

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>

F04_a

观摩：观摩Android的 ListView框架设计(a)

By 高煥堂

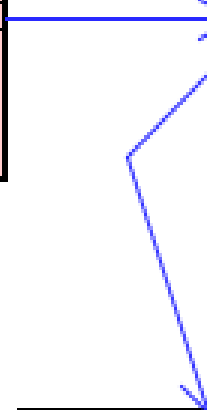
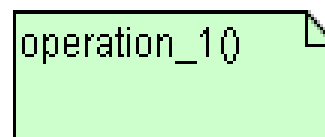
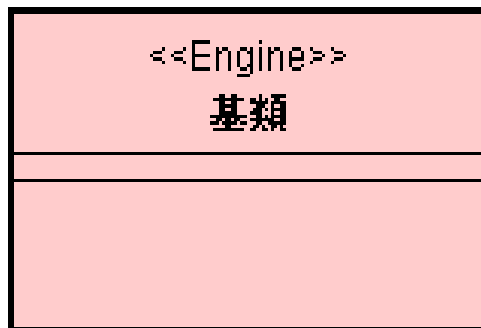
内容

1. 复习：两种知识的分析
2. ListView的接口设计
3. 兼顾通用性与特殊性的设计
4. BaseAdapter接口的使用范例

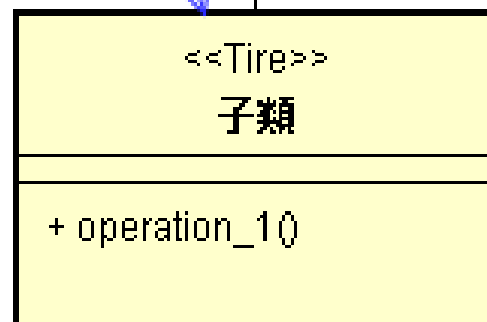
1、复习：两种知识的分析

- 需求知识分为两种：
 - 第一种需求知识→成为基类的内涵
 - 第二种需求知识→成为子类的内涵

第一种
需求知识



第二种
需求知识



两种需求知识的时间差距

- 这两种知识的取得，会有时间落差。
- 第一种需求知识→先取得
- 第二种需求知识→后取得

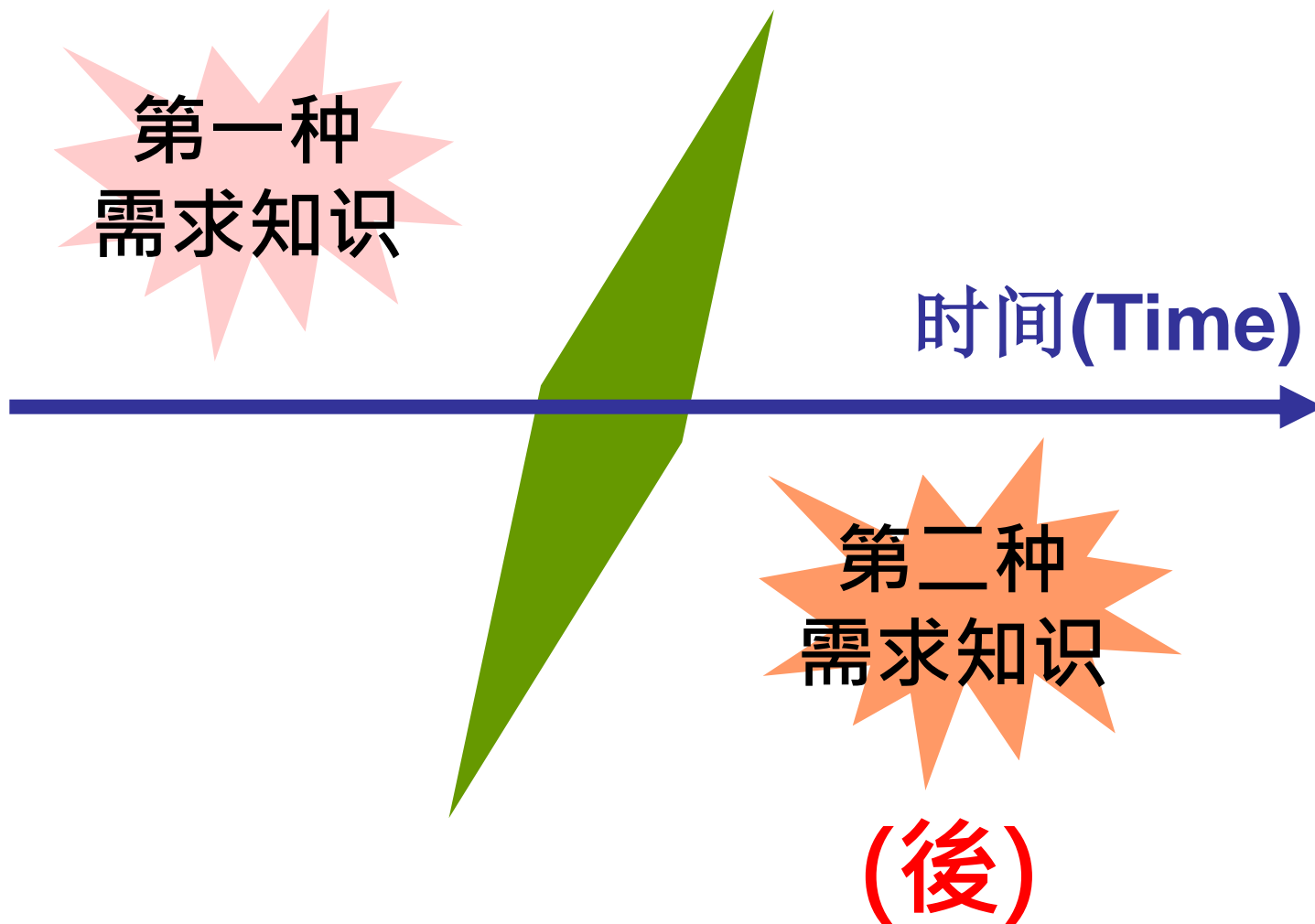
(先)

第一种
需求知识

时间(Time)

第二种
需求知识

(後)



时间先后的分界点：买主来到

- 以<买主来到>的时间点加以切分。

第一种：买主来之前已知的需求知识

第二种：买主(来到时才)提供的需求知识

买主来之前
已知的需求
知识



買主來了

Time

买主提供的
需求知识

两种知识分别写入<基类>和<子类>

- 第一种知识写入基类<E>
- 第二种知识写入子类<T>

<<E>>

基類

买主来之前的
需求知识



買主來了

Time

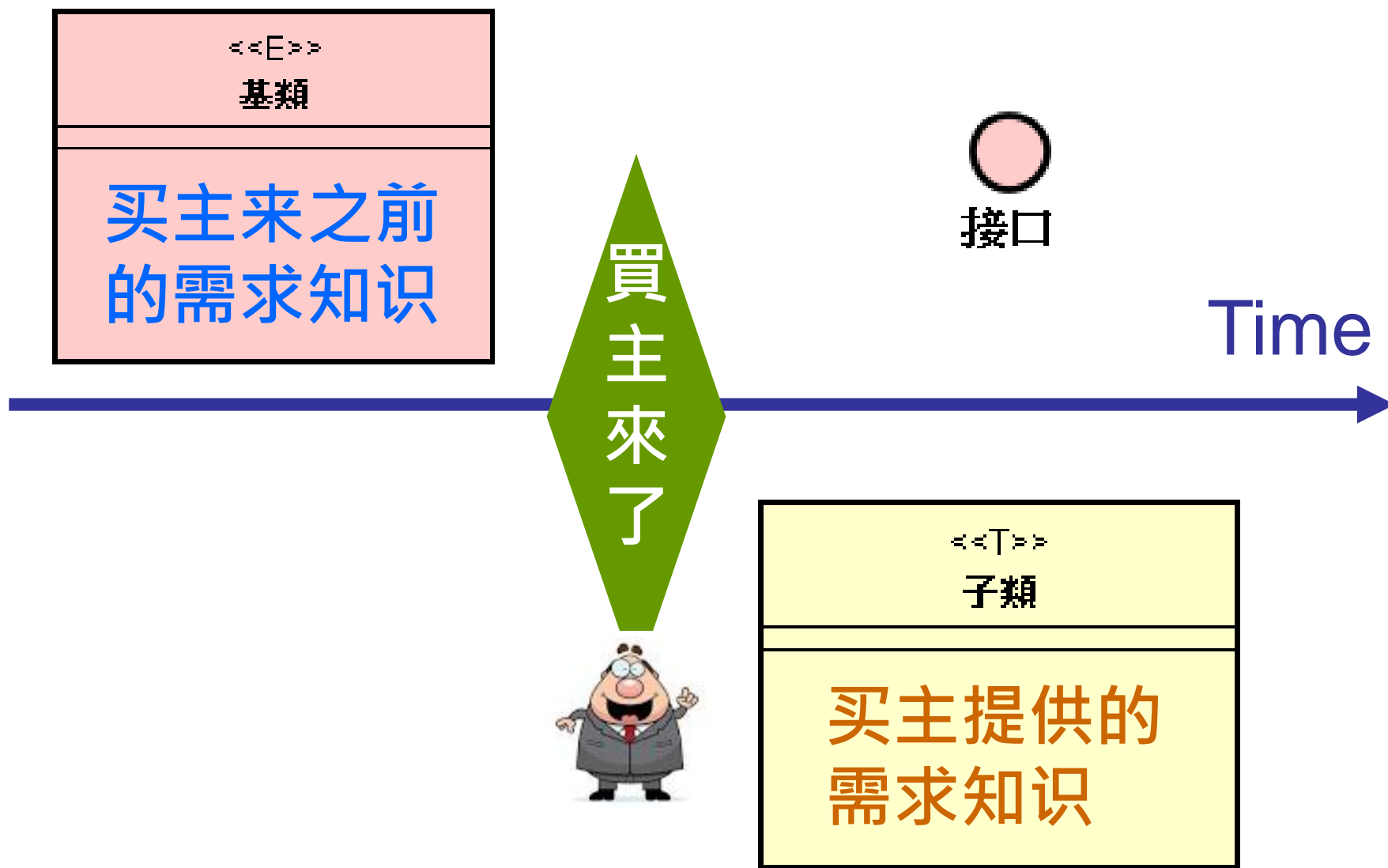
<<T>>

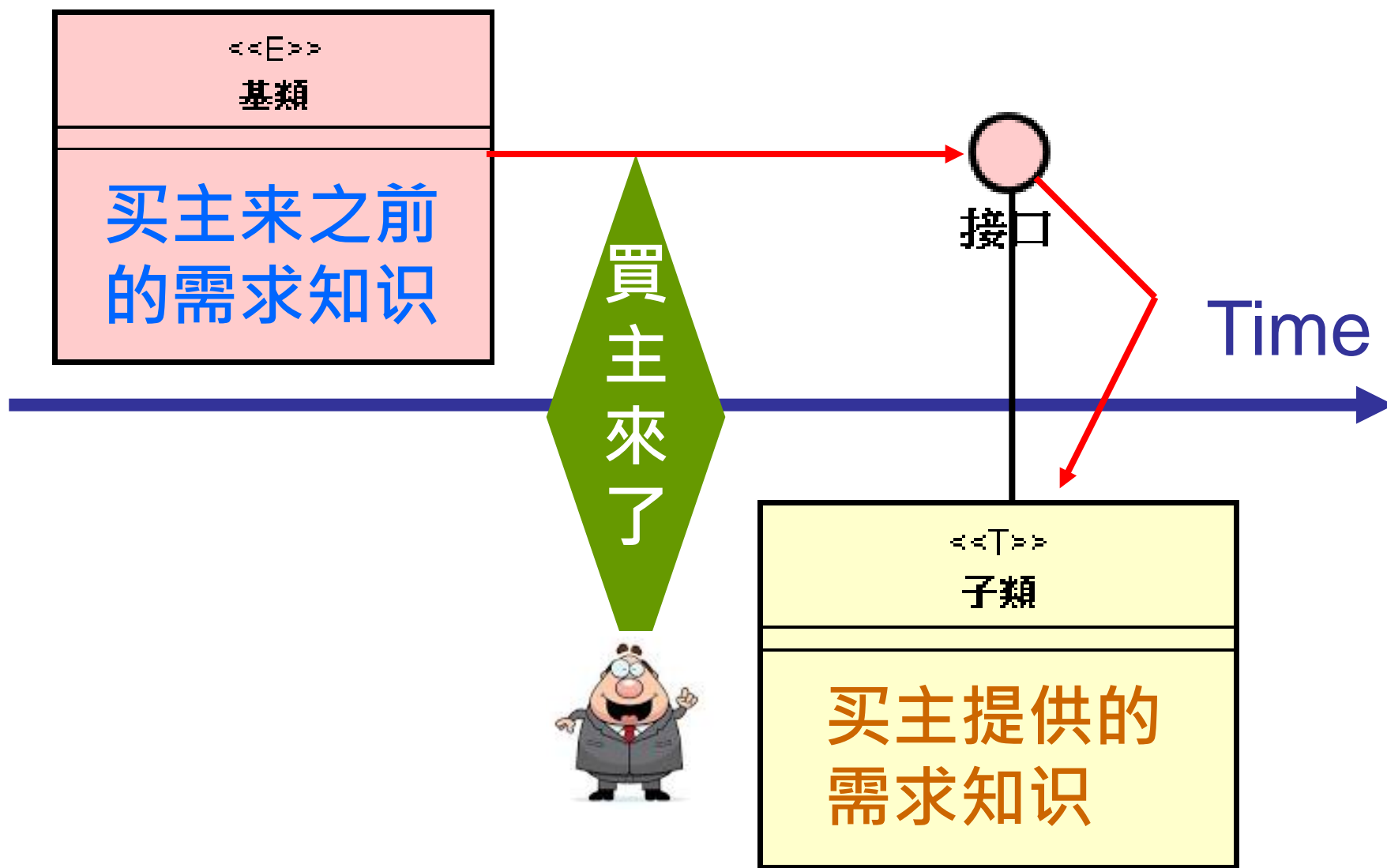
子類

买主提供的
需求知识

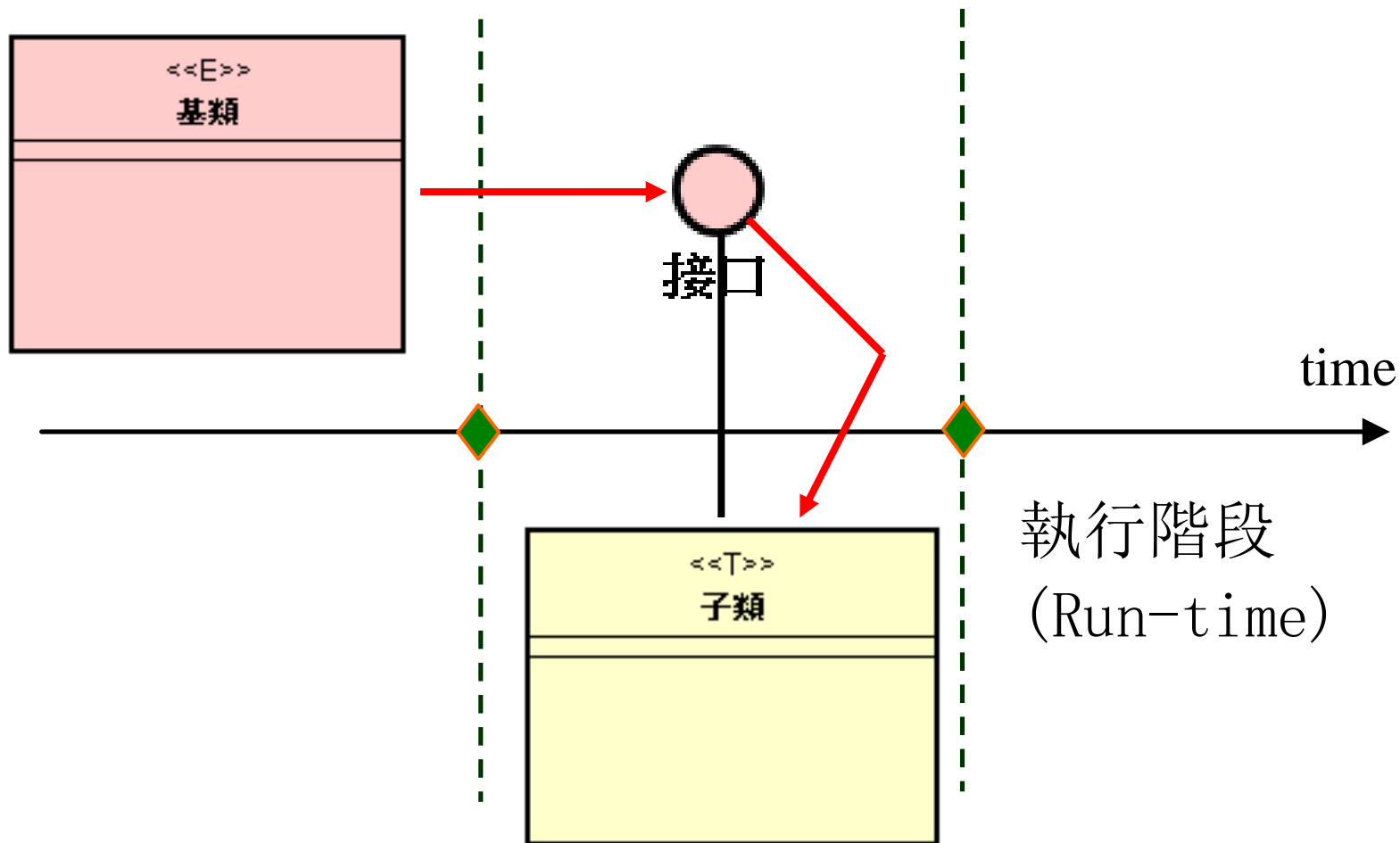
两种知识将必须汇合起来

- 设计接口 <I>，准备将基类与子类汇合起来
- 于是，就能将两种知识汇合起来了

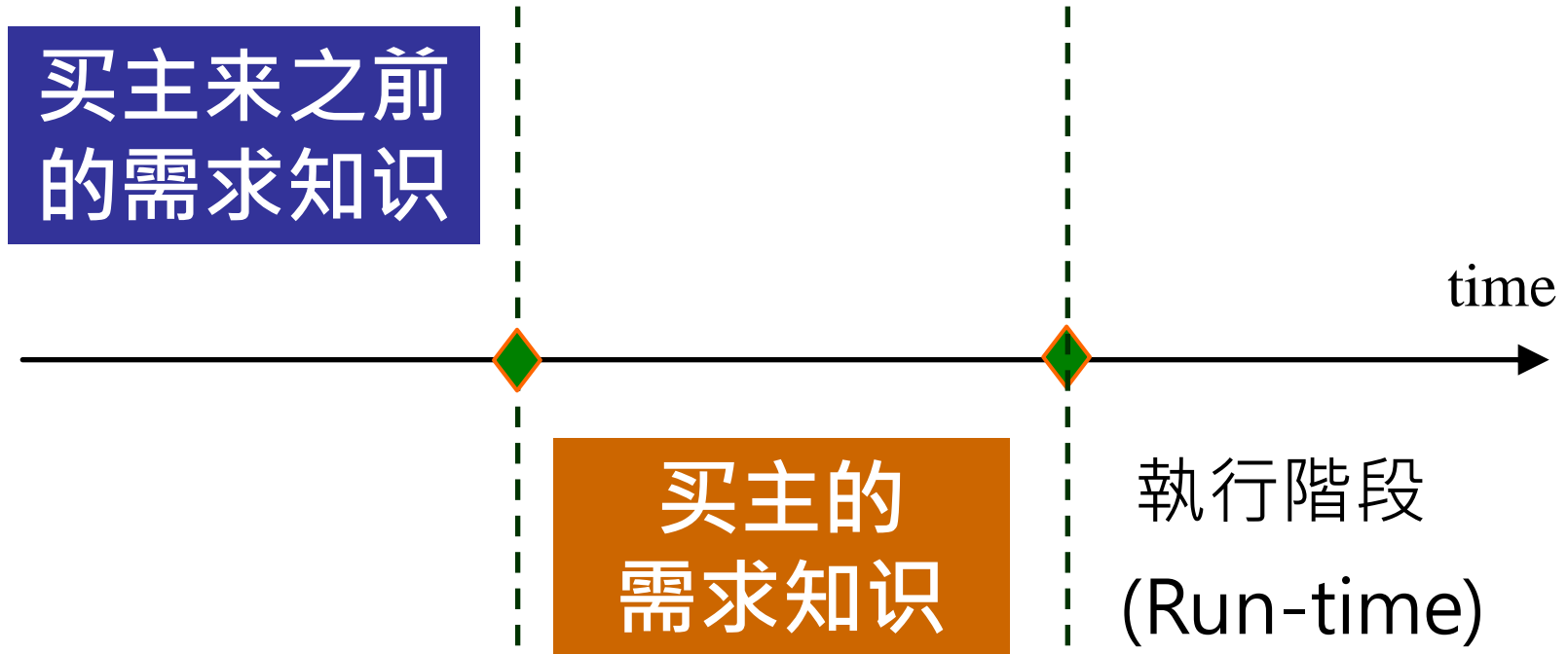




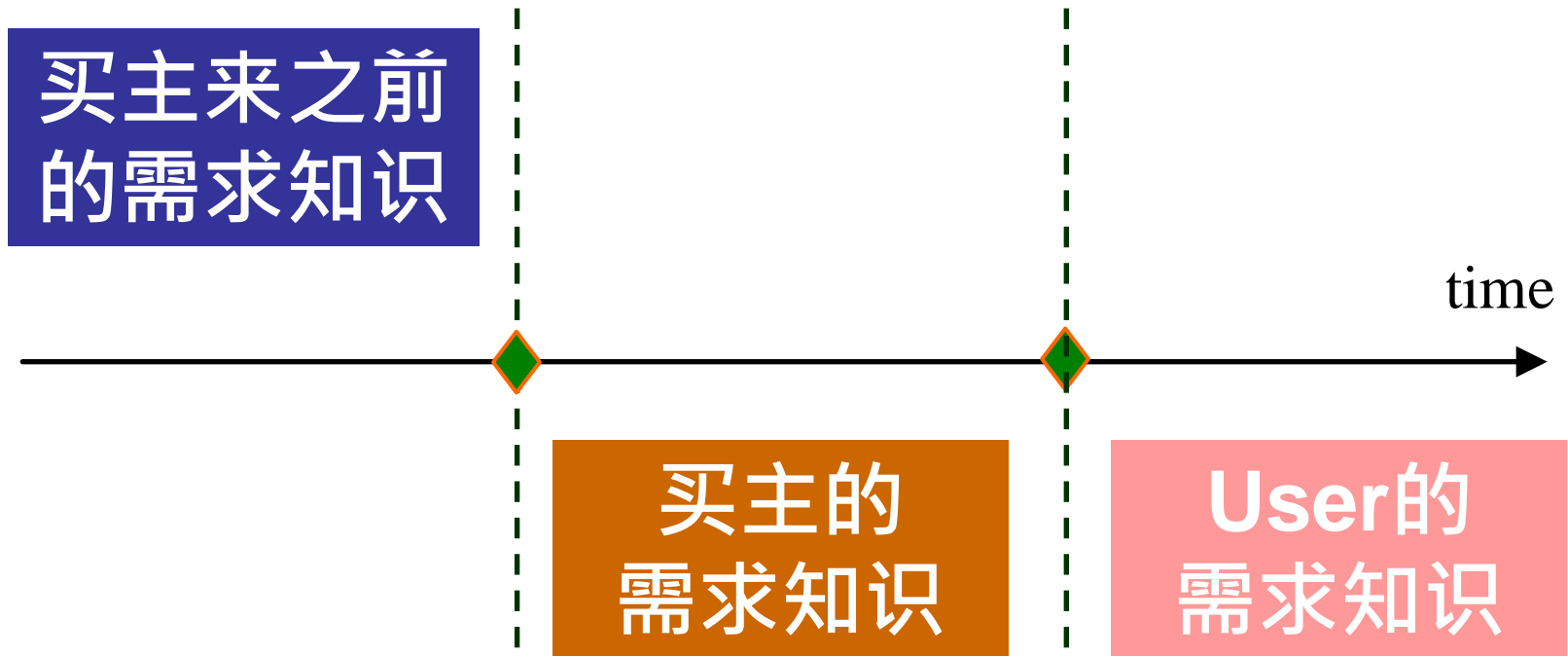
買主來了



買主來了

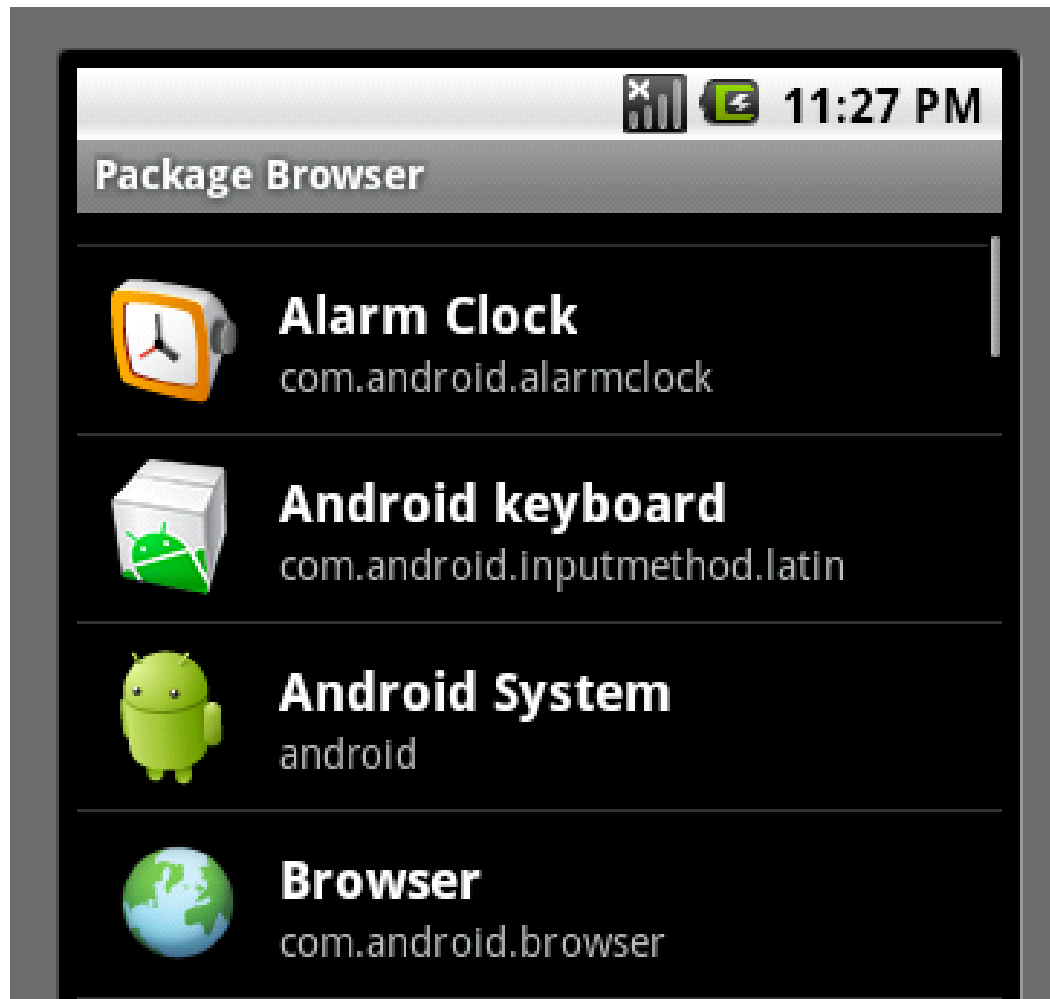


買主來了



2、ListView的接口设计

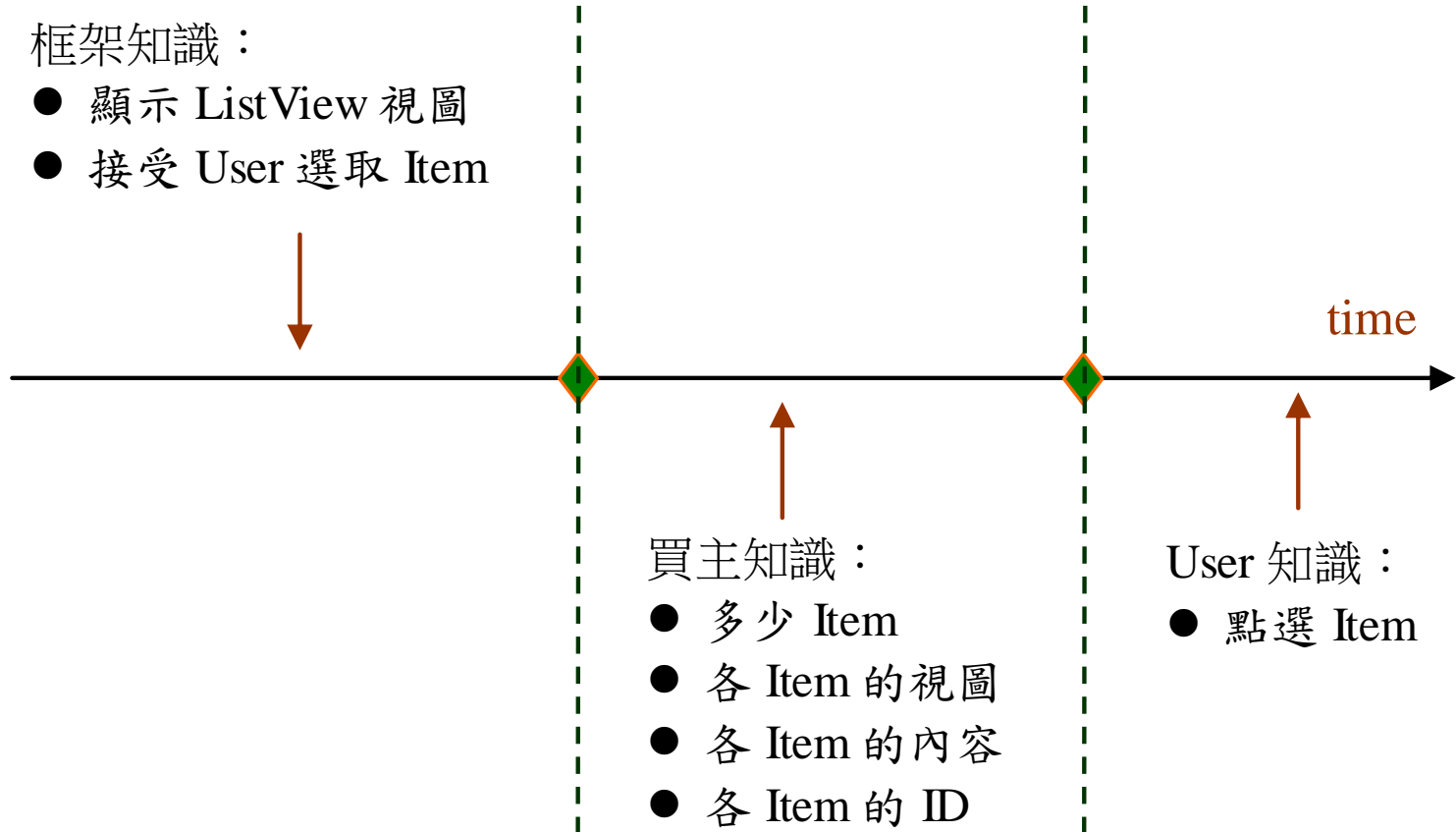
- 大家熟悉的ListView长相，如下：



- 当你看到这个面貌时，必须仔细想一想：买主来了，他会提出那些特殊的要求。
- 如同你正要开设一家新的餐馆，你必须想好如何规划点菜单，也就是点菜单上，你想让买主填写些甚么呢？
- 想好后，得出下述需求时间表：

框架知識：

- 顯示 ListView 視圖
- 接受 User 選取 Item



買主知識：

- 多少 Item
- 各 Item 的視圖
- 各 Item 的內容
- 各 Item 的 ID

User 知識：

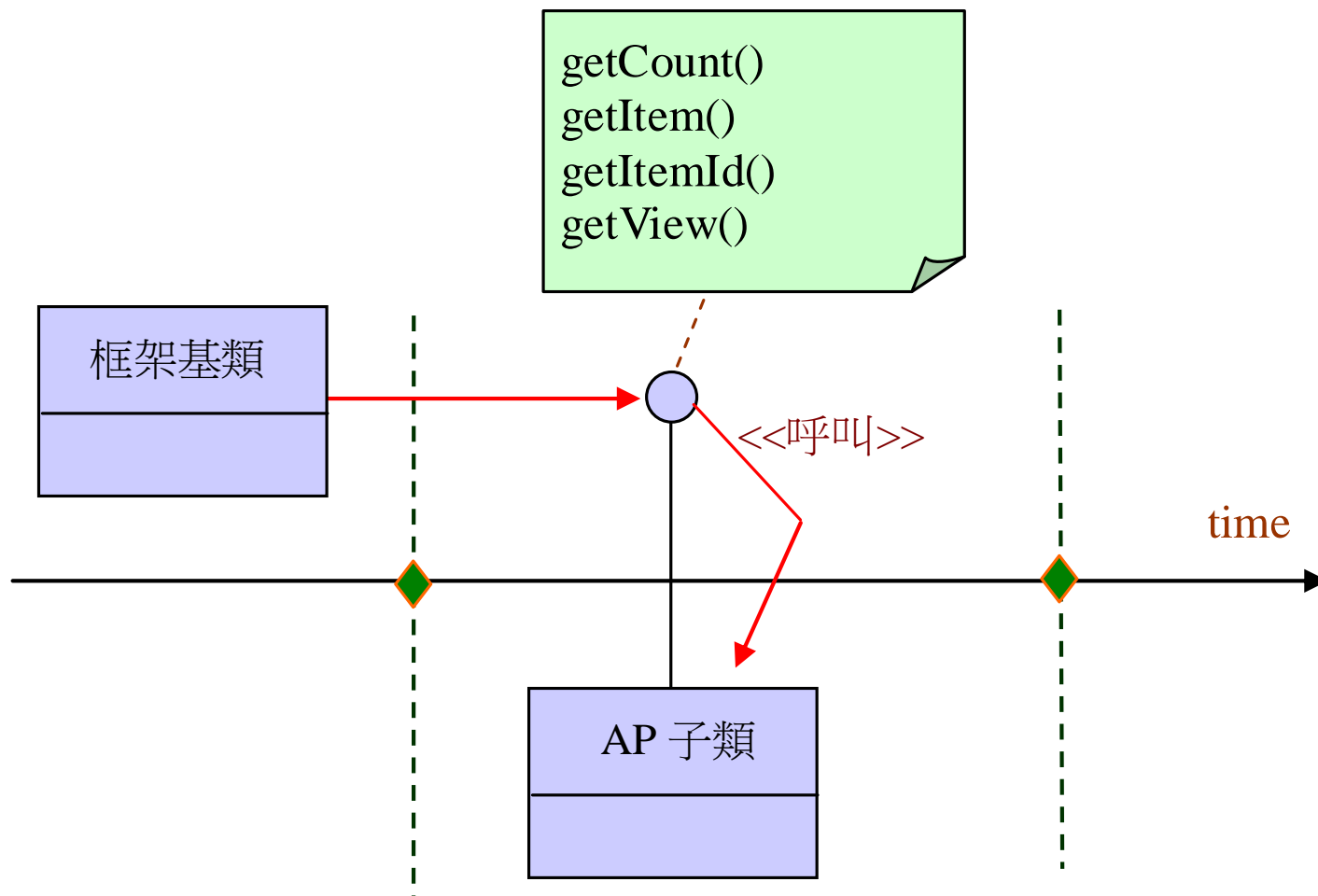
- 點選 Item

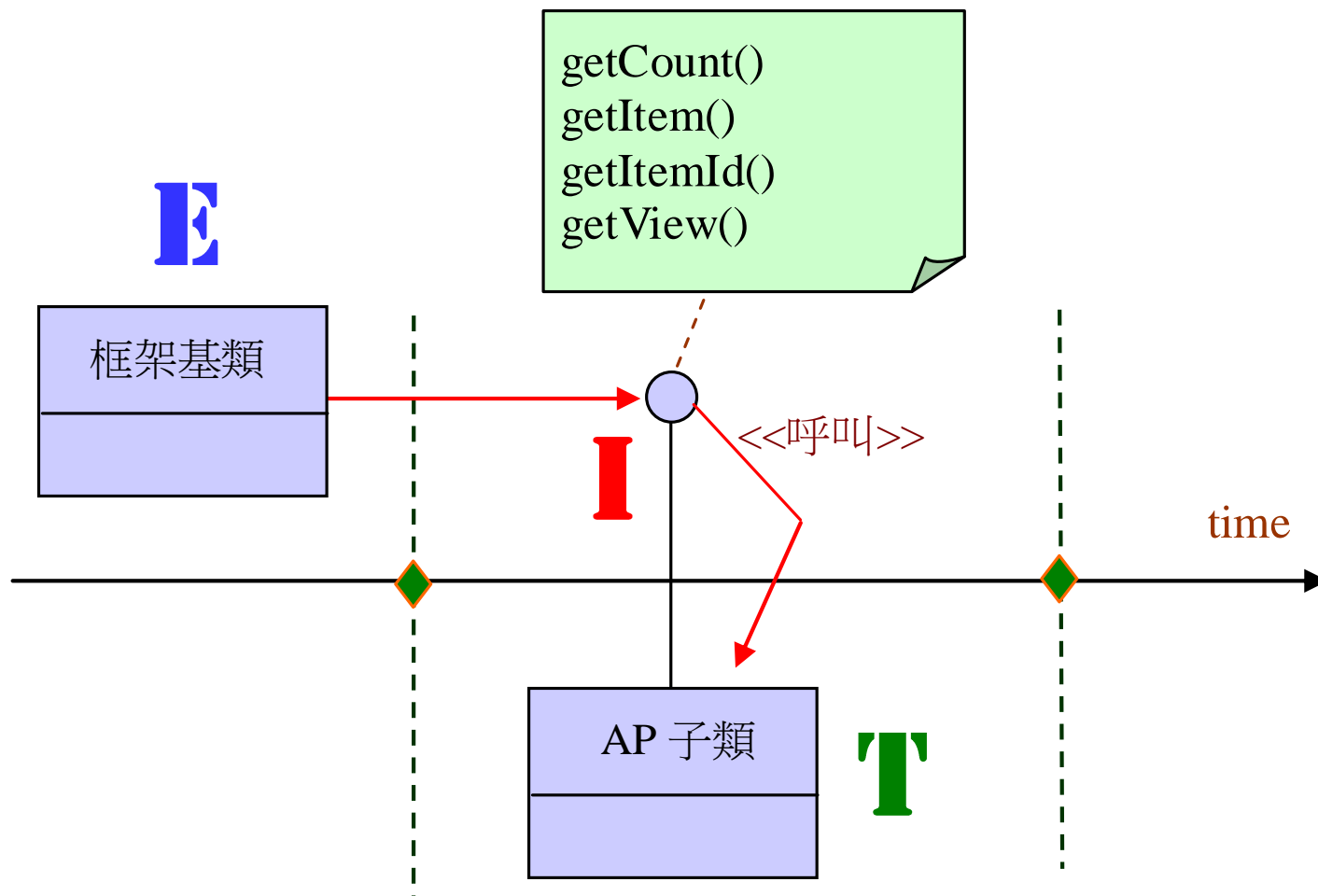
此图说明了，你必须设计
“点菜单”，让买主(Buyer)表达：

- ListView里想显示出多少行选项(Item)，让用户(User)于执行阶段来做选择。
- 每一行选项的视图(Item的图像)是甚么？
- User挑取选项，要响应给User甚么内函？
- User挑取选项，要响应给User甚么ID值？

接口设计

- 将上述买主知识点归于App子类，其余部分归于框架基类。
- 接着，设计接口函数，让基类能调用子类的函数，将子类里的买主知识(点菜单内容)汇集到框架基类(食谱)里。于是，得到接口设计如下：





总共分为两个阶段

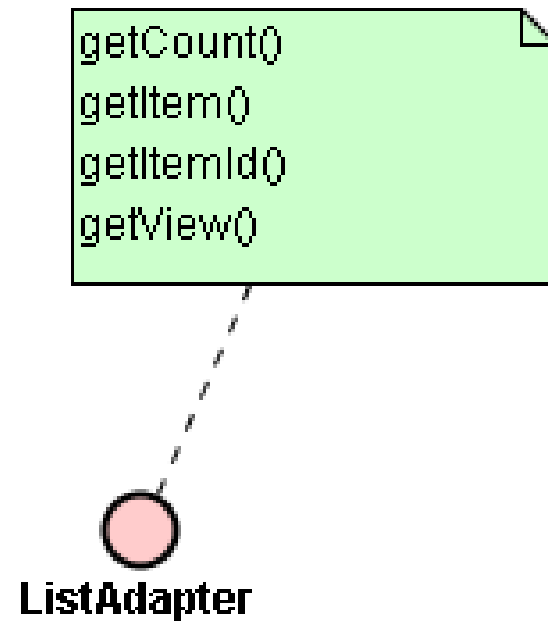
1. 视图显示阶段(User点选之前)

- <E> 透过接口的getCount()从子类里取得买主所做的决定：多少行选项(如N行)。
- 继续调用N次getView()，从子类里取得各行的视图。

2. 选项响应阶段(User点选之后)

- 基类可透过接口的getItem()函数从子类里取得买主所做的决定：响应什数据或对象。
- 也可调用getItemId()函数，从子类里取得各选项的ID值。

- 基于上述的初步规划，继续做细部接口<I>细部设计，包括上述函数的名称和参数、回传值的型态都得详细推敲，才能定案。
- 于是，得到详细的接口设计，如下图所示：



```
public int getCount();  
public T getItem( int position );  
public long getItemId( int position );  
public View getView( int position, View convertView,  
                    ViewGroup parent );  
// 其它
```

说明

- `<E>` 调用 `getCount()` 函数而取得买主心中的 `N` 值(多少个 `Item`)。
- 调用 `N` 次 `getView(position : int)` 函数，每次取得第 `position` 个 `Item` 的貌相，它就是该 `Item` 的代表视图(一个 `View` 对象)。

- 调用N次getItem()函数，每次回传买主所设定第position个Item的内涵(例如每次从Coll[]里取得第position个对象)。
- 调用N次getItemId()函数，每次回传买主所设定的第position个Item的ID值。



~ Continued ~