C# 泛型

Ian Chen

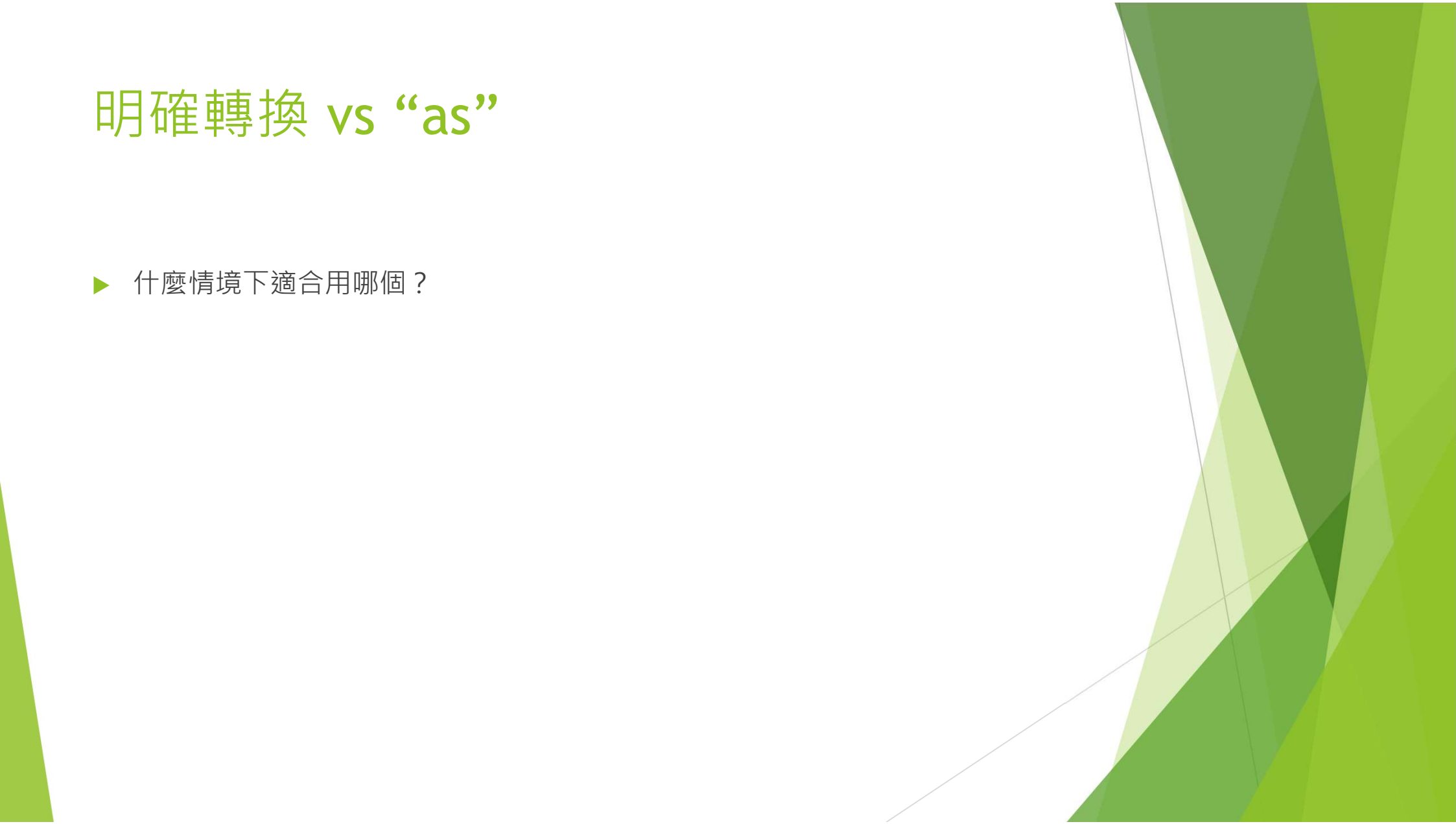
2019/11/11

繼承 vs 介面

- 什麼情境下適合用哪個？

明確轉換 vs “as”

- 什麼情境下適合用哪個？



Generics

泛型

什麼是泛型？

GenericSample_1

如果沒有泛型？

GenericSample_2

ArrayList

GenericSample_3

可能造成什麼問題？

GenericSample_4

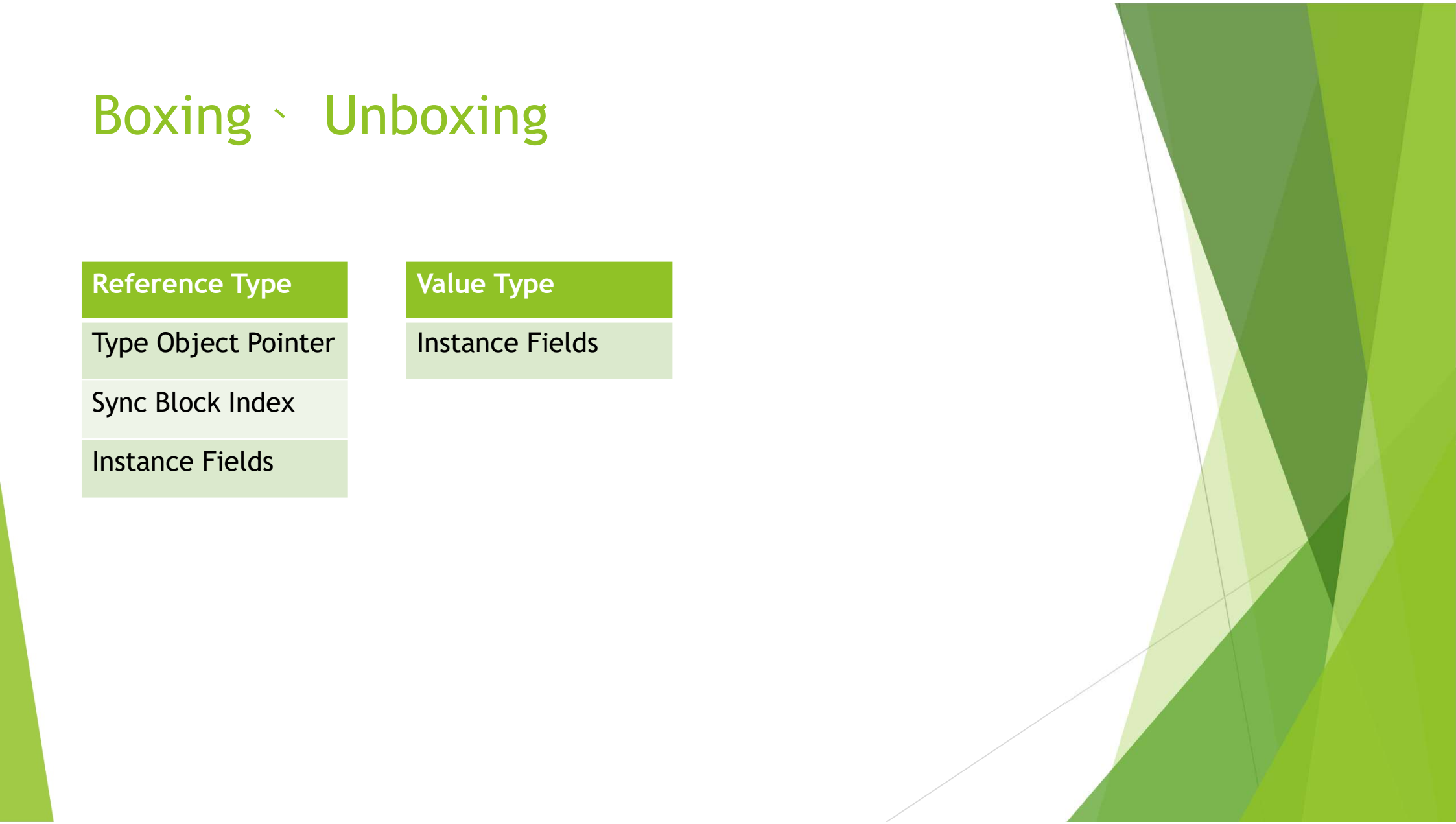
ArrayList

- Boxing
- Unboxing

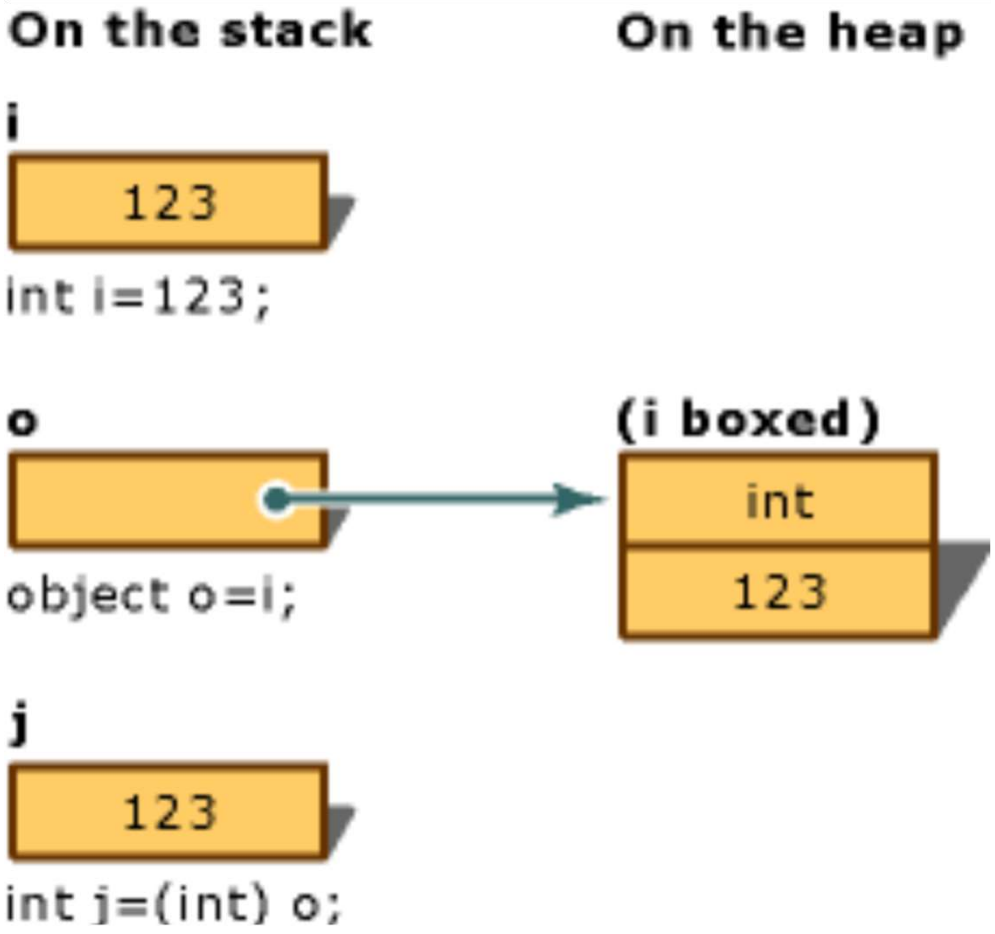
Boxing、Unboxing

Reference Type
Type Object Pointer
Sync Block Index
Instance Fields

Value Type
Instance Fields



Boxing、 Unboxing



Boxing、 Unboxing

- 造成的效能問題？
- 使用上的問題？

GenericSample_5

泛型

- Member
  - Class
  - Interface
  - Method
  - Delegate
  - Field
  - Property

GenericSample_6

泛型

- <T>
- Type Parameter
- Type Argument

GenericSample_7

泛型

- 自動推斷<T>的類型

GenericSample_8

不同type的<T>是相同型別嗎？

- Ex:
- MyClass<int>
- MyClass<string>

泛型

- Static Constructors

GenericSample_9

如果方法裡面會用到特定Type的Member...

GenericSample_13

泛型條件約束

- where

泛型條件約束

- new()

GenericSample_10

泛型條件約束

- where T : class
- where T : <Base Class Name>
- where T : <Interface Name>
- where T : U

GenericSample_11

泛型條件約束

- where T : struct
- where T : enum (C# 7.3+)
- where T : delegate
- where T : notnull

GenericSample_11

Func

Func<TResult> 代理人

命名空間: System

組件: System.Runtime.dll, mscorlib.dll, netstandard.dll, System.Core.dll

封裝沒有參數並傳回 `TResult` 參數所指定之型別值的方法。

```
C#
public delegate TResult Func<out TResult>();
```

類型參數

`TResult`

這個委派所封裝之方法的傳回值之型別。

傳回值

`TResult`

這個委派所封裝之方法的傳回值。

繼承 Object → Delegate → Func<TResult>

FuncSample_1

Func

- Method

FuncSample_2

Func

- 方法參數

FuncSample_3

怎麼實做一個Where？

FuncSample_3

Action

Action<T> 代理人

命名空間: System

組件: System.Runtime.dll, mscorlib.dll, netstandard.dll

封裝具有單一參數的方法，並且不會傳回值。

```
C#
public delegate void Action<in T>(T obj);
```

類型參數

T

這個委派所封裝之方法的參數型別。

參數

obj

這個委派所封裝之方法的參數。

繼承 Object → Delegate → Action<T>

ActionSample_1

End