《基于web的编程技术》**实验讲义**

**张丽翠**

**实验一、**验证各种HTML标记

**1．1 实验目的**

学会HTML注释项、声明部分的使用。

**1．2 实验内容**

1、加入HTML注释、隐藏注释、脚本注释。。

2、编写html文档，设计一个具有文本框、列表、复选框及提交等功能的学生信息提交页面，页面中要求输入包括学生姓名、电话、QQ号、专业等信息其中专业从下拉菜单中选。下拉菜单中的专业有通信、信息、测控、电子等。页面要以表格的形式介绍学院的各个专业。页面要有标题、图像等

**1．3 基础知识**

HTML 是用来描述网页的一种语言。HTML 指的是超文本标记语言 (Hyper Text Markup Language)，它不是一种编程语言，而是一种标记语言 (markup language) 。标记语言是一套标记标签 (markup tag) HTML 使用标记标签来描述网页。

例：h2.htm

<html>

<head>

<title>第一个Html文档</title>

</head>

<body>

欢迎访问<a href="http://www.jb51.net">脚本之家</a>!

</body>

</html>

<html>这个标签告诉浏览器这是HTML文档的开始(见html标签).HTML文档的最后一个标签是</html>,这个标签告诉浏览器这是HTML文档的终止。在<head>和</head>标签之间的文本是头信息.在浏览器窗口中,头信息是不被显示在页面上的(见head标签)。在<title>和</title>标签之间的文本是文档标题,它被显示在浏览器窗口的标题栏(见title标签)。在<body>和</body>标签之间的文本是正文,会被显示在浏览器中(见body标签)。在<p>和</p>标签代表段落(见p标签)。<a>和</a>定义了一个超链接HTML文档可以包含不同的内容，比如文本，链接，图片，列表，表格，表单，框架等。

* 文本 :HTML对文本的支持是最丰富的，可以设置不同级别的标题，分段和换行，可以指定文本的语义和外观，可以说明文本是引用自其它的地方等。
* 链接 :链接用来指出内容与另一个页面或当前页面某个地方有关。
* 图片 :图片用于使页面更加美观，或提供更多的信息。
* 列表 :列表用于说明一系列条目是彼此相关的。
* 表格 :表格是按行与列将数据组织在一起的形式。也有不少人使用表格进行页面布局。
* 表单:表单通常由文本输入框，按钮，多选框，单选框，下拉列表等组成，使HTML页面更有交互性。
* 框架 :框架使页面里能包含其它的页面。

文本：<font>：它用于改变字体，字号，文字颜色。

<font size="6">6</font>

<font size="4">4</font>

<font color="red" size="5">红色的5</font>

<font face="黑体">黑体的字</font>

加粗，下划线，斜体字也是常用的文字效果，它们分别用<b>,<u>,<i>表示：

<b>Bold</b>

<i>italic</i>

<u>underline</u>

表示标题的标签有：<h1>,<h2>,<h3>,<h4>,<h5>,<h6>，它们分别表示一级标题，二级标题，三级标题...。<hr>标签用于在页面上添加横线。可以通过指定width和color属性来控制横线的长度和颜色。如： <hr width="90%" color="red" />。<img>标签用于在页面上添加图片，src属性指定图片的地址，如果无法打开src指定的图片，浏览器通常会在页面上需要显示图片的地方显示alt属性定义的文本。如：<img src="http://www.w3.org/Icons /valid-xhtml10" alt="图片简介" /> 。超级链接用<a>标签表示，href属性指定了链接到的地址。<a>标签可以包含文本，也可以包含图片。

<a href="http://deerchao.net">Unibetter大学生论坛</a>

<a href="http://validator.w3.org"><img src="http://www.w3.org/Icons /valid-xhtml10" alt="验证HTML" /></a>

表格：HTML文档在浏览器里通常是从左到右，从上到下地显示的，到了窗口右边就自动换行。为了实现分栏的效果，很多人使用表格（<table>）进行页面排版（虽然HTML里提供表格的本意不是为了排版）。<table>标签里通常会包含几个<tr>标签，<tr>代表表格里的一行。<tr>标签又会包含<td>标签，每个<td>代表一个单元格。<tr>标签还可以被<table>里的<thead>或<tbody>或<tfoot>包含。它们分别代表表头，表正文，表脚。在打印网页的时候，如果表格很大，一页打印不完，<thead>和<tfoot>将在每一页出现。<th>和<td>非常相似，也用在<tr>里边，不同的是<th>代表这个单元格是它所在的行或列的标题。

例：

<table>

  <thead>

    <tr><th>时间</th><th>地点</th></tr>

  </thead>

  <tbody>

    <tr><td>2000</td><td>悉尼</td></tr>

    <tr><td>2004</td><td>雅典</td></tr>

    <tr><td>2000</td><td>北京</td></tr>

  </tbody>

</table>

列表：表格用于表示二维数据（行，列），一维数据则用列表表示。列表可以分为无序列表（<ul>），有序列表（<ol>）和定义列表（<dl>）。前两种列表更常见一些，都用<li>标签包含列表项目。无序列表表示一系列类似的项目，它们之间没有先后顺序。

<ul>

  <li>苹果</li>

  <li>桔子</li>

  <li>桃</li>

</ul>

有序列表中各个项目间的顺序是很重要的，浏览器通常会自动给它们产生编号。

例：<ol>

  <li>打开冰箱门</li>

  <li>把大象赶进去</li>

  <li>关上冰箱门</li>

</ol>

form 标记的使用：在Web开发中，form标记是最为常用的标记之一。可以通过form标记以及一些包含在form中的其他标记，获取用户的输入信息，从而实现客户端和服务器端的交互。form中文本框的使用：

<HTML>

<HEAD>

<TITLE>文本框练习</TITLE>

</HEAD>

<BODY>

<form>

姓名：

<input type = "text" name = "textbox1"> <br>

主页地址：

<input type = "text" name = "textbox2" value="http://"> <br>

密码：

<input type = "pastword" name = "textbox3"> <br>

</form>

</BODY>

</HTML>

**form中复选框的使用**

<HTML>

<HEAD>

<TITLE>复选框范例</TITLE>

</HEAD>

<BODY>

请选择你喜欢的水果：

<form>

<input type = "checkbox" name = "check1"> <br>

Banana <p>

<input type = "checkbox" name = "check2" checked>

Apple<p>

<input type = "checkbox" name = "check3" value = "橘子">

Orange <p>

</form>

</BODY>

</HTML>

**form中复选框的使用**

<HTML>

<HEAD>

<TITLE>单选框范例</TITLE>

</HEAD>

<BODY>

请选择你喜欢的水果（智能选一种）：

<form>

<input type = "redio" name = "水果"> <br>

Banana <p>

<input type = "redio" name = "水果" checked>

Apple<p>

<input type = "redio" name = "水果">

Orange <p>

</form>

</BODY>

</HTML>

**重置和提交按钮的使用**

文件名称：button.htm

<HTML>

<HEAD>

<TITLE>BUTTON</TITLE>

</HEAD>

<BODY>

<form>

<input type = "text" name = "txt1" value="文本框"> <br>

<input type = "submit" name = "submit" value="提交">

<input type = "reset" name = "reset" value="重置">

</form>

</BODY>

</HTML>

**列表框的使用**

文件名称：select.htm

<HTML>

<HEAD>

<TITLE>列表框范例</TITLE>

</HEAD>

<BODY>

<form>

<select name = "fruits">

<option> Banana </option>

<option selected> Apple </option>

<option> Orange </option>

</select> <p>

<input type="submit">

<input type="reset">

</form>

</BODY>

</HTML>

**列表框的多选功能的使用**

文件名称：select.htm

<HTML>

<HEAD>

<TITLE></TITLE>

</HEAD>

<BODY>

<form>

<select name = "fruits" size ="5" multiple>

<option selected> Banana

<option selected> Apple

<option> Orange

<option selected>Peach

</select> <p>

<input type="submit">

</form>

</BODY>

</HTML>

**多行文本框的使用**

<HTML>

<HEAD>

<TITLE>Textarea</TITLE>

</HEAD>

<BODY>

<form>

< textarea name="comment" rows="5" cols="60"> </textarea>

<p>

<input type = "submit" >

<input type = "reset">

</form>

</BODY>

</HTML>

**1.4 实验步骤**

1．按照实验要求编写实验程序。

2．调试并运行程序。

3．完成实验报告。

**实验二：Java程序的环境及组成训练**

**2．1 实验目的**

Java程序的环境及组成训练。

**2．2 实验内容**

1、查看及设置环境变量

2、练习Java的变量声明、表达式计算、输入输出、控制流程。

3、编写一个简单的Java程序，完成n个连续自然数的阶乘和。1!+2!-3!+4!-5!+...+n!

4、编程验证课上的例子程序。

**2．3 基础知识**

Java所有程序都是以class（类）为单位的，每个Java程序都必须放在一个类中。每一个Java应用程序都要有且只有一个main（）方法，因为main方法是程序进入的起点。

编译时输入命令 c:\> javac HelloWorld.java

执行时输入命令 c:\>java HelloWorld

Java程序的元素：

**注释**：Java中包含三种注释

// 单行注释

/\* .........\*/ 多行注释

/\*\* ........\*/ 文档注释

**标识符**：为变量、方法、类和对象其的名字称为标识符。标识符以字符、下划线或$符号开头，后面可以包含字母、数字、下划线或$。

**关键字**：Java中的关键字是由系统使用的、具备特定含义的字符。

**Java的数据类型**：包括基本数据类型和引用数据类型

**基本数据类型**：布尔、字符、整数、浮点数

**引用数据类型**：类（class）、接口（interface）

例子程序：文件名称：TestVar.java

public class TestVar{

public static void main(String args[]){

boolean flag = true;

byte b = 33;

char c = 'a';

float f = 123.456f;

System.out.println("flag=" +flag);

System.out.println("b =" +b);

System.out.println("c=" +c);

System.out.println("f=" +f);

}

}

**2.4 实验步骤**

1．按照实验要求编写实验程序。

2．调试并运行程序。

3．完成实验报告。

**实验三：常量、变量表达式及String类训练**

**3．1 实验目的**

常量、变量表达式及String类训练

**3．2 实验内容**

1、学会使用字符串常量及变量。

2、学会String类及其各种实例函数的功能，编译并执行Java程序，实现在某一段文字中查找并替换单词。

编写Java程序，从键盘输入一个单词1和单词2，在一段文本中查找单词1，若找到则将其替换成单词2，若没有则告知在文本中单词1不存在。

**3．3 基础知识**

String类的方法

（1）静态方法：只要定义了类就可以通过类名调用的方法。调用静态方法的一般格式是： 类名.方法名（）

ValueOf（）是String类的一个静态方法，其作用是将数字类型的数据转换为字符串数据。

（2）实例方法：指那些要通过类的对象才能调用的方法。调用实例方法的一般格式是 对象名.方法名（）

toUpperCase()方法是Sting类的一个实例方法，作用是读取对象保存的字符串后再将字符串转换为大写形式。、

（3）数字转换为String：使用String类的静态方法String.valueOf( ，例如：

String s = String.valueOf(45); 或 String s = String.valueOf(57.65);

（4）String转换为数字：因为用户通过键盘输入的数字的类型为String型，所以必须将其转换为数字，有两个静态方法可以实现

用Integer类的parseInt( )方法，将String代表的整数型数字转换为int型数值，其格式为 Integer.parseInt（String）

用Double类的parseDouble( )静态方法，将String代表的浮点型数字转换为double型数值，其格式为 Double.parseDouble(String )

（5）length()方法：计算字符串的长度，并返回一个int类型的值。

（6）indexOf()方法：在字符串中定位某个特定字符串，并返回其位置编号。如果小括号中是只有一个字母的字符串，则返回该字母的编号位置。如果是多个字母的字符串，则返回该字符串中第一个字母的编号位置。如果没有 找到特定的内容，则返回值-1。字符串中字母的位置编号从0开始依次增加1。

（7）substring（）方法：取子串。截取字符串中的一部分，并作为一个新的字符串返回。该方法有两种形式：

（a）substring（位置编号）：返回一个从特定位置编号开始的新字符串。位置编号必须为int类型。 例：“abcdefg”.substring(3) 表示返回“defg”。

（b）substring（位置编号1，位置编号2）：返回一个从位置编号1开始到位置编号2结束的新字符串。其中包含位置编号1的字符，不包含位置编号2的字符。

（8） 判断字符串相等。String提供了两种方法

（a）equals（String型对象）方法：比较两个String类型数据是否完全相同。相同则返回true，否则返回false。 例：s.equal(“hello”)

（b）equalsIgnoreCase（String型对象）方法：在忽略大小写的情况下比较相等。

**3.4 实验步骤**

1．按照实验要求编写实验程序。

2．调试并运行程序。

3．完成实验报告。

**实验四：基本算法及流程控制训练**

**4．1 实验目的**

基本算法及流程控制训练

**4．2 实验内容**

用顺序机构及分支结构进行程序设计。

用顺序机构及循环结构进行程序设计。

用嵌套的循环结构完成n行的右上左下直角三角形及2n-1行的菱形图形的输出。

**4．3 基础知识**

算法：指解决一个特定问题采用的特定的、有限的方法和步骤。一般程序有三种结构：顺序结构、选择结构、循环结构。

选择结构语句：if ；if ......else.... ；switch......case

循环结构语句：for 、while 、do ....while

break语句：常用于循环语句和switch语句，作用是使程序从一个语句体的内部跳出去，继续执行该与具体下面的语句。

continue语句：通常只用在循环语句中，其作用是终止当前一轮的循环，不在执行其下面的语句，直接进入下一轮循环。

循环可以嵌套

**4.4 实验步骤**

1．按照实验要求编写实验程序。

2．调试并运行程序。

3．完成实验报告。

**实验五：数组和方法训练**

**5．1 实验目的**

学会数组及方法的声明、初始化及使用。

**5．2 实验内容**

声明各种类型的数组并对其进行构造及初始化

编程实现有n个元素的数组元素的升序排序

编程实现高数成绩的统计，要求统计出各个分数段的人数。

**5．3 基础知识**

**数组及其特点：**

1. 数组是相同数据类型元素的集合。

2. 数组中的各个元素在内存中按照先后顺序连续存放在一起。

3. 每个数组元素用其所在数组的名字和其在数组中的顺序位置（下标）表示。

4. 数组的下标值从0开始，可以是int类型的数据、变量和算术表达式。

5. 每个数组都有长度，即其所能含有元素的个数。

**数组的使用：**

1. 使用数组的步骤：定义、创建、使用

（1）定义数组：定义数组的名称和数组中元素的数据类型。先定义后使用。有两种形式数组

元素类型[] 数组名称；

数组元素类型 数组名称[]；

创建数组：指定数组的长度，并分配相应的内存空间。用关键字new来完成。

使用数组：通过下标来标识数组的各个元素。下标从0开始，到长度减一。

数组的初始化有两种方法：

（a）按照顺序依次给每个元素赋值。例如 result[0] = 87; result[1] = 90; ...result[49] = 9;

（b）在定义数组的同时直接给数组赋初值，初值的个数是数组的长度，初值必须用大括号括起来，用逗号分隔开.

**5.4 实验步骤**

1．按照实验要求编写实验程序。

2．调试并运行程序。

3．完成实验报告。

**实验六：面向对象程序设计训练**

**6．1 实验目的**

面向对象程序设计训练

**6．2 实验内容**

1. 定义一个时间类，组织好类的实例变量和实例方法

2. 定义一个主函数，在主函数中声明并创建前面定义的时间类的对象，通过新建的对象调用并验证各种实例方法。

3. 定义并使用一个类来区分静态变量、实例变量、静态方法、实例方法。

定义一个类，编程实现多个不同专业学生成绩管理、要求可以修改专业，录入、显示学生姓名、学号、高数、英语、计算机三科的成绩。专业生成静态变量。

**6．3 基础知识**

**1. 常用术语：**

（1）对象（Object）：由数据以及对数据进行处理的方法组成，是对现实世界的抽象描述。每一个对象都是由状态和行为两个部分组成。虽然同一类对象的行为是一样的，但每个对象的状态都是独立于其他对象的。每个对象有自己的空间。

（2）类（class）：描述对象的蓝图，同一类对象的集合和抽象。类是一种特殊的数据类型。对象相当于用类声明的变量。一个对象被称作其类的一个实例。是类的实例化的结果。类还可以有子类，子类除了具有类的所有状态和行为外，还具有自己特有的状态和行为。

（3）域（field）：域是类或者对象的状态属性的总称。可以是普通数据类型的变量，也可以是其他类的对象。一般使用实例变量来表达类或对象的状态，并且这些变量必须在类方法的外边定义，作用于整个类。

（4） 方法（Method）：类的行为属性的总称。一个类可以有多个方法，表示该类所具有的功能和操作。通过对象调用类中的方法就可以改变对象域中变量的值。

**2. 面向对象的编程方法具有四个基本特征：抽象、继承、封装、多态**

（1）抽象：抽象就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象并不打算了解全部问题，而只是选择其中的一部分，暂时不用部分细节。比如，我们要设计一个学生成绩管理系统，考察学生这个对象时，我们只关心他的班级、学号、成绩等，而不用去关心他的身高、体重这些信息。抽象包括两个方面，一是过程抽象，二是数据抽象。过程抽象是指任何一个明确定义功能的操作都可被使用者看作单个的实体看待，尽管这个操作实际上可能由一系列更低级的操作来完成。数据抽象定义了数据类型和施加于该类型对象上的操作，并限定了对象的值只能通过使用这些操作修改和观察。

（2）继承：继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。对象的一个新类可以从现有的类中派生，这个过程称为类继承。新类继承了原始类的特性，新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派生类可以从它的基类那里继承方法和实例变量，并且类可以修改或增加新的方法使之更适合特殊的需要。这也体现了大自然中一般与特殊的关系。继承性很好的解决了软件的可重用性问题。比如说，所有的Windows应用程序都有一个窗口，它们可以看作都是从一个窗口类派生出来的。但是有的应用程序用于文字处理，有的应用程序用于绘图，这是由于派生出了不同的子类，各个子类添加了不同的特性。

（3）封装：封装是面向对象的特征之一，是对象和类概念的主要特性。封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象。一旦定义了一个对象的特性，则有必要决定这些特性的可见性，即哪些特性对外部世界是可见的，哪些特性用于表示内部状态。在这个阶段定义对象的接口。通常，应禁止直接访问一个对象的实际表示，而应通过操作接口访问对象，这称为信息隐藏。事实上，信息隐藏是用户对封装性的认识，封装则为信息隐藏提供支持。封装保证了模块具有较好的独立性，使得程序维护修改较为容易。对应用程序的修改仅限于类的内部，因而可以将应用程序修改带来的影响减少到最低限度。

（4）多态性：　多态性是指允许不同类的对象对同一消息作出响应。比如同样的加法，把两个时间加在一起和把两个整数加在一起肯定完全不同。又比如，同样的选择编辑-粘贴操作，在字处理程序和绘图程序中有不同的效果。所以说，由于多态，一个对象可以有独特的表现方式，而对另一个对象有另一种表现方式。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好的解决了应用程序函数同名问题

**3.域**

域是类或者对象的状态属性的总称。它可以是普通数据类型的变量，也可以是其他类的对象类型变量。域常见的有实例变量、静态变量和最终变量3中形式。

（1） 实例变量：

用来存储某个类对象的状态值。它必须在类内、任何方法外被定义，一般位于整个类语句体的最前端。其修饰符可以是public或者private，但不能使用static。实例变量的适用范围是整个类，及可以在类中的任何方法内被使用。实例变量是依据其对象存在的，在运行程序创建对象的同时，创建了其实例变量，当程序运行完成，对象消失，其实例变量也同时消失。

（2）静态变量：

用static修饰的变量成为静态变量，也叫做类变量。静态变量是类的变量，是在类内、类中方法之外定义的变量，但静态变量不属于任何一个类的具体对象实例。它不保存在某个对象实例的内存空间中，而是保存在类的内存空间的公共存储单元中。即不论一个类具有多少个对象，静态变量只有一个，任何一个类的对象访问它，取得的都是相同的数值。同样任何一个类的对象去修改它，也都是在对同一个内存单元进行操作。静态变量可以通过类名直接访问，也可以通过对象来调用，两种访问方法的结果相同。

静态变量和实例变量的区别：

静态变量和实例变量虽然都是在类内、类中方法之外定义的变量，但它们有着本质的区别：

实例变量：一般用public或者private作为修饰符。每个类的对象都具有一套属于自己的、在类中定义的实例变量。每个对象只能修改或者访问其自身的实例变量，不会影响其他对象的实例变量。

静态变量：使用static修饰符，每个类的所有对象共同拥有一套类的静态变量。类和其对象都可以修改或者访问静态变量。

（3） 最终变量：

使用修饰符final修饰的变量是最终变量，其保存的数据在程序的整个执行过程中都不会改变。通常在final前面再加上修饰符static，则这个变量实际就是常量。

如 static final double = 3.141592653589763；

在程序中，给最终变量赋值后，就不可以再以任何方式改变其值。如果在定义最终变量时没有给其赋值，则按该类型数据的默认值进行初始化。final修饰符可以修饰局部变量，必须且只能赋值一次，它的取值在变量存在期间不会改变。

4.构造方法

构造方法也叫构造函数，是一种特殊的方法。在创建对象的同时调用这个对象所属类的构造方法来完成对象实例变量的初始化。构造方法的语法形式：

public 类名( 参数表)

{

语句体；

}

构造方法的特点：

（1）构造方法的名字要与类的名字相同。

（2）虽然构造方法没有返回值，但是也不能有void关键字。

（3）构造方法的修饰符总是public。

（4）构造方法的主要作用是完成对类对象实例变量的初始化工作。

（5）如果在定义类对象时没有定义任何构造方法，系统会自动产生一个构造方法，叫默认构造方法。默认构造方法不带任何参数，并且方法体为空。

除了构造方法外，类中还有许多描述类行为的方法，这些方法可以分为静态方法和实例方法两种。

（1）实例方法：实例方法属于每个对象，只能通过类的对象调用。实例方法用来定义某个类的行为，即类的对象所能进行的操作。实例方法的修饰符可以是public或private，但不能用static。如果希望实例方法在类之外被访问则用public，如果希望实例方法只在类内被访问，则用private。

（2）静态方法：与静态变量类似，静态方法的本质是属于整个类的，而不是属于某个实例变量。静态方法属于类本身，所以只要定义了类，它的静态方法就存在，就可以直接使用类来调用。使用静态方法时要注意以下几个问题：

使用static修饰的静态方法是属于某个类的，它在内存空间中的内容将随着类的定义而进行分配和装载，不被任何一个对象所单独拥有。

静态方法不能操纵和处理属于某个对象的实例变量，而只能处理属于整个类的静态变量，即只能处理静态变量或调用静态变量。

因为静态方法不能访问实例变量，所以在静态方法中不能使用关键字this。

需要调用某个静态方法时，可以使用其所属类的名称直接调用，也可以用类的某个具体对象名调用，其形式为：

类名.静态方法名(参数)

或 对象名.静态方法名(参数)

调用构造方法：必须用new关键字。

访问方法

对象名.方法名的（参数表）

**6.4 实验步骤**

1．按照实验要求编写实验程序。

2．调试并运行程序。

3．完成实验报告。

**实验七：**类的继承和多态训练

**7．1 实验目的**

学会使用继承机制和多态。

**7．2 实验内容**

1.定义一个学生类，包括学生姓名、年龄、成绩实例变量和构造方法、显示姓名、年龄和成绩的方法、修改成绩的方法和设置参数的方法。

2. 在前面定义的类的基础上，定义一个研究生类，加上学院和专业两个实例变量，并加上构造方法、显示信息方法（显示姓名、年龄、成绩和专业）、设置参数的方法。

3. 定义一个含有main方法的类，创建学生对象和研究生对象，设置学生和研究生信息并显示出来。

**7．3 基础知识**

**1. 类继承的概念**

类继承也叫类派生，是指一个类可以继承其他类的所有内容，包括变量和方法。被继承的类称为父类，继承后产生的类称为子类或派生类。派生类除了可以继承父类的所有变量和方法外，还要添加自己特有的变量或方法。

**2. 类继承的实现**

（1）创建子类

在类的声明中加入extends关键字，其形式如下：

[修饰符] class 子类名 extends 父类名

{

语句体

}

如果父类又是某个类的子类，则子类同时也是该类的间接子类。子类可以继承其所有父类的内容。

（2）类继承的传递性

类继承具有传递性，也就是说子类具有其父类和父类的父类，一直到Object类的所有类的内容。例如B类继承了A类，而C类又继承了B类，则C类包含A类和B类的所有内容。

1. Super关键字

uper关键字指的是父类。在类的继承中具有重要的作用。

（a）使用super关键字访问父类的变量或者方法

有时会遇到子类中的实例变量或者实例方法与父类中的实例变量或者实例方法同名。因为子类中的实例变量或实例方法名具有更高的优先级，所以子类中的同名实例变量和实例方法就隐藏了父类的实例变量或实例方法，在此情况下，如果想要使用父类中的某个实例变量或实例方法，就需用super关键字。

（b）使用父类的构造方法

想要使用父类的构造方法，则应使用下面的形式： super（参数）

子类在构造方法中，可以用关键字super来调用父类的构造方法。例如，在子类Student的构造方法中调用父类Person的构造方法。

多态是指程序的多种表现形式。除了方法的重载，子类对父类方法的覆盖或者子类对抽象父类中的抽象方法的具体定义也是类多态的一种表现形式。

覆盖是指在定义子类的方法时，使用与其父类中相同的名称和参数。在执行程序时，将执行子类的方法，而覆盖父类的方法。

**7.4 实验步骤**

1．按照实验要求编写实验程序。

2．调试并运行程序。

3．完成实验报告。

**实验八：包的训练**

**8．1 实验目的**

理解包的作用，学会定义包和导入包。

**8．2 实验内容**

1. 定义一个包，在包中放入实验六中定义的类

2. 在程序中导入包中的类

3.编程实现10个学生的成绩统计，统计每科的最高分和最低分

**8．3 基础知识**

包的作用与文件夹类似，不具有任何特别的功能，只是用来存放一个或者多个类和接口的空间。使用包可以避免同名类发生冲突。包可以嵌套使用。

1. 定义包

在未特别指定的情况下，Java源程序属于默认包，对于默认包中的所有类都可以相互之间引用不是private修饰的变量或者方法。但由于默认包是没有名字的，所以它不能被其他包中的类使用import语句导入，也就不能为其他的类所使用。为了解决这一问题，可以定义具有名字的包，其语法如下：

package 包名；

定义包的语句必须放在整个源程序的第一行，而且前面不能有注释语句和空格。使用这个语句就可以创建具有指定名字的包，并且当前程序中的所有类都属于这个包。

当在程序中定义包后，需要在Windows系统中创建相应的文件夹。一般默认包对应保存程序的文件夹，及DOS中的当前工作目录。如果是新创建的包，则需要在默认包所在的文件夹中，创建一个与包名称相同的子文件夹，用来存放这个包中所含有的类程序。

当需要使用另一个包中的类时，也需要先导入，导入形式为：

import 包名.类名；

其中包名可以用符号“.”来表明包的层次，如果要从一个包中引入多个类，则可以用符号“\*”来代替类名。因为符号“\*”只能表示本层次包中的所有类，不包括其子层次包中的类。所以必须多次使用import语句导入所需要的类。例如：经常用两条import语句来引入两个层次的类：

import java.awt.\*；

import java.awt.event.\*;

在编译Java程序时，系统会自动为程序导入java.lang包，因此不必用import语句导入其包含的类。

Java语言本身固有的类，被保存在若干个包中，这些包就是前面介绍过了API（Application Program Interface）。所有API中的包都以“java.”开头，以便区分用户自行创建的包。Java中一些常见的包如下：

（1）java.lang包：提供了在编写Java程序时所需要的最基本的类。例如，各种普通数据类型的类，Math类，String类，System类和Pachage类，其中最主要的是Object类，它是所有其他类的父类。

（2）java.applet包：提供了用来创建Applet的必须类，它仅包含少量几个接口和一个非常有用的类：java.applet.Applet。该类用来与Applet中的组件进行交流。

（3）java.awt包：是Java语言用来构建图形用户界面的类库，包括了所有创建用户界面所要使用的类以及绘制图形和编辑图片所需要的类。如Button类、Lable类和TextField类等。

（4）java.swt.event包：提供了用来控制不同类型的事件的类和接口，并使每个图形界面的组件本身可以拥有处理它上面事件的能力，如ActionLister接口、ActionEvent类等。

（5）java.io包：提供了系统输入和输出数据流时所需要的类，例如BufferedReader类、BufferedWriter类、FileInputStream类、FileOutputStream类。

（6）java.until包：提供了Java语言中一些实用的类。例如，处理时间的Date类和Timer类，处理数组的Arrays类和Vector类。

（7）java.awt.image包：提供用来穿件和编辑图片的类。

（8）java.net包。提供用来实现网络功能的类。如Socket类和ServerSocket类等。

（9）java.sql包。提供了可以访问和处理保存在数据库中的数据的类，可以通过传递SQL语句来读取、写入数据以及管理数据库

2. 访问控制符

访问控制符是用来控制类、变量和方法的访问权限的。常用的4中访问控制符有：public、protected、private、和friendly。

（1）public修饰符：类的修饰符一般为public，表示该类为公共类，可以被所有的其他类访问和使用。这并不表示类中的变量和方法也都是公共的。不过只有在类为public的条件下，才可以定义其中的变量和方法为public。

用public修饰的变量为公共变量，可以被其他所有的类调用。因此会造成安全性和数据封装性的下降，一般应少用，而改用访问方法来读取和写入变量数据。

（2）protected修饰符：使用此修饰符的变量和方法，可以被同一个包中的其他类、类本身以及类的子类访问和调用。子类既可以是在同一个包中的类，也可以是在不同包中的类。

3）private修饰符：使用本修饰符的变量和方法，只可以被类本身访问和使用。

（4）friendly修饰符：是Java默认的修饰符。当一个变量或者方法没有任何修饰符修饰时，系统默认其修饰符为friendly。使用该修饰符的变量和方法，可以被同一个包中的其他类和类本身访问和使用。如果没有修饰符，则表示该类只可以被同一个包中的其他类访问和使用。

3. 接口

接口是用来组织类内容的一种程序结构，一个类可以有多个接口。通过接口使得处于不同层次、互不相关的类可以具有相同的行为。

在Java语言中，一个对象的行为是与其具体实现细节分开的。即在开发程序时，一方面，可以在不了解某个对象具体实现细节的情况下，使用该对象；另一方面，可以在不知道对象被如何应用的情况下，编写其功能的具体实现细节。这种对象行为与行为的具体实现细节分离的设计方法称为抽象。

抽象的数据结构只定义对象所支持的操作，而不具体描写这些操作是如何实现的。在Java语言中，使用接口来定义某个类所支持的操作。因此，在接口中，所有的方法均为抽象方法。接口定义的仅仅是实现某一特定功能的接口和其规范，而并没有真正实现这个功能。这个功能的真正实现是在应用这个接口的类中完成，要由类来具体定义接口中各个抽象方法的语句体。

创建接口的语法形式：

public interface 接口名

{

public static final 常量名 = 常量数据；

public abstract 方法类型 方法名（参数）；

}

接口的唯一访问控制修饰符就是public。如果接口没有任何访问控制修饰符，则表示只有与该接口定义在同一个包中的类才可以访问这个接口。接口中只能是常量形式的量，方法也必须是抽象方法，因为接口中的方法均是使用修饰符public和关键字abstract修饰的，所以在书写是可以省略这两个单词。

**8.4 实验步骤**

1．按照实验要求编写实验程序。

2．调试并运行程序。

3．完成实验报告。