# 小山的 C# 教學

第 48 課 介面 (Interface)

#### Creature

string say()
void injured(int damage)
string getName()

繼承

# Player

string say()
void injured(int damage)
string getName()

# Villager

string say()
void injured(int damage)
string getName()

### Monster

string say()
void injured(int damage)
string getName()

# 新設計! 玩家可以跟怪物戰鬥!

但村民不行…

#### Creature

string say()
void injured(int damage)
string getName()

繼承

# Player

string say()
void injured(int damage)
string getName()

void attack(Creature target)

## Villager

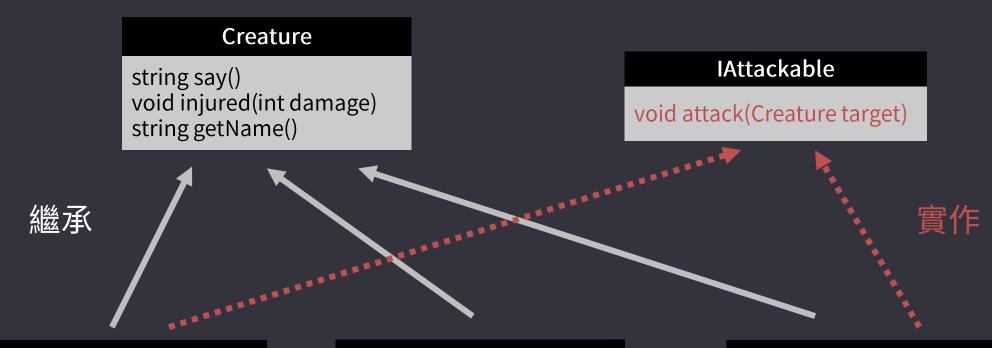
string say()
void injured(int damage)
string getName()

#### Monster

string say()
void injured(int damage)
string getName()
void attack(Creature target)

# 使用 interface 吧!

像是契約,要求實作的 class 必須要具備的行為



# Player

string say()
void injured(int damage)
string getName()
void attack(Creature target)

### Villager

string say()
void injured(int damage)
string getName()

#### Monster

string say()
void injured(int damage)
string getName()
void attack(Creature target)

# 新新設計! 玩家跟村民可以對話!

但怪物不行…

#### Creature

string say()
void injured(int damage)
string getName()

#### **ITalkable**

string talkTo(Creature target)

#### **IAttackable**

void attack(Creature target)



實作

實作

## Player

string say()
void injured(int damage)
string getName()

void attack(Creature target)
string talkTo(Creature target)

### Villager

string say()
void injured(int damage)
string getName()
string talkTo(Creature target)

#### Monster

string say()
void injured(int damage)
string getName()
void attack(Creature target)

# 總結一下 Interface

- Interface 像是契約,要求實作的 class 需要具備指定的能力
- 命名習慣以大寫Ⅰ開頭
- 跟繼承不同,一個 class 可以實作多個 interface
- 實作 interface 定義的 method 必須
  - ◆ 不可以是 static
  - ◆ 必須是 public
  - ◆ Signature (輸入值、名稱與輸出值) 必須完全相同
- 繼承不同 base class 的 class 也能實作相同 interface