

# 第十章 類別的繼承

## 本章學習目標

- 學習繼承的基本概念
- 了解子類別與父類別之間的關係
- 認識method的改寫
- 區分super() 與this() 的用法
- 認識Object類別

1

## 10.1 繼承的基本概念

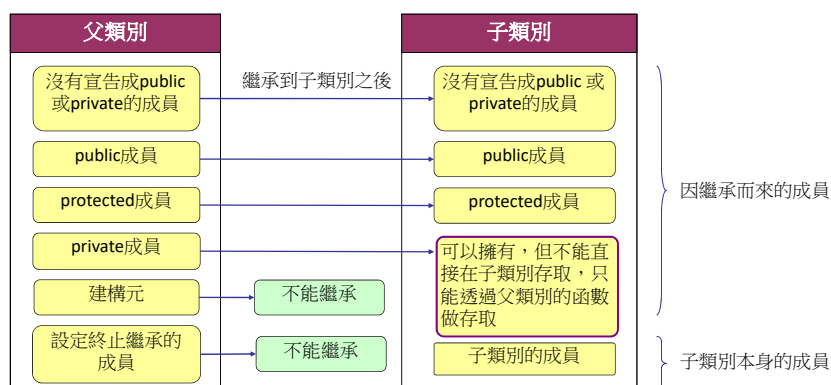
2

## 10.1 繼承的基本概念

- 繼承：根據既有類別為基礎，進而衍生出另一類別。
- 既有的類別稱為父類別（super class）或基底類別（basis class）。
- 衍生出的類別稱為子類別（sub class）或衍生類別（derived class）。
- 每一個類別只能有一個父類別（單一繼承）。
- 一個父類別可以擁有一個以上的子類別。

3

類別成員的繼承關係可用下圖來表示：



4

## 類別的繼承的語法：

### 格式10.1.1 類別繼承的格式

**class** 父類別名稱

父類別

```
{ // 父類別裡的成員 }
```

**class** 子類別名稱 **extends** 父類別名稱

子類別

```
{ // 子類別裡的成員 }
```

5

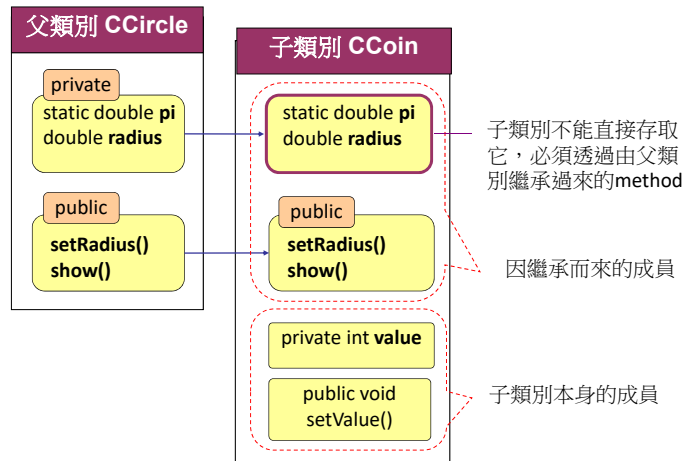
```
app10_1包含了原有的CCircle類別，以及繼承而來的CCoin類別
01 // app10_1, 簡單的繼承範例
02 class CCircle                                // 父類別CCircle
03 {
04     private static double pi=3.14;
05     private double radius;
06     public CCircle()                          // CCircle()建構元
07     {
08         System.out.println("CCircle() constructor called ");
09     }
10     public void setRadius(double r)
11     {
12         radius=r;
13         System.out.println("radius="+radius);
14     }
15     public void show()
16     {
17         System.out.println("area="+pi*radius*radius);
18     }
19 }
20
21 class CCoin extends CCircle                  // 子類別CCoin，繼承自CCircle類別
22 {
23     private int value;                        // 子類別的資料成員
24     public CCoin()                           // 子類別的建構元
25     {
26         System.out.println("CCoin() constructor called ");
27     }
28     public void setValue(int t)               // 子類別的setValue() method
29     {
30         value=t;
31         System.out.println("value="+value);
32     }
33 }
34 }
```

### 10.1.1 簡單的繼承範例

```
/* app10_1 OUTPUT-----
CCircle() constructor called
CCoin() constructor called
radius=2.0
area=12.56
value=5
-----*/
```

```
35 public class app10_1
36 {
37     public static void main(String args[])
38     {
39         CCoin coin=new CCoin(); // 建立coin物件
40         coin.setRadius(2.0);    // 呼叫由父類別繼承而來的setRadius()
41         coin.show();           // 呼叫由父類別繼承而來的show()
42         coin.setValue(5);      // 呼叫子類別的setValue()
43     }
44 }
```

本例的繼承關係圖繪製如下



7

課堂舉例...下述程式可以執行嗎？

```

class P
{
    private int a=2;
}
class S extends P
{
    public void show()
    {
        System.out.println("a="+a);
    }
}
public class test
{
    public static void main(String argv[])
    {
        S obj=new S();
        obj.show();
    }
}
  
```

```

class P
{
    private int a=2;
    public void show_1()
    {
        System.out.println("a="+a);
    }
}
class S extends P
{
    public void show()
    {
        show_1();
    }
}
public class test
{
    public static void main(String argv[])
    {
        S obj=new S();
        obj.show();
    }
}
  
```

8

## 下列幾點重要的觀念：

1. 透過**extends**關鍵字，可將父類別的成員（包含資料成員與**method**）繼承給子類別。
2. **Java**在執行子類別的建構元之前，會先呼叫父類別的建構元。

9

### 10.1.2 建構元的呼叫

下面的範例是透過**super()**來呼叫父類別中特定的建構元：

10-10

```

01 // app10_2, 呼叫父類別中特定的建構元
02 class CCircle // 定義父類別CCircle
03 {
04     private static double pi=3.14;
05     private double radius;
06
07     public CCircle() // 父類別裡沒有引數的建構元
08     {
09         System.out.println("CCircle() constructor called");
10     }
11     public CCircle(double r) // 父類別裡有一個引數的建構元
12     {
13         System.out.println("CCircle(double r) constructor called");
14         radius=r;
15     }
16     public void show()
17     {
18         System.out.println("area="+pi*radius*radius);
19     }
20 }

```

```

/* app10_2 OUTPUT
CCircle() constructor called
CCoin() constructor called
CCircle(double r) constructor called
CCoin(double r, int v) constructor called
area=0.0
area=19.625

```



```

21 class CCoin extends CCircle // 定義子類別CCoin，繼承自CCircle類別
22 {
23     private int value;
24     public CCoin() // 子類別裡沒有引數的建構元
25     {
26         System.out.println("CCoin() constructor called");
27     }
28     public CCoin(double r, int v) // 子類別裡有兩個引數的建構元
29     {
30         super(r); // 呼叫父類別裡有引數的建構元,即第11行所定義的建構元
31         value=v;
32         System.out.println("CCoin(double r, int v) constructor called");
33     }
34 }

```

```

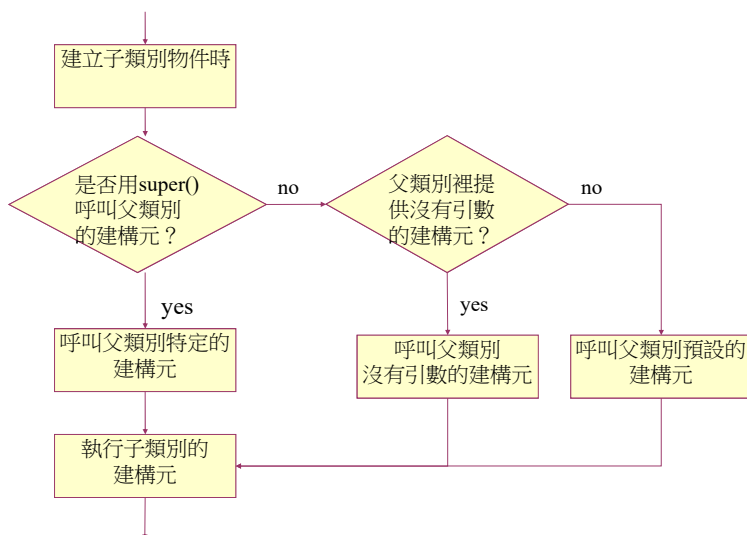
35 public class app10_2
36 {
37     public static void main(String args[])
38     {
39         CCoin coin1=new CCoin(); // 建立物件,並呼叫第24行的建構元
40         CCoin coin2=new CCoin(2.5,10); // 建立物件,並呼叫第28行的建構元
41         coin1.show();
42         coin2.show();
43     }
44 }

```

10

### 10.1.3 使用建構元常見的錯誤

下圖是建構元呼叫的流程圖：



11

### (1) 請問下述會出現什麼結果?

```

// app10_2, 呼叫父類別中特定的建構元
class CCircle // 定義父類別CCircle
{private double radius;
  public CCircle(double r) // 父類別裡有一個引數的建構元
  {
    System.out.println("A");
  }
}
class CCoin extends CCircle // 定義子類別CCoin，繼承自CCircle類別
{
  public CCoin() // 子類別裡沒有引數的建構元
  {
    super(2.0);
    System.out.println("B");
  }
}
public class test
{
  public static void main(String args[])
  {
    CCoin coin1=new CCoin(); // 建立物件，並呼叫第24行的建構元
  }
}
  
```

12

下面是錯誤的建構元使用範例：

```

01 // app10_3, 建構元錯誤的範例
02 class CCircle // 定義類別CCircle
03 {
04     private static double pi=3.14;
05     private double radius;
06
07     public CCircle(double r) // 有引數的建構元
08     {
09         radius=r;
10     }
11     public void setRadius(double r)
12     {
13         radius=r;
14         System.out.println("radius="+radius);
15     }
16 }
17
18 class CCoin extends CCircle // 定義CCoin類別，繼承自CCircle類別
19 {
20     private int value;
21
22     public CCoin(double r, int v) // CCoin()有兩個引數的建構元
23     {
24         setRadius(r); // 透過setRadius() method來設定radius成員
25         value=v; // 設定value成員
26     }
27 }
28
29 public class app10_3
30 {
31     public static void main(String args[])
32     {
33         CCoin coin1=new CCoin(2.5,10); // 建立物件,並呼叫有兩個引數的建構元
34     }
35 }

```

執行時會得到如下的錯誤訊息：  
cannot find symbol constructor CCircle ()

13

更正app10\_3的錯誤：

```

01 // app10_4, 修正app10_3的錯誤
02 class CCircle // 定義類別CCircle
03 {
04     private double pi=3.14;
05     private double radius;
06
07     public CCircle() // 沒有引數的建構元
08     {
09     }
10     public CCircle(double r) // 有一個引數的建構元
11     {
12         radius=r;
13     }
14     public void setRadius(double r)
15     {
16         radius=r;
17         System.out.println("radius="+radius);
18     }
19 }
20 // 將app10_3中，類別CCoin的定義置於此處
21 // 將app10_3中，類別app10_3的定義置於此處

```

```

/* app10_4 OUTPUT---
radius=2.5
-----*/

```

14

## (2)請問下述會出現什麼結果？

```

class CCircle // 定義父類別CCircle
{ private double radius;
  public CCircle(double r) // 父類別裡有一個引數的建構元
  {
    System.out.println("A");
  }
}
class CCoin extends CCircle // 定義子類別CCoin，繼承自CCircle類別
{
  public CCoin() // 子類別裡沒有引數的建構元
  {
    System.out.println("B");
  }
}
public class test
{
  public static void main(String args[])
  {
    CCoin coin1=new CCoin(); // 建立物件，並呼叫第24行的建構元
  }
}

```

15

## (3)更改之後會出現什麼結果？

```

class CCircle // 定義父類別CCircle
{ private double radius;
  public CCircle(double r) // 父類別裡有一個引數的建構元
  {
    System.out.println("A");
  }
}
class CCoin extends CCircle // 定義子類別CCoin，繼承自CCircle類別
{
  public CCoin() // 子類別裡沒有引數的建構元
  {
    super(2.0);
    System.out.println("B");
  }
}
public class test
{
  public static void main(String args[])
  {
    CCoin coin1=new CCoin(); // 建立物件，並呼叫第24行的建構元
  }
}

```

16



## this() 與 super() 的比較

- ✓ this() 是在同一類別內呼叫其它的建構元。
- ✓ super() 則是從子類別的建構元呼叫其父類別的建構元。

this()與super()還是有其相似之處：

1. this() 與super() 均可多載。
2. this() 與super() 均必須撰寫在建構元內的第一行，因此 this()與super() 無法同時存在同一個建構元內。

17

## ex10\_1\_1.java

於下面的程式碼中，請設計一子類別**Cbbb**，繼承自**Caaa** 類別，並加入**set\_num()** method，可用來設定從父類別繼承而來的成員**num1** 與**num2**，並於**Cbbb**建立一個**show()** method，用來顯示**num1** 與**num2** 的值，並試測試您的程式：

```

02 class Caaa
03 {
04     public int num1;
05     public int num2;
06 }
07 //請於此處設計子類別Cbbb
08 public class ex10_1_1
09 {
10     public static void main(String args[])
11     {
12         Cbbb bb=new Cbbb();
13         bb.set_num(5,10);
14         bb.show();
15     }
16 }

```

```

/* output---
num1=5
num2=10
-----*/

```

18

## ex10\_1\_2.java

於下面的程式碼中，第11行嘗試存取由父類別繼承過來的私有成員n2，因而編譯時會發生錯誤。試嘗試修改這個程式，使得show() method 裡也能印出成員n2的值(n2的private屬性不能修改)：

提示:透過Caaa中新增一個shown2()的方法,以印出private的n2,並在Cbbb中呼叫shown2()的方法

```
01 // ex10_4, 簡單的繼承範例
02 class Caaa
03 { public int n1=2;
04   private int n2=4;}
05 class Cbbb extends Caaa
06 {
07   public void show()
08   {
09     System.out.println("n1="+n1+", n2="+n2); // 錯誤
10   }
11 }
12 public class ex10_4
13 {
14   public static void main(String args[])
15   {
16     Cbbb bb=new Cbbb();
17     bb.show();
18   }
19 }
```

```
/* output-----
n1=2
n2=4
-----*/
```

19

## ex10\_1\_3.java

下列程式碼的第26行嘗試建立bb物件，但編譯時發生錯誤，試指出錯誤之所在，並修改之，使得第26行可以正確的執行，且第27行的輸出為num=2。提示:請在Caaa中使用this

```
02 class Caaa
03 {
04   private int num;
05   public Caaa(int n)
06   {
07     num=n;
08   }
09   public int get()
10   {
11     return num;
12   }
13 }
14 class Cbbb extends Caaa
15 {
16   public void show()
17   {
18     System.out.println("num="+get());
19   }
20 }
21 public class ex10_1_3
22 {
23   public static void main(String args[])
24   {
25     Cbbb bb=new Cbbb();
26     bb.show();
27   }
28 }
```

```
/* output-----
num=2
-----*/
```

20

## 10.1 回家作業

### 繼承的基本概念

hw10\_1\_1.java

接續習題 ex10\_1\_1.java，並逐步完成下面的程式設計：

- 試在 Caaa 類別裡加入一個沒有引數的建構元 Caaa()，它可用來把 num1 和 num2 設定初值為 1。
- 試在 Caaa 類別裡加入另一個有引數的建構元 Caaa(int a, int b)，它可用來把 num1 設值為 a，把 num2 設值為 b。
- 試在 Cbbb 類別內裡加入建構元 Cbbb(int a, int b)，可用來呼叫父類別的建構元 Caaa(int a, int b)。
- 在 main() method 裡撰寫程式碼，並測試 (a)~(c) 的結果。

```
/* output-----
num1=1, num2=1
num1=3, num2=9
-----*/
```

21

## 10.1 回家作業

### 繼承的基本概念

hw10\_1\_2.java

假設有一 CRectangle 類別，用來表示長方形，其資料成員及成員函數如下：

```
class CRectangle
{
    private int length;
    private int width;
    private void show()
    {
        System.out.print("length="+length);
        System.out.print(", width="+width);
    }
}
```

請於 main() method 裡加入下列敘述，並完成下面的程式設計：

```
CData obj1=new CData(3,8);
CData obj2=new CData();
obj1.area();
obj2.area();
```

```
/* output-----
length=3, width=8, area=24
length=2, width=2, area=4
-----*/
```

- 試在 CRectangle 類別裡加入一個沒有引數的建構元 CRectangle()，它可用來把 length 和 width 設定初值為 2。
- 試在 CRectangle 類別裡加入另一個有引數的建構元 CRectangle(int l, int w)，它可用來把 length 設值為 l，把 width 設值為 w。
- 建立一個 CData 類別，使得 CData 繼承自 CRectangle 類別。在 CData 類別內裡加入建構元 CData(int l, int w)，可用來呼叫父類別的建構元 CRectangle(int l, int w)。
- 加入 area() method，使得它可以計算、印出長方形面積，同時印出 length 及 width 的內容。

22

## 10.2 保護成員

23

### 10.2 保護成員（protected members）

如果在子類別內直接存取private的資料成員，則在編譯時將出現錯誤。

若把[app10\\_2](#)中的28~33行改寫成如下的敘述：

完整程式碼在  
下一頁

```
28 public CCoin(double r, int v)
29 {
30     radius=r; // 錯誤,radius為private成員,無法在CCircle類別外部存取
31     value=v;
32     System.out.println("CCoin(double r, int v) constructor called");
33 }
```

編譯時將出現下列的錯誤訊息：

```
radius has private access in CCircle
radius=r;
```

24

```

01 // app10_2
02 class CCircle
03 {
04     private static double pi=3.14; // 將pi宣告成private
05     private double radius;        // 將radius宣告成private
06
07     public void show()
08     {
09         System.out.println("area="+pi*radius*radius);
10     }
11 }
12 class CCoin extends CCircle    // 定義CCoin類別，繼承自CCircle類別
13 {
14     private int value;
15
16     public CCoin(double r, int v)
17     {
18         radius=r; // 錯誤,radius為private成員,無法在CCircle類別外部存取
19         value=v;
20         System.out.println("radius="+radius+", value="+value);
21     }
22 }
23 public class app10_5
24 {
25     public static void main(String args[])
26     {
27         CCoin coin=new CCoin(2.5,10);
28         coin.show();
29     }
30 }

```

```

/* app10_5 OUTPUT-----
radius=2.5, value=10
area=19.625
-----*/

```

25

若在CCircle類別裡把radius與pi這兩個成員宣告成protected：

```

protected static double pi=3.14;
protected double radius;

```

radius與pi不僅可以在CCircle類別裡直接取用，同時也可以在繼承CCircle而來的CCoin類別裡存取。

26

下面的範例將radius與pi這兩個成員設為protected：

```

01 // app10_5, protected成員的使用
02 class CCircle
03 {
04     protected static double pi=3.14; // 將pi宣告成protected
05     protected double radius;         // 將radius宣告成protected
06
07     public void show()
08     {
09         System.out.println("area="+pi*radius*radius);
10     }
11 }
12 class CCoin extends CCircle // 定義CCoin類別，繼承自CCircle類別
13 {
14     private int value;
15
16     public CCoin(double r, int v)
17     {
18         radius=r; // 在子類別裡可直接取用父類別裡的protected成員
19         value=v;
20         System.out.println("radius="+radius+", value="+value);
21     }
22 }
23 public class app10_5
24 {
25     public static void main(String args[])
26     {
27         CCoin coin=new CCoin(2.5,10);
28         coin.show();
29     }
30 }

```

```

/* app10_5 OUTPUT-----
radius=2.5, value=10
area=19.625
-----*/

```

27

## 思考

如果本題中num1與num2均宣告成protected，則於本題中，您撰寫的set\_num()與show() method 是否可以存取到它們？為什麼？

```

02 class Caaa
03 {
04     public int num1;
05     public int num2;
06 }
07
08 class Cbbb extends Caaa
09 {
10     public void set_num(int a,int b)
11     {
12         num1=a;
13         num2=b;
14     }
15     public void show()
16     {
17         System.out.println("num1="+num1);
18         System.out.println("num2="+num2);
19 }

```

```

22 public class ex10_2
23 {
24     public static void main(String args[])
25     {
26         Cbbb bb=new Cbbb();
27         bb.set_num(5,10);
28         bb.show();
29     }
30 }

```

28

## 再思考

您覺得把n1和n2宣告成public較好，還是protected？

```

01 // ex10_7,
02 class Caaa
03 {
04     protected int n1=2;
05     protected int n2=4;
06 }
07 class Cbbb extends Caaa
08 {
09     public void show()
10 {
11     System.out.println("n1="+n1+", n2="+n2);
12 }
13 }
14 public class ex10_7
15 {
16     public static void main(String args[])
17 {
18     Cbbb bb=new Cbbb();
19     bb.show();
20 }
21 }

```

29

## ex10\_2\_1.java

假設有一 CTriangle 類別,用來表示三角形,其資料成員及成員函數如下:

```

class CTriangle
{
    protected int base;
    protected int height;
    protected void show()
    {
        System.out.println("base="+base+", height="+height);
    }
}

```

- (a) 試建立 CData 類別,繼承自 CTriangle 類別。
- (b) 試在 CData 類別裡加入 area() method,可以利用 show() 印出三角形的 base 及height 成員,再印出三角形的面積。
- (c) 試於 main() method 裡加入下列敘述:

```

CData obj=new CData(3,8);
obj.area();

```

```

/* output-----
base=3, height=8
area=12.0
-----*/

```

30

## 10.2 回家作業 保護成員

hw10\_2\_1.java

撰寫程式碼完成下列敘述：

- (a) 定義一個矩形類別CRectangle，含有 length、width 兩個保護資料成員，以及計算面積的方法CalArea()。
- (b) 定義一個立方體類別CCube繼承矩形類別CRectangle的所有成員，並新增一個求面積的方法GetFaceArea()及一私有資料成員height。
- (c) 在主程式main()中，分別建立CRectangle類別物件與CCube類別物件，並分別顯示矩形的面積與立方體的表面積。

```
/* output-----  
長= 20, 寬= 10  
矩形面積=200  
長= 20, 寬= 10,高 = 5  
立方體面積=700  
-----*/
```

31

## 10.3 改寫

32



## 10.3 改寫

改寫（overriding）與多載，均是多型（polymorphism）的技術之一。

### 10.3.1 改寫父類別的method

下面是改寫父類別之method的範例：

33

```

01 // app10_6, method的「改寫」範例
02 class CCircle // 父類別CCircle
03 {
04     protected static double pi=3.14;
05     protected double radius;
06
07     public CCircle(double r)
08     {
09         radius=r;
10     }
11     public void show() // 父類別裡的show() method
12     {
13         System.out.println("radius="+radius);
14     }
15 }
16

```

```

/* app10_6 OUTPUT---
radius=2.0, value=5
-----*/

```

```

17 class CCoin extends CCircle // 子類別CCoin
18 {
19     private int value;
20
21     public CCoin(double r,int v)
22     {
23         super(r);
24         value=v;
25     }
26     public void show() // 子類別裡的show() method
27     {
28         System.out.println("radius="+radius+", value="+value);
29     }
30 }
31
32 public class app10_6
33 {
34     public static void main(String args[])
35     {
36         CCoin coin=new CCoin(2.0,5);
37         coin.show(); // 呼叫show() method
38     }
39 }

```

34

## 「改寫」與「多載」的比較

「多載」：

**overloading**，它是在相同類別內，定義名稱相同，但引數個數或型態不同的**method**，Java依據引數的個數或型態，呼叫相對應的**method**。

「改寫」：

**overriding**，它是在子類別當中，定義名稱、引數個數與傳回值的型態均與父類別相同的**method**，用以改寫父類別裡**method**的功用。

35

### 10.3.2 以父類別的變數存取子類別物件的成員

下面的程式碼，是透過父類別變數cir呼叫show() method：

```
01 // app10_7, 透過父類別變數cir呼叫show() method
02 class CCircle // 父類別CCircle
03 {
04     protected static double pi=3.14;
05     protected double radius;
06
07     public CCircle(double r)
08     {
09         radius=r;
10     }
11     public void show() // 父類別裡的show() method
12     {
13         System.out.println("radius="+radius);
14     }
15 }
```

父類存取子類的物件成員，僅限於改寫的情況發生，亦即父類與子類的method name,引述個數,型態要完全相同,才可以透過父呼叫子的method.

```
16 class CCoin extends CCircle // 子類別CCoin
17 {
18     private int value;
19
20     public CCoin(double r,int v)
21     {
22         super(r);
23         value=v;
24     }
25     public void show() // 子類別裡的show() method
26     {
27         System.out.println("radius="+radius+", value="+value);
28     }
29     public void showValue() // showValue() method,此函數只存在於子類別
30     {
31         System.out.println("value="+value);
32     }
33 }
```

```
34 public class app10_7
35 {
36     public static void main(String args[])
37     {
38         CCircle cir=new CCoin(2.0,5); // 宣告父類別變數cir，並將它指向物件
39         cir.show(); // 利用父類別變數cir呼叫show()
40         // cir.showValue(); 若某個method只存在於子類，則無法存取。
41     }
42 }
```

```
/* app10_7 OUTPUT---
radius=2.0, value=5
-----*/
```

36

## 透過父類別存取子類別物件的成員，有什麼好處呢？

- app10\_7 只建立一個子類別的物件，好處不明顯。但若好幾個子類別繼承自同一個父類別，共同改寫父類別的某個函數時，即可看到其好處。
- 以父類別存取子類別物件的成員，可以有效簡化程式碼，尤其是在許多子類別繼承自同一個父類別時。
- (此部分之練習在兩題回家作業中可看出差異)

37

## ex10\_3\_1

完成下面的程式設計：

- 在父類別Caaa 裡加入一個display() method，它可顯示出 "printed from Caaa class" 字串。
- 在子類別Cbbb 裡加入display() method，用來改寫父類別的display()。子類別的 display() method 可用來顯示出 "printed from Cbbb class" 字串。
- 接續 (a) 與 (b)，如果我們以下面的敘述建立物件bb  
`Caaa bb=new Cbbb();`  
 則以物件bb 呼叫display() method 時，則是父類別的display() method 會被呼叫，還是子類別的display() method 會被呼叫？

```
/* output-----
printed from Cbbb class
-----*/
```

38

## ex10\_3\_2.java

請逐行瞭解下面的程式碼，並編譯之。編譯之後，你會得到哪些錯誤？請試著由錯誤訊息中指出錯誤之所在：

```
02 class Caaa{}
03 class Cbbb extends Caaa{}
04 class Cccc extends Cbbb{}
05 public class ex10_3_2
06 {
07     public static void main(String args[])
08     {
09         Cbbb b1=new Cbbb();
10         Cbbb b2=new Cccc();
11         Cccc c1=new Caaa();
12         Cccc c2=new Cbbb();
13     }
14 }
```

39

## 10.3 回家作業 改寫

hw10\_3\_1.java

下面的程式碼定義了 CShape 父類別，請先閱讀它，然後回答接續的問題：

```
class CShape // 父類別 CShape
{
    public double area()
    {
        return 0.0;
    }
}
```

```
/* output-----
circle(1.0)=3.14
circle(2.0)=12.56
square(1.0)=1.0
square(2.0)=4.0
triangle(3,6)=9.0
triangle(4,10)=20.0
The largest area is 20.0
-----*/
```

- 試定義一個圓形類別 CCircle，它繼承自 CShape 類別，並改寫了父類別的 area() method，可用來傳回圓形物件的面積。
- 試定義一個正方形類別 CSquare，它也繼承自 CShape 類別，同時改寫了父類別的 area() method，可用來傳回正方形物件的面積。
- 試定義一個三角形類別 CTriangle，它也繼承自 CShape 類別，同時改寫了父類別的 area() method，可用來傳回三角形物件的面積。
- 試利用 (a)~(c) 所定義的類別，建立圓形、正方形與三角形物件各兩個，其中所有的引數（如半徑、邊長或三角形的底和高等）請自行設定。
- 試撰寫一個 largest() method，可用來找出 (d) 中的所有物件裡，面積的最大值。於本題中，您是否察覺到在找出最大的面積時，在程式的撰寫上有些許不便？

40

## 10.3 回家作業 改寫

hw10\_3\_2.java

試修改習題hw10\_3\_1.java裡的 (d) 與 (e) 小題，使得於 (d) 中，所有的子類別物件均是由父類別的陣列變數指向它。與習題 hw10\_3\_1.java 相比，您覺得以父類別的變數存取子類別物件的成員，可帶來哪些好處？

```
/* output-----  
area of shp[0]=3.14  
area of shp[1]=12.56  
area of shp[2]=1.0  
area of shp[3]=4.0  
area of shp[4]=9.0  
area of shp[5]=20.0  
The largest area is 20.0  
-----*/
```

42

## 10.4 再談super and this

44

## 10.4 再談super 與this

super後面也可加上資料成員或method的名稱：

super.資料成員名稱    // 存取父類別的資料成員

super.method名稱    // 存取父類別的method

45

下面的範例說明了super關鍵字的用法：

```

01 // app10_8, 透過super關鍵字來存取父類別的變數
02 class Caaa
03 {
04     protected int num; // 父類別的資料成員num
05
06     public void show()
07     {
08         System.out.println("Caaa_num="+num);
09     }
10 }
11 class Cbbb extends Caaa
12 {
13     int num=10; // 子類別的資料成員num
14
15     public void show()
16     {
17         super.num=20; // 設定父類別的資料成員num為20
18         System.out.println("Cbbb_num="+num);
19         super.show(); // 呼叫父類別的show() method
20     }
21 }
22
23 public class app10_8
24 {
25     public static void main(String args[])
26     {
27         Cbbb b=new Cbbb();
28         b.show();
29     }

```

```

/* app10_8 OUTPUT--
Cbbb_num=10
Caaa_num=20
-----*/

```

46

下面的範例是利用**this**來呼叫**實例變數**：

```

01 // app10_9, 用this來呼叫實例變數
02 class Caaa
03 {
04     public int num=10; // num是實例變數
05
06     public void show()
07     {
08         int num=5; // num是區域變數，其有效範圍僅限於在show()內
09         System.out.println("this.num="+this.num); // 印出實例變數
10         System.out.println("num="+num); // 印出區域變數
11     }
12 }
13 public class app10_9
14 {
15     public static void main(String args[])
16     {
17         Caaa a=new Caaa();
18         a.show();
19     }
20 }

```

**\*\*Instance variable(實例變數):** 物件各自儲存資料成員的記憶體空間，不與其它物件共用。

**\*\*static variable(類別變數):** static的宣告，所有物件共享

```

/* app10_9 OUTPUT---
this.num=10
num=5
-----*/

```

47

### ex10\_4\_1.java

請試著在子類別**Cbbb**裡撰寫程式碼，呼叫父類別裡沒有引數的建構元**Caaa()**來設定初值。

```

02 class Caaa
03 {
04     public int num1;
05     public int num2;
07     public Caaa()
08     {
09         num1=1;
10         num2=1;
11     }
12     public Caaa(int a,int b)
13     {
14         num1=a;
15         num2=b;
16     }
17 }
19 class Cbbb extends Caaa
20 {
21     //此處撰寫程式碼
27     public void show()
28     {
29         System.out.println("num1="+num1+",
num2="+num2);
30     }
31 }
33 public class ex10_4_1
34 {
35     public static void main(String args[])
36     {
37         Cbbb b1=new Cbbb();
38         b1.show();
39     }
40 }

```

```

/* output-----
num1=1, num2=1
-----*/

```

48

## ex10\_4\_2.java

請在子類別Cbbb 裡撰寫程式碼，呼叫父類別裡有引數的建構元Caaa(int a, int b) 來設定初值，並測試之。

```

02 class Caaa                                19 class Cbbb extends Caaa
03 {                                          20 {
04 public int num1;                          //此處撰寫程式碼
05 public int num2;                          27 public void show()
07 public Caaa()                             28 {
08 {                                          29 System.out.println("num1="+num1+",
09 num1=1;                                  num2="+num2);
10 num2=1;                                  30 }
11 }                                          31 }
12 public Caaa(int a,int b)                 32
13 {                                          33 public class ex10_3
14 num1=a;                                  34 {
15 num2=b;                                  35 public static void main(String args[])
16 }                                          36 {
17 }                                          //此處撰寫程式碼
                                           42 }
                                           43 }
                                           /* output-----
                                           num1=3, num2=9
                                           -----*/

```

49

## 10.4 回家作業

### super and this

## hw10\_4\_1.java

下面的程式碼定義了CAdd父類別，請先閱讀它，然後完成題目的要求：

```

class CAdd
{
    public int sum=0;
    public void add()
    {
        for(int i=1;i<=10;i++)
            sum+=i;
        System.out.println("1+2+...+10="+sum);
    }
}

```

```

/* output-----
1+2+...+10=55
1+2+...+15=120
-----*/

```

(a) 試定義一個類別CAddto，它繼承自CAdd類別，並改寫了父類別的add() method，可用來計算並列印出1+2+...+n的結果。

(b) 如果我們用下面的敘述建立物件ad1、ad2

```

CAddto ad1=new CAddto();
CAddto ad2=new CAddto(15);

```

並以物件ad1、ad2呼叫add() method時，希望能出現如下程式的執行結果，請將完整的程式寫出。



## 10.4 回家作業 super and this

hw10\_4\_2.java

下面的程式碼定義了CMux父類別，請先閱讀它，然後完成題目的要求：

```
class Mux
{
    public int sum=1;
    public void mux()
    {
        for(int i=1;i<=5;i++)
            sum*=i;
        System.out.println("1*2*3*...*5="+sum);
    } }
```

```
/* output-----
1*2*3*...*4=24
1*2*3*...*6=720
-----*/
```

(a) 是定義一個類別CMuxto，它繼承自CMux類別，並改寫了父類別的mux() method，可用來計算並列印出1\*2\*...\*n的結果。

(b) 如果我們用下面的敘述建立物件ad1、ad2

```
CMuxto ad1=new CMuxto();
```

```
CMuxto ad2=new CMuxto(6);
```

並以物件ad1、ad2呼叫mux() method時，希望能出現如下程式的執行結果，請將完整的<sub>51</sub>程式寫出。

## 10.5 設定終止

### 10.5 設定終止繼承

如果想設定終止繼承，可利用**final**關鍵字，如下面的範例：

```

01 // app10_10, 設定終止繼承
02 class Caaa
03 {
04     public final void show()    // 父類別的show()已被設為終止繼承
05     {
06         System.out.println("show() method in class Caaa called");
07     }
08 }
09 class Cbbb extends Caaa
10 {
11     public void show()          // 錯誤，改寫父類別的show() method
12     {
13         System.out.println("show() method in class Cbbb called");
14     }
15 }
16 public class app10_10
17 {
18     public static void main(String args[])
19     {
20         Cbbb b=new Cbbb();
21         b.show();
22     }
23 }

```

於app10\_10，在程式編譯時將產生如下的錯誤訊息：

```
show() in Cbbb cannot override show() in Caaa; overridden method is final
```

53

### final的其它用法：

**static final double** PI=3.14; // 設定PI值不能再被修改

**final class** CCircle // 設定CCircle類別不能被其它類別繼承

54

想一想...下述程式之執行結果？

```

02 class Caaa
03 {
04     public final int num1;
05     public final int num2;
06 }
07
08 class Cbbb extends Caaa
09 {
10     public void set_num(int a,int b)
11     {
12         num1=a;
13         num2=b;
14     }
15     public void show()
16     {
17         System.out.println("num1="+num1);
18         System.out.println("num2="+num2);
19     }}
22 public class ex10_5_1
23 {
24     public static void main(String args[])
25     {
26         Cbbb bb=new Cbbb();
27         bb.set_num(5,10);
28         bb.show();
29     }
30 }

```

無法繼承num1 and num2  
因為變數已經設定成final

55

```

package e;
package ex;
class Caaa
{
    public int num1;
    public int num2;
}

class Cbbb extends Caaa
{
    /*public void set_num(int a,int b)
    {
        num1=a;
        num2=b;
    } */
    public void show()
    {
        System.out.println("num1="+num1);
        System.out.println("num2="+num2);
    }}
public class ex
{
    public static void main(String args[])
    {
        Cbbb bb=new Cbbb();
        //bb.set_num(5,10);
        bb.show();
    }
}

```

num1=0  
num2=0

56

## ex10\_5\_1.java

假設有一Car 類別，用來表示車子的資訊，其資料成員及成員函數如下：

```
class Car
{
    protected String owner;
    protected String id;
    final void show()
{
    System.out.println("車主姓名:"+owner);
    System.out.println("車牌號碼:"+id);
}}
```

請於main() method 裡加入下列敘述，並完成下面的程式設計：

```
CColor mycar=new CColor("Riaan","A1-2345","Black");
mycar.show();
```

```
/* output-----
車主姓名:Riaan
車牌號碼:A1-2345
車身顏色:Black
-----*/
```

- 試在Car 類別裡加入一個有引數的建構元 Car(String own, String s)，它可用來owner 設值為 own，把id 設值為s。
  - 建立一個CColor 類別，使得CColor 繼承自Car 類別，資料成員為String 型態的color用來記錄車子的顏色。
  - 在CColor 類別內裡加入建構元CColor(String own, String s, String col)，可用來呼叫父類別的建構元 Car(String own, String s)。
  - 在CColor 類別裡加入show() method，用來改寫父類別的show() method。
- ```
public void show()
{
    System.out.println("車主姓名:"+owner);
    System.out.println("車牌號碼:"+id);
    System.out.println("車身顏色"+color);
}
```
- 程式編譯後，發生了什麼問題？在不更動父類別的程式敘述的原則下，該如何解決，使得程式可以執行？

57

## 10.6 類別之源: object類別

58

### 10.6 類別之源--Object類別

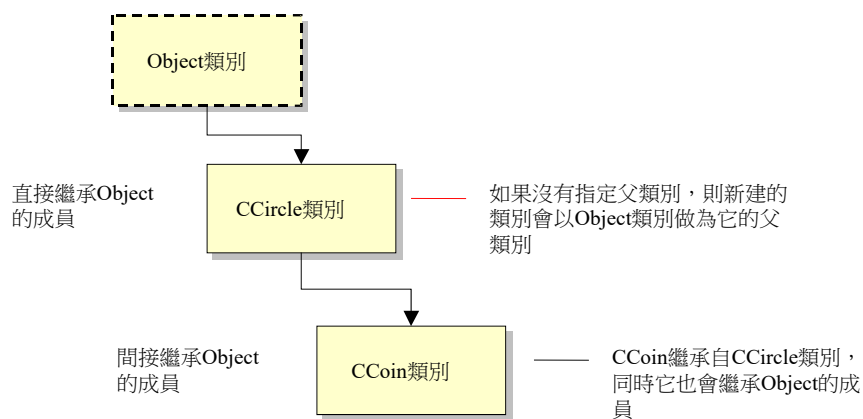
如果沒指定父類別，則Java會自動設定Object這個類別為它的父類別。

```
class CCircle
{
    ...
}
```

當沒有指定父類別時，便會以 `java.lang.Object` 類別做為它的父類別，而自己變成它的子類別。

59

Object類別可說是類別之源，所有的類別均直接或間接繼承它：



60

下表列舉了Object類別裡三個常用的method：

表10.6.1 Object類別裡常用的method

| Method名稱                   | 功能說明                    |
|----------------------------|-------------------------|
| Class getClass()           | 取得呼叫getClass() 的物件所屬之類別 |
| Boolean equals(Object obj) | 兩個類別變數所指向的是否為同一個物件      |
| String toString()          | 將呼叫toString() 的物件轉成字串   |

### getClass() method的使用

想知道某個物件obj是屬於哪個類別時，可用：

obj.getClass()

的語法來查詢。

61

下面的程式是getClass() 簡單的使用範例：

```

01 // app10_11, 利用getClass()取得呼叫物件所屬的類別
02 class Caaa // 定義Caaa類別
03 {
04     private int num;
05
06     public Caaa(int n)
07     {
08         num=n;
09     }
10 }
11 public class app10_11
12 {
13     public static void main(String args[])
14     {
15         Caaa a=new Caaa(5);
16         Class ca=a.getClass(); // 用變數a呼叫getClass() 找出a物件是屬於哪一個class
17         System.out.println("Class of obj = "+ca);
18     }
19 }

```

```

/* app10_11 OUTPUT-----
Class of obj = class Caaa
-----*/

```

62

**equals() method的使用**

**equals() method**可用來比較兩個類別變數是否指向同一個物件：

```

01 // app10_12, 利用equals()來判別兩個類別變數所指向的是否為同一個物件
02 class Caaa // 定義Caaa類別
03 {
04     private int num;
05
06     public Caaa(int n)
07     {
08         num=n;
09     }
10 }
11 public class app10_12
12 {
13     public static void main(String args[])
14     {
15         Caaa a=new Caaa(5);
16         Caaa b=new Caaa(5);
17         Caaa c=a; // 宣告類別變數c，並讓它指向變數a所指向的物件
18         boolean br1=a.equals(b); // 測試a與b是否指向同一物件
19         boolean br2=a.equals(c); // 測試a與c是否指向同一物件
20         System.out.println("a.equals(b)="+br1);
21         System.out.println("a.equals(c)="+br2);
22     }
23 }

```

```

/* app10_12 OUTPUT--
a.equals(b)=false
a.equals(c)=true
-----*/

```

63

**toString() method的使用**

**toString()** 的功用是將物件的內容轉換成字串，如下面的範例：

```

01 // app10_13, Object類別裡的toString() method
02 class Caaa
03 {
04     private int num;
05
06     public Caaa(int n)
07     {
08         num=n;
09     }
10 }
11 public class app10_13
12 {
13     public static void main(String args[])
14     {
15         Caaa a=new Caaa(2);
16         System.out.println(a.toString()); // 印出物件a的內容
17     }
18 }

```

```

/* app10_13 OUTPUT--
Caaa@5d87b2
-----*/

```

68

改寫toString() method的方法如下面的範例：

```

01 // app10_14, 改寫Object類別裡的toString() method
02 class Caaa
03 {
04     private int num;
05
06     public Caaa(int n)
07     {
08         num=n;
09     }
10     public String toString() // 改寫toString() method
11     {
12         String str="toString() called, num="+num;
13         return str;
14     }
15 }
16 public class app10_14
17 {
18     public static void main(String args[])
19     {
20         Caaa a=new Caaa(2);
21         System.out.println(a.toString()); // 印出物件a的內容
22     }
23 }

```

```

/* app10_14 OUTPUT-----
toString() called, num=2
-----*/

```

69

## ex10\_6\_1.java

試修改app10\_5，在子類別CCoin 中加入toString() method，使得當子類別的物件coin呼叫toString() 時，會印出"radius=xxx, value=yyy" 字串，其中xxx 與yyy分別代表物件coin 資料成員radius 與value 的值。

```

/* output-----
radius=2.5, value=10
-----*/

```

70



## 10.6 回家作業

### 類別之源—Object 類別

hw10\_6\_1.java

在習題 ex10\_6\_1 中，子類別 CCoin 裡的 toString() method 改寫了父類別 Object 的 toString() method。試修改習題，使得在 main() method 裡以子類別的物件 coin 呼叫 toString() method 時，必須滿足下面 (a) 與 (b) 兩種狀況：

- (a) 當 toString() 裡沒有任何引數時，則習題中，子類別 CCoin 裡所定義的 toString() method 會被呼叫。
- (b) 當 toString() 裡的引數為整數 0 時，則父類別 CCircle 裡繼承類別 Object 而來的 toString() method 會被呼叫；若引數不為整數 0，則傳回 "Argument Error" 字串。
- (c) 於 (a) 和 (b) 中，哪一個 toString() method 改寫了父類別 CCircle 的 toString() method？
- (d) 於 (a) 和 (b) 中所定義的兩個 toString() method 得以同時存在於一個類別內，是利用 OOP 裡的哪一種技術？

```
/* output-----
radius=2.5, value=10
CCoin@d9f9c3
-----*/
```

71