

编程入门基础

——内部类与匿名类

主讲教师：耿宇航



内部类

- 一个器件可能会包含**专用的**内部器件
- 内部类的作用相当于这些内部器件
- 内部类在某个类的内部定义
- 包含这个内部类的类称为它的外部类

使用内部类的理由

- 第一个理由
 - 避免名字的泛滥
 - 生成的.class模式为：外部类名\$内部类名
- 第二个理由
 - 内部类可以直接使用其外部类的私有变量
 - 避免调用内部类的时候，传递冗长的参数

内部类的特性

- 必须获得了外部类的实例，才能创建内部类的实例。
 - 内部类对象，依赖于外部类的对象。这好比一种宿主和寄生关系
- 在内部类的方法中，能拿到两个this指针，一个是本对象的指针，一个是它所寄生的外部类对象的指针

内部类内存

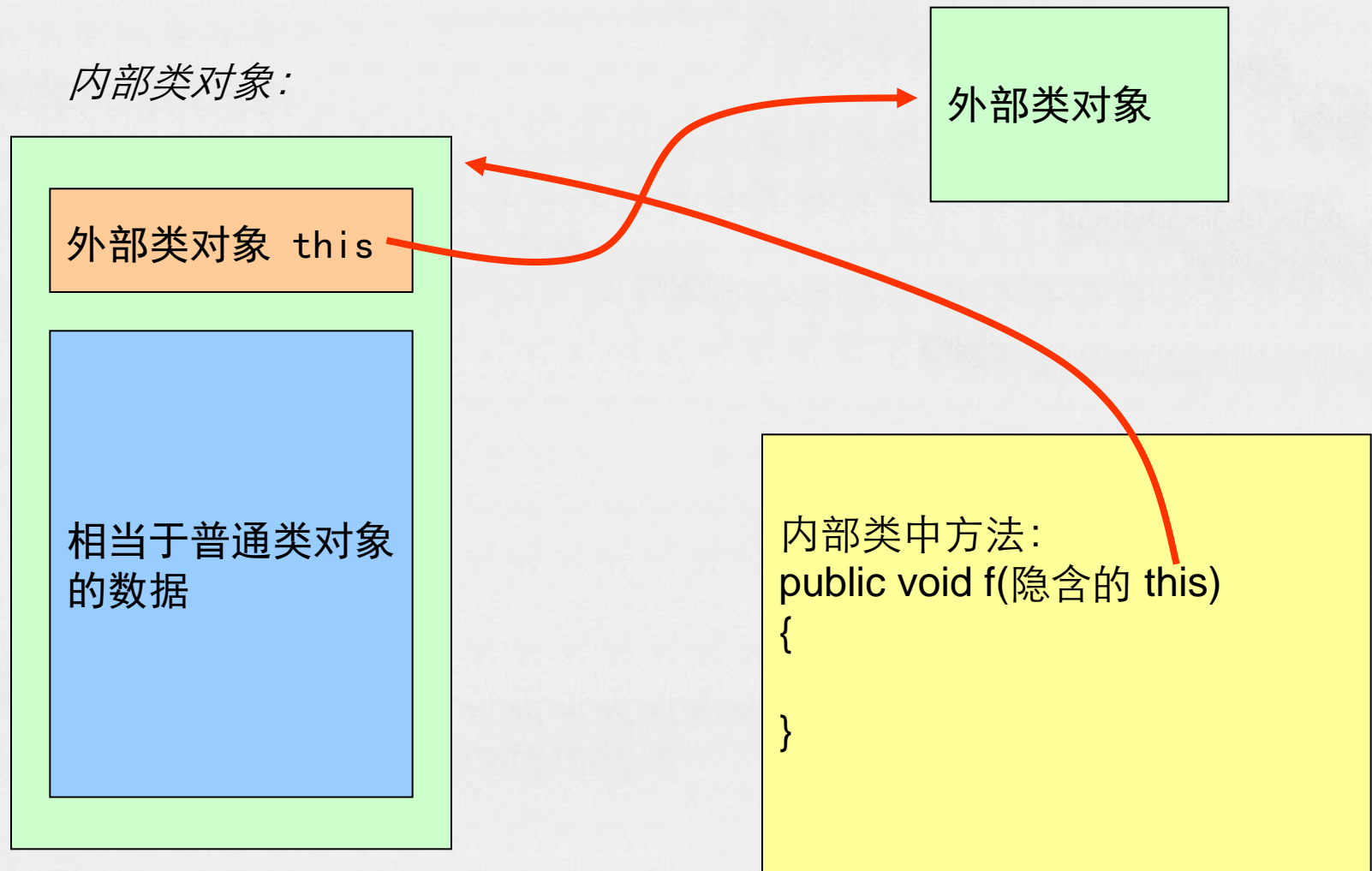
- 在创建内部类对象的时候，就拿到了外部类对象的指针

```
class 外部类  
{  
  class 内部类 { ... }  
  ...  
}
```

为什么会出现如此怪异的语法？
为什么说内部类对象有双**this**引用？

```
外部类 a = new 外部类();  
外部类.内部类 b = a.new 内部类();
```

内部类对象表示



内部类的语法

- 定义
- 在其寄生的外部类中的使用
- 在其它类中的使用

匿名对象

- 如果某个对象创建后，只调用一次方法，就可以不为它起名字

匿名类

- 如果对现存的某个类的行为，只是做轻微的、临时的修正，就不必为这个类命名，而是修正后直接创建对象了事。

谢 谢 ！