

언리얼C++ 설계 I - 인터페이스

(Unreal C++ Design I - Interface)

강의 내용

언리얼 C++ 인터페이스의 선언과 활용

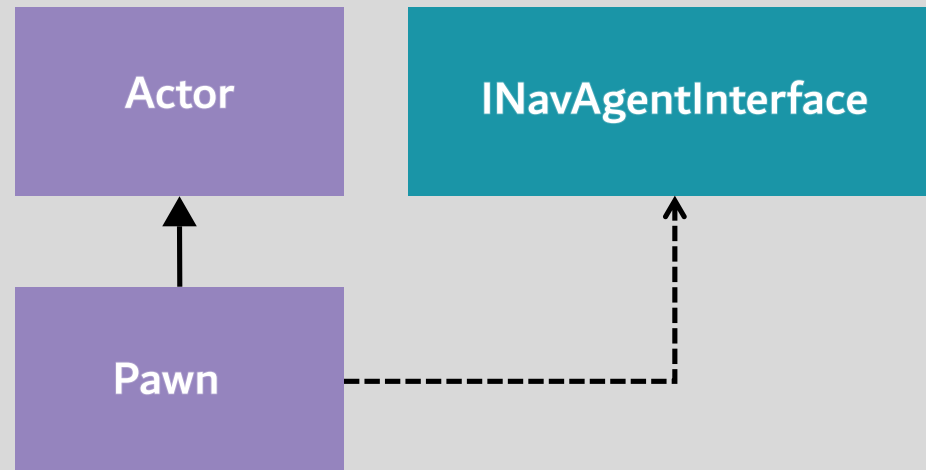
강의 목표

- 언리얼 C++ 인터페이스 클래스를 사용해 보다 안정적으로 클래스를 설계하는 기법의 학습

언리얼 C++ 인터페이스

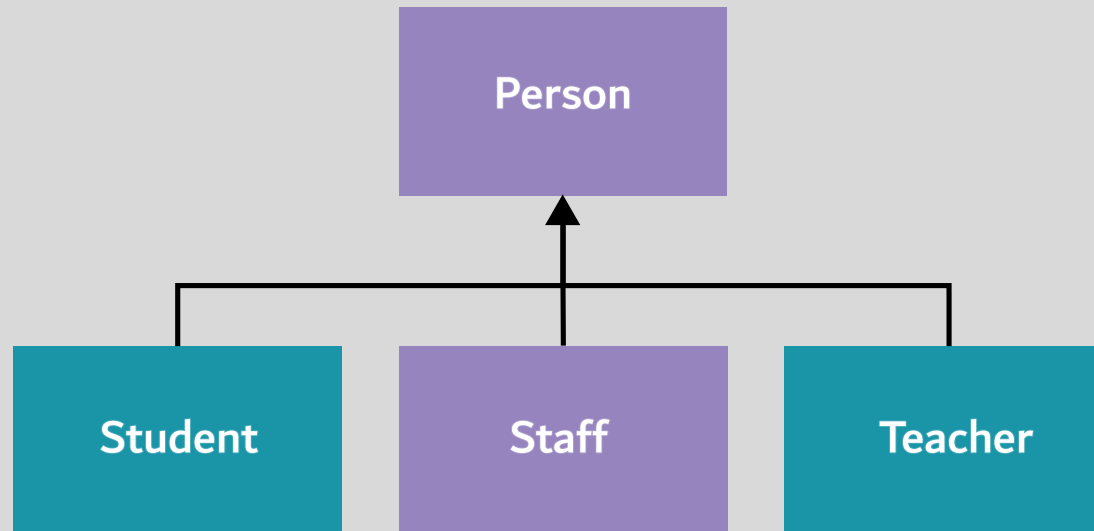
언리얼 C++ 인터페이스

- 인터페이스란?
 - 객체가 **반드시** 구현해야 할 행동을 지정하는데 활용되는 타입
 - 다형성(Polymorphism)의 구현, 의존성이 분리(Decouple)된 설계에 유용하게 활용.
- 언리얼 엔진에서 게임 콘텐츠를 구성하는 오브젝트의 설계 예시
 - 월드에 배치되는 모든 오브젝트. 안 움직이는 오브젝트를 포함 (Actor)
 - 움직이는 오브젝트 (Pawn)
 - 길찾기 시스템을 **반드시** 사용하면서 움직이는 오브젝트 (INavAgentInterface 인터페이스를 구현한 Pawn)



예제를 위한 클래스 다이어그램

- 수업에 참여하는 사람과 참여하지 않는 사람의 구분
 - 수업에 **반드시** 참여해야 하는 학교 구성원 : 학생, 선생
 - 수업에 참여하지 않는 학교 구성원 : 교직원
 - 수업 행동에 관련된 인터페이스 : `ILessonInterface`



언리얼 C++ 인터페이스 특징

- 인터페이스를 생성하면 두 개의 클래스가 생성됨
 - U로 시작하는 타입 클래스
 - I로 시작하는 인터페이스 클래스
- 객체를 설계할 때 I 인터페이스 클래스를 사용
 - U타입 클래스 정보는 런타임에서 인터페이스 구현 여부를 파악하는 용도로 사용됨.
 - 실제로 U타입 클래스에서 작업할 일은 없음.
 - 인터페이스에 관련된 구성 및 구현은 I 인터페이스 클래스에서 진행
- C++ 인터페이스의 특징
 - 추상 타입으로만 선언할 수 있는 Java, C#과 달리 언리얼은 인터페이스에도 구현이 가능함.

UInterface

클래스 타입정보의 제공

IInterface

실질적인 설계 및 구현

정리

언리얼 C++ 인터페이스

1. 클래스가 반드시 구현해야 하는 기능을 지정하는데 사용함.
2. C++은 기본적으로 다중상속을 지원하지만, 언리얼 C++의 인터페이스를 사용해 가급적 축소된 다중상속의 형태로 구현하는 것이 향후 유지보수에 도움된다.
3. 언리얼 C++ 인터페이스는 두 개의 클래스를 생성한다.
4. 언리얼 C++ 인터페이스는 추상 타입으로 강제되지 않고, 내부에 기본 함수를 구현할 수 있다.

언리얼 C++ 인터페이스를 사용하면,
클래스가 수행해야 할 의무를 명시적으로 지정할 수 있어
좋은 객체 설계를 만드는데 도움을 줄 수 있다.