# TSet

和TMap类似，但速度更快，无需单独提供键值进行关联，不允许有重复的键值

TSet也是值类型，支持常规的赋值，赋值和析构函数的操作。以及其元素较强的所有权。集合被销毁时，其元素也将被销毁。

同质容器，只允许存储相同类型的数据

# UObject

UE4对于C++进行了深度调整，靠拢了其他虚拟机语言，为了方便**对象内存管理（去掉指针）**和**映射数据操作（提供和蓝图交互的基本依据）**，UE4对于UECPP进行了更加严谨的语法定义，所有UE中使用的对象大部分继承自UObject，这样系统管理更加轻松，不必设计更多的不定内容，UObject具备如下的优点

1. 垃圾回收
2. 引用更新（更新受限于GC回收机制，需要使用UPROPERTY宏标记）
3. 序列化（场景中的Actor被保存时发生）
4. 默认属性变化更新

如果一个类，用蓝图进行了延展，当我们在源码修改了一个类的成员属性值，则蓝图中的此属性也会随之修改（蓝图之前没有调整过该属性）

1. 自动属性初始化
2. 自动编辑器整合
3. 运行时类型信息可用

运行时使用Cast可以进行类型信息投射检查

1. 网络复制

# UCLASS（）

这是一个宏。用于标记从UObject派生的类，使得UObject处理系统识别到它们。

该宏为描述其是基于虚幻引擎的类型。对于标记UCLASS的类，引擎可以进行识别。

# 继承关系中的头文件描述

#include "HeroActor.generated.h"

必须是最后一个被引入的头文件，默认由UE模版生成，此头文件需要了解其他类，所以需要放到最后。

GENERATED\_BODY()

宏不获取参数，但会对类进行设置，以支持引擎要求的基础结构。宏将成员访问等级设置为“public”，而非设置为“private”访问域。但是建议大家进行访问域提示的时候明确指出访问域级别。

# 创建UObject类

NewObject（）是最简单的UObject工厂模式函数。他需要可选的外部对象和类，并会创建拥有自动生成的名称新实例。

NewObject<AHeroActor>(AHeroActor::StaticClass());

返回该类的具体实例指针

# UE中的描述宏

## UCLASS()类描述宏

这是一个宏。用于标记从UObject派生的类，使得UObject处理系统识别到它们。

##### 常用标记

标记方式

UCLASS(描述指令， 描述指令， …)

描述指令是已知的，虚幻为我们提供的

BlueprintType

此类可以作为蓝图中的一种变量类型使用，类默认均可被蓝图访问，一般我们用此标记描述结构体，提供给蓝图访问。

Blueprintable/NotBlueprintable

用来表明当前类是否可以被蓝图继承，默认是可以被继承，标记关系向子类传递，子类可以覆盖描述关系

Abstract

Abstract修饰符讲该类声明为“抽象基类”，这样会组织用户在虚幻编辑器中想这个世界中添加这个类的Actor，或者在游戏过程中创建这个类的实例。

Const

本类中的所有属性及函数均为常态，并应作为常量导出。该标识由子类继承。

Config

将类内的成员变量数据信息保存到本地配置文件中，需要现实调用函数SaveConfig使用，并配合UPROPERTY宏操作

ClassGroup

用来配置组件在添加时的分组情况

## UFUNCTION()

用来标记当前函数，将函数进行序列化映射，提供给蓝图使用

##### 结构

UFUNCTION(指令， 指令， 指令， …， meta（key=value）)

##### 常用标记

BlueprintCallable

标记此函数可以被蓝图调用

**注意：如果函数参数是引用类型，则在蓝图中调用被当作输出针脚，如果传入参数是const修饰的引用类型，则在蓝图中被当作输入针脚**