MICROOH 麦可网

Android-从程序员到架构师之路

出品人：Sundy
讲师：高焕堂（台湾）

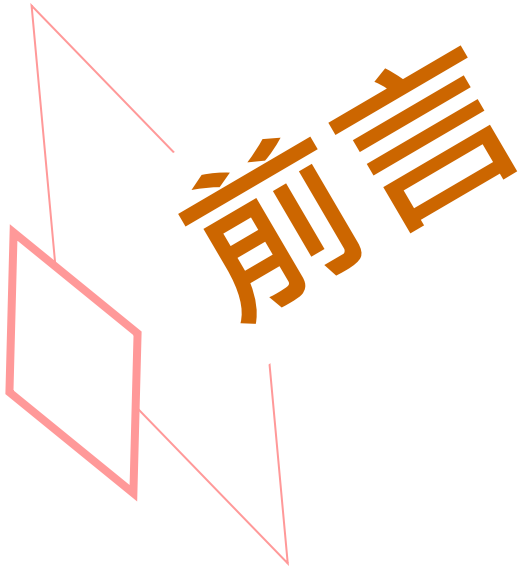
http://www.microoh.com

H05_b

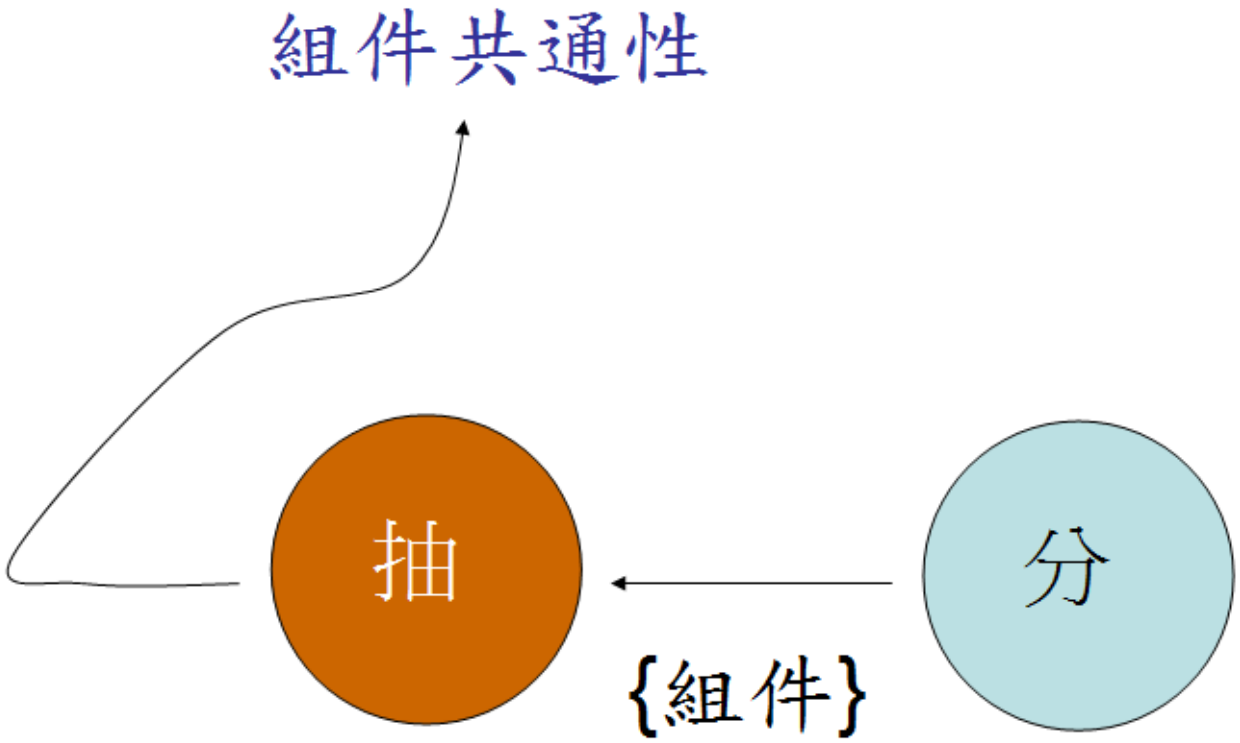
A段架构师：组合思维(b)

By 高煥堂

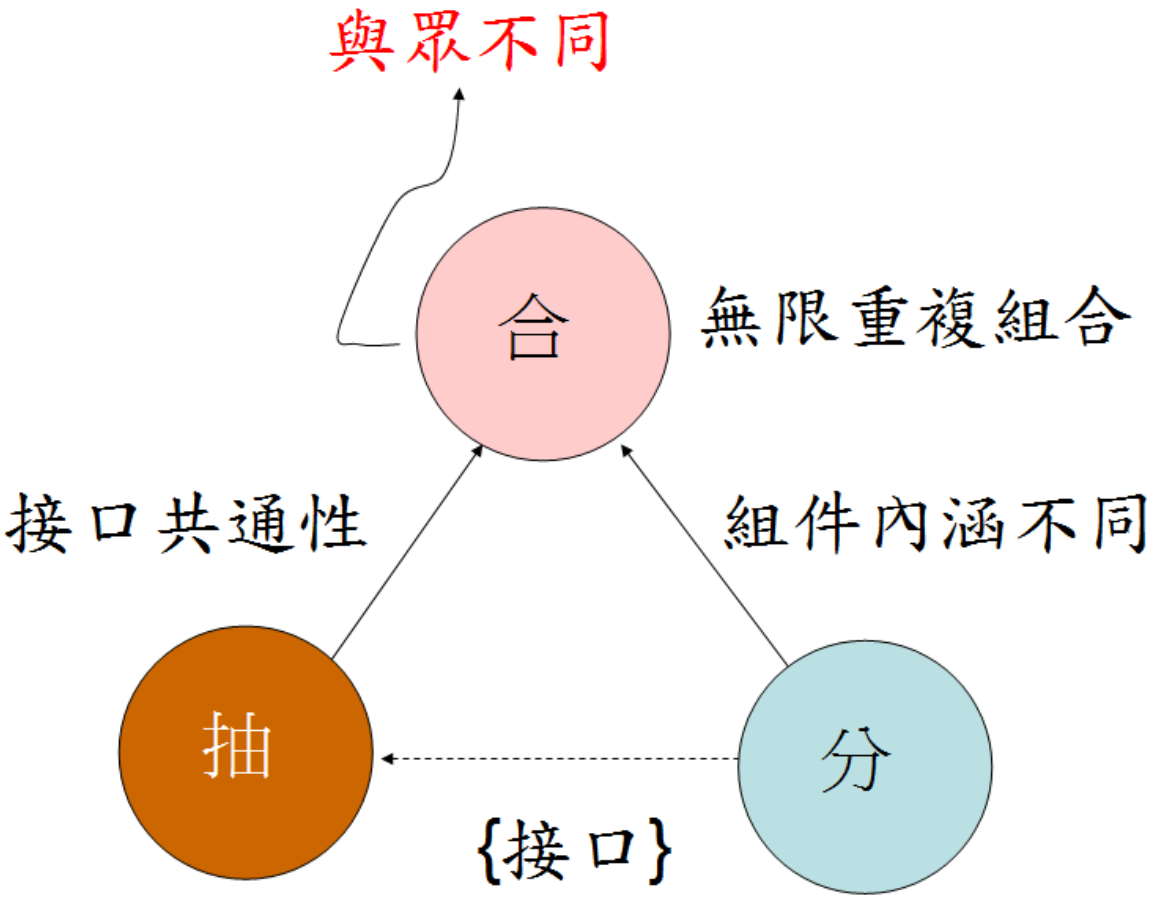
两派的”抽象视角”不同



抽象思维派



创新组合派



- 其实，分与合，两者都是一种抽象
- 抽象派与组合派，两者的差异只在于：抽象视角的不同而已。
- 无论那一种视角，抽象的目的都是：想得到简单。
- 想从简单中掌握复杂。

- 于是，
- 目的一样：想得到简单；然后从简单中掌握复杂。
- 手段一样：抽象(Abstraction)。
- 视角不同：由于对本质(Essence)的认知不同。

第1种认知

- 第1种认知：”本質(Essence)”這個字眼的涵意是“道”。而道不变，本质不变，真理是简单的。所以我们在开发软件时，就好比建一栋房子，下面的地基不变，上面的房屋是可变的。所以架构师是建造平台(地基)的人，是要找不变的人。

第2种认知

- 第2种认知：” 本質(Essence)” 涵意是 “不可或缺的” 特性，例如在牛津字典里，这个字眼的涵义是 “不可或缺的” ，并非” 稳定不变” 的。所以两种认知有很大的差异。

- 无论那一种认知，都想要：从复杂中得到简单，然后从简单中掌握复杂。
- 因为我们在遇到复杂的时候总是先找简单，因为简单不会让我们害怕。那么怎样找到简单？就是我们怎样从复杂中得到简单？人类最擅长的做法就是抽象。

抽象视角-1：基于本质是”简单不变的”

- 把很多具象的东西去掉差异性的东西，这叫抽象。例如把一群猫都抓过来你会发现它们的胡须都不一样，就对胡须视而不见而抽象掉；而尾巴也都不一样，就把尾巴去掉，就对尾巴视而不见而抽象掉。在心中留下来的，就是猫不变的本质了。
- 只是，一只没有胡须、没有尾巴的东西还称为<猫>吗？

抽象视角-2：基于本质是”不可或缺的”

- 如果本质(Essence)”是指不可或缺的，又怎样才能达到简单呢？只要想一想，女士们为什么要带皮包？把猫(不可或缺的)尾巴放进皮包里不就得到简单了么？把不可或缺的复杂特性妥善地包装起来，也是一种抽象。

- 中国古代就有这个理念，孙悟空会72变，大闹天宫，唐三藏怎么处理？就是在孙悟空的头上安一个紧箍咒，念紧箍咒孙悟空就会头疼，所以唐僧还是达到用简单去控制复杂，但他没有去伤害孙悟空。

软件本质是复杂? 还是简单?

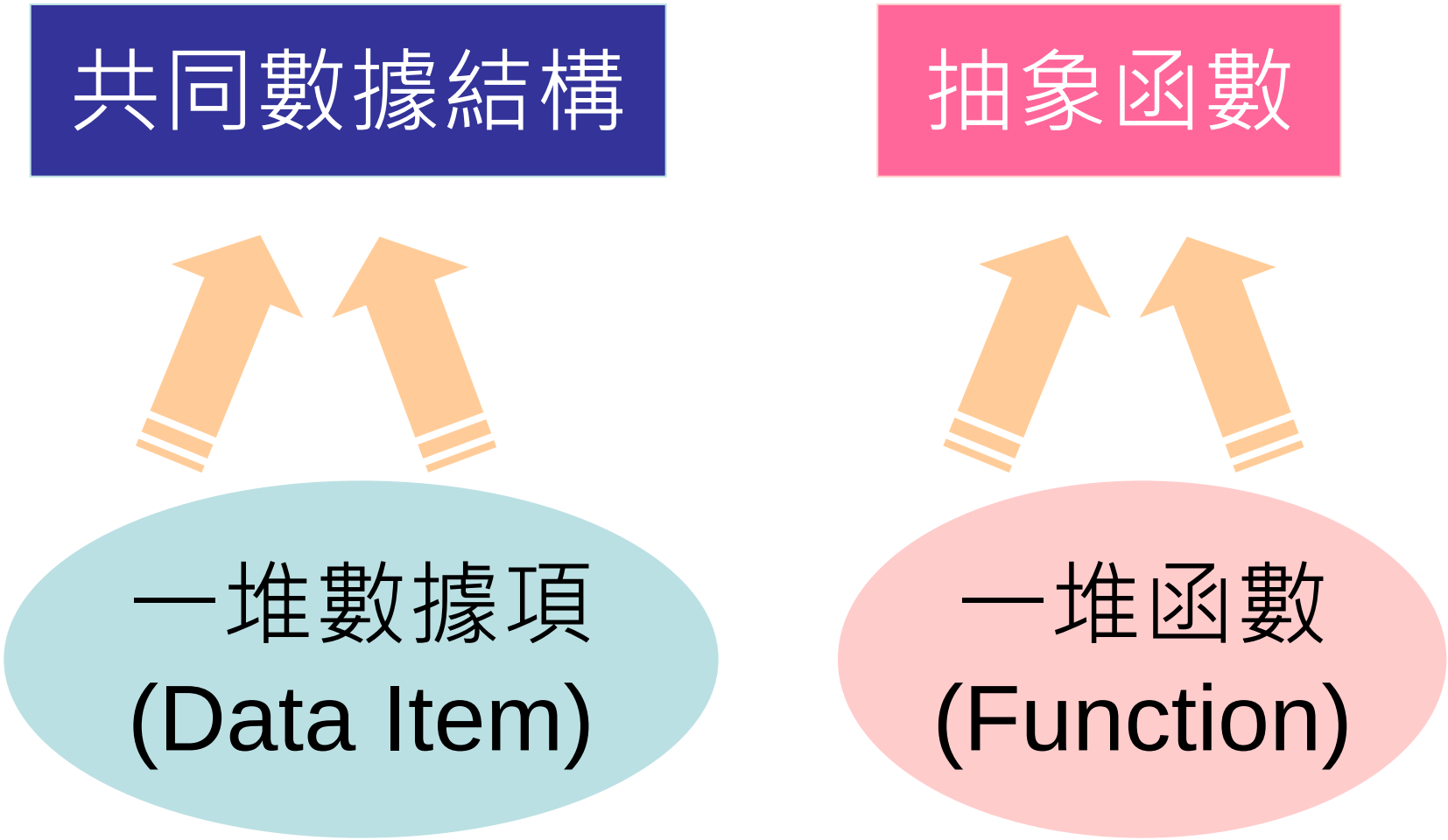
- 一位著名的软件专家Fred Brooks，40年前就在他的《人月神话》一书里说道："软件的复杂是本质性的，并非表象而已" (The complexity of software is an essential property, not an accidental one.)。

- 另外，著名软件架构师 周爱民先生在他的《大道至简》一书里是说：“软件的本质是简单的”。
- 其实这是一体的两面，分别是两个命题，软件有复杂的一面，也有简单的一面。

抽象演练：视角1

- 从一堆软件函数(Function)中抽象出 "抽象函数 "。
- 也从一堆软件数据(Data)中抽象出 "共同数据结构"。

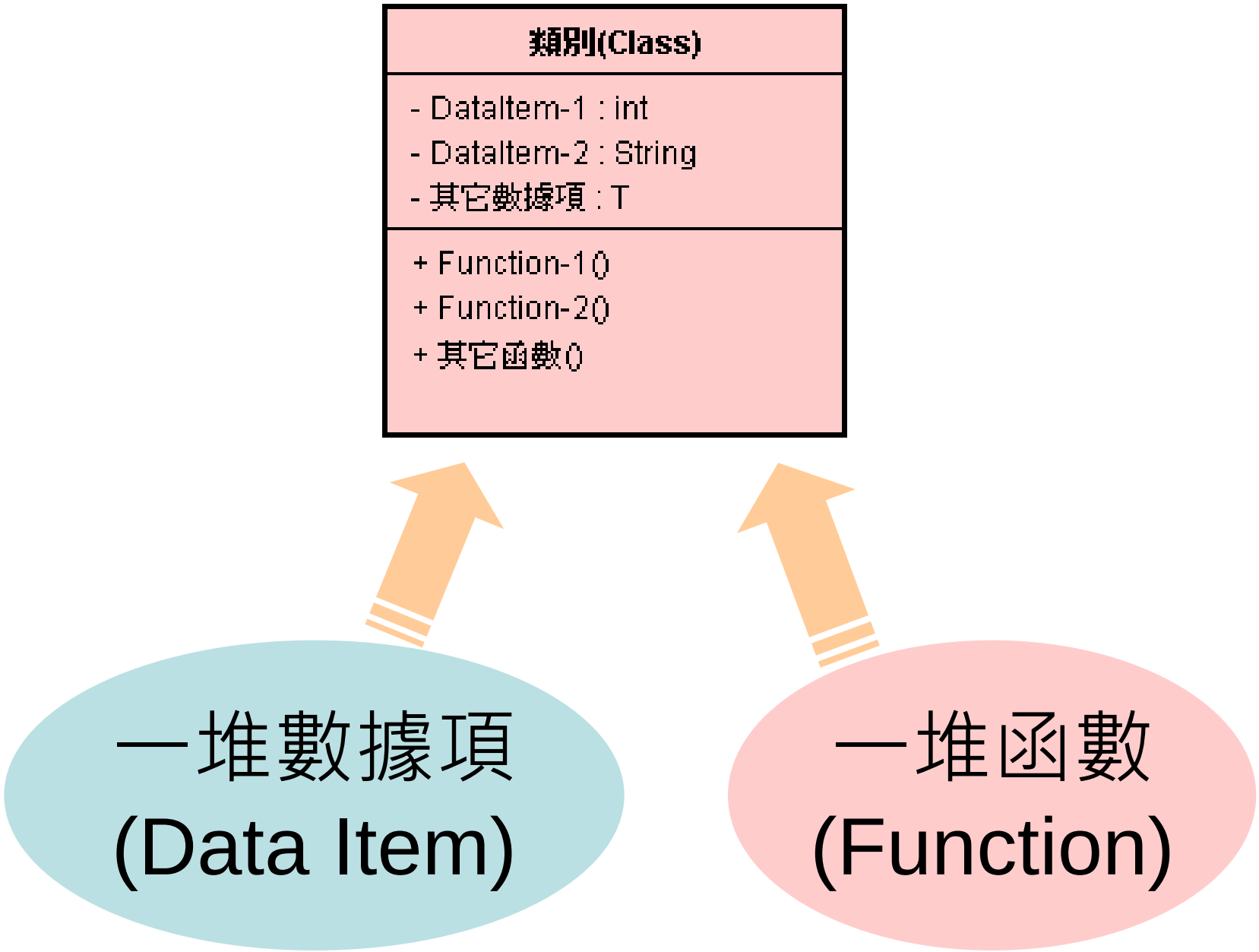
可能抽掉
不可或缺的複雜


共同數據結構
抽象函數
一堆數據項
(Data Item)
一堆函數
(Function)

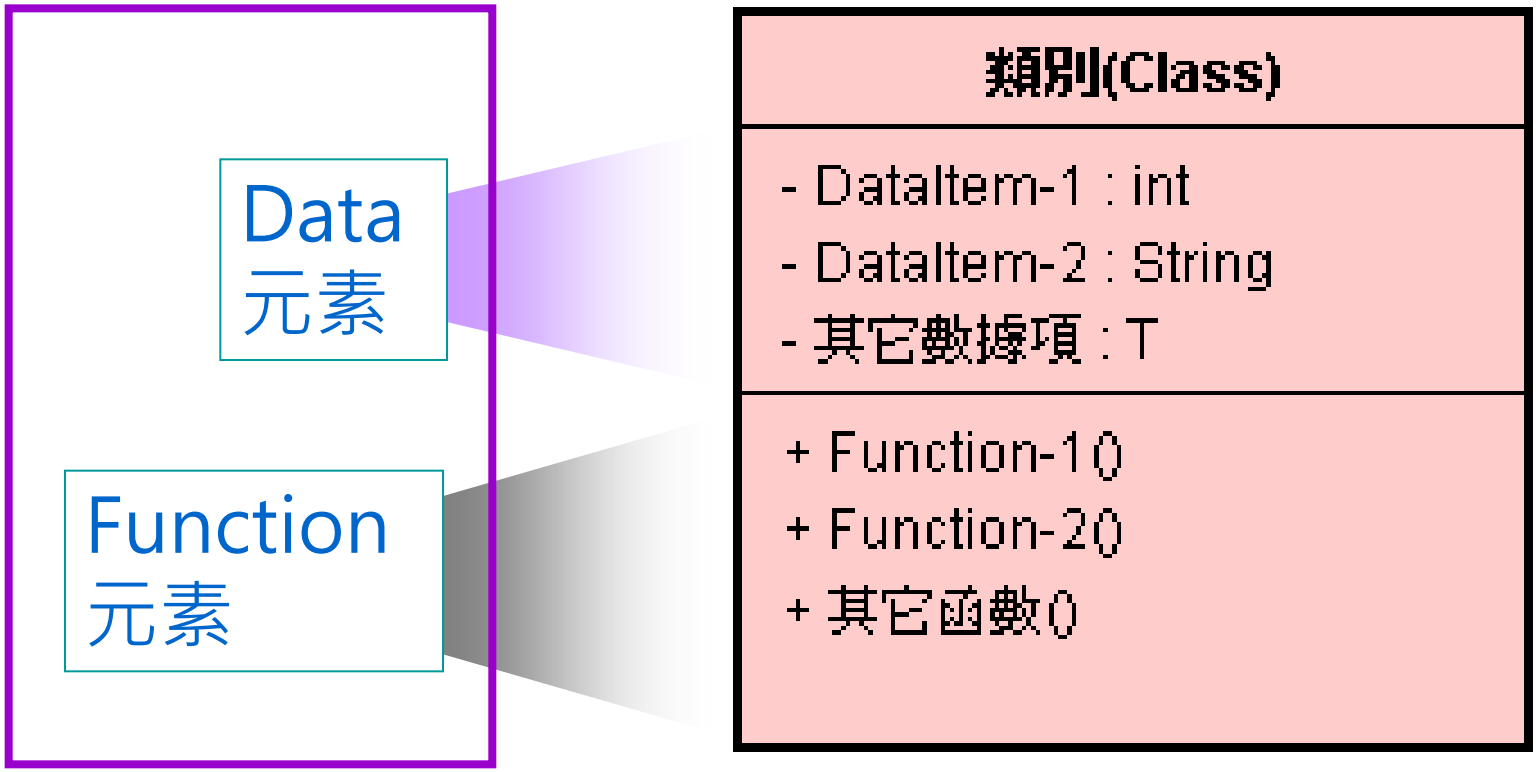
抽象演练：视角2

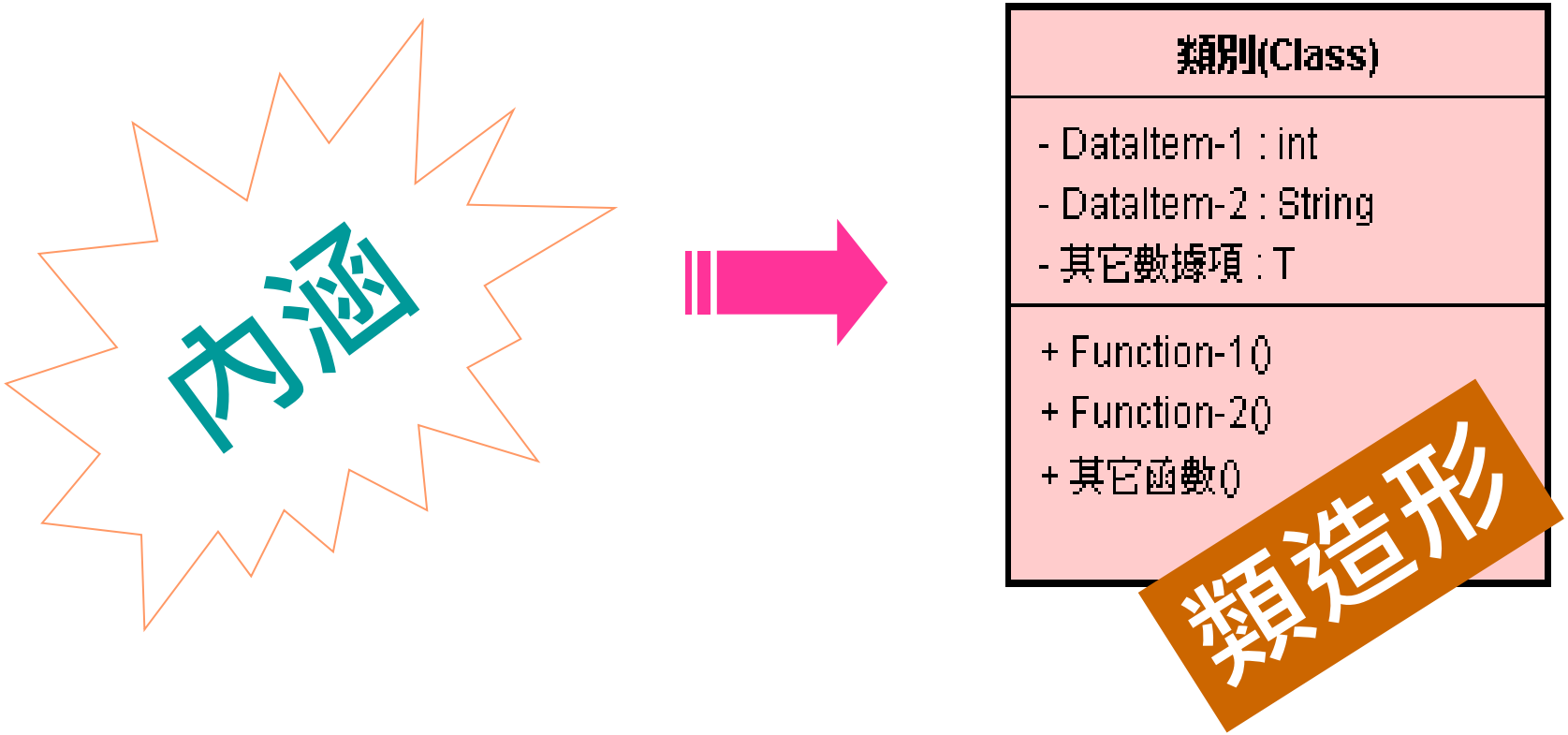
- 从具象的一堆函数和一堆数据之中，抽象或设计出 "类(Class)结构" 来包容具象或抽象的函数&数据。

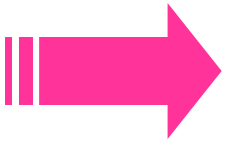
容納各項
不可或缺的複雜


類別(Class)
- DataItem-1 : int
- DataItem-2 : String
- 其它數據項 : T
+ Function-1()
+ Function-2()
+ 其它函數()
一堆數據項
(Data Item)
一堆函數
(Function)

Class造形


Data
元素
Function
元素
類別(Class)
- DataItem-1 : int
- DataItem-2 : String
- 其它數據項 : T
+ Function-1()
+ Function-2()
+ 其它函數()


內涵
類別(Class)
- DataItem-1 : int
- DataItem-2 : String
- 其它數據項 : T
+ Function-1()
+ Function-2()
+ 其它函數()
類造形



類別(Class)
- DataItem-1 : int - DataItem-2 : String - 其它數據項 : T
+ Function-1() + Function-2() + 其它函數()

類造形

楓葉類

楓葉
- 顏色 : int - 大小 : int - 種類 : char
+ 飛() : void + 其它() : void

鸚鵡
(內涵)

類別(Class)

- DataItem-1 : int
- DataItem-2 : String
- 其它數據項 : T

+ Function-1()
+ Function-2()
+ 其它函數()

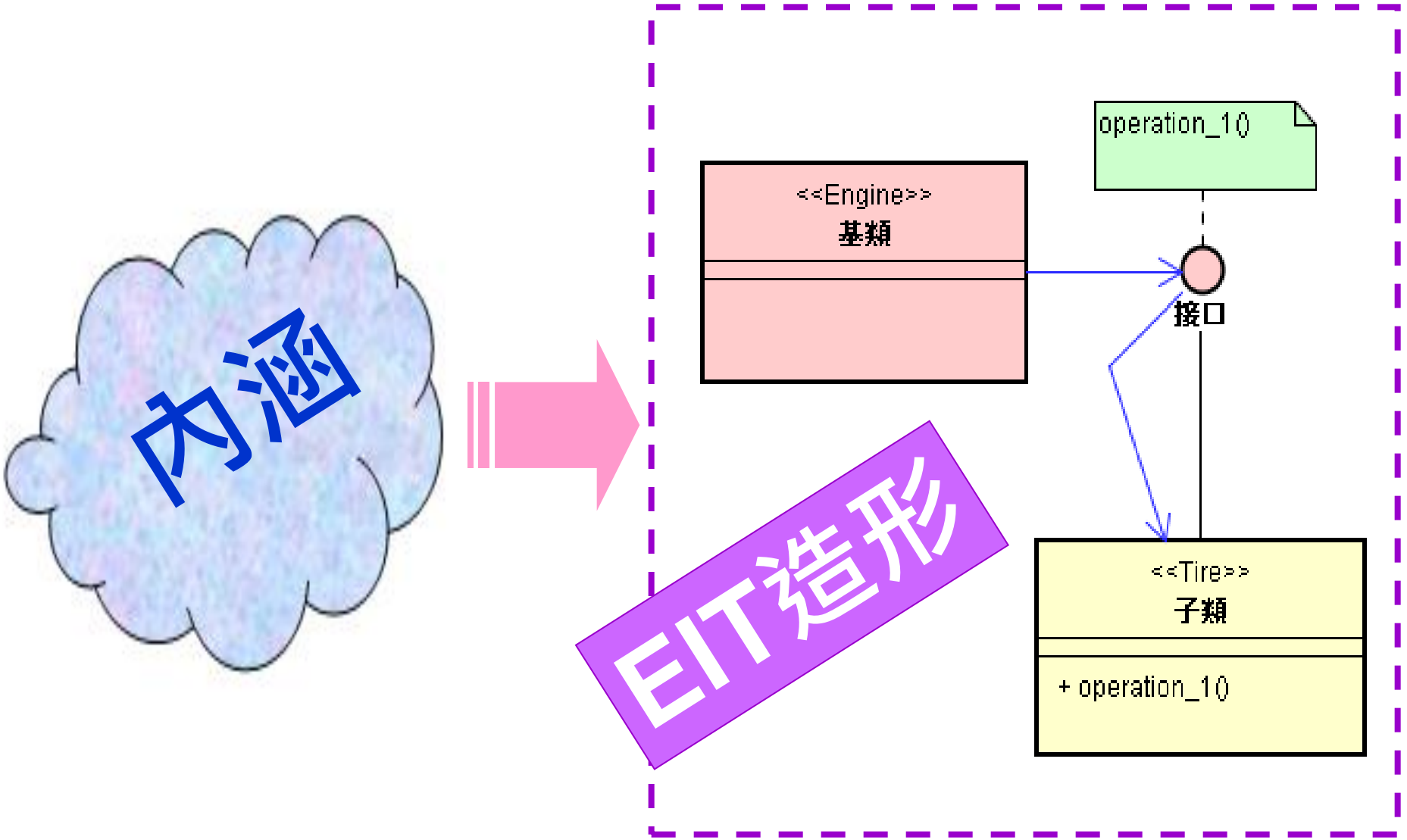
類造形

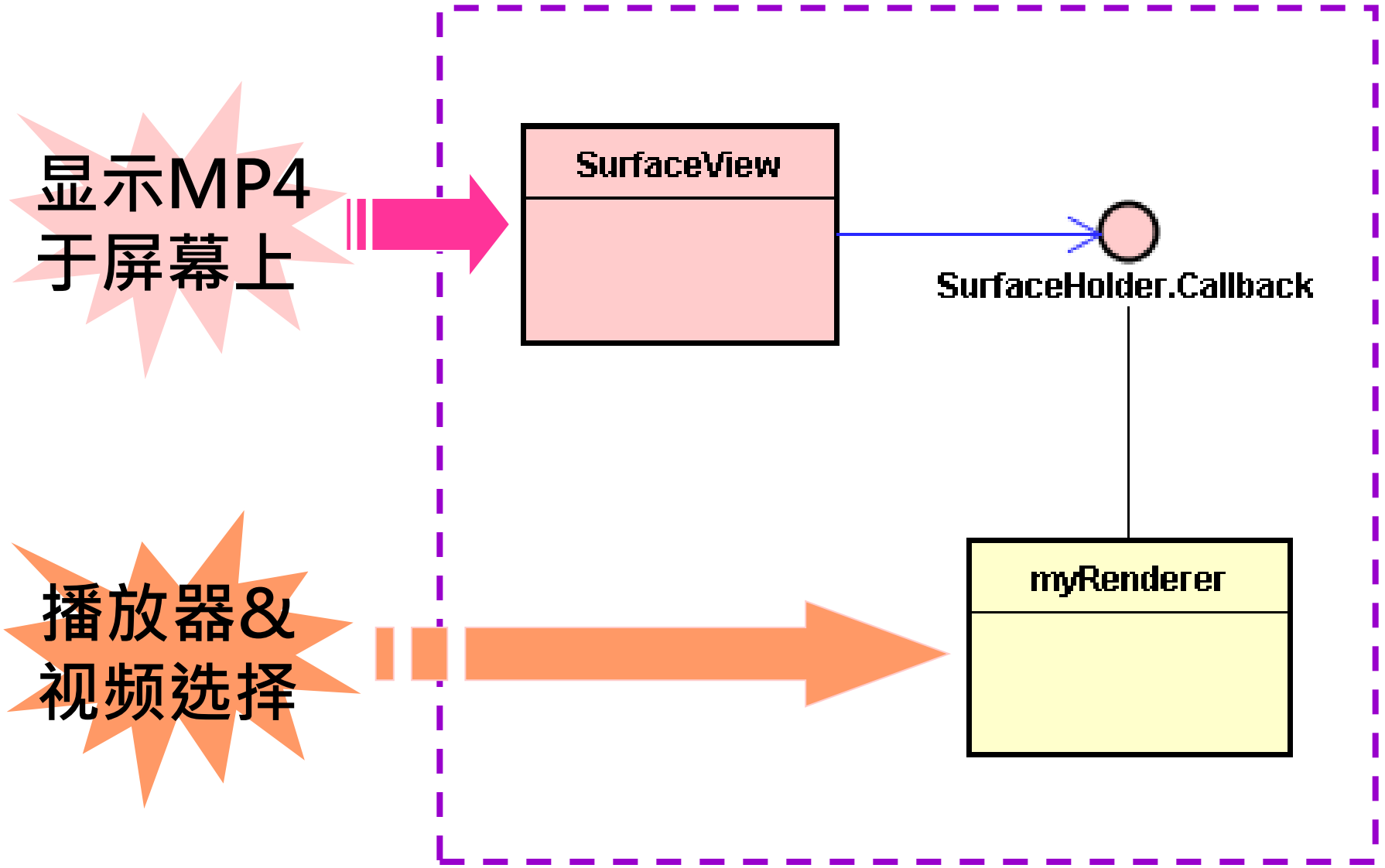
鸚鵡

- 品種() : char
- 價錢 : double
- 其它 : void

+ 講話() : void
+ 其它() : void

鸚鵡類


內涵
EIT造形
operation_10()
<<Engine>>
基類
接口
<<Tire>>
子類
+ operation_10()

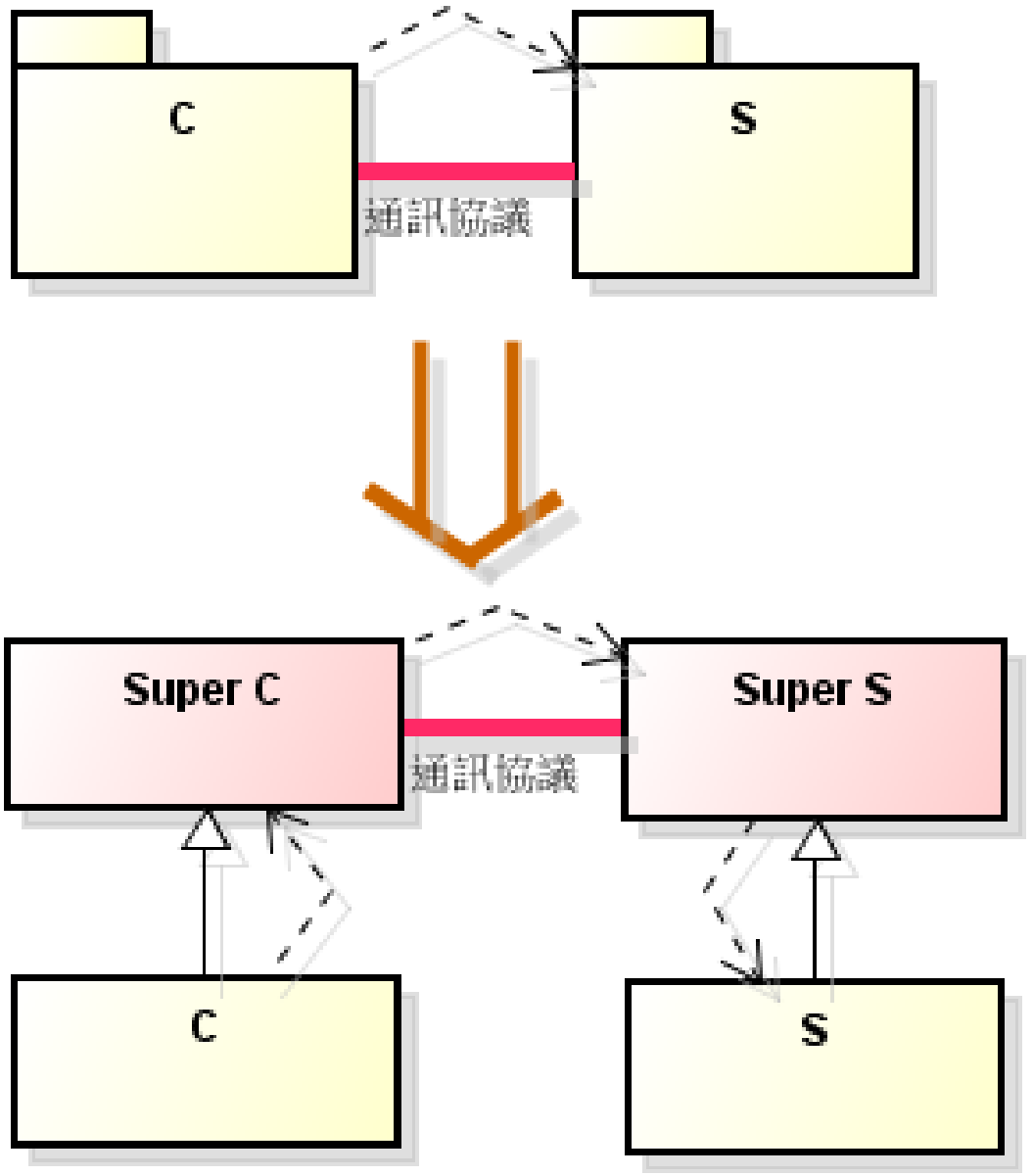

显示MP4
于屏幕上
SurfaceView
SurfaceHolder.Callback
播放器&
视频选择
myRenderer

未来性：包容改变

- 架构师清晰的Vision并非要准确预测世界的未来景象，未来世界是不可知的。但是我们的目前决策会影响世界未来的发展轨迹。
- 所以Vision是期望的未来景象；对架构师而言，它是手段，不是目的。真正目的是：要找出有助于实现愿景的目前决策，这样的决策才真正具有未来性。

決策要有未來性

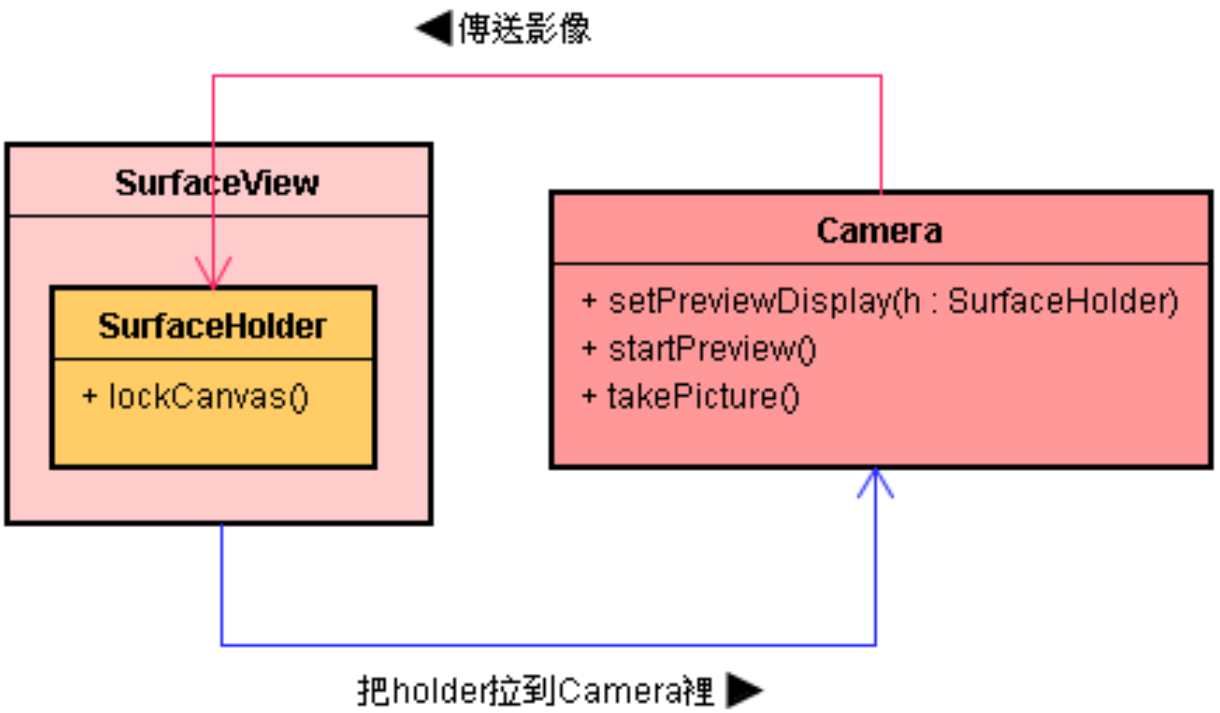
设计出未来性：
例如，包容通信
的<未来变化>



设计出未来性：
例如，底层可抽换、上层跨平台。

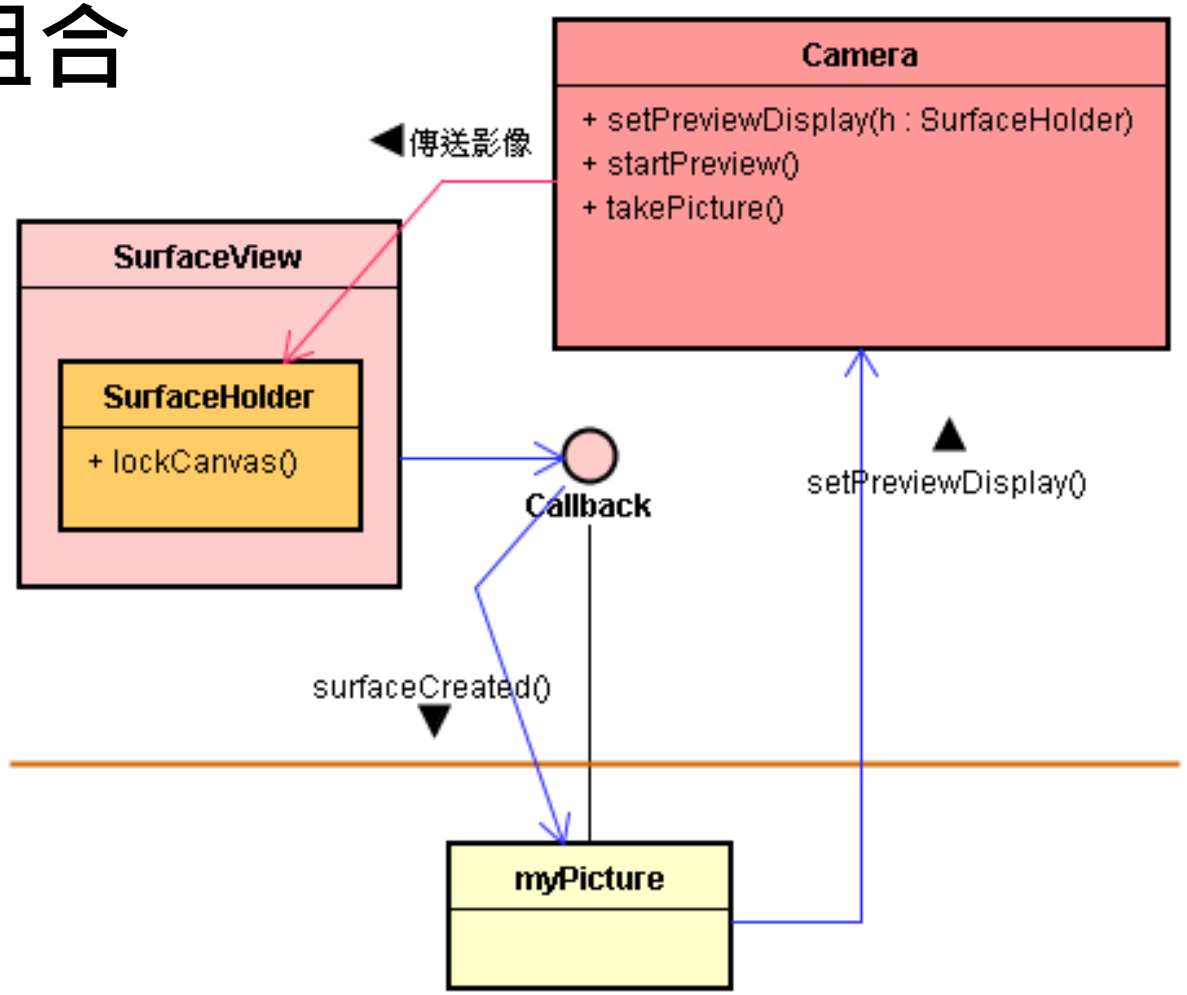
例如，Android终端设备里的芯片常来自不同的供货商；此时设计一个架构来：1. 确保底层芯片的可抽换性；2. 同时实现了上层应用的跨平台(跨芯片)。

例如：缺乏未来性的架构

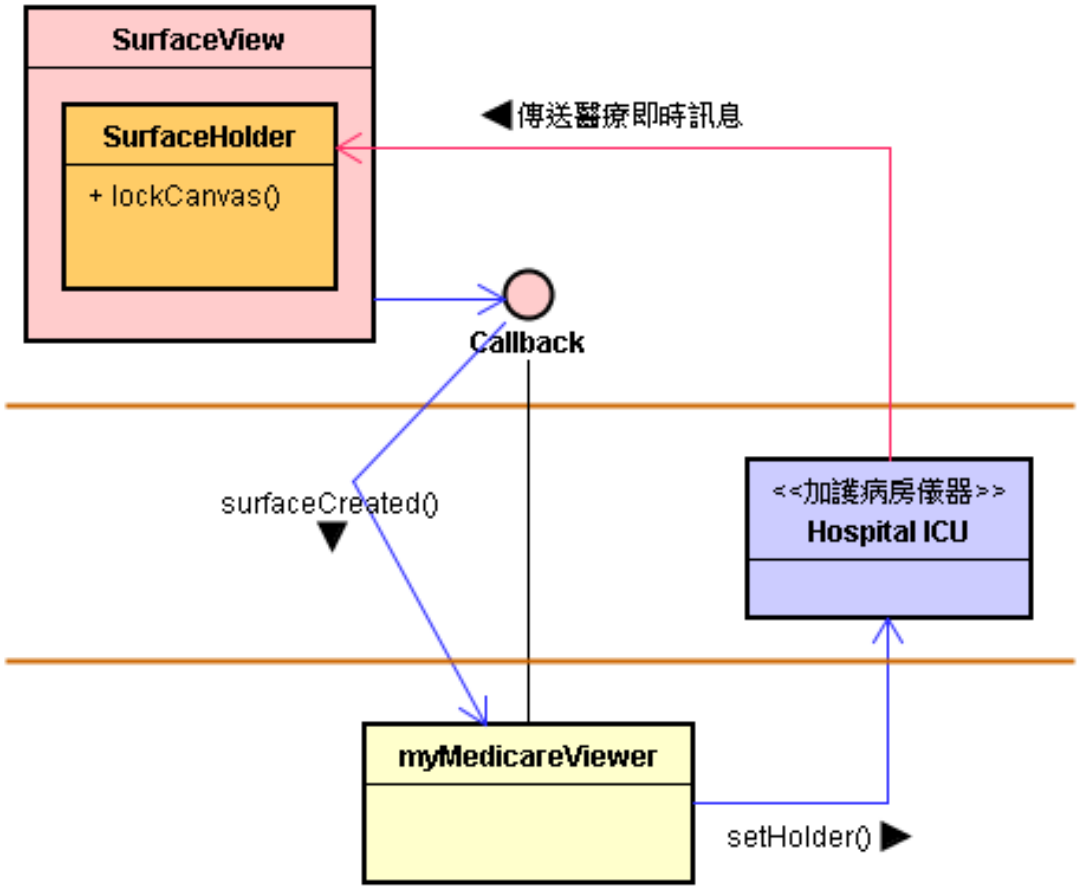

傳送影像
SurfaceView
SurfaceHolder
+ lockCanvas()
Camera
+ setPreviewDisplay(h : SurfaceHolder)
+ startPreview()
+ takePicture()
把holder拉到Camera裡

设计出未来性：

例如，设计插件、
创新组合



抽換插件，创新组合。


SurfaceView
SurfaceHolder
+ lockCanvas()
傳送醫療即時訊息
Callback
surfaceCreated()
<<加護病房儀器>>
Hospital ICU
myMedicareViewer
setHolder()

- 有些人認為，架構設計是要尋覓系統的共通性。 其實，架構是獨一無二的，架構設計是追求獨特性的、氣象萬千的、與眾不同的嶄新組合。蘋果公司 喬布斯說：「創造無非就是把事物聯結起來，...即若是非凡的創意通常也不過是對已有事物進行的新<<組合>>而已。」

Thanks...

高煥堂