

第十二章 大型程式的發展與常用的類別庫

本章學習目標

- 學習如何分割檔案
- 認識類別庫，以及如何取用在不同類別庫裡的類別
- 建構package的階層關係
- 學習Java裡常用的類別庫

1

app12_2.java

```

02 package pack1;    // 宣告以下程式碼所定義的類別均納入package pack2中
03 class CCircle     // CCircle類別已納入package pack2中
04 {
05     public void show()
06     {
07         System.out.println("show() method called");
08     }
09 }
10 public class app12_2 // app12_2類別也納入package pack2中
11 {
12     public static void main(String args[])
13     {
14         CCircle cir=new CCircle();
15         cir.show();
16     }
17 }

```

```

/* app12_2 OUTPUT---
show() method called
-----*/

```

package pack1


CCircle類別

app12_2類別

2


12.1 檔案的分割

分割檔案的實作：

 CCircle.java

```
01 // CCircle.java, 本檔案置於c:\Java\pack1資料夾內
02 class CCircle // 定義類別CCircle
03 {
04     public void show()
05     {
06         System.out.println("show() method called");
07     }
08 }
```

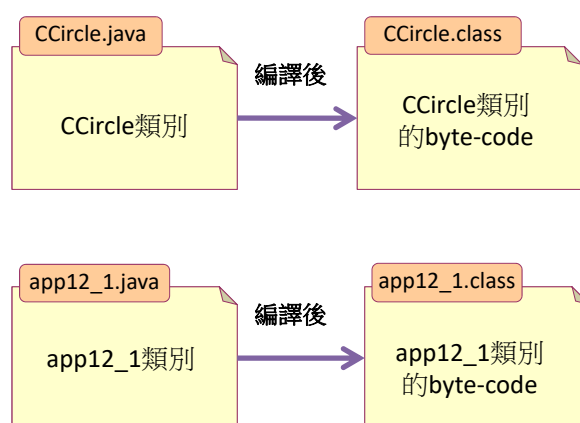
3

 app12_1.java

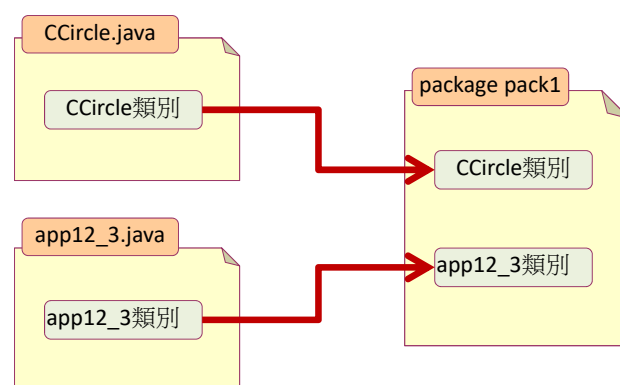
```
01 // app12_1.java, 本檔案置於c:\Java\pack1資料夾內
02 public class app12_1
03 {
04     public static void main(String args[])
05     {
06         CCircle cir=new CCircle();
07         cir.show();
08     }
09 }
```

4

編譯後會分別產生CCircle.class與app12_1.class兩個檔案，
如下圖：



5



6

課堂練習

繳交檔案為：**CWindow.java + ex12_1.java**

下面的程式碼包含了**CWindow**與**ex12_1**兩個類別。試將**CWindow**與**ex12_1**兩個類別分別存成兩個檔案，但這兩個檔案均在同一個package之內(pack12_1)，然後各別編譯並執行之。

```

02 class CWindow
03 {
04     private int width;
05     private int height;
06     private String name;
07
08     public CWindow(int w, int h, String s)
09     {
10         width=w;
11         height=h;
12         name=s;
13     }
14     public void show()
15     {
16         System.out.println("Name="+name);
17         System.out.println("W="+width+", H="+height);
18     }
19 }

20 public class ex12_1
21 {
22     public static void main(String args[])
23     {
24         CWindow cw=new CWindow(3,5,"Big windows");
25         cw.show();
26     }
27 }

/* output-----
Name=Big windows
W=3, H=5
-----*/

```

7

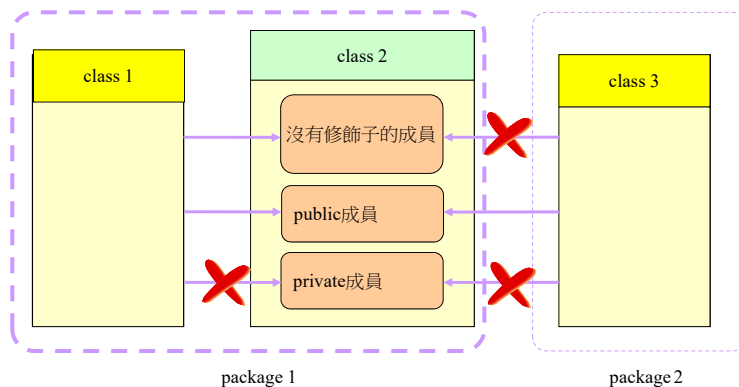
12.3 存取在不同package裡的類別

- ✓ 若某個類別需要被存取時，此類別必須宣告成**public**。
- ✓ 若要存取不同package內某個public類別的成員時，在程式碼內必須明確的指明「被存取package的名稱.類別名稱」。

8

public與private

- 若省略**public**與**private**，則成員只能在同一個**package**裡被存取
- 如果冠上**public**的話，則成員可以被任何一個**package**所存取
 - 類別內成員的修飾子與存取等級的關係圖：



9

12.3.2 public、private與protected修飾子的角色

下面列出了修飾子相對於類別成員的存取所扮演的角色：

表12.3.1 類別與介面所使用的修飾子

修飾子	說 明
沒有修飾子	只能讓同一個 package 裡的類別來存取
public	其它 package 裡的類別也可以存取此一類別裡的成員

表12.3.2 成員與建構元所使用的修飾子

修飾子	說 明
沒有修飾子	成員或建構元只能被同一個 package 內的程式所存取
public	如果所屬的類別也宣告成 public ，則成員或建構元可被不同 package 內所有的類別所存取。若所屬類別不是宣告成 public ，則成員或建構元只能被同一個 package 內的程式所存取
private	成員或建構元只在同一個類別內存取
protected	成員或建構元只能被位於同一 package 內的類別，以及它的子類別來存取

10

app12_2.java

```

02 package pack1;    // 宣告以下程式碼所定義的類別均納入package pack1中
03 class CCircle     // CCircle類別已納入package pack2中
04 {
05     public void show()
06     {
07         System.out.println("show() method called");
08     }
09 }
10 public class app12_2 // app12_2類別也納入package pack2中
11 {
12     public static void main(String args[])
13     {
14         CCircle cir=new CCircle();
15         cir.show();
16     }
17 }

```

注意找
找看!!!

```

/* app12_2 OUTPUT---
show() method called
-----*/

```

package pack1

CCircle類別

app12_2類別

11

12.3.1 簡單的範例 下面的範例說明了如何存取在不同package裡的類別：

CCircle.java

```

01 // CCircle.java, package的使用(三),此檔案存放在pack4b資料夾內
02 package pack4b;
03 public class CCircle // 將CCircle類別納入package pack4b當中
04 {
05     public void show()
06     {
07         System.out.println("show() method called");
08     }
09 }

```

```

/* app12_4 OUTPUT---
show() method called
-----*/

```

app12_4.java

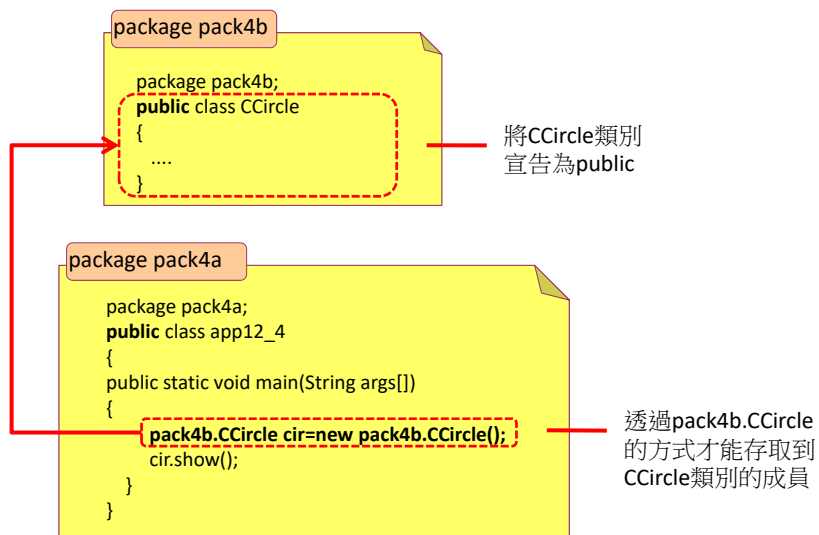
```

01 // app12_4.java, package的使用(三),此檔案存放在pack4a資料夾內
02 package pack4a;
03 public class app12_4 // 將app12_4類別納入package pack4a當中
04 {
05     public static void main(String args[])
06     {
07         pack4b.CCircle cir=new pack4b.CCircle();
08         cir.show();
09     }
10 }

```

12

下圖說明了如何在package pack4a裡存取package pack4b裡的成員：



13

存取存放在不同package裡的類別，可以透過
被存取的package名稱.類別名稱
的語法。

app12_4.java的第7行即是：

被存取的package名稱

```
pack4b . CCircle cir = new pack4b . CCircle();
```

類別名稱

14

課堂練習

繳交檔案為：CRectangle.java + app8_1.java

- 請參考app8_1，試將CRectangle類別與app8_1類別分別存於aaa與bbb這兩個資料夾，然後個別編譯並執行之。提示：aaa.CRectangle rect1=new aaa.CRectangle();

```

01 // app8_1, 建立物件與field的存取
02 class CRectangle // 定義CRectangle類別
03 {
04     int width; // 宣告資料成員width
05     int height; // 宣告資料成員height
06 }
07
08 public class app8_1
09 {
10     public static void main(String args[])
11     {
12         CRectangle rect1;
13         rect1=new CRectangle(); // 建立新的物件
14
15         rect1.width=10; // 設定矩形rect1的寬
16         rect1.height=5; // 設定矩形rect1的高
17
18         System.out.println("width="+rect1.width); // 印出rect1.width
19         System.out.println("height="+rect1.height); // 印出rect1.height
20     }
21 }
/* output-----
width=10
height=5
-----*/

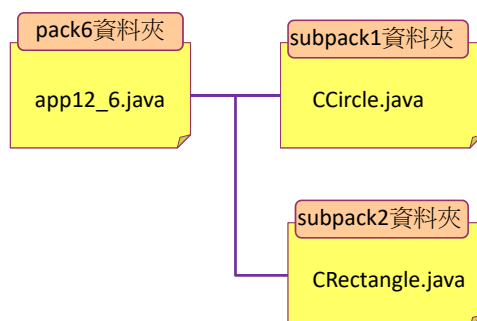
```

15

12.4 建構package的階層關係

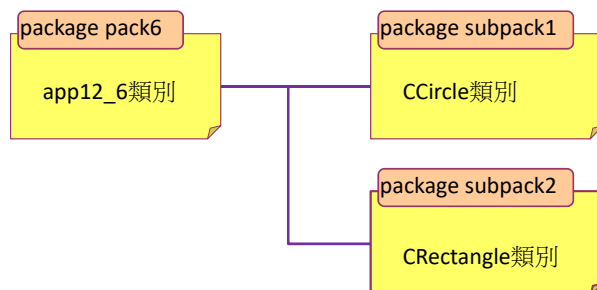
把packages劃分為階層的關係，程式碼可更加容易維護。

下面的範例為資料夾與所存放之Java原始檔的階層關係圖：



16

上圖的資料夾階層關係圖可化成package的階層關係圖：



要宣告某個類別是屬於某個sub-package，可用下面的語法來宣告：

package package名稱.sub-package名稱;

格式12.4.1
sub-package的宣告格式

17

透過import指令，即可將某個package內的特定類別匯入。

import package名稱.類別名稱;

格式12.3.1
匯入package裡的某個類別

18

下面是import指令的範例：

CCircle.java

```
01 // CCircle.java, package的使用(四),此檔案置於pack5b資料夾內
02 package pack5b;
03 public class CCircle // 將CCircle類別納入package pack5b當中
04 {
05     public void show()
06     {
07         System.out.println("show() method called");
08     }
09 }
```

```
/* app12_5 OUTPUT---
show() method called
-----*/
```

app12_5.java

```
01 // app12_5.java, package的使用(四),此檔案置於pack5a資料夾內
02 package pack5a;
03 import pack5b.CCircle; // 載入pack5b package裡的CCircle類別
04
05 public class app12_5
06 {
07     public static void main(String args[])
08     {
09         CCircle cir=new CCircle(); // 不用再寫package的名稱了
10         cir.show();
11     }
12 }
```

19

CCircle.java

```
01 // CCircle.java, 此檔案置於pack6\subpack1資料夾內
02 package pack6.subpack1; // 將CCircle類別納入pack6.subpack1中
03 public class CCircle
04 {
05     public void show()
06     {
07         System.out.println("show() method of class CCircle called");
08     }
09 }
```

```
/* app12_6 OUTPUT-----
show() method in class CCircle called
show() method in class CRectangle called
-----*/
```

CRectangle.java

```
01 // CRectangle.java, 此檔案置於pack6\subpack2資料夾內
02 package pack6.subpack2; // 將CRectangle類別納入pack6.subpack2中
03 public class CRectangle
04 {
05     public void show()
06     {
07         System.out.println("show() method of class CRectangle called");
08     }
09 }
```

app12_6.java

```
01 // app12_6.java,此檔案置於pack6資料夾內
02 package pack6; // 將app12_6類別納入package pack6當中
03 import pack6.subpack1.CCircle; // 載入pack6.subpack1裡的CCircle類別
04 import pack6.subpack2.CRectangle; // 載入pack6.subpack2裡的CRectangle類別
05 public class app12_6
06 {
07     public static void main(String args[])
08     {
09         {
10             CCircle cir=new CCircle();
11             CRectangle rect=new CRectangle();
12             cir.show();
13             rect.show();
14         }
15     }
16 }
```

20

課堂練習

(繳交檔案為：**CCircle.java**, **CRectangle.java**, **app12_6.java**, **CStyle.java**)

接續**app12_6**，假設在**package subpack1** 裡有另一個**package subsubpack**，裡面定義了**CStyle** 類別，其程式碼如下：

01 // CStyle.java, 此檔案置於pack6\subpack1\subsubpack 資料夾內

02 package pack6.subpack1.subsubpack;

03 public class CStyle

04 {

05 public int style;

06

07 public CStyle(int n)

08 {

09 style=n;

10 System.out.println("style"+style);

11 }

12 }

試撰寫相關的程式碼，使得在**main()** method 裡可以建立一個**CStyle** 物件，並將**style** 成員的值設為**1**。

/* output-----

style1

show() method of class CCircle called

show() method of class CRectangle called

-----*/

21

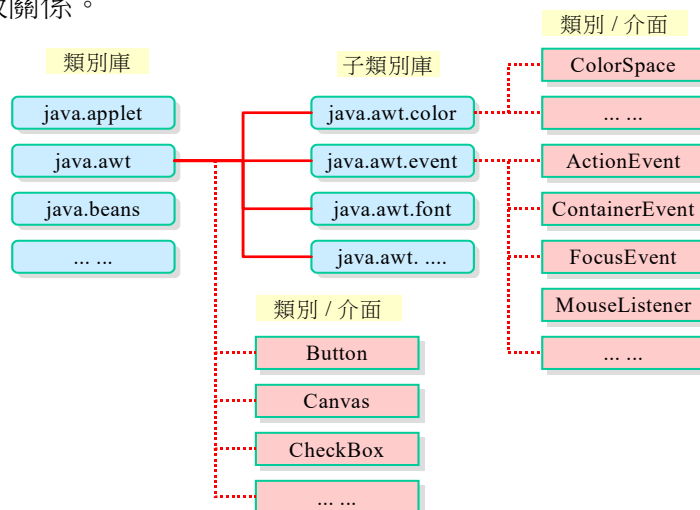
12.6 Java常用的類別庫

✓Java的package是用來放置類別與介面的地方。

✓package可譯為「類別庫」。

22

下面的簡圖為是類別庫、子類別庫、類別與介面之間的層級關係。



23

下表列出了Java常用的類別庫。

表12.6.1 Java常用的類別庫

類別庫名稱	所包含類別之主要功能
java.applet	與applet相關的類別，applet是嵌在網頁裡的小程式，用來執行特定的功能
java.awt	與Java早期視窗元件設計有關的類別
java.awt.event	與事件(event)觸發相關的類別
java.lang	Java最基本的類別，此類別會自動載入
java.io	與輸入/輸出相關的類別
java.net	與網路元件與連線相關的類別
java.util	Java utility相關的類別，如Array、Vector等

24

要匯入java.awt類別庫裡的Button類別，可用下列的語法：

```
import java.awt.Button;
```

如果要匯入類別庫裡的所有類別時，可以透過萬用字元「*」來匯入：

```
import java.awt.*;
```

25

12.6.1 有關字串的類別庫

String類別是放置在java.lang類別庫內。

建立字串物件（String object）

建立String物件的方式：

```
String str = "abc";  
char data[]={ 'a', 'b', 'c' };    // data為字元組成的陣列  
String str=new String(data); // 利用建構元來產生字串  
String str = new String("abc");    //利用建構元建立字串
```

26

下表列出了第二種與第三種所使用的建構元之格式：

表12.6.2 String類別建構元的格式

建構元格式	主要功能
<code>String()</code>	沒有引數的 String() 建構元
<code>String(byte[] bytes)</code>	以 byte 陣列建立字串
<code>String(byte[] bytes, int offset, int length)</code>	取出 byte 陣列裡，從陣列的第 offset 位置開始，長度為 length 來建立字串
<code>String(char[] value)</code>	利用字元陣列來產生字串物件
<code>String(char[] value, int offset, int count)</code>	取出字元陣列裡，從陣列的第 offset 位置開始，長度為 count 來建立字串
<code>String(String original)</code>	利用原始字串（ original string ）產生字串物件

27

字串類別所提供的method

下表列出了一些常用的method：

表12.6.3 String類別常用的method

method	主要功能
<code>byte[] getBytes()</code>	將字串轉換成 byte 型態的陣列
<code>char charAt(int index)</code>	取得 index 位置的字元
<code>boolean equals(String str)</code>	測試字串是否與 str 相同
<code>int indexOf(char ch)</code>	根據字元 ch 找出第一個在字串出現的位置
<code>int length()</code>	取得字串的長度
<code>String substring(int index)</code>	取出 index 之後的子字串
<code>String substring(int ind1, int ind2)</code>	取出位於 ind1 和 ind2 之間的字串
<code>boolean startsWith(String prefix)</code>	測試字串是否以 prefix 字串為開頭
<code>String toLowerCase()</code>	將字串轉換成小寫
<code>String toUpperCase()</code>	將字串轉換成大寫

28

下面的範例列舉了幾個method的用法：

```

01 // app12_9, String類別使用的範例
02 public class app12_9
03 {
04     public static void main(String args[])
05     {
06         String str="Easier said than done.";
07         System.out.println("length="+str.length());
08         System.out.println("charAt(8)="+str.charAt(8)); //index=8的字元
09         System.out.println("sub string="+str.substring(7)); //index>=7之後的子字串
10         System.out.println("start with \"th\"="+str.startsWith("th")); //字串起始於...
11         System.out.println("upper case="+str.toUpperCase()); //轉換大寫
12     }
13 }

/* app12_9 OUTPUT-----
length=22
charAt(8)=a
sub string=said than done.
start with "th"=false
upper case=EASIER SAID THAN DONE.
-----*/

```

29

12.6.2 StringBuffer類別庫

要修改字串，必須使用StringBuffer類別：

表12.6.4 StringBuffer類別常用的method

method	主要功能
StringBuffer append(char c)	將字元c附加到字串之後
StringBuffer append(String str)	將字串str附加到字串之後
StringBuffer deleteCharAt(int index)	刪除字串第index位置的字元
StringBuffer insert(int k, char c)	將字串的第k個位置插入字元c
StringBuffer insert(int k, String str)	將字串的第k個位置插入字串str
int length()	取得字串的長度
StringBuffer replace(int m,int n,String str)	將字串第m(包括)到n(不包括)之間以字串str取代
StringBuffer reverse()	將字串反向排列
String toString()	將StringBuffer型態的字串轉換成String型態

30

下面的程式碼為**String**使用的典型範例：

```

01 // app12_10, StringBuffer類別使用的範例
02 public class app12_10
03 {
04     public static void main(String args[])
05     {
06         StringBuffer str=new StringBuffer("Black & White");
07
08         System.out.println(str);
09         System.out.println("length="+str.length());
10         System.out.println(str.replace(0,5,"cats")); //0<= 取代<5
11         System.out.println(str.replace(7,12,"dogs"));
12         System.out.println(str.reverse());
13         System.out.println(str);
14     }
15 }

```

**/* app12_10 OUTPUT-
Black & White
length=13
cats & White
cats & dogs
sgod & stac
sgod & stac
-----*/**

31

12.6.3 wrapper class

下表列出了原始資料型態與相對應的wrapper class：

表12.6.5 原始資料型態與其wrapper class

原始資料型態	wrapper class
boolean	Boolean
byte	Byte
char	Character
short	Short
int	Integer
long	Long
float	Float
double	Double

基於效率考量，原始資料型態(primitive) 如:boolean, byte, char, short, int, long, float, double等均不被看成是物件。但Java還是提供一些特殊的類別，讓原始資料型態在使用上有如物件一般，這些特殊的類別，我們稱為wrapper class。

32

下表列出了各種類別常用的轉換函數：

表12.6.6 各種類別常用的轉換函數

類別	method	主要功能
Byte	static byte parseByte(String s)	將字串s轉換成byte型態的值
Byte	static String toString(byte b)	將byte型態的數值b轉換成字串
Character	static String toString(char c)	將字元c轉換成字串
Short	static short parseShort(String s)	將字串s轉換成短整數
Short	static String toString(short s)	將短整數s轉換成字串
Integer	static int parseInt(String s)	將字串s轉換成整數
Integer	static String toString(int i)	將整數i轉換成字串
Long	static long parseLong(String s)	將字串s轉換成長整數
Long	static String toString(Long i)	將長整數i轉換成字串
Float	static float parseFloat(String s)	將字串s轉換成浮點數
Float	static String toString(float f)	將浮點數f轉換成字串
Double	static double parseDouble(String s)	將字串s轉換成倍精度浮點數
Double	static String toString(double d)	將倍精度浮點數d轉換成字串

33

下面的程式碼是Integer類別使用的範例：

```

01 // app12_11,Integer class method的應用
02 public class app12_11
03 {
04     public static void main(String args[])
05     {
06         String str;
07         int inum;
08
09         inum=Integer.parseInt("654")+3;          // 將字串轉成整數後，再加3
10         System.out.println(inum);
11         str=Integer.toString(inum)+"3"; // 將 "3" 附加在字串後面
12         System.out.println(str);
13     }
14 }

/* app12_11 OUTPUT---
657
6573
-----*/

```

34

12.6.4 使用Math 類別

Math類別提供的method可用來計算相關的數學函數。

下表列出了Math 類別所提供的「類別變數」：

表12.6.7 Math類別所提供的類別變數

method	主要功能
public static final double E	尤拉常數 (Euler's constant)
public static final double PI	圓周率， π

35

常用的數學函數列表如下：

表12.6.8 Math類別所提供的method

method	主要功能
public static double sin(double a)	正弦函數，計算 $\sin(a)$
public static double cos(double a)	餘弦函數，計算 $\cos(a)$
public static double tan(double a)	正切函數，計算 $\tan(a)$
public static double asin(double a)	反正弦函數，計算 $\sin^{-1}(a)$
public static double acos(double a)	反餘弦函數，計算 $\cos^{-1}(a)$
public static double atan(double a)	反正切函數，計算 $\tan^{-1}(a)$
public static double exp(double a)	自然指數函數，計算 $\exp(a)$
public static double log(double a)	自然對數函數，計算 $\log(a)$
public static double sqrt(double a)	開根號函數，計算 \sqrt{a}
public static double ceil(double a)	傳回大於a的最小整數
public static double floor(double a)	傳回小於a的最大整數
public static double pow(double a, double b)	計算a的b次方

36

method	主要功能
<code>public static int round(float a)</code>	傳回最接近a的整數
<code>public static double random()</code>	傳回0.0~1.0之間的亂數
<code>public static type abs(type a)</code>	計算a的絕對值，其中type可為int、long、float或是double
<code>public static int max(int a, int b)</code>	找出a與b中較大者
<code>public static int min(int a, int b)</code>	找出a與b中較小者

37

下面的範例說明數學函數的使用：

```

01 // app12_12,數學函數的使用
02 public class app12_12
03 {
04     public static void main(String args[])
05     {
06         System.out.println("ceil(3.9)= "+Math.ceil(3.9)); //大於a的最小整數
07         System.out.println("sin(PI/2)= "+Math.sin(Math.PI/2));
08         System.out.println("max(8,2)= "+Math.max(8,2));
09     }
10 }

```

/* app12_12 OUTPUT-

```

ceil(3.9)= 4.0
sin(PI/2)= 1.0
max(8,2)= 8

```

-----*/

38

課堂練習

ex12_6_1.java

試撰寫一程式，利用**String** 類別將字串 "Milk" 反向排列。

```
/* output---  
KliM  
-----*/
```

ex12_6_2.java

試將倍精度浮點數**67.834** 轉換成字串，並計算字串長度。

```
/* output---  
67.834  
length=6  
-----*/
```

ex12_6_3.java

試利用**String** 類別的**substring()** method，從字串 "Two heads are better than one." 取出 "better" 子字串。

```
/* output-----  
new string=better  
-----*/
```

39