

## 第二章 类别的设计与对象的使用

### 本章说明与大纲

面向对象的观念是从类别实作对象，再利用其方法与属性。

- 除了只有一個 class 可以有 main()方法外，其他的類別只能有一個和類別名稱一樣的方法，此方法稱之為**建構子**。
- 要電腦聽命作事的步驟要放在**建構子**中。
- 請記住名言『作事需要方法』。
- 利用上↑下↓鍵可以重複使用在命令提示字元已經打入的指令。

### 2.1 编辑第二个程序

第一步：执行 WordPad 并将路径选择在 c:\Myjava，打开第一个程序 Myfirstjava.java 内容如下

```
class Myfirstjava {  
    void main()  
    {  
        System.out.println("My first java program!");  
    }  
}
```

第二步：修改程序，将 first 改为 second

```
class Mysecondjava {  
    void main() {  
        System.out.println("My first second  
program!");  
    }  
}
```

第三步：另存新文件并将路径选择在 c:\Myjava，将档名改为 Mysecondjava.java

注意：檔名 Mysecondjava 与类别名称 Mysecondjava 一致，扩展名为 java。

### 2.2 继续编辑程序\_作加法运算

目的: 计算两个数字的相加, 例如  $x=1, y=2$  则  $z=x+y$  结果  $z=3$ 。

第一步: 修改程序, 增加一行指令 `x=1;`

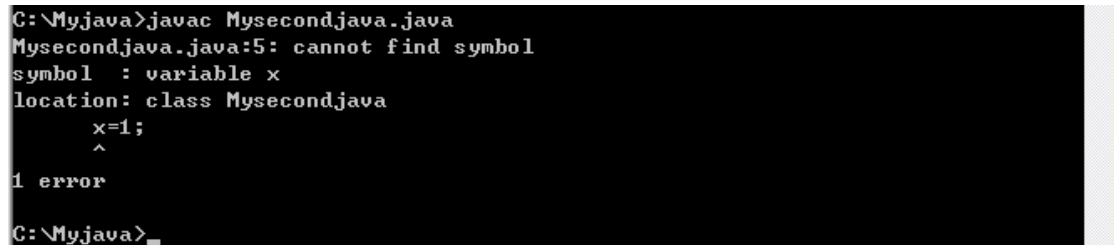
```
class Mysecondjava {  
    void main()  
    {  
        x=1;  
        System.out.println("My first java program!");  
    }  
}
```

第二步: 储存档案(档案→储存档案)

第三步: 在命令提示字符中打字 `cd c:\Myjava`

第四步: 在命令提示字符中打字 `javac MySecondjava.java`

结果出现错误讯息



```
C:\Myjava>javac Mysecondjava.java  
Mysecondjava.java:5: cannot find symbol  
symbol : variable x  
location: class Mysecondjava  
    x=1;  
    ^  
1 error  
C:\Myjava>
```

找不到 `x`

想一想: 为什么找不到 `x`, 我们已经打了 `x=1`, 为什么找不到 `x`?

原因: 计算机是到内存空间中去找变量 `x`, 可是我们没有指定一个空间给 `x`。

## 2.3 继续编辑程序\_分配内存空间

第一步: 修改程序, 增加一行指令 `int x;`

```
class Mysecondjava {  
    void main()  
    {  
        int x;  
        x=1;  
        System.out.println("My second java program!");  
    }  
}
```

第二步: 储存档案(档案→储存档案)

第三步: 在命令提示字符中打字 `javac Mysecondjava.java`

```
C:\Myjava>javac Mysecondjava.java
```

```
C:\Myjava>
```

如果没有错误讯息恭喜您！

第四步：在命令提示字符中打字 java Mysecondjava

```
C:\Myjava>java Mysecondjava
```

```
Exception in thread "main" java.lang.NoSuchMethodError: main
```

```
C:\Myjava>
```

有错误讯息！和 main 有关

想一想：为什么和 main 有关？

原 因：main 是一种必须有自变量的方法

## 2.4继续编辑程序\_main(String args[])

第一步：修改程序，修改第二行指令 public static void main(String args[])

```
class Mysecondjava {  
    public static void main(String args[])  
    {  
        int x;  
        x=1;  
        System.out.println("My second java program!");  
    }  
}
```

第二步：在命令提示字符中打字 javac Mysecondjava.java

```
C:\Myjava>javac Mysecondjava.java
```

```
C:\Myjava>
```

如果没有错误讯息恭喜您！

第四步：在命令提示字符中打字 java Mysecondjava

```
C:\Myjava>java Mysecondjava
```

```
My second java program!
```

```
C:\Myjava>
```

如果出现讯息 My second java program! 恭喜您！

## 2.5继续编辑程序\_分配内存空间

第一步：修改程序，新增两行指令 int y;以及 int z;

```
class Mysecondjava {  
    public static void main(String args[])  
    {  
        int x;
```

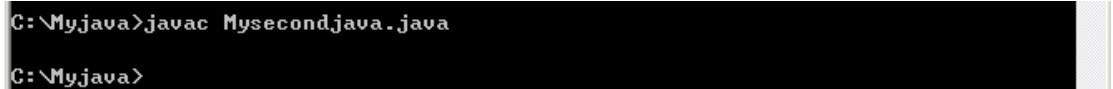
```
int y;  
int z;  
x=1;  
System.out.println("My second java program!");  
}  
}
```

第二步：修改程序，新增两行指令 `y=2;`以及 `z=x+y;`

```
class Mysecondjava {  
    public static void main(String args[])  
    {  
        int x;  
        int y;  
        int z;  
        x=1;  
        y=2;  
        z=x+y;  
        System.out.println("My second java program!");  
    }  
}
```

第三步：储存档案

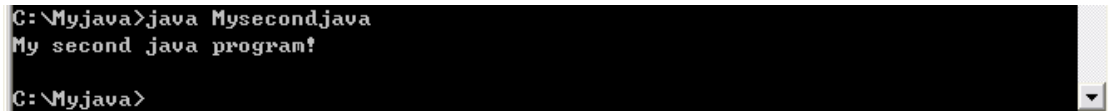
第四步：在命令提示字符中打上 `javac Mysecondjava.java`



```
C:\Myjava>javac Mysecondjava.java  
C:\Myjava>
```

如果没有错误讯息恭喜您！

第五步：在命令提示字符中打上 `java Mysecondjava`



```
C:\Myjava>java Mysecondjava  
My second java program!  
C:\Myjava>
```

想一想：为什么没有看到作加法运算之后的结果呢？

原因：因为没有把 `z` 打印出来，所以看不到结果。

## 2.6继续编辑程序\_打印结果

第一步：修改程序，把 `System.out.println("My second java program!");`

改成 `System.out.println("My second java program!" + z);`

```
class Mysecondjava {  
    public static void main(String args[])  
    {  
        int x;  
        int y;  
        int z;  
        x=1;  
        y=2;  
        z=x+y;
```

```

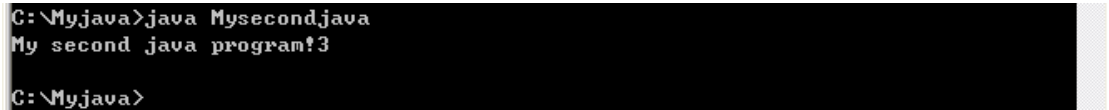
        System.out.println("My second java program!" + z);
    }
}

```

第二步：储存档案，并在命令提示字符中打上 `javac Mysecondjava.java`

如果没有错误讯息恭喜您！

第三步：在命令提示字符中打上 `java Mysecondjava`



```

C:\Myjava>java Mysecondjava
My second java program!3
C:\Myjava>

```

如果出现讯息 `My second java program!3` 恭喜您！

## 2.7 继续编辑程序\_类别

第一步：修改程序，在 `class Mysecondjava` 外面，新增一个类别 `class Addop`

```

class Mysecondjava {
    public static void main(String args[])
    {
        int x;
        int y;
        int z;
        x=1;
        y=2;
        z=x+y;
        System.out.println("My second java program!" + z);
    }
}

class Addop {

}

```

第二步：每个class都会有一个跟class一样名称的方法，此方法称为建构子

修改程序，在 `class Addop` 里面新增一行程序 `Addop()`，只要是方法就必须给予一组大括号“{}”

```

class Addop {
    Addop() {

    }

}

```

第三步：然后把 `int x;`、`int y;`、`int z;` 剪下，然后建构子之前一行贴上

```

class Addop {
    int x;
    int y;
    int z;
    Addop() {

    }
}

```

第四步：然后把x=1; 、 y=2; 、 z=x+y;剪下，然后在建构子里面贴上

```

class Addop {
    int x;
    int y;
    int z;
    Addop() {
        x=1;
        y=2;
        z=x+y;
    }
}

```

第五步：接着为了要印出结果，把System.out.println("My second java program!"+z);复制到建构子里面，并把main里面的System.out.println("My second java program!"+z);改成System.out.println("My second java program!");，然后把建构子里面的System.out.println("My second java program!"+z);改成System.out.println("z="+z);

```

class Mysecondjava {
    public static void main(String args[])
    {
        System.out.println("My second java program!");
    }
}

class Addop {
    int x;
    int y;
    int z;
    Addop() {
        x=1;
        y=2;
        z=x+y;
    }
}

```

```

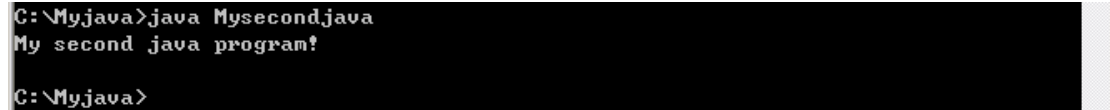
        System.out.println("z="+z);
    }
}

```

第六步：储存档案，并在命令提示字符中打上 `javac Mysecondjava.java`

如果没有错误讯息恭喜您！

第七步：在命令提示字符中打上 `java Mysecondjava`



```

C:\Myjava>java Mysecondjava
My second java program!
C:\Myjava>

```

想一想：为什么没有看到 `z=3` 呢？而只有看到 `My second java program!`

原因：没有实作 `Addop` 类别的对象，所以没有执行到里面的程序

## 2.8 继续编辑程序\_实作对象

第一步：修改程序，新增一行指令 `Addop result = new Addop();`

```

class Mysecondjava {
    public static void main(String args[])
    {
        Addop result = new Addop();
        System.out.println("My second java program!");
    }
}

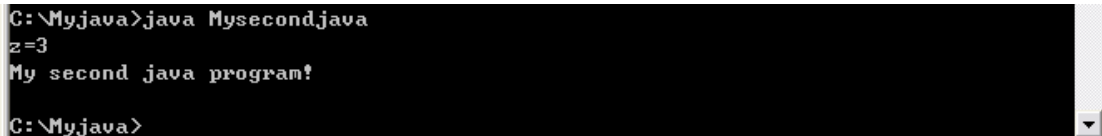
```

利用这个指令实作了一个对象 `result`

第二步：储存档案，并在命令提示字符中打上 `javac Mysecondjava.java`

如果没有错误讯息恭喜您！

第三步：在命令提示字符中打上 `java Mysecondjava`



```

C:\Myjava>java Mysecondjava
z=3
My second java program!
C:\Myjava>

```

## 2.9 继续编辑程序\_实作对象

在 `Addop` 类别里面的变量称为 `Addop` 类别的属性

第一步：修改程序，把 `System.out.println("My second java program!");` 改成

`System.out.println("My second java program!" + result.z);`

```

class Mysecondjava {
    public static void main(String args[])
    {

```



```

Addop result = new Addop();
System.out.println("My second java program!" + result.z);
}
}

```

result.z 这个指令的"."相对于中文的"的"也就是呼叫 result 这个对象的 z 属性

第二步：储存档案，并在命令提示字符中打上 javac Mysecondjava.java

如果没有错误讯息恭喜您！

第三步：在命令提示字符中打上 java Mysecondjava



恭喜您！

## 2.10 批注与说明：

- 第二个程序代表java程序的最简单形式！
- 最简单形式的java程序内容包含 class 文件名 及 一对大括号{}。在这一对大括号{}中，必须有一个没有回传数值的 void 方法 main()及另一对大括号{}，在这一对大括号{}中要有指令并以分号;做结尾。
- System.out.println("My first java program!");是输出一个字符串My first java program! 的指令。

## 2.11 补充说明

- 利用上↑下↓键可以重复使用在命令提示字符已经打入的指令。

## 2.12 本章重点回顾

1. 本章说明了利用 Java 类别的**建构子**来建构对象，并打印出对象的属性。

## 2.13习题

1. 设计class Multi() 可以计算两个数的乘法并将结果印出。