컴파일 Error!!
27      }
28  };
29
30
31  /+
```

00 % 1 0

오류 목록

전체 솔루션 | 2 오류 | 0 경고 | 0/2 메시지 | 빌드 + IntelliSense

	코드	설명
abc	E0265	멤버 "A::num3" (선언됨 줄 16)에 액세스할 수 없습니다.
⊗	C2248	'A::num3': private 멤버('A' 클래스에서 선언)에 액세스할 수 없습니다.

방금 예시와는 다르게 protected 변수에 대해서는 상속받을 경우 접근이 가능함을 확인할 수 있다.

public, protected, private 변수를 사용하는 이유

방금 보여드린 예제는 상속 관계일 때 자식 클래스인 B에서 A에 대한 접근을 했을 때의 모습입니다.

이것들을 접근제어 지시자라고 하며 이렇게 접근을 막고 외부로부터 노출을 줄이는 것을 정보은닉이라고 부릅니다.

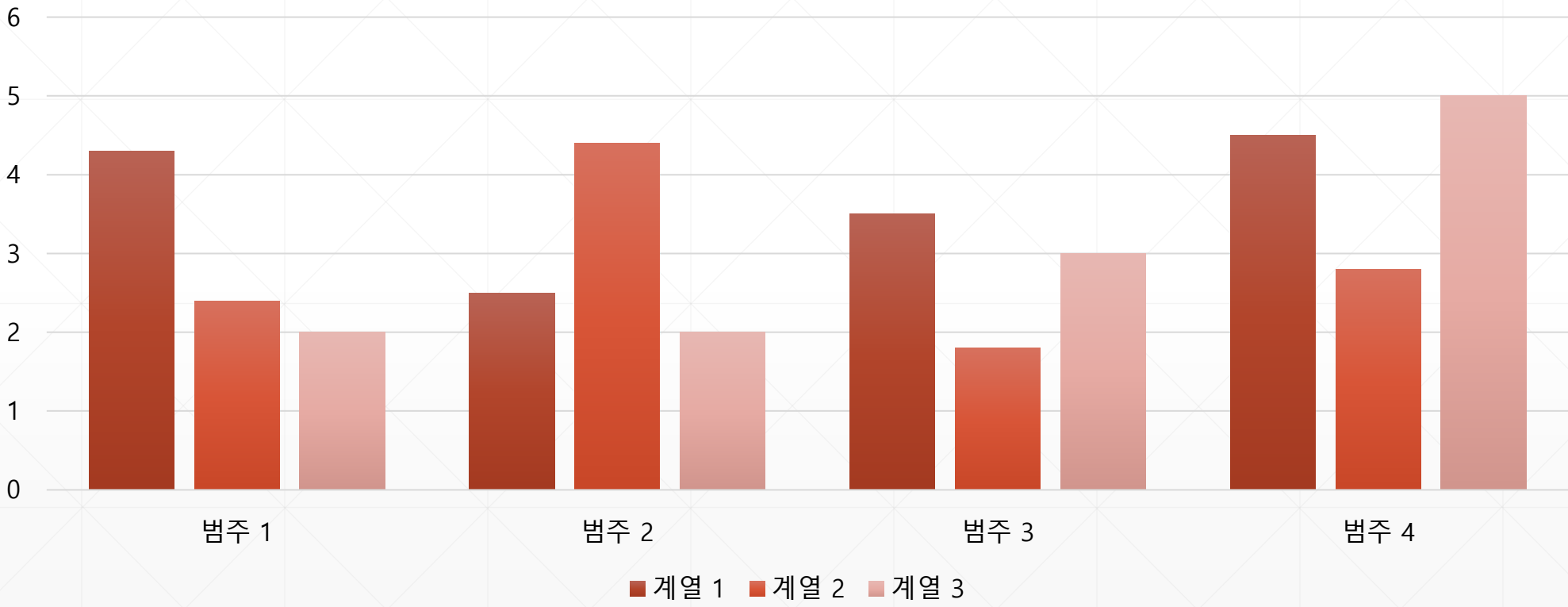
이는 객체지향 언어의 특징인데, 의도치 않은 코드의 수정을 막고 객체와 객체 간의 간섭이 최대한 없도록 하기 위함 입니다.

그리고 여러 class를 정의하고 사용하다 보면 필요한 기능이 있는 함수만을 갖고 사용하는 면이 많아집니다.

따라서 주 기능들만 외부로 들어내고 나머지 기능들은 외부로 노출을 시키지 않는 것입니다.

출처 : https://hwan-shell.tistory.com/25

차트를 사용한 제목 및 내용 레이아웃



표를 사용한 두 개의 내용 레이아웃

- 여기에 첫 번째 글머리 기호
- 여기에 두 번째 글머리 기호
- 여기에 세 번째 글머리 기호

클래스	그룹 1	그룹 2
클래스 1	82	95
클래스 2	76	88
클래스 3	84	90

SmartArt가 있는 제목 및 내용 레이아웃

슬라이드 제목 추가 - 3

슬라이드 제목 추가 - 4

슬라이드 제목 추가 - 5