

专业语言:JAVA的类和接口 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/273/2021\\_2022\\_\\_E4\\_B8\\_93\\_E4\\_B8\\_9A\\_E8\\_AF\\_AD\\_E8\\_c104\\_273309.htm](https://www.100test.com/kao_ti2020/273/2021_2022__E4_B8_93_E4_B8_9A_E8_AF_AD_E8_c104_273309.htm)

Java是一种面向对象(OOP)的编程语言；在Java中，面向对象是以类的形式出现的，而对象则是类的实例，所以要学习Java的类首先必须了解面向对象。面向对象编程不同于过程语言或结构化程序设计，他存在对象这个概念，面向对象中的对象不仅是数据结构，还包含和数据结构有关的方法。

一：类

1.类的声明及实例化

类是面向对象关键，可以说没有类就没有对象；类具有两个优点，首先，他允许封装；即将数据和方法封装起来使得这方面的信息和其他程序有效的分开，封装的好处是如果修改了这个类，只要没改变这个类中的方法就不需要动别的程序；类的第二个优点是允许继承。类由两部分组成，第一部分是类的变量的声明，到二部分类中的方法；以下是创建一个叫test的类：

```
public class test{ //声明一个类
String name. //类的变量的声明
changtest(){ //类的方法 name="text". }
```

在上面声明一个类时使用了修饰符"public".在声明类时可以使用三中说明符：public,final和abstract。public类意味这个类可以被任何对象使用和扩展，而且与他的包无关；final类则表明这个类没有任何子类；abstract类即说明这个类是一个抽象类；如果类声明前没有任何修饰符，则默认为protected类，即只有同一个包中的对象才能使用该类。前面已经说了类有个优点是允许继承，这就说明新建的类可以使用已建类的变量和方法，这样通过基于简单的类建立较大的类的方法，可以减少许多代码的编写，也减少了在程序中查找和修改错误的工作；以

下例子建立一个类MyClass，他继承了类test中所有的变量和方法：`public class MyClass extends test{}` 其中类MyClass被称为子类，类test被称为父类或超类。当创建了类后，就可以用new来创建类的一个实例，并不是所有的类都需要创建他的实例，如果要使用的类中的方法或变量是静态的，就可以不用创建这个类的实例，但是静态的方法只能引用静态的类变量(System类的out变量是一个例外)；以下例子创建类test的一个实例：`test mytest=new test()`。创建类的实例是为了要引用这个类中的变量或方法，前面已经为类test创建了一个实例mytest，如果类test中包含了一个方法changtest()，可以用如下方式来引用：`mytest.changtest()`。

### 2.类的变量

声明类的变量和声明类类似，前面可以跟上修饰符；变量有以下几种修饰符：`public`--变量在所有类中可视；包括包中的类和所有子类。`protected`--变量可以被当前类中的所有方法访问，但是在当前包以外的类中不可见。`private`--变量只能被当前类中的所有方法访问。`static`--声明此变量为静态变量。`final`--该修饰符说明此变量的值在运行期间不能被修改。为什么不把所有变量都声明成全局变量呢？有两个原因：第一个原因是这样做将浪费大量的内存；第二个原因是如果把所有变量都声明成全局变量会使得程序难以阅读。在类中有两个特殊的变量：`this`和`super`；在以下两种情况中需要使用`this`变量：在类中有两个变量同名，其中一个属于当前类而另一个属于某个特定的方法。当某个类需要将类作为参数传递给某个方法。以下情况需要使用`super`变量：当覆盖方法时，有时候希望使用老方法中的代码，这是可以使用`super`变量来获取原来的代码。

### 3.有时候为了做一个项目需要建立一大堆的类，这些类文

件就象没有子目录或文件夹的硬盘一样是杂乱无章的，为了方便管理所以引入了包，这样就可以将有着相关的类放在一个包中，包就象是文件夹或子目录。要想使某个类成为包的成员，必须使用package来声明他，要注意的是package语句必须是文件的第一行，package语句前面除了空白和注释不能有任何东西；例如：`package packagename.` 如果要引用包packagename中的类则用以下方式：`import packagename.*.` 以上命令是引用包packagename中的所有类，如果你只引用这个包中的某个类可以用以下方式：`import packagename.classname.` 也许有人会问引用整个包中的类不好吗？为什么还要指定是某个类呢？原因有以下几点：首先当引用不在本地机器上的整个包，程序将通过网络将包的所有文件都拖过来，如果这个包有上千个类，而你只使用其中很少的部分，将浪费很多资源。如果引用的两个包中有共享的文件名（例如包pack1中有个叫classname的类，pack2中也有个叫classname的类，如果你使用`import pack1.*.import pack2.*`），你的系统就要崩溃了。4.下面就用一个例子来详细介绍如何创建，实例一个类：

```
*****  
***** //声明类tools是属于包mypack  
package mypack. //声明一个公共的类tools public class tools {  
public tools() {} /** *声明类的方法replace() *此方法在一个字符串里查找某个特定的字符串，如果找到，则将这个字符串替换成另外一个字符串 *如果找到返回替换后的字符串，如果未找到则返回原字符串 *replace(整个字符串，被替换的字符串，替换后的字符串) *值得说明的是由于我是使用while，所以
```

如果原字符串为"ababcdcd"，要将字符串"ab"替换成"e"，则输出为"eecdcd" \*如果只是想将原字符串中的第一个被找到的字符串替换掉，则将while换成if即可 \*/

```
public String replace(String allstr,String replstr1,String replstr2) {
while(allstr.indexOf(replstr1)!=-1)
allstr=allstr.substring(0,allstr.indexOf(replstr1)) replstr2
allstr.substring(allstr.indexOf(replstr1) replstr1.length()). return
allstr. } public static void main(String[] args) { //生成类tools的实例t
tools t=new tools(). //用类tools的方法replace()将字符串"abcd"替换成"adcd"并输出到控制台
System.out.print(t.replace("abcd","b","d")). } }
```

\*\*\*\*\*

\*\*\*\*\* 二：接口 1.前面讲了类有个重要的特性就是继承,但是一个类只能有一个父类,为了实现象C 中的多继承性,就必须使用接口.接口和类非常相似,接口用于定义几个类具有的但又不在这些类中定义的功能,通过在接口中设置这些方法,描述出共同的特性,至于如何实现则是类本身的事。由于接口的这个特性，就意味着在接口中只能定义抽象的方法而不能有方法体；这又于类的定义相违背了，这就造成了接口不能扩展类。另外接口还有一个与类不同的地方是接口声明的变量必须是静态变量。下面语句定义了一个接口： public interface interf 值得注意的是，如果使用了扩展的接口，必须将老接口的方法和新接口的方法全部覆盖掉。在覆盖方法时方法名和参数都必须和老方法一致，如果只是方法名相同而参数不同那将重载该方法，但不会覆盖该方法。例如：

\*\*\*\*\*

```
***** public interface interf { .....  
public void meth() { ..... } public void meth(int a) { ..... } .....
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问  
[www.100test.com](http://www.100test.com)