

언리얼 컨테이너 라이브러리 II – 구조체와 Map

(Unreal Container Library II – Struct & Map)

강의 내용

언리얼 구조체의 특징을 이해하고,
다양한 컨테이너 라이브러리에서 구조체를 활용하기

강의 목표

- 언리얼 구조체의 선언과 특징 이해
- 언리얼 대표 컨테이너 라이브러리 TMap의 내부 구조 이해
- 세 컨테이너 라이브러리의 장단점을 파악하고, 알맞게 활용하는 방법의 학습

언리얼 구조체

언리얼 구조체



<https://bit.ly/uestructkr>

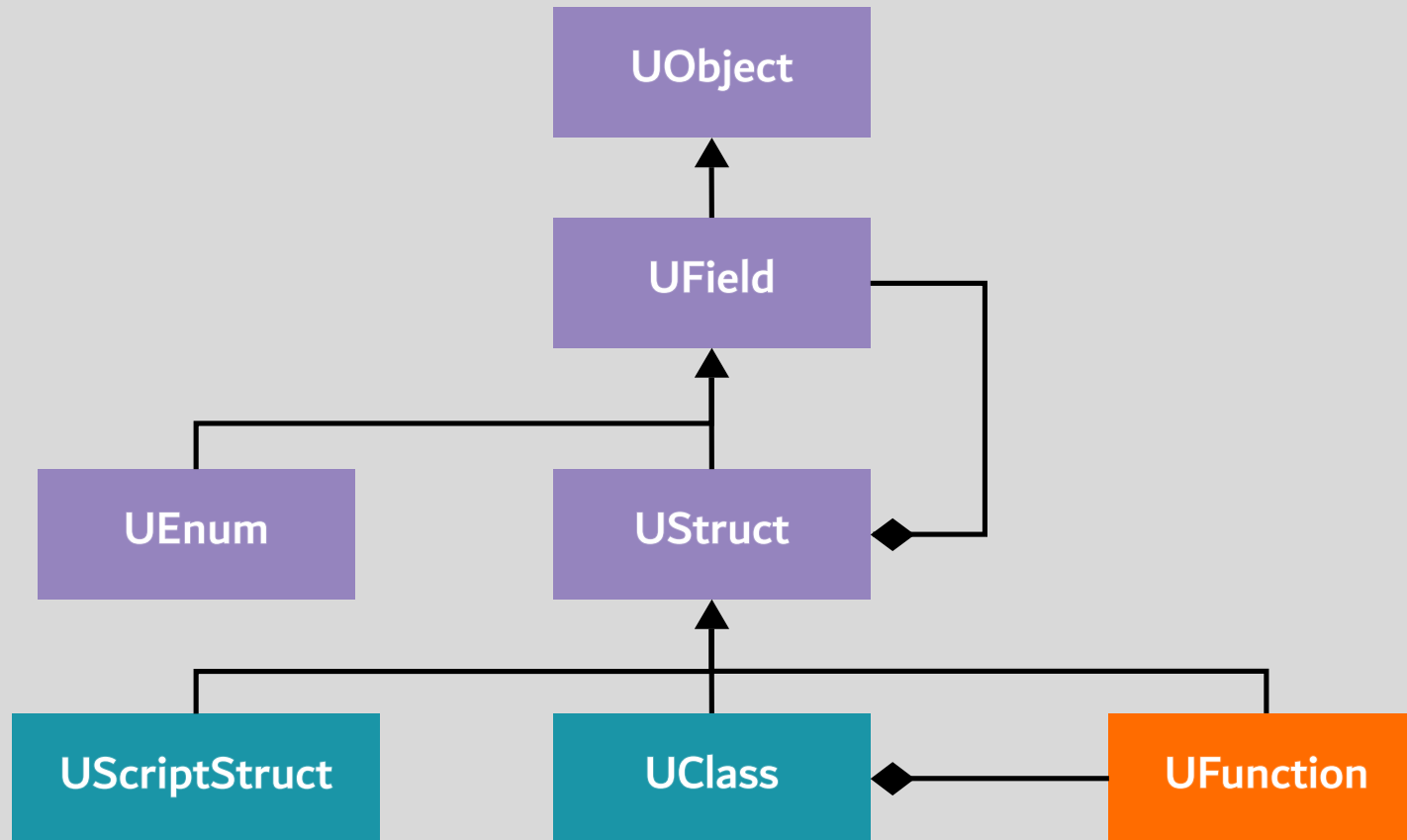
<https://docs.unrealengine.com/5.1/ko/using-structs-in-unreal-cpp/>

언리얼 구조체 UStruct

- 데이터 저장/전송에 특화된 가벼운 객체
- 대부분 GENERATED_BODY 매크로를 선언해준다.
 - 리플렉션, 직렬화와 같은 유용한 기능을 지원함.
 - GENERATED_BODY를 선언한 구조체는 UScriptStruct 클래스로 구현됨.
 - 이 경우 제한적으로 리플렉션을 지원함
 - 속성 UPROPERTY만 선언할 수 있고 함수 UFUNCTION은 선언할 수 없음
- 언리얼 엔진의 구조체 이름은 F로 시작함.
 - 대부분 힙 메모리 할당(포인터 연산) 없이 스택 내 데이터로 사용됨.
 - NewObject API를 사용할 수 없음.

언리얼 리플렉션 관련 계층 구조

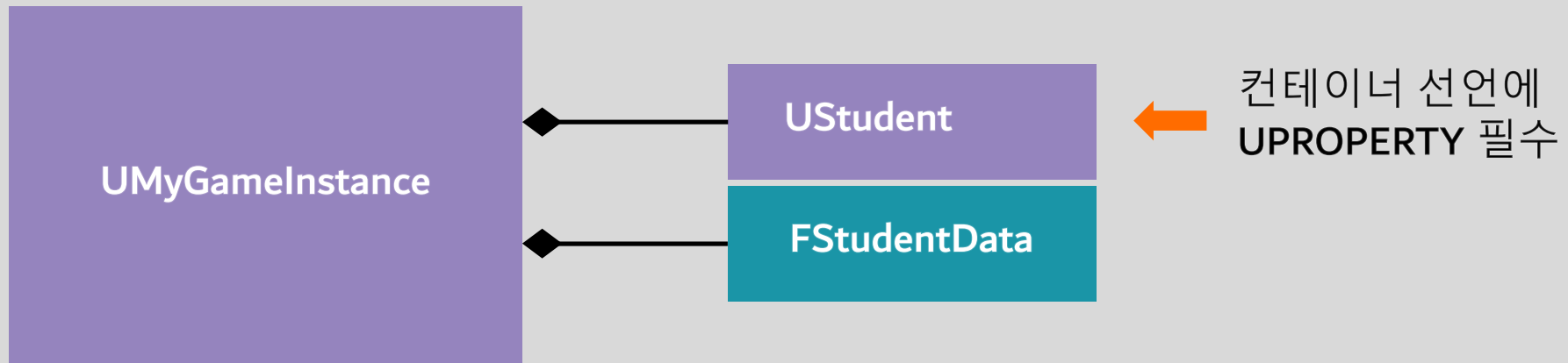
- 리플렉션에 관련된 언리얼 오브젝트의 계층 구조



TArray의 활용2

객체의 동적 배열 관리를 위한 예제 다이어그램

- 언리얼 오브젝트 학생의 동적 배열 관리 방법
- 언리얼 구조체 학생 정보의 동적 배열 관리 방법



컨테이너 자료의 배치 처리

- 언리얼 엔진이 제공하는 알고리즘 라이브러리의 활용
- 컨테이너 요소 대상으로 다양한 작업을 할 수 있음
 - 총합 구하기 (Sum)
 - 최대, 최소값 찾기 (MinElementBy, MaxElementBy)
 - 새로운 컨테이너로 옮기기 (Transform)
 - 기타 등등..
- 총합 구하는 예시

```
#include "MyGameInstance.h"  
#include "Algo/Accumulate.h"
```



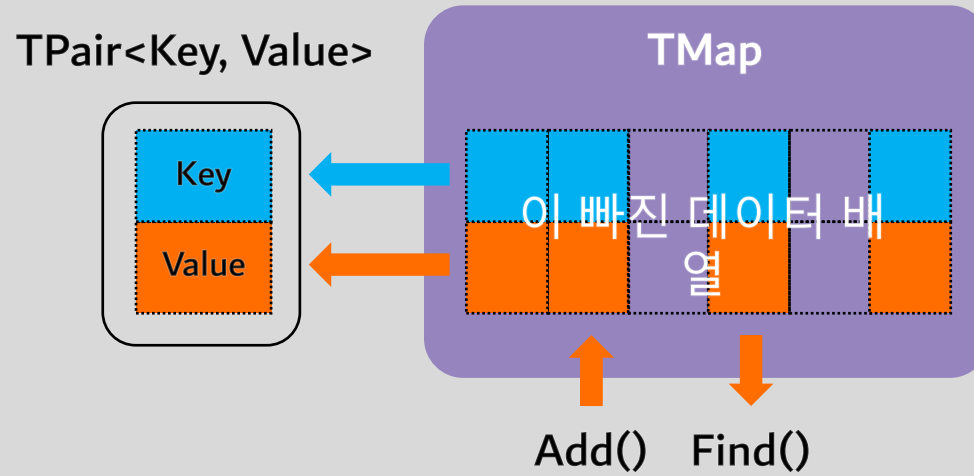
```
// 원기 관용 루프  
int32 Sum = 0;  
for (const int32& IntElem : IntegerArray1)  
{  
    Sum += IntElem;  
}  
ensure(Sum == 30);  
  
// 언리얼 C++ 알고리즘 라이브러리를 활용한 총합 계산  
int32 SumByAlgo = Algo::Accumulate(IntegerArray1, 0);  
ensure(Sum == SumByAlgo);
```

TMap의 구조와 활용

TMap의 특징

- STL map과 TMap의 비교
 - STL map의 특징
 - STL map은 STL set과 동일하게 이진 트리로 구성되어 있음.
 - 정렬은 지원하지만, 메모리 구성이 효율적이지 않으며, 데이터 삭제시 재구축이 일어날 수 있음.
 - 모든 자료를 순회하는데 적합하진 않음.
 - 언리얼 TMap의 특징
 - 키, 밸류 구성의 튜플(Tuple) 데이터의 TSet 구조로 구현되어 있음
 - 해시테이블 형태로 구축되어 있어 빠른 검색이 가능함.
 - 동적 배열의 형태로 데이터가 모여있음.
 - 데이터는 빠르게 순회할 수 있음.
 - 데이터는 삭제해도 재구축이 일어나지 않음.
 - 비어있는 데이터가 있을 수 있음.
 - TMultiMap을 사용하면 중복 데이터를 관리할 수 있음.
- 동작 원리는 STL unordered_map과 유사함.
- 키, 밸류 쌍이 필요한 자료구조에 광범위하게 사용됨.

TMap의 내부 구조



TMap



<https://bit.ly/uetmapkr>

<https://docs.unrealengine.com/5.1/ko/map-containers-in-unreal-engine/>

TMap의 활용

자료구조의 시간 복잡도 비교

- 각 자료구조의 시간복잡도(Time Complexity)

	TArray	TSet	TMap	TMultiMap
접근	$O(1)$	$O(1)$	$O(1)$	$O(1)$
검색	$O(N)$	$O(1)$	$O(1)$	$O(1)$
삽입	$O(N)$	$O(1)$	$O(1)$	$O(1)$
삭제	$O(N)$	$O(1)$	$O(1)$	$O(1)$

빈틈없는 메모
가장 높은 접근성
가장 높은 순회성

빠른 중복 감지

중복 불허
키 밸류 관리

중복 허용
키 밸류 관리

정리

구조체와 언리얼 컨테이너 라이브러리

1. TArray, TSet, TMap 컨테이너 라이브러리 내부 구조와 활용 방법
2. 언리얼 구조체의 선언 방법
3. TSet과 TMap에서 언리얼 구조체를 사용하기 위해 필요한 함수의 선언과 구현 방법