



# 設計範式 (Design Patterns) in Java Library

侯捷

資訊教育, 技術傳播, 大學授課



Template Method  
Iterator  
Composite  
Decorator  
Observer  
Listener  
Adapter  
Factory  
Singleton  
Undoable

## Template Method

- Definition in "GoF"
- UML Class Diagram
- Template Method in Java Lib.

# Template Method

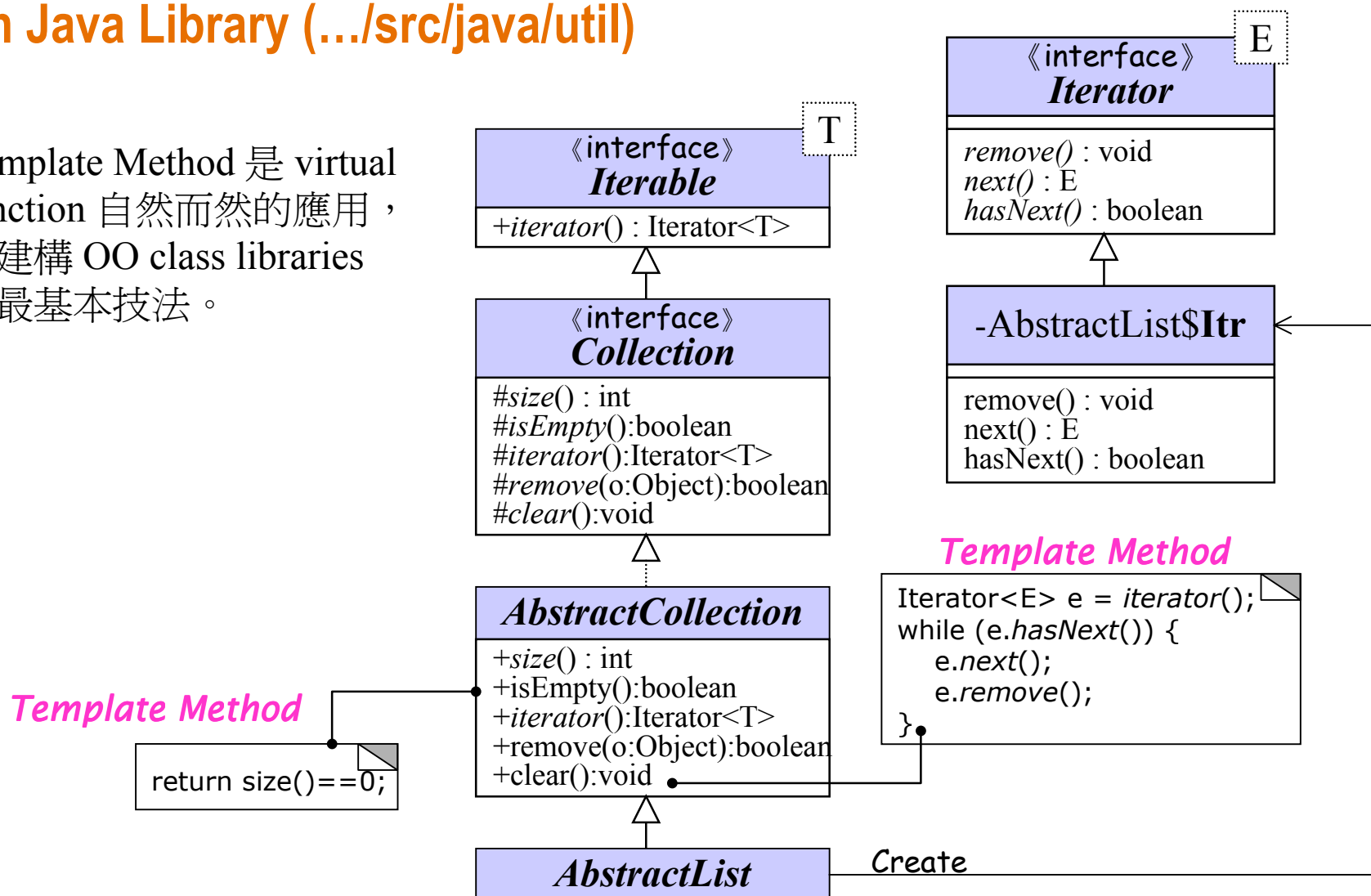
## Definition in "GoF"

- Define the skeleton of an algorithm in an operation, deferring some steps to subclasses. Template Method lets subclasses redefine certain steps of an algorithm without changing the algorithm's structure.
- 定義演算法骨幹，延緩其中某些步驟，使它們在 subclasses 中執行。Template Method 使 subclasses 得以重新定義演算法內的某些動作而不需改變演算法的總體結構。

# Template Method

in Java Library (.../src/java/util)

Template Method 是 virtual function 自然而然的應用，是建構 OO class libraries 的最基本技法。



# Iterator

- Definition in "GoF"
- UML Class Diagram
- Iterator in Java Lib.

# Iterator

## Definition in "GoF"

- Provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation.
- 提供一種巡訪「聚合物內各元素」的通用介面，並且無需曝露聚合物的底層表述（內部細節）。

# 使用 Iterator

用法：

Generics Java 只允許在 collection 內放置 class object，不允許放置 primitive type data. 所以不得寫為：LinkedList<int> il=new...;

```
LinkedList<Integer> il = new LinkedList<Integer>();
il.add(new Integer(0));
il.add(new Integer(1));
il.add(new Integer(5));
il.add(new Integer(2));

Iterator ite = il.iterator();
while(ite.hasNext()) {
    System.out.println(ite.next());
}
```

自從JDK5.0提供 box/unbox 之後，  
可寫為：

```
il.add(0);
il.add(1);
il.add(5);
il.add(2);
```

每一種 Java collection 都有個 iterator()

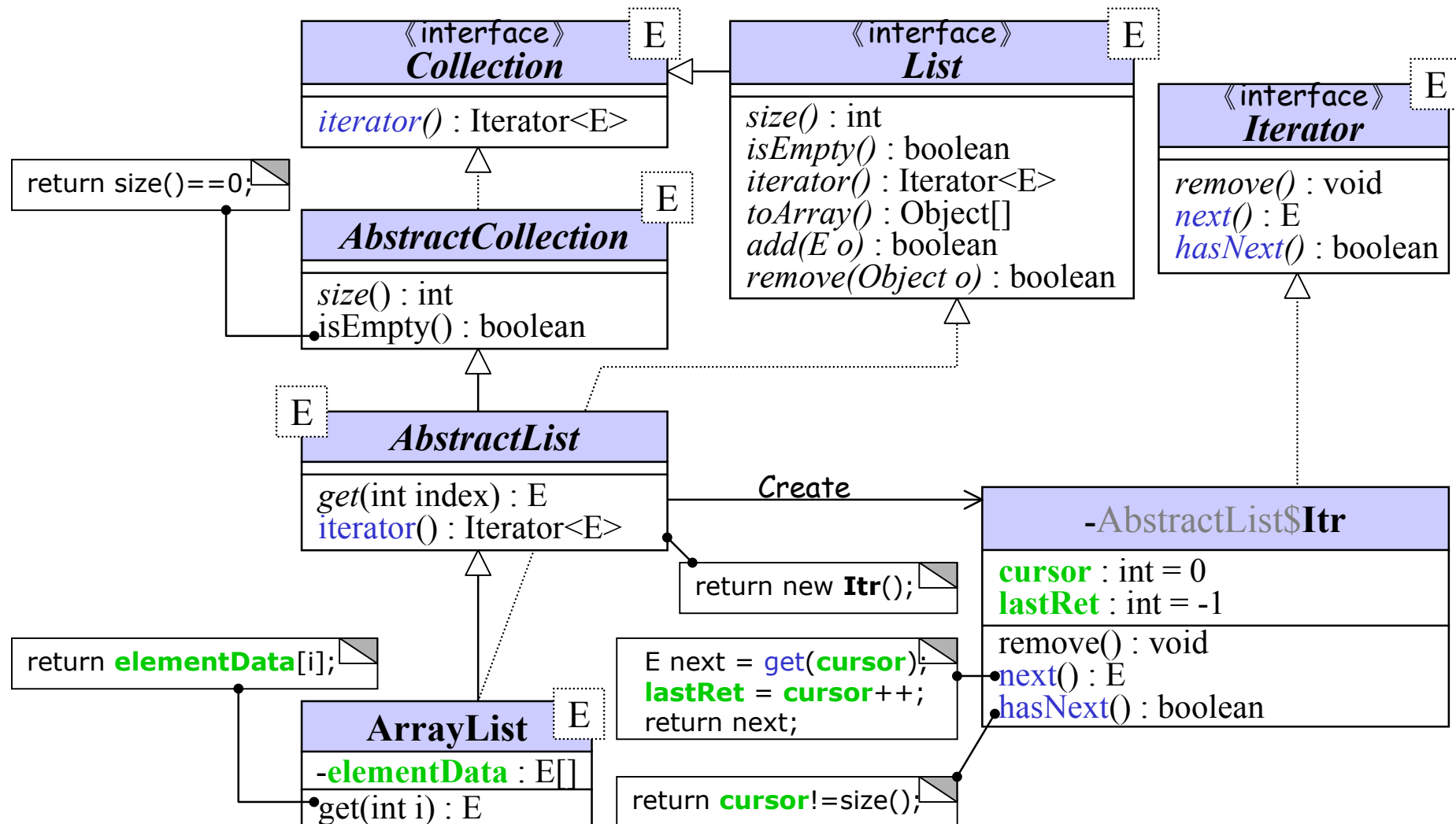
每一個 iterator 都有這樣的 methods.



# ArrayList's Iterator

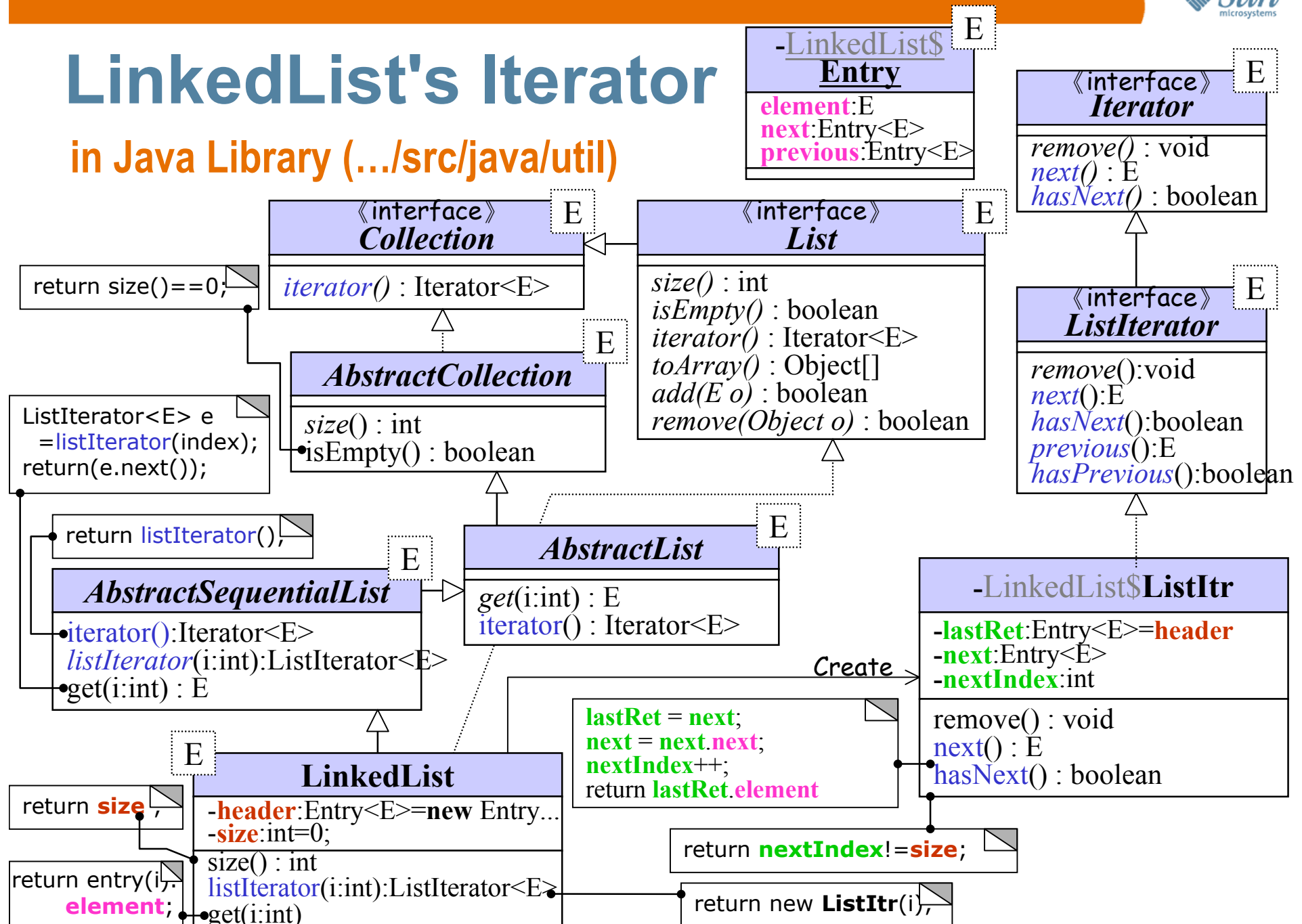
in Java Library (.../src/java/util)

附帶補充：Java Library 內的 *AbstractXXX* 都是爲了局部（分層）實作。



# LinkedList's Iterator

in Java Library (.../src/java/util)



# Composite

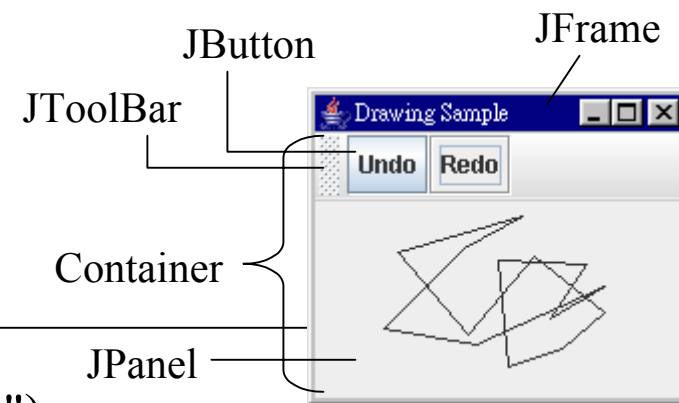
- Definition in "GoF"
- UML Class Diagram
- Composite in Java Lib.

# Composite

## Definition in "GoF"

- Compose objects into tree structures to represent part-whole hierarchies. Composite lets clients treat individual objects and compositions of objects uniformly.
- 將 **objects** 組成爲樹狀結構，用以表示 "局部-全部" 階層體系。讓 **clients** 以一致的方式對待「個別物件」和「合成物件」。

# 使用 Composite



```
public static void main(String args[]) {
    JFrame frame = new JFrame("Drawing Sample");
    ...
    UndoableDrawingPanel drawingPanel = new UndoableDrawingPanel();
    ...

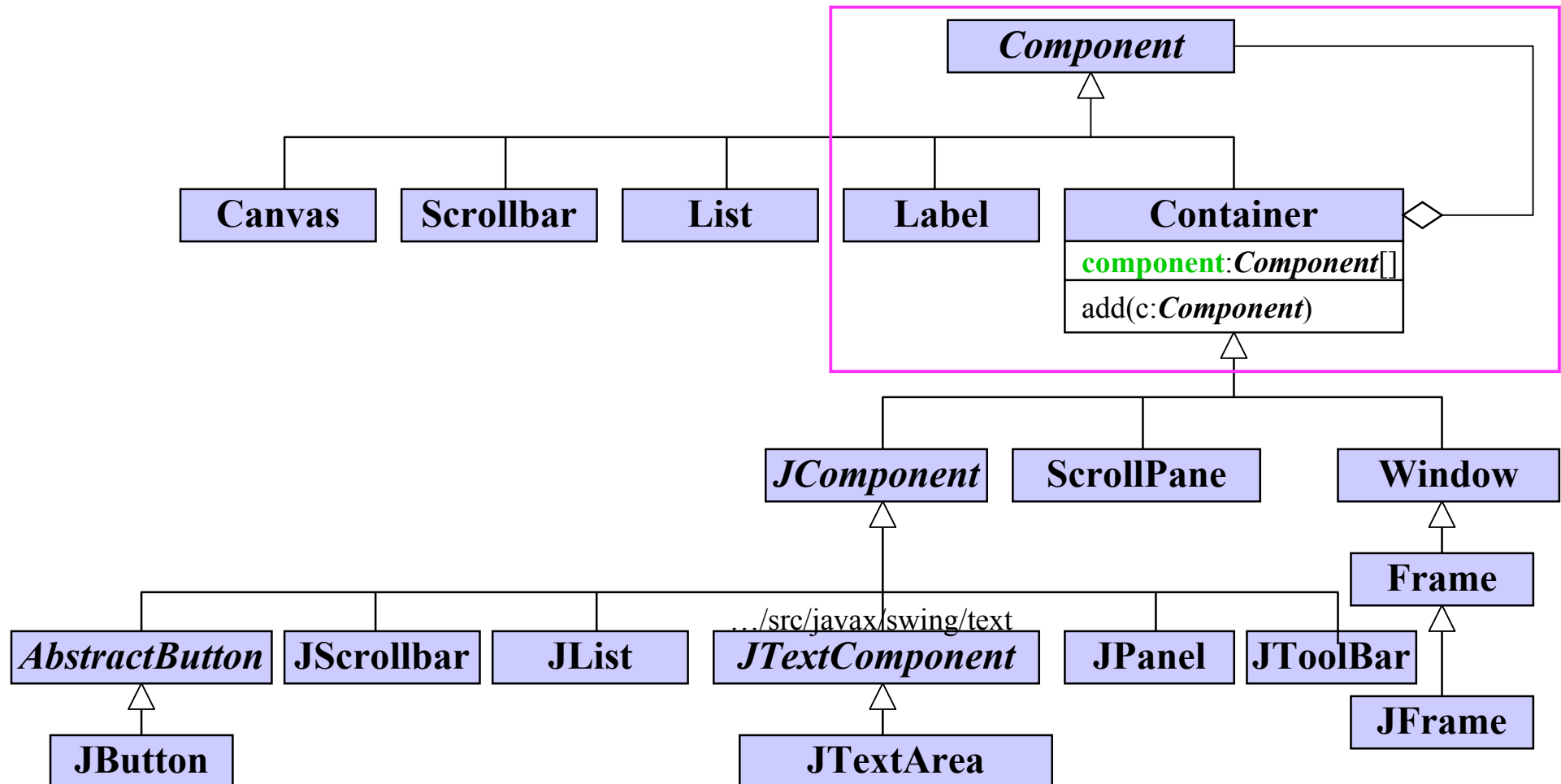
    JToolBar toolbar = new JToolBar();
    toolbar.add(...); //Adds a new JButton which dispatches the action
    toolbar.add(...);

    Container content = frame.getContentPane();
    content.add(toolbar, ...);
    content.add(drawingPanel, ...);
    ...
}
```

# Composite

in Java Library (.../src/java/awt, .../src/javax/swing)

Composite



# Decorator

- Definition in "GoF"
- UML Class Diagram
- Decorator in Java Lib.

# Decorator

## Definition in "GoF"

- Attach additional responsibilities to an object dynamically. Decorators provide a flexible alternative to subclassing for extending functionality.
- 將額外的「權與責」以動態方式附著於 object 身上，使不必進行 subclassing 就能擴展功能。



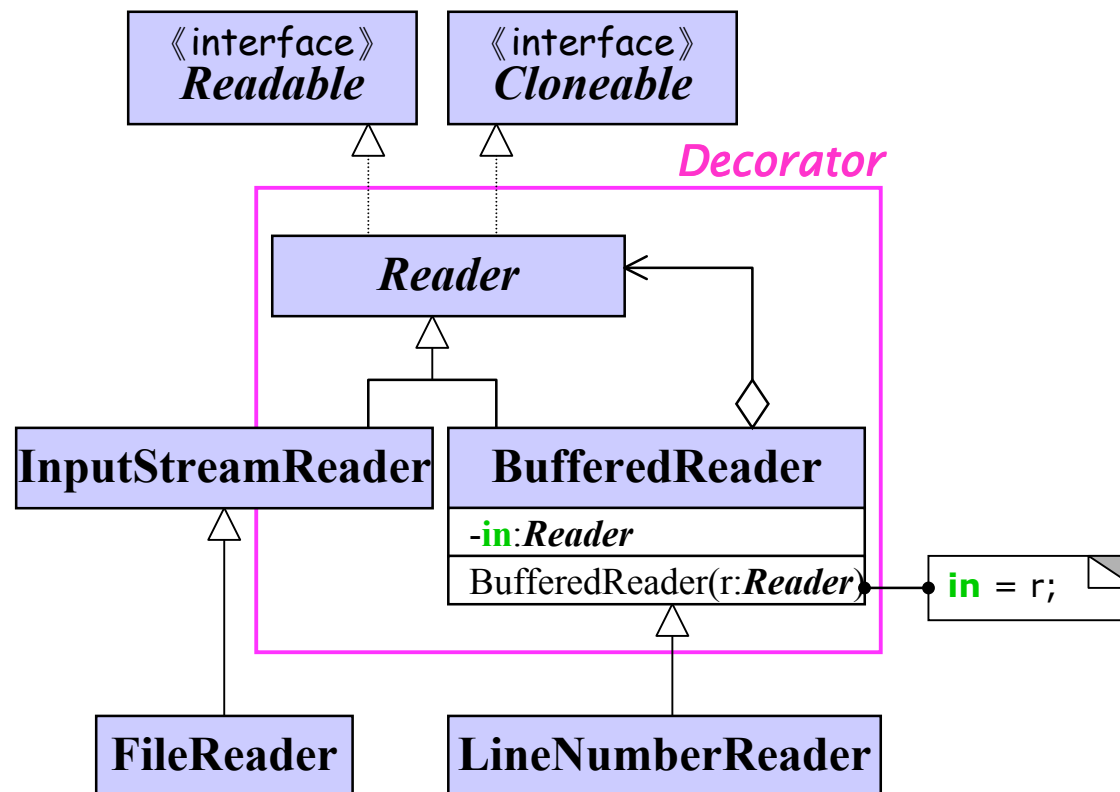
# 使用 **Decorator**

```
// Reading input by lines:  
BufferedReader in =  
    new BufferedReader(  
        new FileReader("IODemo.java"));  
String s, s2 = new String();  
while((s = in.readLine()) != null)  
    s2 += s + "\n";  
in.close();
```

```
// Reading standard input:  
BufferedReader stdin =  
    new BufferedReader(  
        new InputStreamReader(System.in));  
System.out.print("Enter a line:");  
System.out.println(stdin.readLine());
```

# Decorator

in Java Library (.../src/java/io)

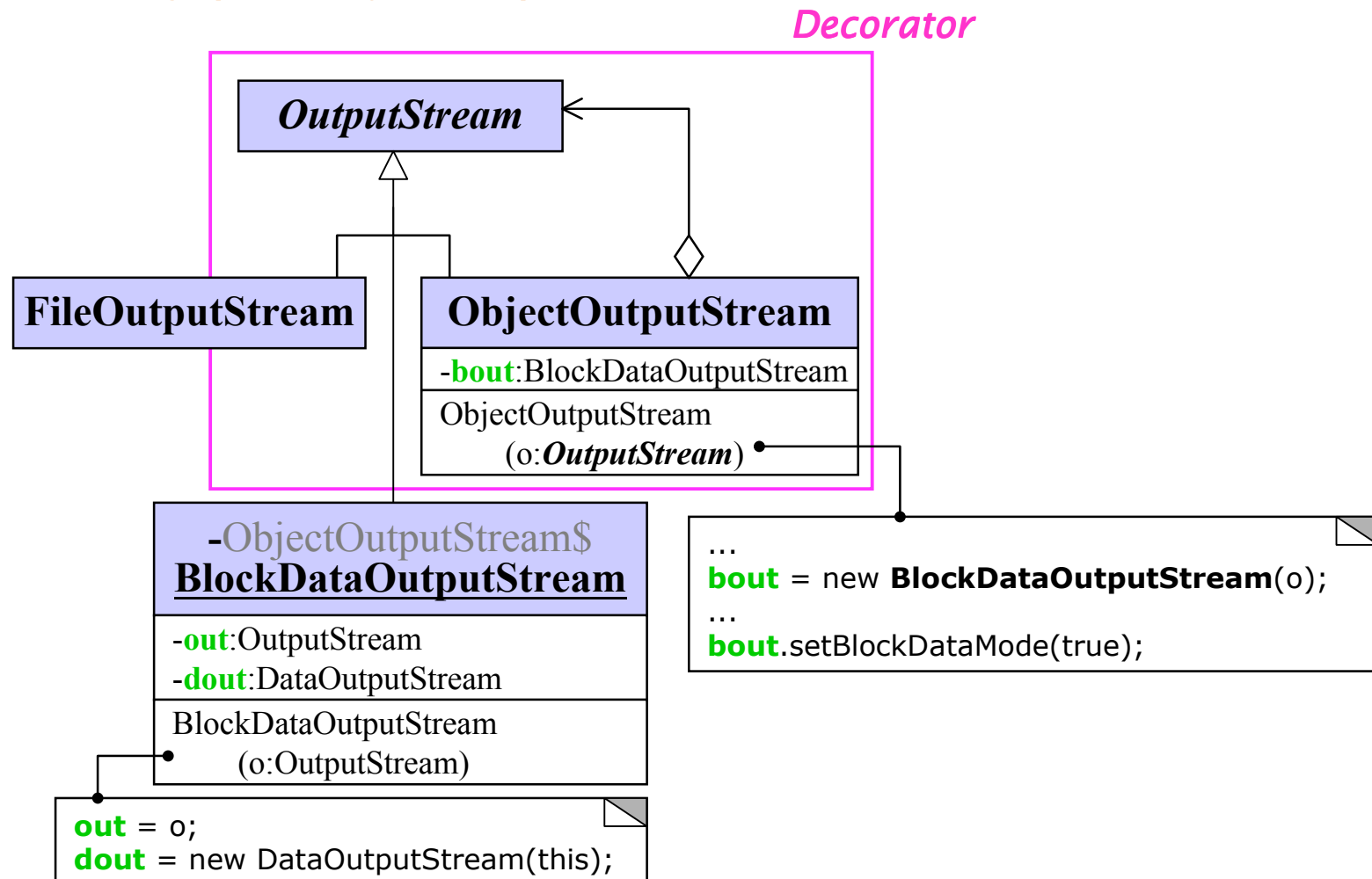


# 使用 Decorator

```
public class Worm implements Serializable { ... }  
Worm w = new Worm(6, 'a');  
ObjectOutputStream out =  
    new ObjectOutputStream(  
        new FileOutputStream("worm.out"));  
out.writeObject("Worm storage");  
out.writeObject(w);  
out.close(); // Also flushes output
```

# Decorator

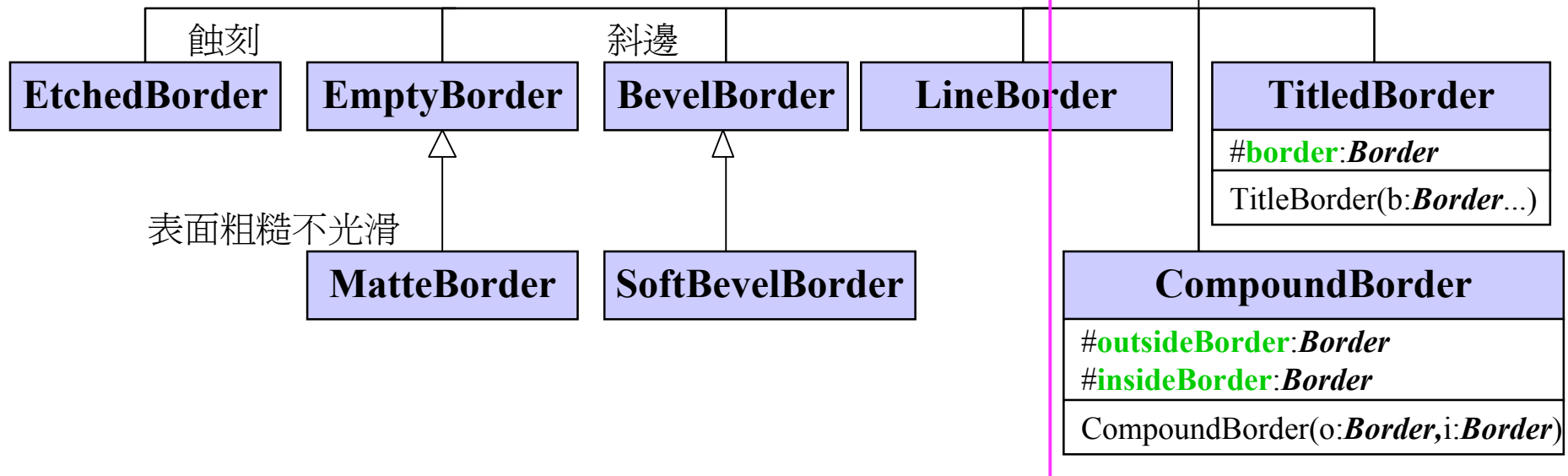
in Java Library (.../src/java/io)



# Decorator

in Java Library (.../src/javaw/swing/border)

<http://java.sun.com/docs/books/tutorial/uiswing/misc/border.html>,  
"How to Use Borders"



## Observer

- Definition in "GoF"
- UML Class Diagram
- Observer in Java Lib.

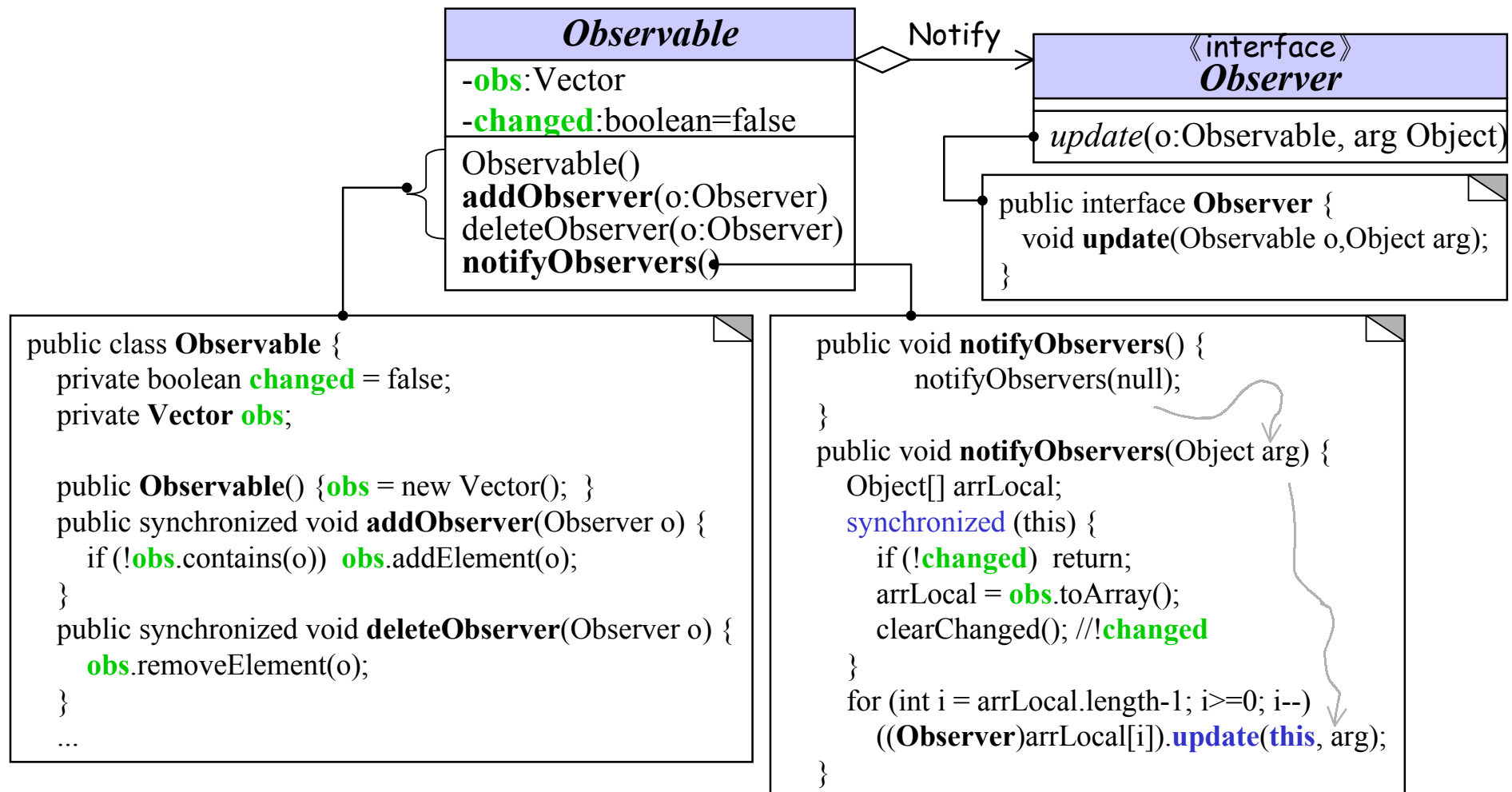
# Observer

## Definition in "GoF"

- Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.
- 在 objects 之間定義 "一對多" 依存性，使得當 object 改變狀態時，它所依存的所有 objects 都會獲得通知並自動更新。
  - > Observer 是被動地被通知，而不是主動地觀察，所以另一個名稱 publish-subscribe 也許比較更合適些。

# Observer

in Java Library (.../src/java/util)





# 使用 Observer+Observable

```
import java.util.Observable;
import java.util.Observer;

// A simple demo of Observable->Observer
public class ObservDemo extends Object {
    MyView view;
    MyModel model;

    public ObservDemo() {
        view = new MyView();
        model = new MyModel();
        model.addObserver(view);
    }

    public static void main(String[] av) {
        ObservDemo me = new ObservDemo();
        me.demo();
    }

    public void demo() {
        model.changeSomething();
    }
}
```

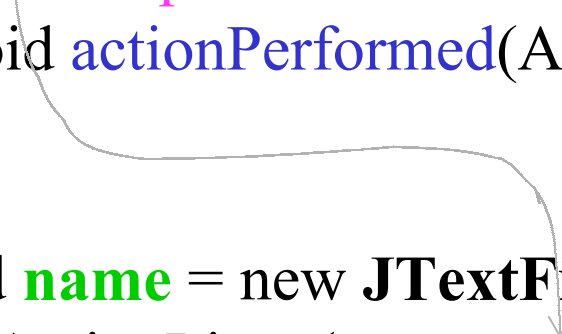
```
E:\handout>java ObservDemo
update(ObservDemo$MyModel@1cd2e5f,null);
```

```
/** The Observer normally maintains a view on the data */
class MyView implements Observer {
    /** For now, we just print the fact that we got notified. */
    public void update(Observable obs, Object x) {
        System.out.println("update(" + obs + ", " + x + ");");
    }
}

/** The Observable normally maintains the data */
class MyModel extends Observable {
    public void changeSomething() {
        // Notify observers of change
        setChanged();
        notifyObservers();
    }
}
}
```

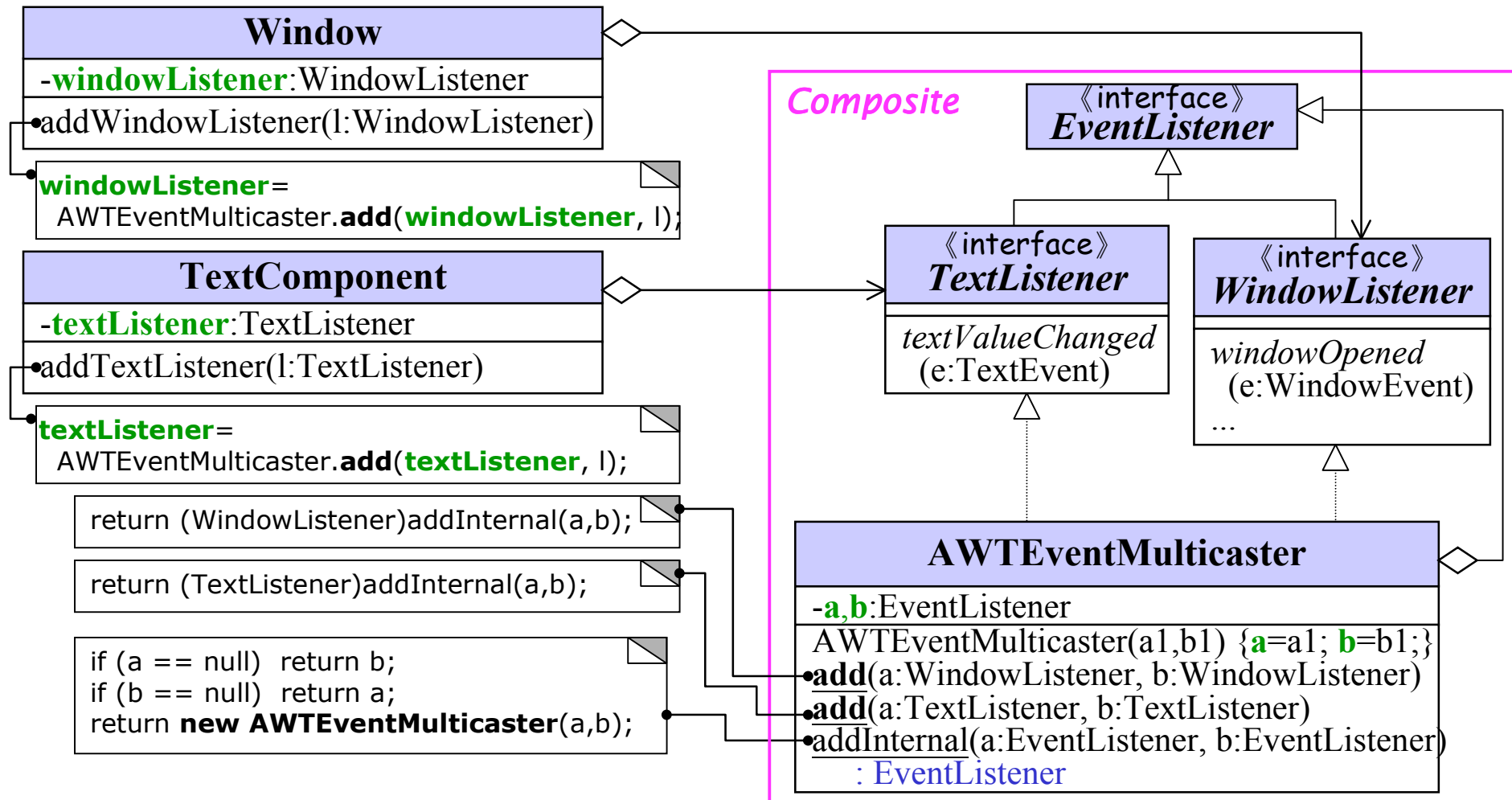
# 使用 Listener

```
class NameL implements ActionListener { //監聽器
    public void actionPerformed(ActionEvent e) { ... }
}
...
JTextField name = new JTextField(25);
name.addActionListener(new NameL()); //產生 listener object 並註冊
```



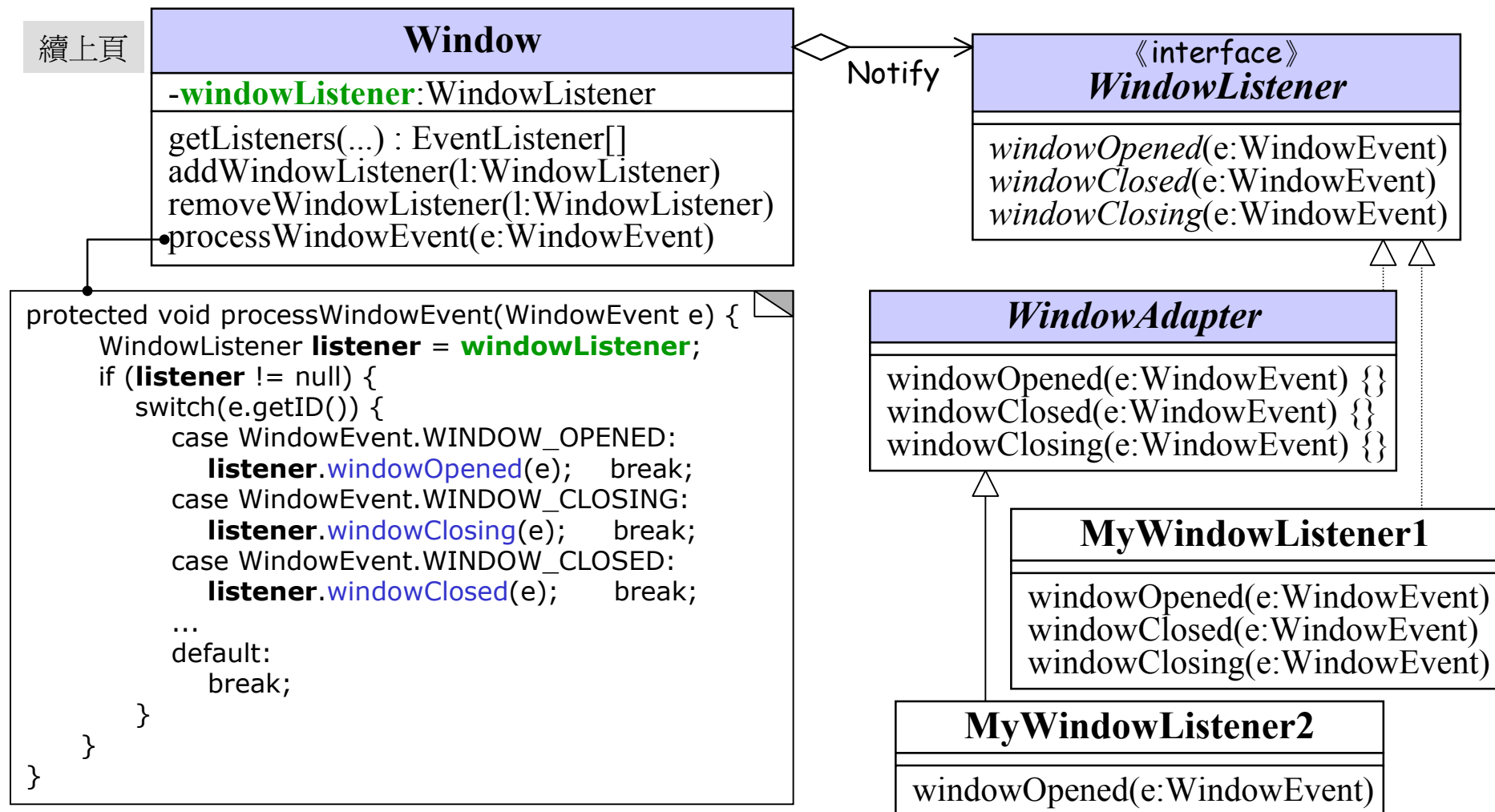
# Listener

in Java Library (.../src/java/awt, .../src/java/awt/event)



# Listener

in Java Library (.../src/java/awt, .../src/java/awt/event)



## Adapter (for Listener)

- Definition in "GoF"
- UML Class Diagram
- Adapter in Java Lib.

# Adapter

## Definition in "GoF"

- Convert the interface of a class into another interface clients expect. Adapter lets classes work together that couldn't otherwise because of incompatible interfaces.
- 轉換 class 的介面使其為 client 所期望。Adapter 使得原本因「介面不相容」而無法合作的 classes 變得可以合作。

# Adapter

## in Java Library

- Java Lib. 提供所謂的 listener。每一個 Swing component 都有兩個函式：addXXXListener() 和 removeXXXListener()，其中 XXX 代表 event 種類。這樣一來 App. 就可以藉由植入 listener 而監聽 component 發生什麼事。每個 listener 都是一個 object，其 class 必須實現特定之 interface。

# Adapter

## in Java Library

- Listener interface（例如 **WindowListener**）若有多個 methods，listener class 必須全部實作出來。這有時候形成困擾。因此某些 listener interface 搭配了所謂的 adapter，爲其對應之 listener interface 的每一個 methods 實現出空函式。Java Adapter classes 的中心思想是要簡化 listener class 的撰寫工程。



# 使用 Adapter

使用 Listener Class :

```
class NameL implements ActionListener { //監聽器
    public void actionPerformed(ActionEvent e) { ... }
}
...
JTextField name = new JTextField(25);
name.addActionListener(new NameL()); //產生 listener object 並註冊
```

本來應該 implements WindowListener，改爲 extends WindowAdapter

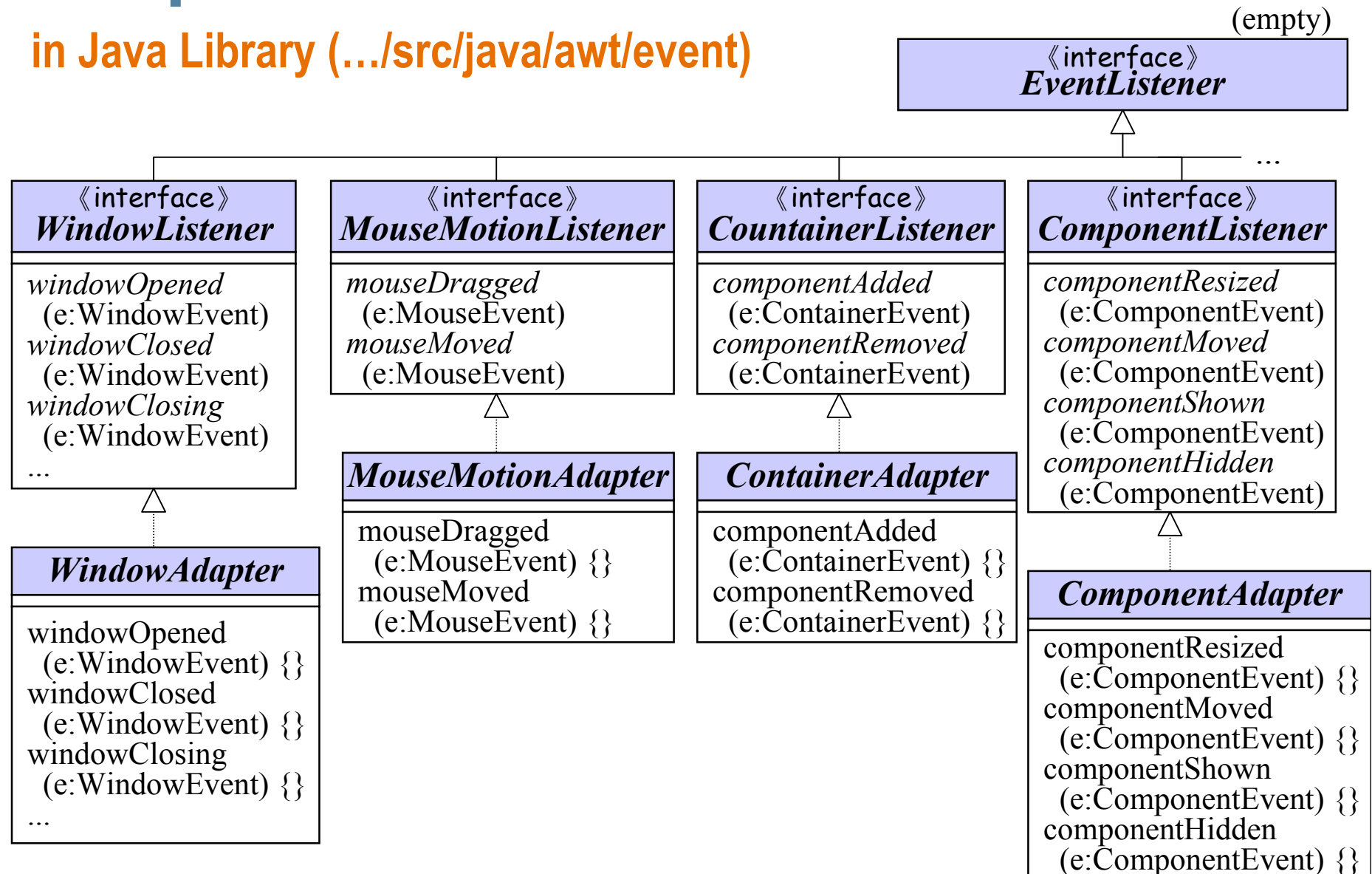
使用 Adapter Class :

```
class MyWindowListener
    extends WindowAdapter { //listener adapter
    public void windowClosing(WindowEvent e) { ... }
}
```

注意：萬一寫成 WindowClosing() 就麻煩了！編譯器不報錯。

# Adapter

in Java Library (.../src/java/awt/event)



## Object Factory

- Definition in "GoF"
- UML Class Diagram
- Object Factory in Java Lib.

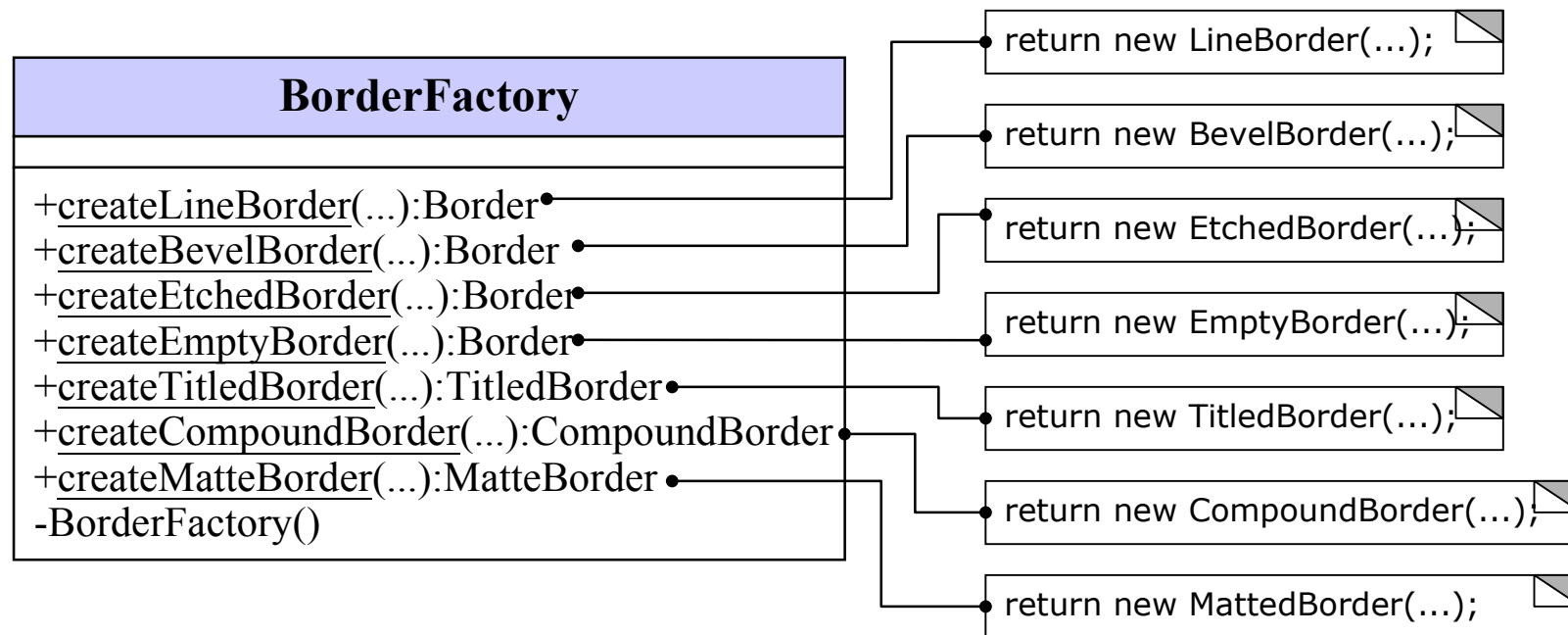
# Object Factory

## Definition in "GoF"

- Define an interface for creating an object, but let subclasses decide which class to instantiate. Factory Method lets a class defer instantiation to subclasses.
- 定義一個用來 creating object 的介面，但讓 subclasses 決定最終要具現出哪一種 object。Factory Method 使 class 得將具現行為延緩至 subclasses 再進行。

# Border Factory(?!)

in Java Library (.../src/javaw/swing/border)



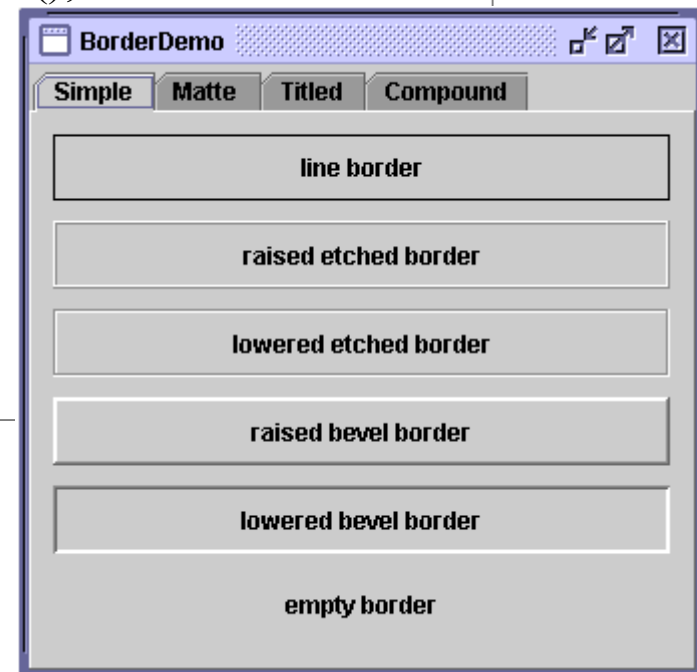
<http://java.sun.com/docs/books/tutorial/uiswing/misc/border.html>,  
*"How to Use Borders"*

# 使用 Border Factory

Border blackline, raisedetched, loweredetched, raisedbevel, loweredbevel, empty;

```
blackline = BorderFactory.createLineBorder(Color.black);  
raisedetched = BorderFactory.createEtchedBorder(EtchedBorder.RAISED);  
loweredetched = BorderFactory.createEtchedBorder(EtchedBorder.LOWERED);  
raisedbevel = BorderFactory.createRaisedBevelBorder();  
loweredbevel = BorderFactory.createLoweredBevelBorder();  
empty = BorderFactory.createEmptyBorder();
```

```
//Simple borders  
jComp1.setBorder(blackline);  
jComp2.setBorder(raisedbevel);  
jComp3.setBorder(loweredbevel);  
jComp4.setBorder(empty);
```



# 使用 Border Factory

```

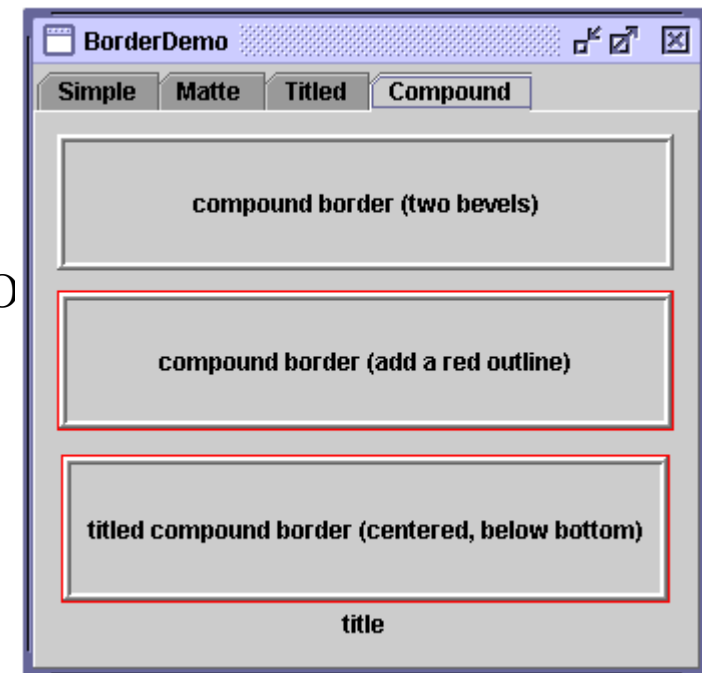
Border compound;
Border redline = BorderFactory.createLineBorder(Color.red);

//This creates a nice frame.
compound = BorderFactory.createCompoundBorder(raisedbevel, loweredbevel);
jComp13.setBorder(compound);

//Add a red outline to the frame.
compound = BorderFactory.createCompoundBorder(redline, compound);
jComp14.setBorder(compound);

//Add a title to the red-outlined frame.
compound = BorderFactory.createTitledBorder(
    compound, "title",
    TitledBorder.CENTER,
    TitledBorder.BELOW_BOTTOM,
    jComp15.setBorder(compound);

```



# Singleton

- Definition in "GoF"
- UML Class Diagram
- Singleton in Java Lib.



# Singleton

## Definition in "GoF"

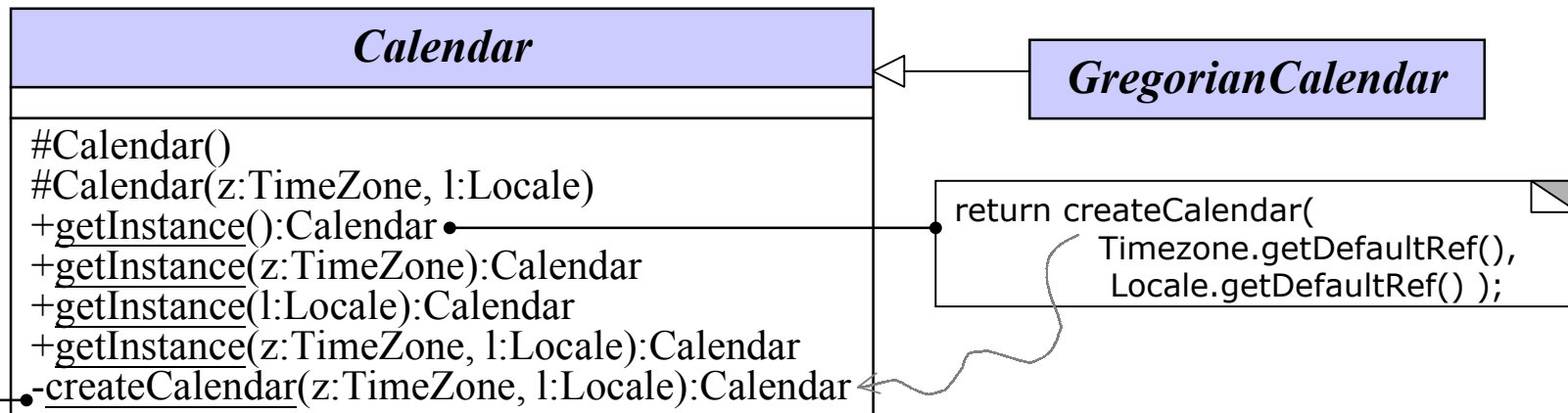
- Ensure a class only has one instance, and provide a global point of access to it.
- 確保某 **class** 只能生成唯一一個實體，並為它提供單一的全域存取窗口。

# Singleton ?!

in Java Library (.../src/java/util)

Calendar's getInstance method returns a Calendar object whose calendar fields have been initialized with the current date and time:

**Calendar** rightNow = **Calendar**.getInstance();



```

private static Calendar createCalendar(TimeZone zone, Locale aLocale)
{
    // If the specified locale is a Thai locale, returns a BuddhistCalendar instance
    if ("th".equals(aLocale.getLanguage())
        && ("TH".equals(aLocale.getCountry())) {
        return new sun.util.BuddhistCalendar(zone, aLocale); // 佛曆
    }
    // else create the default calendar
    return new GregorianCalendar(zone, aLocale); // (羅馬教皇) 格里高里曆
}
    
```

## Undoable

- Command Definition in "GoF"
- UML Class Diagram
- Undoable in Java Lib.

# Undoable with Command

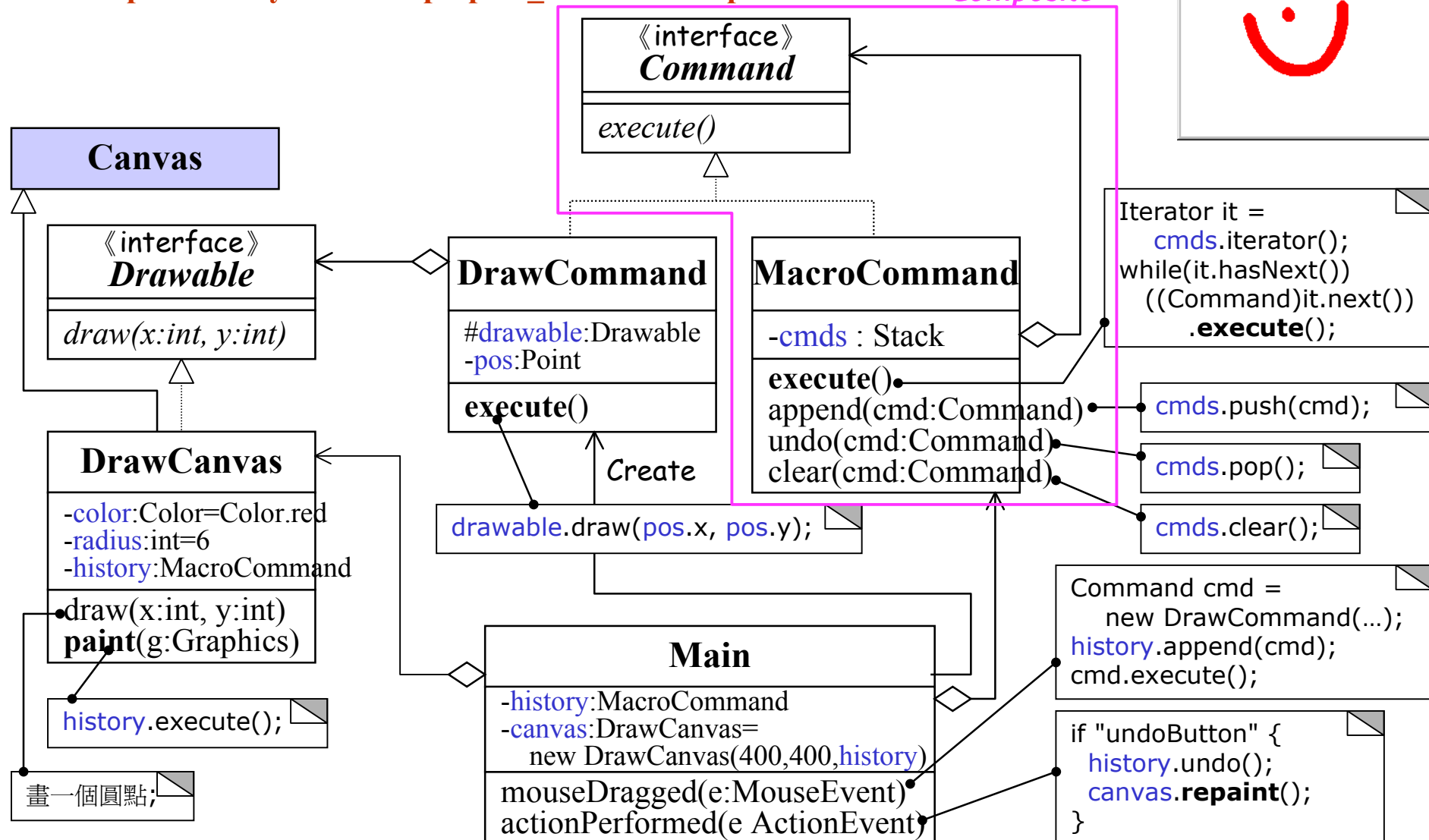
## Command Definition in "GoF"

- Encapsulate a request as an object, thereby letting you parameterize clients with different requests, queue or log requests, and support **undoable operations**.
- 將 request 封裝為 object，讓你得以運用不同的 requests（甚至是一系列 requests）對 client 參數化，並支援 **undo** 操作。

# Undoable with Command

[http://www.hyuki.com/dp/dpsrc\\_2004-05-26.zip](http://www.hyuki.com/dp/dpsrc_2004-05-26.zip)

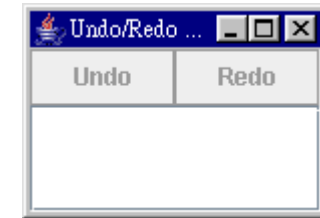
Composite



# Undoable (for TextArea)

in Java Library (.../src/javaw/swing/undo)

<http://www.java2s.com/ExampleCode/Swing-JFC/Undoredotextarea.htm>

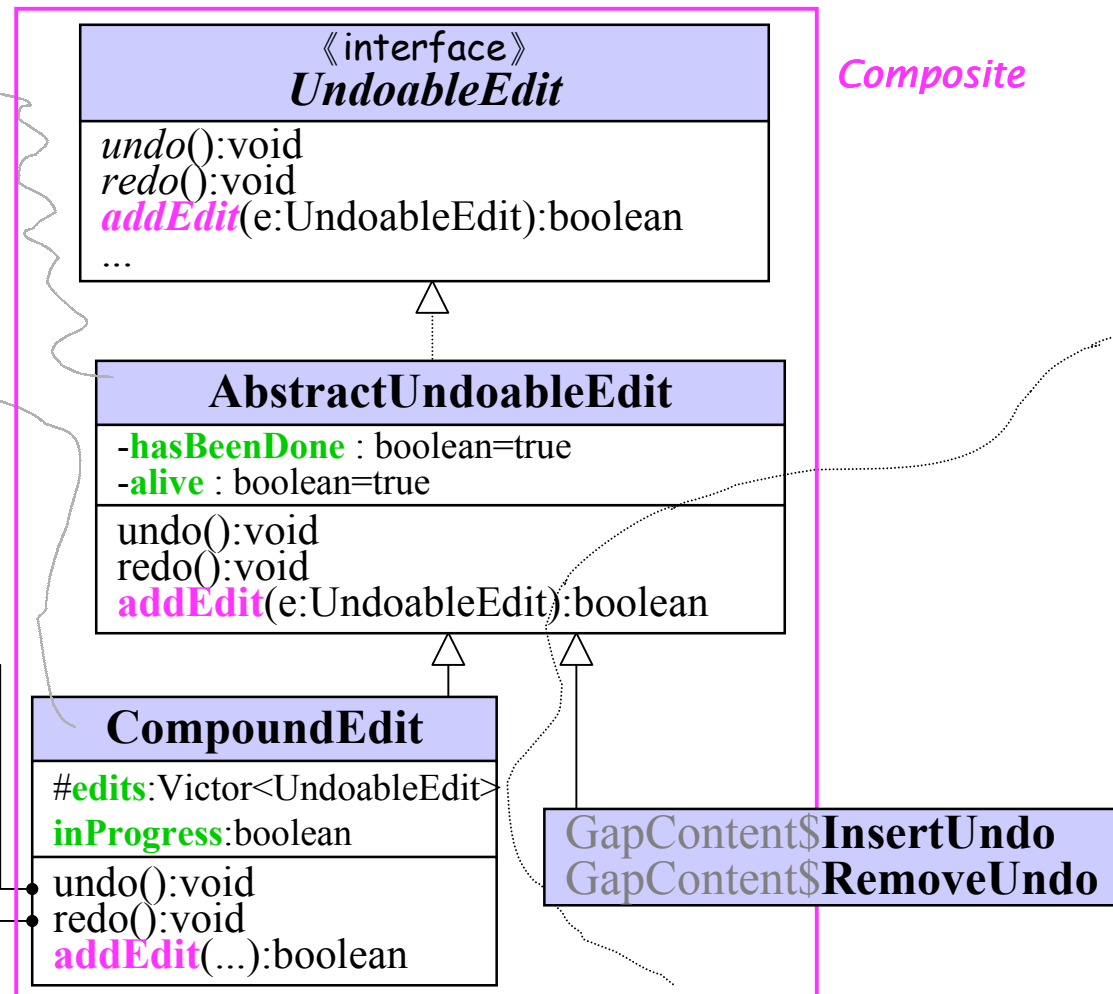


Subclasses 應該覆寫(override) undo().  
覆寫時應該首先呼叫 super。

用來組合 little  
UndoableEdits  
成為大的。

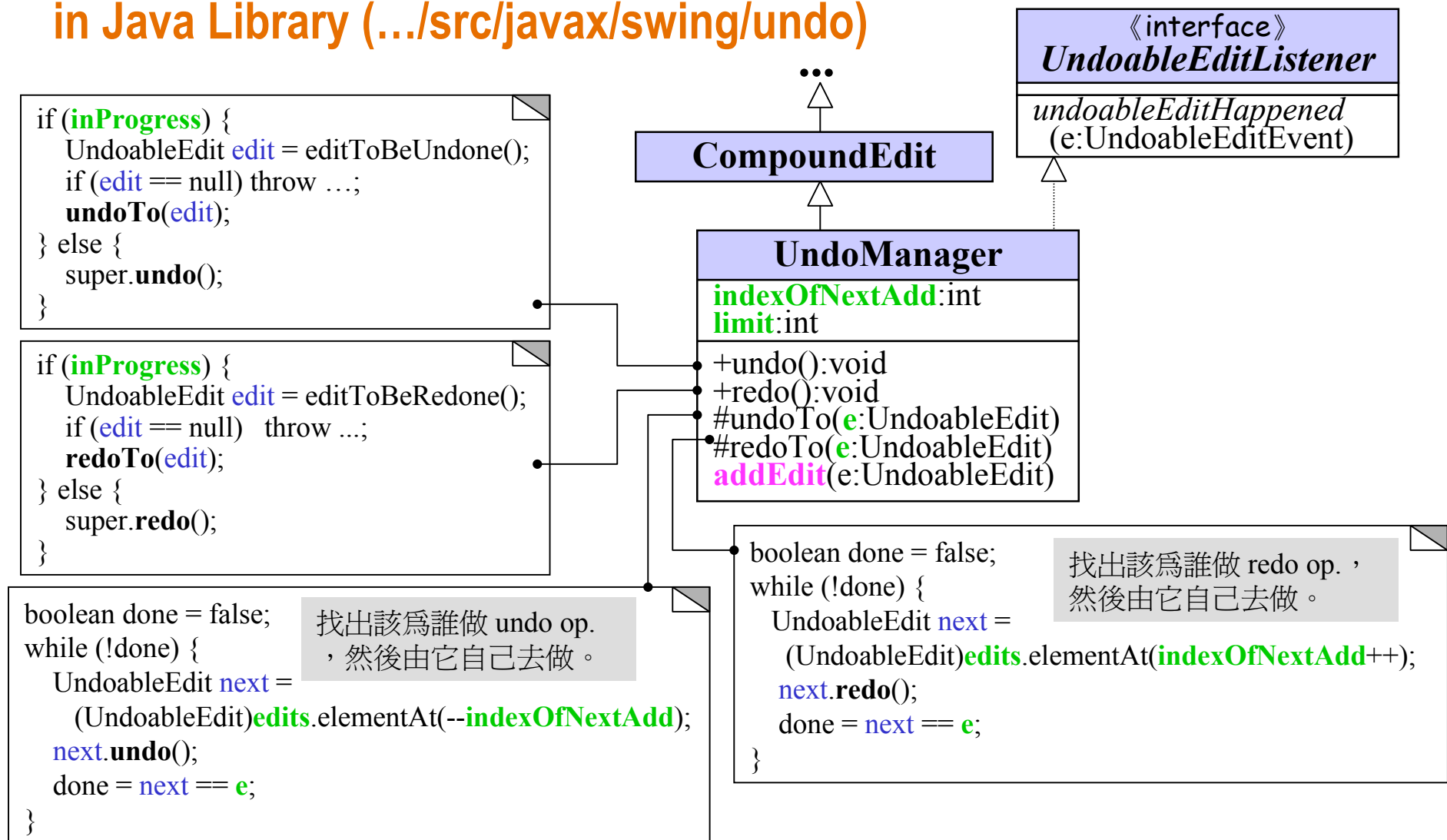
```
super.undo();
int i = edits.size();
while (i-- > 0) {
    UndoableEdit e =
        (UndoableEdit)edits.elementAt(i);
    e.undo();
}
```

```
super.redo();
Enumeration cursor = edits.elements();
while (cursor.hasMoreElements()) {
    ((UndoableEdit)cursor.
        nextElement()).redo();
}
```



# Undoable (for TextArea)

in Java Library (.../src/javaw/swing/undo)



# Undoable (for TextArea) 運行剖析,1



javax.swing.undo.UndoManager@6c585a hasE  
inProgress: true edits:

- ① [[javax.swing.text.GapContent\$InsertUndo@11ca803 hasBeenDone: true alive: true offset:0 length:1 string:null posRefs:null],
  - ② [[javax.swing.text.GapContent\$InsertUndo@5a67c9 hasBeenDone: true alive: true offset:1 length:1 string:null posRefs:null],
  - ③ [[javax.swing.text.GapContent\$InsertUndo@766a24 hasBeenDone: true alive: true offset:2 length:1 string:null posRefs:null],
  - ④ [[javax.swing.text.GapContent\$InsertUndo@32784a hasBeenDone: true alive: true offset:3 length:1 string:null posRefs:null],
  - ⑤ [[javax.swing.text.GapContent\$InsertUndo@15c07d8 hasBeenDone: true alive: true offset:4 **length:4** string:null posRefs:null]]
- limit: 100 indexOfNextAdd: 5

0 1 2 3 4  
侯捷臺灣侯捷臺灣  
(copy/paste)



按一次 [Undo] :

javax.swing.undo.UndoManager@6c585a hasE  
inProgress: true edits:

- ① [[javax.swing.text.GapContent\$InsertUndo@11ca803 hasBeenDone: true alive: true offset:0 length:1 string:null posRefs:null],
  - ② [[javax.swing.text.GapContent\$InsertUndo@5a67c9 hasBeenDone: true alive: true offset:1 length:1 string:null posRefs:null],
  - ③ [[javax.swing.text.GapContent\$InsertUndo@766a24 hasBeenDone: true alive: true offset:2 length:1 string:null posRefs:null],
  - ④ [[javax.swing.text.GapContent\$InsertUndo@32784a hasBeenDone: true alive: true offset:3 length:1 string:null posRefs:null],
  - ⑤ [[javax.swing.text.GapContent\$InsertUndo@15c07d8 hasBeenDone: **false** alive: true offset:4 length:4 string:侯捷臺灣  
posRefs:[javax.swing.text.GapContent\$UndoPosRef@13c7378]]]
- limit: 100 indexOfNextAdd: 4

0 1 2 3 4  
侯捷臺灣侯捷臺灣

只有在 undo 後這  
些string 才有內容



# Undoable (for TextArea) 運行剖析,2

續上頁，按一次 [BackSpace]：

javax.swing.undo.UndoManager@53fb57 has  
inProgress: true edits:

- 0 [[javax.swing.text.GapContent\$InsertUndo@19a32e0 hasBeenDone: true alive: true offset:0 length:1 string:null
  - 1 posRefs:null], [javax.swing.text.GapContent\$InsertUndo@8238f4 hasBeenDone: true alive: true offset:1 length:1 string:null
  - 2 posRefs:null], [javax.swing.text.GapContent\$InsertUndo@1b5340c hasBeenDone: true alive: true offset:2 length:1 string:null
  - 3 posRefs:null], [javax.swing.text.GapContent\$InsertUndo@16c163f hasBeenDone: true alive: true offset:3 length:1 string:null
  - 4 posRefs:null], [javax.swing.text.**GapContent\$RemoveUndo@15e0873** hasBeenDone: true alive: true offset:3 length:1
- string:灣 posRefs:[javax.swing.text.GapConten]]  
limit: 100 indexofNextAdd: 5



按一次 [Undo]：

javax.swing.undo.UndoManager@53fb57 ha  
inProgress: true edits:

- 0 [[javax.swing.text.GapContent\$InsertUndo@19a32e0 hasBeenDone: true alive: true offset:0 length:1 string:null posRefs:null],
  - 1 [javax.swing.text.GapContent\$InsertUndo@8238f4 hasBeenDone: true alive: true offset:1 length:1 string:null posRefs:null],
  - 2 [javax.swing.text.GapContent\$InsertUndo@1b5340c hasBeenDone: true alive: true offset:2 length:1 string:null posRefs:null],
  - 3 [javax.swing.text.GapContent\$InsertUndo@16c163f hasBeenDone: true alive: true offset:3 length:1 string:null posRefs:null],
  - 4 [javax.swing.text.**GapContent\$RemoveUndo@15e0873** hasBeenDone: **false** alive: true offset:3 length:1 string:**null** posRefs:null]
- limit: 100 indexofNextAdd: 4



# Undoable (for TextArea) 運行剖析,3

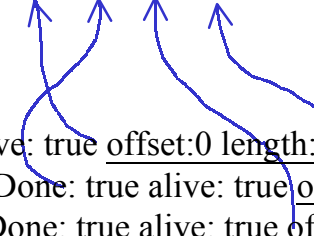
續上頁，按一次 [Undo]：

javax.swing.undo.UndoManager@453807 hasBeenDone: true alive: true inProgress: true edits:

- ① [[javax.swing.text.GapContent\$InsertUndo@618d26 hasBeenDone: true alive: true offset:0 length:1 string:null posRefs:null], [javax.swing.text.GapContent\$InsertUndo@79e304 hasBeenDone: true alive: true offset:1 length:1 string:null posRefs:null], [javax.swing.text.GapContent\$InsertUndo@3fa5ac hasBeenDone: true alive: true offset:2 length:1 string:null posRefs:null], [javax.swing.text.GapContent\$InsertUndo@95cfbe hasBeenDone: **false** alive: true offset:3 length:1 string:灣 posRefs:[javax.swing.text.GapContent\$UndoPosRef@15c07d8]],
  - ④ [javax.swing.text.GapContent\$RemoveUndo@1878144 hasBeenDone: **false** alive: true offset:3 length:1 string:null posRefs:null]
- limit: 100 indexOfNextAdd: 3



0 1 2 3  
侯捷臺灣



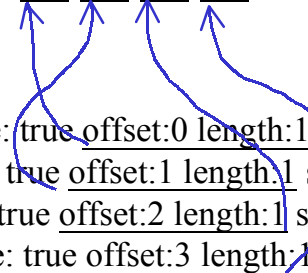
輸入 '北'：

javax.swing.undo.UndoManager@453807 hasBeenDone: true alive: true inProgress: true edits:

- ① [[javax.swing.text.GapContent\$InsertUndo@618d26 hasBeenDone: true alive: true offset:0 length:1 string:null posRefs:null], [javax.swing.text.GapContent\$InsertUndo@79e304 hasBeenDone: true alive: true offset:1 length:1 string:null posRefs:null], [javax.swing.text.GapContent\$InsertUndo@3fa5ac hasBeenDone: true alive: true offset:2 length:1 string:null posRefs:null], [javax.swing.text.GapContent\$InsertUndo@13c7378 hasBeenDone: true alive: true offset:3 length:1 string:null posRefs:null]]
- limit: 100 indexOfNextAdd: 4

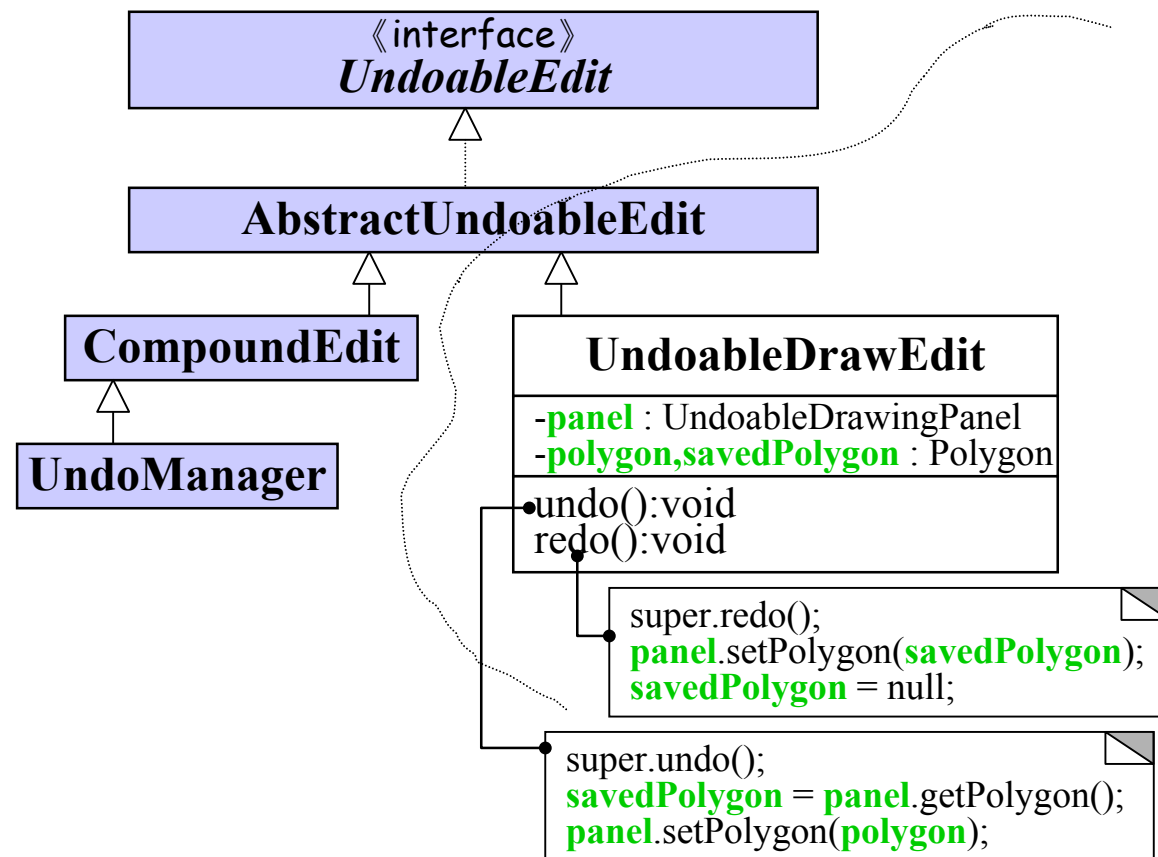
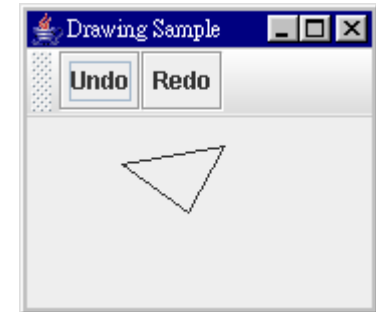


0 1 2 3  
侯捷臺北

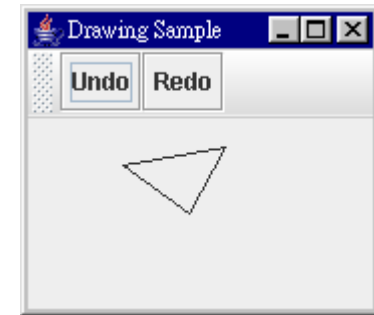


# Undoable (for Drawing)

<http://www.java2s.com/ExampleCode/Swing-JFC/UndoDrawing.htm>



# Undoable (for TextArea) 運行剖析,1



```
javax.swing.undo.UndoManager@119dc16 hasBeenDone: true alive: true
inProgress: true edits:
```

```
[UndoableDrawEdit@c05d3b hasBeenDone: true alive: true
```

```
panel:UndoableDrawingPanel[,0,34,292x89,layout=java.awt.FlowLayout,alignmentX=0.0,alignmentY=0.0,border=,flags
=9,maximumSize=,minimumSize=,preferredSize=]
```

```
polygon:java.awt.Polygon@18f1d7e npoints:0 xpoints[4]. ypoints[4]. bounds:null
```

```
savedPolygon:null,
```

```
UndoableDrawEdit@64883c hasBeenDone: true alive: true
```

```
panel:UndoableDrawingPanel[,0,34,292x89,layout=java.awt.FlowLayout,alignmentX=0.0,alignmentY=0.0,border=,flags
=9,maximumSize=,minimumSize=,preferredSize=]
```

```
polygon:java.awt.Polygon@2c1e6b npoints:1 xpoints[1]. ypoints[1]. bounds:null
```

```
savedPolygon:null,
```

```
UndoableDrawEdit@153f67e hasBeenDone: true alive: true
```

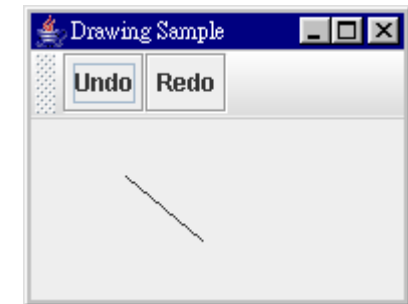
```
panel:UndoableDrawingPanel[,0,34,292x89,layout=java.awt.FlowLayout,alignmentX=0.0,alignmentY=0.0,border=,flags
=9,maximumSize=,minimumSize=,preferredSize=]
```

```
polygon:java.awt.Polygon@15bdc50 npoints:2 xpoints[2]. ypoints[2]. bounds:null
```

```
savedPolygon:null]
```

```
limit: 100 indexOfNextAdd: 3
```

# Undoable (for TextArea) 運行剖析,2



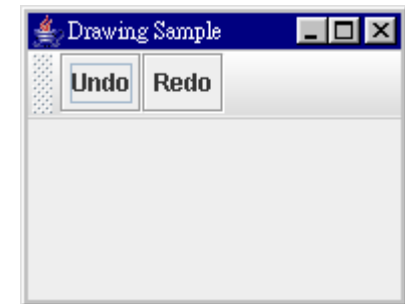
續上頁 [Undo]一次

```

javax.swing.undo.UndoManager@119dc16 hasBeenDone: true alive: true
inProgress: true edits:
[UndoableDrawEdit@c05d3b hasBeenDone: true alive: true
panel:UndoableDrawingPanel[,0,34,292x89,layout=java.awt.FlowLayout,alignmentX=0.0,alignmentY=0.0,border=,flag
s=9,maximumSize=,minimumSize=,preferredSize=]
polygon:java.awt.Polygon@18f1d7e npoints:0 xpoints[4]. ypoints[4]. bounds:null
savedPolygon:null,
UndoableDrawEdit@64883c hasBeenDone: true alive: true
panel:UndoableDrawingPanel[,0,34,292x89,layout=java.awt.FlowLayout,alignmentX=0.0,alignmentY=0.0,border=,flag
s=9,maximumSize=,minimumSize=,preferredSize=]
polygon:java.awt.Polygon@2c1e6b npoints:1 xpoints[1]. ypoints[1]. bounds:null
savedPolygon:null,
UndoableDrawEdit@153f67e hasBeenDone: false alive: true
panel:UndoableDrawingPanel[,0,34,292x89,layout=java.awt.FlowLayout,alignmentX=0.0,alignmentY=0.0,border=,flag
s=9,maximumSize=,minimumSize=,preferredSize=]
polygon:java.awt.Polygon@15bdc50 npoints:2 xpoints[2]. ypoints[2]. bounds:null
savedPolygon:java.awt.Polygon@170bea5 npoints:3 xpoints[3]. ypoints[3]. bounds:null]
limit: 100 indexOfNextAdd: 2

```

# Undoable (for TextArea) 運行剖析,3



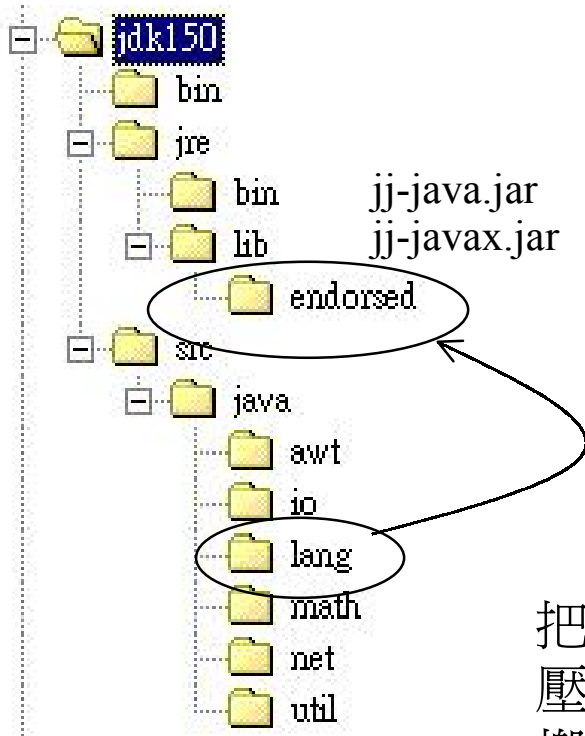
續上頁再 [Undo] 一次

```

javax.swing.undo.UndoManager@119dc16 hasBeenDone: true alive: true
inProgress: true edits:
[UndoableDrawEdit@c05d3b hasBeenDone: true alive: true
panel:UndoableDrawingPanel[,0,34,292x89,layout=java.awt.FlowLayout,alignmentX=0.0,alignmentY=0.0,border=,flag
s=9,maximumSize=,minimumSize=,preferredSize=]
polygon:java.awt.Polygon@18f1d7e npoints:0 xpoints[4]. ypoints[4]. bounds:null
savedPolygon:null,
UndoableDrawEdit@64883c hasBeenDone: false alive: true
panel:UndoableDrawingPanel[,0,34,292x89,layout=java.awt.FlowLayout,alignmentX=0.0,alignmentY=0.0,border=,flag
s=9,maximumSize=,minimumSize=,preferredSize=]
polygon:java.awt.Polygon@2c1e6b npoints:1 xpoints[1]. ypoints[1]. bounds:null
savedPolygon:java.awt.Polygon@9cbd4b npoints:2 xpoints[2]. ypoints[2]. bounds:null,
UndoableDrawEdit@153f67e hasBeenDone: false alive: true
panel:UndoableDrawingPanel[,0,34,292x89,layout=java.awt.FlowLayout,alignmentX=0.0,alignmentY=0.0,border=,flag
s=9,maximumSize=,minimumSize=,preferredSize=]
polygon:java.awt.Polygon@15bdc50 npoints:2 xpoints[2]. ypoints[2]. bounds:null
savedPolygon:java.awt.Polygon@170bea5 npoints:3 xpoints[3]. ypoints[3]. bounds:null]
limit: 100 indexOfNextAdd: 1

```

# Java 源碼修改經驗



```
rem "rejava.bat" usage :
rem (1) enter the directory which you want to rewrite java source.
rem (2) backup the original java source "x.java" to "x.java.ori".
rem (3) modify java source.
rem (4) rejava path filename (no extension name)
rem     ex: rejava javax\swing\text GapContent

del c:\jdk150\jre\lib\endorsed\jj-javax.jar
del c:\jdk150\jre\lib\endorsed\jj-java.jar
javac %2.java
move *.class e:\%1
e:
cd e:\
jar cvfM c:\jdk150\jre\lib\endorsed\jj-javax.jar javax
jar cvfM c:\jdk150\jre\lib\endorsed\jj-java.jar java
dir e:\%1
dir c:\jdk150\jre\lib\endorsed
c:
```

把修改後的.java 編譯為.class，  
 壓縮為 xxx.jar（需帶路徑例如 java\lang）並  
 搬移到endorsed，  
 即可被class loader優先讀取。





# The End

侯捷

[jjhou@jjhou.com](mailto:jjhou@jjhou.com)

[www.jjhou.com](http://www.jjhou.com)

