**Java 封装**

在面向对象程式设计方法中，封装（英语：Encapsulation）是指，一种将抽象性函式接口的实作细节部份包装、隐藏起来的方法。

封装可以被认为是一个保护屏障，防止该类的代码和数据被外部类定义的代码随机访问。

要访问该类的代码和数据，必须通过严格的接口控制。

封装最主要的功能在于我们能修改自己的实现代码，而不用修改那些调用我们代码的程序片段。

适当的封装可以让程式码更容易理解与维护，也加强了程式码的安全性。

**实例**

让我们来看一个java封装类的例子：

/\* 文件名: EncapTest.java \*/

public class EncapTest{

private String name;

private String idNum;

private int age;

public int getAge(){

return age;

}

public String getName(){

return name;

}

public String getIdNum(){

return idNum;

}

public void setAge( int newAge){

age = newAge;

}

public void setName(String newName){

name = newName;

}

public void setIdNum( String newId){

idNum = newId;

}

}

以上实例中public方法是外部类访问该类成员变量的入口。

通常情况下，这些方法被称为getter和setter方法。

因此，任何要访问类中私有成员变量的类都要通过这些getter和setter方法。

通过如下的例子说明EncapTest类的变量怎样被访问：

/\* F文件名 : RunEncap.java \*/

public class RunEncap{

public static void main(String args[]){

EncapTest encap = new EncapTest();

encap.setName("James");

encap.setAge(20);

encap.setIdNum("12343ms");

System.out.print("Name : " + encap.getName()+

" Age : "+ encap.getAge());

}

}

以上代码编译运行结果如下:

Name : James Age : 20