//打印金字塔（键入层数）

**package** com.remake;

**import** java.io.\*;

**public** **class** Demo1

{

**public** **static** **void** main(String[] args)

{

// **TODO** 自动生成的方法存根

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

**while**(**true**)

{

System.***out***.println("请输入金字塔的层数");

**int** x=0;

**try** {

x = Integer.*parseInt*(br.readLine());

} **catch** (NumberFormatException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}

Jinzita j=**new** Jinzita();

j.dayin(x);

}

}

}

**class** Jinzita

{

**int** x;

**public** **void** dayin(**int** x)

{

**for**(**int** i=0;i<x;i++)

{

**for**(**int** j=0;j<x-i-1;j++)

{

System.***out***.print(" ");

}

**for**(**int** k=0;k<(i+1)\*2-1-1;k++)

{

System.***out***.print("\*");

}

System.***out***.println("\*");

}

}

}

**package** com.remake;

**import** java.io.\*;

**public** **class** Demo2

{

**public** **static** **void** main(String[] args)

{

// **TODO** 自动生成的方法存根

**int** x=0,y=0,z=0;

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

System.***out***.println("1.2.3.4一次代表+ - \* / ");

**while**(**true**)

{

**try** {

System.***out***.print("请输入想进行的计算;输入99退出");

x=Integer.*parseInt*(br.readLine());

**if**(x==99)

{

System.*exit*(0);

}

System.***out***.print("请输入第一个数");

y=Integer.*parseInt*(br.readLine());

System.***out***.print("请输入第二个数");

z=Integer.*parseInt*(br.readLine());

} **catch** (NumberFormatException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}

Jisuan j=**new** Jisuan(x);

j.start(y,z);

}

}

}

**class** Jisuan

{

**int** x;

**public** Jisuan(**int** n)

{

**this**.x=n;

}

**public** **void** start(**int** y,**int** z)

{

**if**(x==1)

{

System.***out***.println(y+"+"+z+"="+(y+z));

}

**else** **if**(x==2)

{

System.***out***.println(y+"-"+z+"="+(y-z));

}

**else** **if**(x==3)

{

System.***out***.println(y+"\*"+z+"="+(y\*z));

}

**else** **if**(x==4)

{

System.***out***.println(y+"/"+z+"="+(y/z));

}

**else**

{

System.***out***.println("输入错误，请看清提示，亲！");

}

}

}

//猜拳游戏

**package** com.l;

**import** java.io.\*;

**public** **class** Caiquan {

**public** **static** **void** main(String[] args)**throws** Exception {

// **TODO** 自动生成的方法存根

System.***out***.println("猜拳游戏开始！");

System.***out***.println("0代表石头；1代表剪刀；2代表布。输入0、1、2以外的数结束游戏");

System.***out***.println("请输入！");

Cai c=**new** Cai();

**while**(**true**)

{

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

**int** x;

x=Integer.*parseInt*(br.readLine());

**if**(x==0||x==1||x==2)

{

c.quan(x);

}

**else**

{

System.***out***.println("猜拳游戏结束！");

**break**;

}

}

}

}

**class** Cai

{

**public** **void** quan(**int** x)

{

**int** y=(**int**)(Math.*random*()\*3);

**if**(y==0)

{

**if**(x==0)

{

System.***out***.print("电脑出石头");

System.***out***.print("你出的石头");

System.***out***.println("平手");

}

**if**(x==1)

{

System.***out***.print("电脑出石头");

System.***out***.print("你出的剪刀");

System.***out***.println("你输了");

}

**if**(x==2)

{

System.***out***.print("电脑出石头");

System.***out***.print("你出的布");

System.***out***.println("你赢了");

}

}

**if**(y==1)

{

**if**(x==0)

{

System.***out***.print("电脑出剪刀");

System.***out***.print("你出的石头");

System.***out***.println("你赢了");

}

**if**(x==1)

{

System.***out***.print("电脑出剪刀");

System.***out***.print("你出的剪刀");

System.***out***.println("平手");

}

**if**(x==2)

{

System.***out***.print("电脑出剪刀");

System.***out***.print("你出的布");

System.***out***.println("你输了");

}

}

**if**(y==2)

{

**if**(x==0)

{

System.***out***.print("电脑出布");

System.***out***.print("你出的石头");

System.***out***.println("你输了");

}

**if**(x==1)

{

System.***out***.print("电脑出布");

System.***out***.print("你出的剪刀");

System.***out***.println("你赢了");

}

**if**(x==2)

{

System.***out***.print("电脑出布");

System.***out***.println("你出的布");

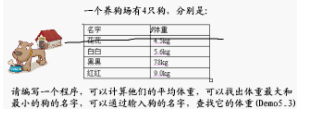
System.***out***.println("平手");

}

}

}

}



**package** com.remake;

**import** java.io.\*;

**public** **class** Demo3

{

**public** **static** **void** main(String[] args)

{

// **TODO** 自动生成的方法存根

Dog dog[]=**new** Dog[4];

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

dog[0]=**new** Dog();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

dog[0].setName("a");

dog[0].setWeight(4.5f);

dog[1]=**new** Dog();

dog[1].setName("b");

dog[1].setWeight(5.6f);

dog[2]=**new** Dog();

dog[2].setName("c");

dog[2].setWeight(7.8f);

dog[3]=**new** Dog();

dog[3].setName("d");

dog[3].setWeight(9.0f);

System.***out***.println("输入1查平均体重");

System.***out***.println("输入1查最大，最小体重");

System.***out***.println("输入3通过名字查出体重");

System.***out***.println("输入99结束");

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

**int** x=0;

String s="",name="";

**while**(**true**)

{

**try** {

x=Integer.*parseInt*(br.readLine());

**if**(x==1)

{

**float** w=dog[1].avg(dog);

System.***out***.println("平均体重是"+w+"kg");

}

**else** **if**(x==2)

{

**float** min=dog[1].min(dog);

**float** max=dog[1].max(dog);

String minname=dog[1].minname(dog);

String maxname=dog[1].maxname(dog);

System.***out***.println("体重最大的狗是"+maxname+"为"+max+"kg;体重最小的狗是"+minname+"为"+min+"kg");

}

**else** **if**(x==3)

{

String string=br.readLine();

**float** weight=dog[1].select(string, dog);

System.***out***.println(string+"的体重为"+weight);

}

**else** **if**(x==99)

{

System.***out***.println("Over");

**break**;

}

**else**

{

System.***out***.println("输入错误，请看清提示，亲！");

}

} **catch** (NumberFormatException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}

}

}

}

**class** Dog

{

**private** String name;

**private** **float** weight;

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **float** getWeight() {

**return** weight;

}

**public** **void** setWeight(**float** weight) {

**this**.weight = weight;

}

**public** **float** avg(Dog a[])

{

**float** sum=0,avg=0;

**for**(**int** i=0;i<a.length;i++)

{

sum=sum+a[i].weight;

}

avg=sum/a.length;

**return** avg;

}

**public** **float** min(Dog a[])

{

**float** b=0;

**for**(**int** i=0;i<a.length;i++)

{

**for**(**int** j=i+1;j<a.length;j++)

**if**(a[i].weight>a[j].weight)

{

b=a[i].weight;

a[i].weight=a[j].weight;

a[j].weight=b;

}

}

**return** a[0].weight;

}

**public** **float** max(Dog a[])

{

**float** b=0;

**for**(**int** i=0;i<a.length;i++)

{

**for**(**int** j=i+1;j<a.length;j++)

**if**(a[i].weight>a[j].weight)

{

b=a[i].weight;

a[i].weight=a[j].weight;

a[j].weight=b;

}

}

**return** a[a.length-1].weight;

}

**public** String minname(Dog a[])

{

**float** b=0;

**int** n=0;

**for**(**int** i=0;i<a.length;i++)

{

**int** time=0;

**for**(**int** j=0;j<a.length;j++)

{

**if**(a[i].weight<a[j].weight)

{

time++;

**if**(time==3)

{

n=i;

**break**;

}

}

}

}

**return** a[n].name;

}

**public** String maxname(Dog a[])

{

**float** b=0;

**int** n=0;

**for**(**int** i=0;i<a.length;i++)

{

**int** time=0;

**for**(**int** j=0;j<a.length;j++)

{

**if**(a[i].weight>a[j].weight)

{

time++;

**if**(time==3)

{

n=i;

**break**;

}

}

}

}

**return** a[n].name;

}

**public** **float** select(String s,Dog a[])

{

**float** weight=0;

**for**(**int** i=0;i<a.length;i++)

{

**if**(a[i].name.equals(s))

{

weight=a[i].weight;

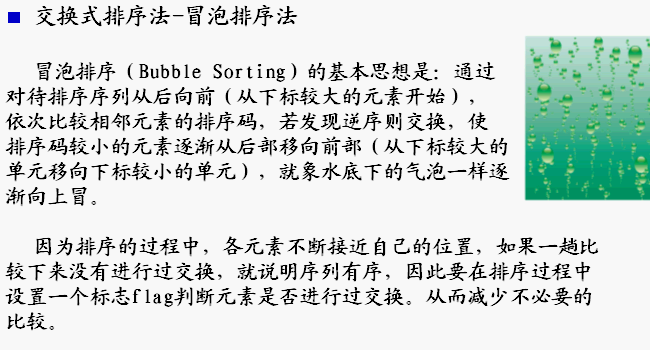
}

}

**return** weight;

}

}



**package** com.remake;

**import** java.io.\*;

**public** **class** Demo4 {

**public** **static** **void** main(String[] args) {

// **TODO** 自动生成的方法存根

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

**int** x=0;

**while**(**true**)

{

**try** {

System.***out***.println("请输入要排序的元素个数");

x=Integer.*parseInt*(br.readLine());

} **catch** (NumberFormatException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}

**int** a[]=**new** **int**[x];

**for**(**int** i=0;i<a.length;i++)

{

**try** {

System.***out***.println("请输入要排序的元素");

a[i]=Integer.*parseInt*(br.readLine());

} **catch** (NumberFormatException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}

}

Maopao m=**new** Maopao();

m.paixu(a);

}

}

}

**class** Maopao

{

**public** **void** paixu(**int** a[])

{

**int** c=0;

**for**(**int** i=0;i<a.length-1;i++)

{

**for**(**int** j=0;j<a.length-i-1;j++)

{

**if**(a[j]>a[j+1])

{

c=a[j];

a[j]=a[j+1];

a[j+1]=c;

}

}

}

**for**(**int** k=0;k<a.length;k++)

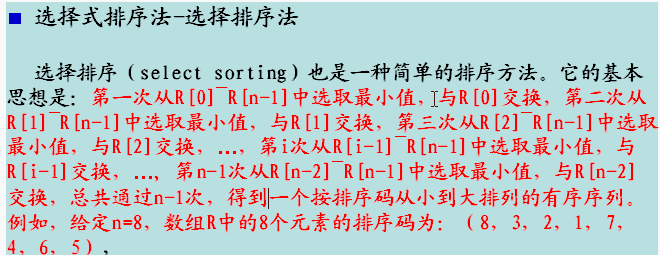
{

System.***out***.print(a[k]+" ");

}

}

}



主要：下标的游离、应用

//选择排序

**package** com.remake;

**import** java.io.\*;

**public** **class** Demo5 {

**public** **static** **void** main(String[] args) {

// **TODO** 自动生成的方法存根

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

**int** x=0;

**while**(**true**)

{

**try** {

System.***out***.println("请输入要排序的元素个数");

x=Integer.*parseInt*(br.readLine());

} **catch** (NumberFormatException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}

**int** a[]=**new** **int**[x];

**for**(**int** i=0;i<a.length;i++)

{

**try** {

System.***out***.println("请输入要排序的元素");

a[i]=Integer.*parseInt*(br.readLine());

} **catch** (NumberFormatException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}

}

Xuanze xuanze=**new** Xuanze();

xuanze.paixu(a);

}

}

}

**class** Xuanze

{

**public** **void** paixu(**int** a[])

{

**for**(**int** i=0;i<a.length-1;i++)

{

**int** index=i;

**int** temp=0;

**for**(**int** j=i+1;j<a.length;j++)

{

**if**(a[j]<a[index])

{

index=j;

}

}

temp=a[i];

a[i]=a[index];

a[index]=temp;

}

**for**(**int** k=0;k<a.length;k++)

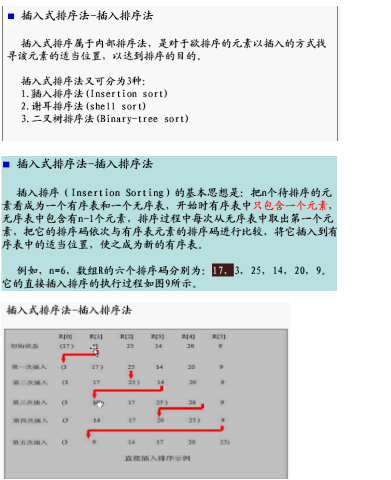
{

System.***out***.print(a[k]+" ");

}

}

}



主要：下标的游离、应用

//插入排序

**package** com.remake;

**import** java.io.\*;

**public** **class** Demo6 {

**public** **static** **void** main(String[] args) {

// **TODO** 自动生成的方法存根

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

**int** x=0;

**while**(**true**)

{

**try** {

System.***out***.println("请输入要排序的元素个数");

x=Integer.*parseInt*(br.readLine());

} **catch** (NumberFormatException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}

**int** a[]=**new** **int**[x];

**for**(**int** i=0;i<a.length;i++)

{

**try** {

System.***out***.println("请输入要排序的元素");

a[i]=Integer.*parseInt*(br.readLine());

} **catch** (NumberFormatException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}

}

Charu c=**new** Charu();

c.paixu(a);

}

}

}

**class** Charu

{

**public** **void** paixu(**int** a[])

{

**for**(**int** i=1;i<a.length;i++)

{

**int** x=i-1;

**int** y=a[i];

**while**(x>=0&&y<a[x])

{

a[x+1]=a[x];

x--;

}

a[x+1]=y;

}

**for**(**int** j=0;j<a.length;j++)

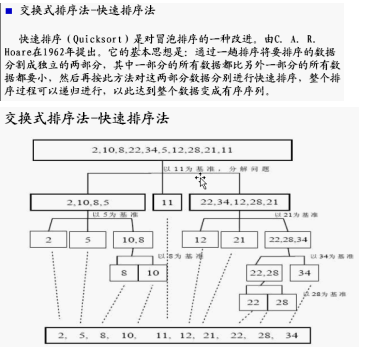
{

System.***out***.print(a[j]+" ");

}

}

}



//快速排序

**package** com.remake;

**public** **class** Demo10 {

**public** **static** **void** main(String[] args) {

// **TODO** 自动生成的方法存根

**int** a[]={226,369,12,58,46};

Quiksort q=**new** Quiksort();

q.sort(a, 0, a.length-1);

**for**(**int** i=0;i<a.length;i++)

{

System.***out***.print(a[i]+" ");

}

}

}

**class** Quiksort

{

**public** **void** sort(**int** a[],**int** low,**int** high)

{

**int** i = low;

**int** j = high;

**int** temp = a[i];

//总的来说就是右边开始第一个小于temp的数和左边开始第一个大于temp的数和temp 这三个数之间的位置转换

**if**( low < high)

{

**while**(i < j)

{

**while**((a[j] >= temp) && (i < j))

{

j--;

}

a[i] = a[j];

**while**((a[i] <= temp) && (i < j))

{

i++;

}

a[j]= a[i];

}

a[i] = temp;

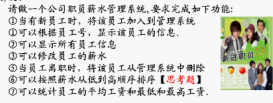
sort(a,low,i-1);

sort(a,j+1,high);

}

}

}



**package** com.remake;

**import** java.io.\*;

**import** java.util.\*;

**public** **class** Demo7 {

**public** **static** **void** main(String[] args) {

// **TODO** 自动生成的方法存根

Manager m=**new** Manager(0,"l",5);

Vector<Manager> vv=**new** Vector<Manager>();

BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.***in***));

**int** x;

**while**(**true**)

{

**try** {

System.***out***.println("员工管理系统：0新增；1查询全部；2按照工号查询；3修改薪水；4删除员工；5薪水排序；6min、max；99退出系统");

x = Integer.*parseInt*(br.readLine());

**if**(x==0)

{

//新增

System.***out***.println("输入新增员工信息");

**int** no=Integer.*parseInt*(br.readLine());

String name=br.readLine();

**float** comm=Float.*parseFloat*(br.readLine());

m.add(no,name,comm,vv);

}

**if**(x==1)

{

//查询全部

m.selectall(vv);

}

**if**(x==2)

{

System.***out***.println("输入要查询员工工号");

**int** no=Integer.*parseInt*(br.readLine());

//根据工号查信息

m.select(no, vv);

}

**if**(x==3)

{

//修改薪水

System.***out***.println("输入要修改薪水员工的工号");

**int** no=Integer.*parseInt*(br.readLine());

**float** comm=Float.*parseFloat*(br.readLine());

m.x(no, comm, vv);

}

**if**(x==4)

{

//删除员工

System.***out***.println("输入要删除的员工的工号");

**int** no=Integer.*parseInt*(br.readLine());

m.delete(no, vv);

}

**if**(x==5)

{

//薪水排序

m.paixu(vv);

}

**if**(x==6)

{

//min、max comm

m.minmax(vv);

}

**if**(x==99)

{

//退出

System.***out***.println("正在退出系统");

**break**;

}

} **catch** (NumberFormatException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

} **catch** (IOException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}

}

}

}

**class** Manager

{

**int** no;

String name;

**float** comm;

**public** **int** getNo() {

**return** no;

}

**public** **void** setNo(**int** no) {

**this**.no = no;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **float** getComm() {

**return** comm;

}

**public** **void** setComm(**float** comm) {

**this**.comm = comm;

}

**public** Manager(**int** no,String name,**float** comm)

{

**this**.no=no;

**this**.name=name;

**this**.comm=comm;

}

**public** **void** add(**int** no,String name,**float** comm,Vector<Manager> vv)

{

Manager m=**new** Manager(no,name,comm);

vv.add(m);

}

**public** **void** selectall(Vector<Manager> vv)

{

**for**(**int** i=0;i<vv.size();i++)

{

Manager ma=**null**;

ma=vv.get(i);

System.***out***.print(ma.getNo()+" "+ma.getName()+" "+ma.getComm());

System.***out***.println(" ");

}

}

**public** **void** select(**int** no,Vector<Manager> vv)

{

**for**(**int** i=0;i<vv.size();i++)

{

Manager ma=**null**;

ma=vv.get(i);

**if**(ma.getNo()==no)

{

System.***out***.print(ma.getNo()+" "+ma.getName()+" "+ma.getComm());

System.***out***.println(" ");

}

}

}

**public** **void** x(**int** no,**float** comm,Vector<Manager> vv)

{

**for**(**int** i=0;i<vv.size();i++)

{

Manager ma=**null**;

ma=vv.get(i);

**if**(ma.getNo()==no)

{

ma.setComm(comm);

}

}

}

**public** **void** delete(**int** no,Vector<Manager> vv)

{

**for**(**int** i=0;i<vv.size();i++)

{

Manager ma=**null**;

ma=vv.get(i);

**if**(ma.getNo()==no)

{

vv.remove(ma);

}

}

}

**public** **void** paixu(Vector<Manager> vv)

{

**int** n=0;

**float** f=0;

String s="";

**for**(**int** i=0;i<vv.size()-1;i++)

{

**for**(**int** j=i+1;j<vv.size();j++)

{

**if**(vv.get(i).getComm()>vv.get(j).getComm())

{

f=vv.get(i).getComm();

vv.get(i).setComm(vv.get(j).getComm());

vv.get(j).setComm(f);

n=vv.get(i).getNo();

vv.get(i).setNo(vv.get(j).getNo());

vv.get(j).setNo(n);

s=vv.get(i).getName();

vv.get(i).setName(vv.get(j).getName());

vv.get(j).setName(s);

}

}

}

**for**(**int** k=0;k<vv.size();k++)

{

Manager ma=**null**;

ma=vv.get(k);

System.***out***.print(ma.getNo()+" "+ma.getName()+" "+ma.getComm());

System.***out***.println(" ");

}

}

**public** **void** minmax(Vector<Manager> vv)

{

**float** f=0;

**for**(**int** i=0;i<vv.size()-1;i++)

{

**for**(**int** j=i+1;j<vv.size();j++)

{

f=vv.get(i).getComm();

vv.get(i).setComm(vv.get(j).getComm());

vv.get(j).setComm(f);

}

}

System.***out***.println("最高薪水是"+vv.get(0).getComm()+"最低薪水是"+vv.get(vv.size()-1).getComm());

}

}

**package** com.t;

**public** **class** Canshu {

**public** **static** **void** main(String[] args) {

// **TODO** 自动生成的方法存根

Canshu c=**new** Canshu();

}

**int** i=9;

**public** Canshu()

{

**this**.f(i);

System.***out***.print(i);

}

**public** **void** f(**int** i)

{

i=100;

}

}

结果：



原因：实参i进入f（）中变成形参得到i=100但是形参100传不出f（） 所以构造函数里的实参i虽然去f（）跑了一片，但却带不走一片落叶，所以实参任然是9

**package** com.t;

**public** **class** Canshu {

**public** **static** **void** main(String[] args) {

// **TODO** 自动生成的方法存根

Canshu c=**new** Canshu();

}

**public** Canshu()

{

**int** i=9;

i=**this**.f(i);

System.***out***.print(i);

}

**public** **int** f(**int** x)

{

x=x\*x;

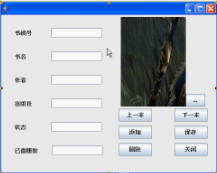
**return** x;

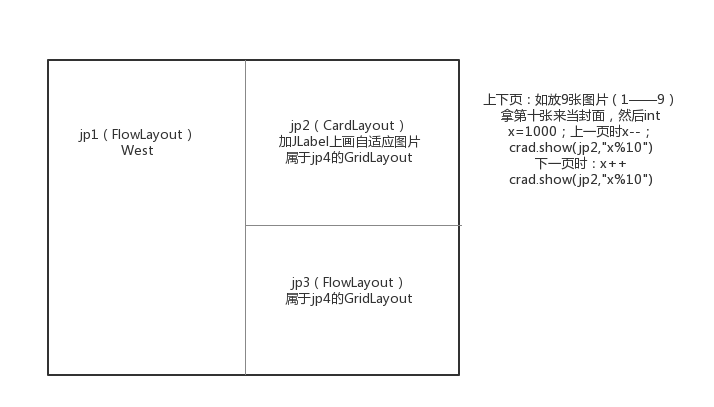
}

}

结果：







**代码：**

Demo8

**package** com.remake;

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** javax.swing.\*;

**public** **class** Demo8 **extends** JFrame **implements** ActionListener{

JPanel jp1,jp2,jp3,jp4,jp11,jp12,jp13,jp14,jp15,jp16,jp31,jp32,jp33;

JLabel jlb11,jlb12,jlb13,jlb14,jlb15,jlb16,jlb21;

JTextField jtf1,jtf2,jtf3,jtf4,jtf5,jtf6;

JButton jb1,jb2,jb3,jb4,jb5,jb6;

CardLayout cl;

**int** x=10;

**public** **static** **void** main(String[] args) {

// **TODO** 自动生成的方法存根

Demo8 d=**new** Demo8();

}

**public** Demo8()

{

jlb11=**new** JLabel("图片编号");

jlb12=**new** JLabel("图片名称");

jlb13=**new** JLabel("图片属性");

jlb14=**new** JLabel("图片格式");

jlb15=**new** JLabel("图片所有者");

jlb16=**new** JLabel("图片著作者");

jtf1=**new** JTextField(7);

jtf2=**new** JTextField(7);

jtf3=**new** JTextField(7);

jtf4=**new** JTextField(7);

jtf5=**new** JTextField(7);

jtf6=**new** JTextField(7);

jp11=**new** JPanel(**new** FlowLayout());

jp12=**new** JPanel(**new** FlowLayout());

jp13=**new** JPanel(**new** FlowLayout());

jp14=**new** JPanel(**new** FlowLayout());

jp15=**new** JPanel(**new** FlowLayout());

jp16=**new** JPanel(**new** FlowLayout());

jp11.add(jlb11,"Left");

jp11.add(jtf1,"Right");

jp12.add(jlb12,"Left");

jp12.add(jtf2,"Right");

jp13.add(jlb13,"Left");

jp13.add(jtf3,"Right");

jp14.add(jlb14,"Left");

jp14.add(jtf4,"Right");

jp15.add(jlb15,"Left");

jp15.add(jtf5,"Right");

jp16.add(jlb16,"Left");

jp16.add(jtf6,"Right");

jp1=**new** JPanel();

jp1.setLayout(**new** GridLayout(6,1));

jp1.add(jp11);

jp1.add(jp12);

jp1.add(jp13);

jp1.add(jp14);

jp1.add(jp15);

jp1.add(jp16);

cl=**new** CardLayout();

jp2=**new** JPanel(cl);

jlb21=**new** JLabel();

Image im0=Toolkit.*getDefaultToolkit*().getImage("image/0.jpg");

Demo8\_1 d0=**new** Demo8\_1(im0);

Image im1=Toolkit.*getDefaultToolkit*().getImage("image/1.jpg");

Demo8\_1 d1=**new** Demo8\_1(im1);

Image im2=Toolkit.*getDefaultToolkit*().getImage("image/2.jpg");

Demo8\_1 d2=**new** Demo8\_1(im2);

Image im3=Toolkit.*getDefaultToolkit*().getImage("image/3.jpg");

Demo8\_1 d3=**new** Demo8\_1(im3);

Image im4=Toolkit.*getDefaultToolkit*().getImage("image/4.jpg");

Demo8\_1 d4=**new** Demo8\_1(im4);

Image im5=Toolkit.*getDefaultToolkit*().getImage("image/5.jpg");

Demo8\_1 d5=**new** Demo8\_1(im5);

Image im6=Toolkit.*getDefaultToolkit*().getImage("image/6.jpg");

Demo8\_1 d6=**new** Demo8\_1(im6);

Image im7=Toolkit.*getDefaultToolkit*().getImage("image/7.jpg");

Demo8\_1 d7=**new** Demo8\_1(im7);

Image im8=Toolkit.*getDefaultToolkit*().getImage("image/8.jpg");

Demo8\_1 d8=**new** Demo8\_1(im8);

Image im9=Toolkit.*getDefaultToolkit*().getImage("image/9.jpg");

Demo8\_1 d9=**new** Demo8\_1(im9);

jp2.add(d0,"0");

jp2.add(d1,"1");

jp2.add(d2,"2");

jp2.add(d3,"3");

jp2.add(d4,"4");

jp2.add(d5,"5");

jp2.add(d6,"6");

jp2.add(d7,"7");

jp2.add(d8,"8");

jp2.add(d9,"9");

jb1=**new** JButton("上一页");

jb1.addActionListener(**this**);

jb2=**new** JButton("下一页");

jb2.addActionListener(**this**);

jb3=**new** JButton("添加");

jb4=**new** JButton("删除");

jb5=**new** JButton("放大");

jb6=**new** JButton("缩小");

jp31=**new** JPanel(**new** FlowLayout());

jp32=**new** JPanel(**new** FlowLayout());

jp33=**new** JPanel(**new** FlowLayout());

jp31.add(jb1,"Left");

jp31.add(jb2,"Right");

jp32.add(jb3,"Left");

jp32.add(jb4,"Right");

jp33.add(jb5,"Left");

jp33.add(jb6,"Right");

jp3=**new** JPanel(**new** GridLayout(3,1));

jp3.add(jp31);

jp3.add(jp32);

jp3.add(jp33);

jp4=**new** JPanel(**new** GridLayout(2,1));

jp4.add(jp2);

jp4.add(jp3);

**this**.setLayout(**new** BorderLayout());

**this**.add(jp1,"West");

**this**.add(jp4,"East");

**this**.setSize(400,600);

**this**.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

**this**.setTitle("图片播放");

**this**.setVisible(**true**);

}

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

// **TODO** 自动生成的方法存根

//上一页

**if**(arg0.getSource()==jb1)

{

x--;

**int** y=x%10;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

cl.show(jp2,""+y+"");

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

//下一页

**if**(arg0.getSource()==jb2)

{

x++;

**int** y=x%10;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

cl.show(jp2,""+y+"");

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

**if**(x==0)

{

x=10;

}

}

}

Demo8\_1

**package** com.remake;

**import** java.awt.\*;

**import** javax.swing.\*;

**public** **class** Demo8\_1 **extends** JPanel{

**int** w;

**int** h;

Image im;

**public** Demo8\_1(Image im)

{

**this**.w=Toolkit.*getDefaultToolkit*().getScreenSize().width;

**this**.h=Toolkit.*getDefaultToolkit*().getScreenSize().height;

**this**.im=im;

//自适应大小

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**this**.setSize(**this**.w,**this**.h);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

**public** **void** paint(Graphics g)

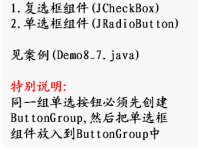
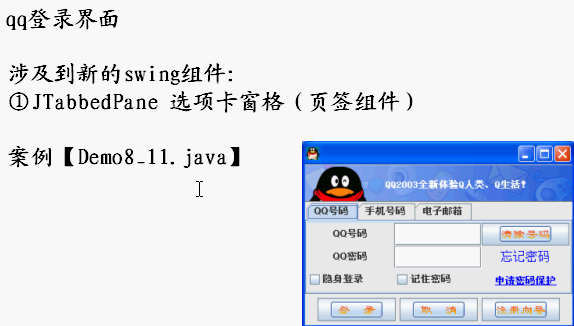
{

**super**.paint(g);

g.drawImage(**this**.im,0,0,**this**.getWidth(), **this**.getHeight(), **this**);

}

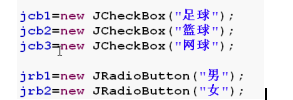
}



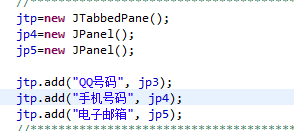
单选复选框的用法：

jcb1=**new** JCheckBox("隐身登录",**false**);

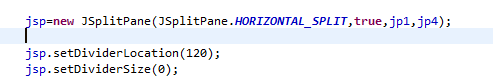
jcb2=**new** JCheckBox("记住密码",**false**);



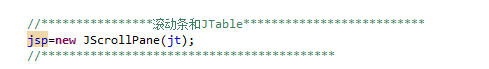
JTabbedPane的用法：



JSplitPane的用法：



JScrollPane的用法：



**代码：**

//QQ登录

package com.remake;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

import javax.swing.\*;

public class Demo9 extends JFrame implements ActionListener{

JPanel jp1,jp2,jp3,jp31,jp32,jp33,jp4,jp5;

JLabel jlb1,jlb31,jlb32,jlb33,jlb34;

JTextField jtf31,jtf32;

// JPasswordField jtf32;

JButton jb31,jb21,jb22,jb23;

JTabbedPane jtp;

JCheckBox jcb1,jcb2;

Connection conn;

PreparedStatement ps;

ResultSet rs;

String s="";

public static void main(String[] args) {

// TODO 自动生成的方法存根

Demo9 demo9=new Demo9();

}

public Demo9()

{

// Image im=Toolkit.getDefaultToolkit().getImage("image/tou.gif");

// Demo8\_1 d1=new Demo8\_1(im);

jlb1=new JLabel(new ImageIcon("image/tou.gif"));

jp1=new JPanel(new BorderLayout());

jp1.add(jlb1);

jb21=new JButton("登录");

jb21.addActionListener(this);

jb22=new JButton("取消");

jb22.addActionListener(this);

jb23=new JButton("注册");

jp2=new JPanel();

jp2.add(jb21);

jp2.add(jb22);

jp2.add(jb23);

jp3=new JPanel(new GridLayout(3,1));

jlb31=new JLabel("QQ号码");

jtf31=new JTextField(10);

jb31=new JButton("清除号码");

jb31.addActionListener(this);

jb31.setForeground(Color.ORANGE);

jlb32=new JLabel("QQ密码");

jtf32=new JTextField(13);

jlb33=new JLabel("忘记密码");

jlb33.setForeground(Color.BLUE);

jcb1=new JCheckBox("隐身登录",false);

jcb2=new JCheckBox("记住密码",false);

jlb34=new JLabel("<html><a href='www.qq.com'>申请密码保护</a>");

jp31=new JPanel(new FlowLayout());

jp32=new JPanel(new FlowLayout());

jp33=new JPanel(new FlowLayout());

jp31.add(jlb31);

jp31.add(jtf31);

jp31.add(jb31);

jp32.add(jlb32);

jp32.add(jtf32);

jp32.add(jlb33);

jp33.add(jcb1);

jp33.add(jcb2,"Left");

jp33.add(jlb34);

jp3.add(jp31);

jp3.add(jp32);

jp3.add(jp33);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

jtp=new JTabbedPane();

jp4=new JPanel();

jp5=new JPanel();

jtp.add("QQ号码", jp3);

jtp.add("手机号码", jp4);

jtp.add("电子邮箱", jp5);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

this.setLayout(new BorderLayout());

this.add(jp1,"North");

this.add(jtp,"Center");

this.add(jp2,"South");

this.setTitle("登录");

Image im1=Toolkit.getDefaultToolkit().getImage("image/qq.gif");

this.setIconImage(im1);

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

this.setSize(350, 260);

this.setLocation(500, 300);

this.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e) {

// TODO 自动生成的方法存根

if(e.getSource()==jb31)

{

jtf31.setText("");

}

if(e.getSource()==jb22)

{

System.exit(0);

}

if(e.getSource()==jb21)

{

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:oracle","scott","tiger");

ps=conn.prepareStatement("select \* from qq where userid=? and password=?");

// ps=conn.prepareStatement("select \* from qq");

ps.setString(1, jtf31.getText().toString());

ps.setString(2, jtf32.getText().toString());

rs=ps.executeQuery();

while(rs.next())

{

s=rs.getString(1);

}

if(s.equals(jtf31.getText()))

{

JOptionPane.showMessageDialog(this,"登陆成功:即将进入您的OICQ","为您提示",JOptionPane.INFORMATION\_MESSAGE);

}

else

{

JOptionPane.showMessageDialog(this,"登陆失败：您的OICQ用户名或密码错误，请检查","为您提示",JOptionPane.INFORMATION\_MESSAGE);

}

} catch (ClassNotFoundException e1) {

// TODO 自动生成的 catch 块

e1.printStackTrace();

} catch (SQLException e1) {

// TODO 自动生成的 catch 块

e1.printStackTrace();

}

}

}

}