**ｔｈｉｓ的使用：**

1. 可以調用的結構：屬性、方法；構造器

２．this調用屬性、方法：  
this理解為：當前對象 或 當前正在創建的對象  
2.1 在類的方法中，我們可以使用"this.屬性"或"this.方法"的方式，調用當前對象屬性或方　　　　  
法。但是，通常情況下，我們都選擇省略"this."。特殊情況下，如果方法的形參和類的屬性同名時，我們必須顯式的使用"this.變量"的方式，表明此變量是屬性，而非形參。

2.2 在類的構造器中，我們可以使用"this.屬性"或"this.方法"的方式，調用當前正在創建的對象屬性或方法。但是，通常情況下，我們都選擇省略"this."。特殊情況下，如果構造器的形參和類的屬性同名時，我們必須顯式

的使用"this.變量"的方式，表明此變量是屬性，而非形參。

３．this調用構造器  
1. 我們在類的構造器中，可以顯式的使用"this(形參列表)"方式，調用本類中指定的其他構造器

2. 構造器中不能通過"this(形參列表)"方式調用自己

3. 如果一個類中有n個構造器，則最多有 n - 1構造器中使用了"this(形參列表)"

4. 規定："this(形參列表)"必須聲明在當前構造器的首行

5. 構造器內部，最多只能聲明一個"this(形參列表)"，用來調用其他的構造器

**ｐａｃｋａｇｅ的使用：**

１．使用說明：  
1.為了更好的實現項目中類的管理，提供包的概念

2.使用package聲明類或接口所屬的包，聲明在源文件的首行

3.包，屬於標識符，遵循標識符的命名規則、規範(xxxyyyzzz)、“見名知意”

4.每"."一次，就代表一層文件目錄。

補充：同一個包下，不能命名同名的接口、類。

不同的包下，可以命名同名的接口、類。

２．舉例：



舉例二：ＭＶＣ設計模式



1.3 JDK的主要包介紹：



**ｉｍｐｏｒｔ的使用：**

import:導入

1. 在源文件中顯式的使用import結構導入指定包下的類、接口

2. 聲明在包的聲明和類的聲明之間

3. 如果需要導入多個結構，則並列寫出即可

4. 可以使用"xxx.\*"的方式，表示可以導入xxx包下的所有結構

5. 如果使用的類或接口是java.lang包下定義的，則可以省略import結構

6. 如果使用的類或接口是本包下定義的，則可以省略import結構

7. 如果在源文件中，使用了不同包下的同名的類，則必須至少有一個類需要以全類名的方式顯示。

8. 使用"xxx.\*"方式表明可以調用xxx包下的所有結構。但是如果使用的是xxx子包下的結構，則仍需要顯式導入

9. import static:導入指定類或接口中的靜態結構:屬性或方法。