# 第一章

## 1.1什么是面向过程的程序设计的特点?

**采用自顶向下的功能分解方法,即一个要解决的问题被划分成成若干个子问题.直至子问题足够简单,可以在相应的子问题中解决.而程序结构也按照功能划分为若干个模块,每个模块用以过程或函数实现.**

## 1.2什么是面向对象的程序设计特点?

**程序模块间关系简单,程序的独立性高,数据安全;显著特性有:封装性,抽象性,继承性,多态性,使软件具有可重用性,开发和维护成本低.**

## 1.3什么是类和对象?

**类:对于相同类型的对象进行分类,得出具有相同的特性而形成类;  
对象:将数据与对数据的操作行为放在一起,作为一个不可分割,相互依存的整体;**

## 1.4什么叫引用变量?

**具有类类型的变量叫引用变量;**

## 1.5什么叫OOP的封装性,继承性,抽象性,多态性?

**封装:就是把对象的属性与操作结合为一个独立的整体,并尽可能的隐藏内部实现细节  
继承性:可以在已有类的基础上派生出新的类,新的来可以吸收已有类的属性与行为,并扩展新能力  
抽象性:类是对于相同类型的对象进行分类抽象,得出具有相同的特性而形成类 多态性:超类中定义的属性与行为,被子类继承后,可以具有不同的数据类型或表现出不同的行为**

# 第二章

## 2.1什么是Java语言的基本特点?

面向对象,平台无关性,可靠性,安全性,多线程,分布式

## 2.2什么是Java程序的基本类型,各有何特点?

应用小程序,应用程序,servlet,bean

## 2.3执行如下代码会打印什么?

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

\*/n\*\*  
\*\*\*  
\*\*\*/n\*\*\*\*

class Test3{

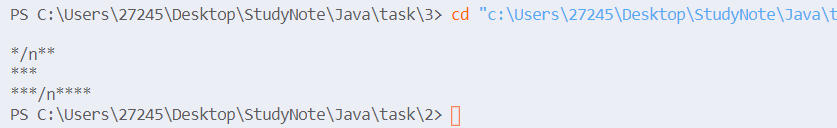
public static void main(String[] args)

{

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

}

}



## 2.4执行如下代码会打印什么?请编制一个完整程序验证其结果;

int a,b;a=20;b=10;

System.out.printf("a=%d,b=%d",a,b);

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

System.out.printf("%s%s","string1","string2");

a=20,b=10\*  
\*\*\*  
string1string2

class Test4{

public static void main(String[] args)

{

int a,b;a=20;b=10;

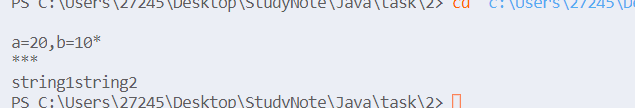
System.out.printf("a=%d,b=%d",a,b);

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

System.out.printf("%s%s","string1","string2");

}

}



## 2.5编写一个程序,输入用户的姓名和性别,输出姓名和性别,并且在他们之间加一个空格.

import java.util.Scanner;

class Test5{

public static void main(String[] args)

{

String s1,s2;

Scanner input=new Scanner(System.in);

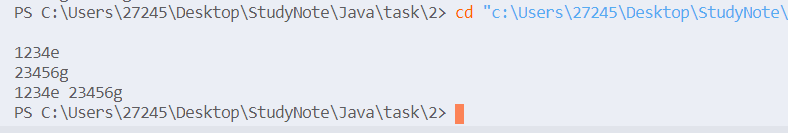
s1=input.next();

s2=input.next();

System.out.println(s1+" "+s2);

}

}



## 2.6编写一个程序,要求输入俩个整数,并显示这俩个数的和,差;

import java.util.Scanner;

class Test6{

public static void main(String[] args){

Scanner input=new Scanner(System.in);

int a,b;

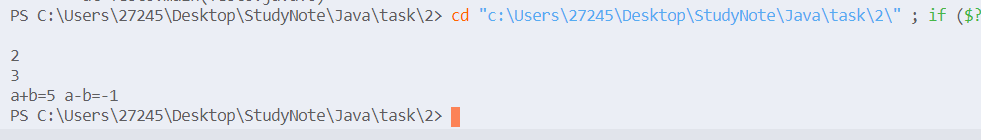
a=input.nextInt();

b=input.nextInt();

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

}

}



## 2.7编写一个applet,要求画一个矩形,并在矩形的上方显示"这是一个矩形"的字符;

**import** java.awt.\*;

**import** javax.swing.\*;

**public** **class** **DrawRectangle** **extends** **JApplet**{

**public** void **paint**(Graphics g)

    {

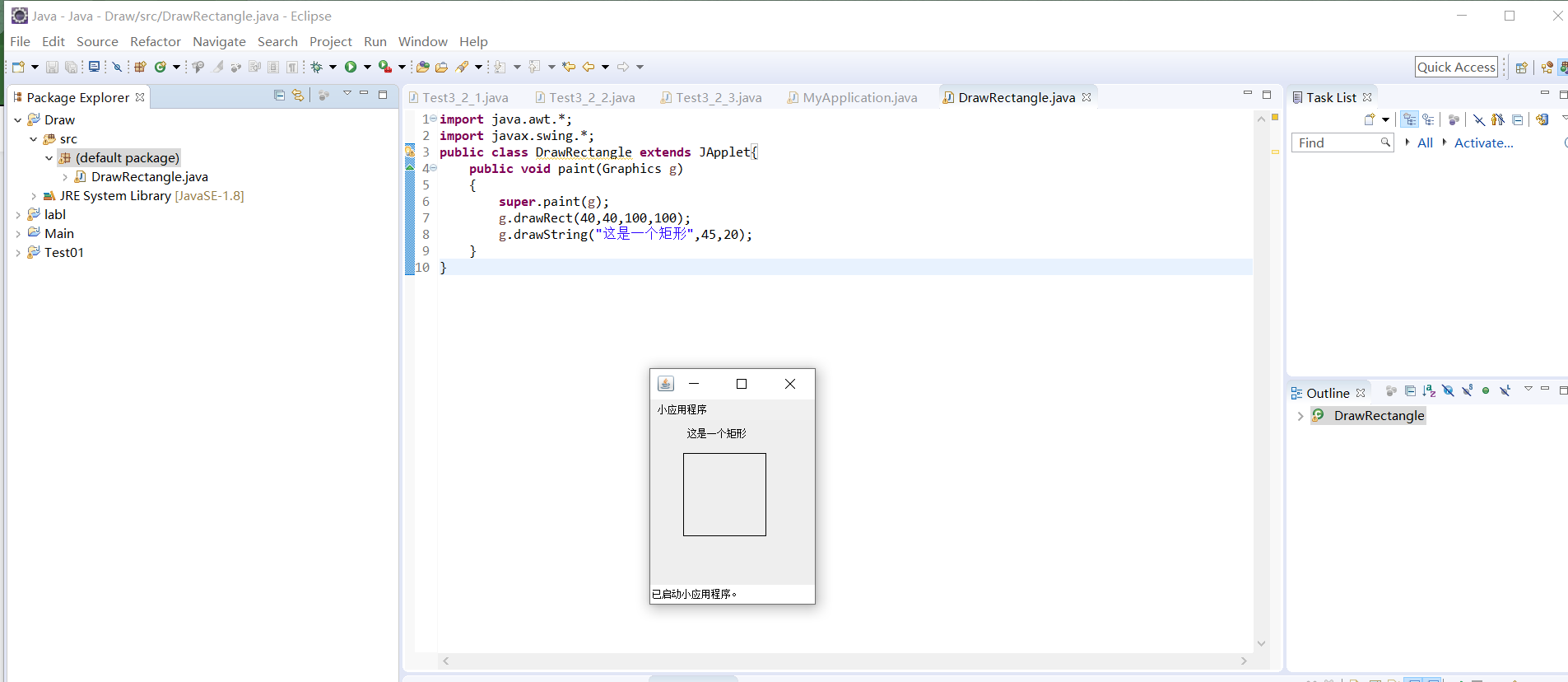
        super**.paint**(g);

        g**.drawRect**(40,40,100,100);

        g**.drawString**("这是一个矩形",45,20);

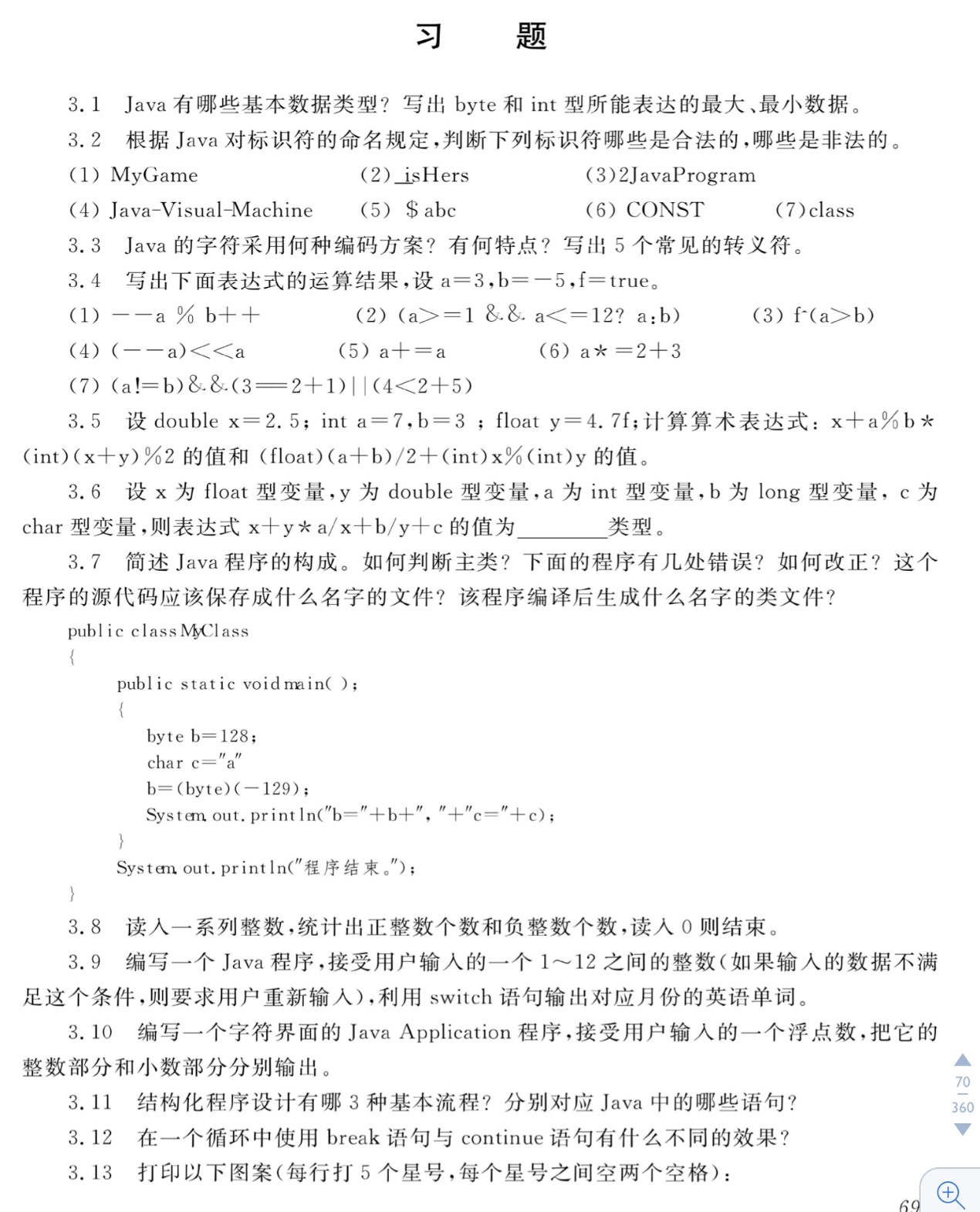
    }

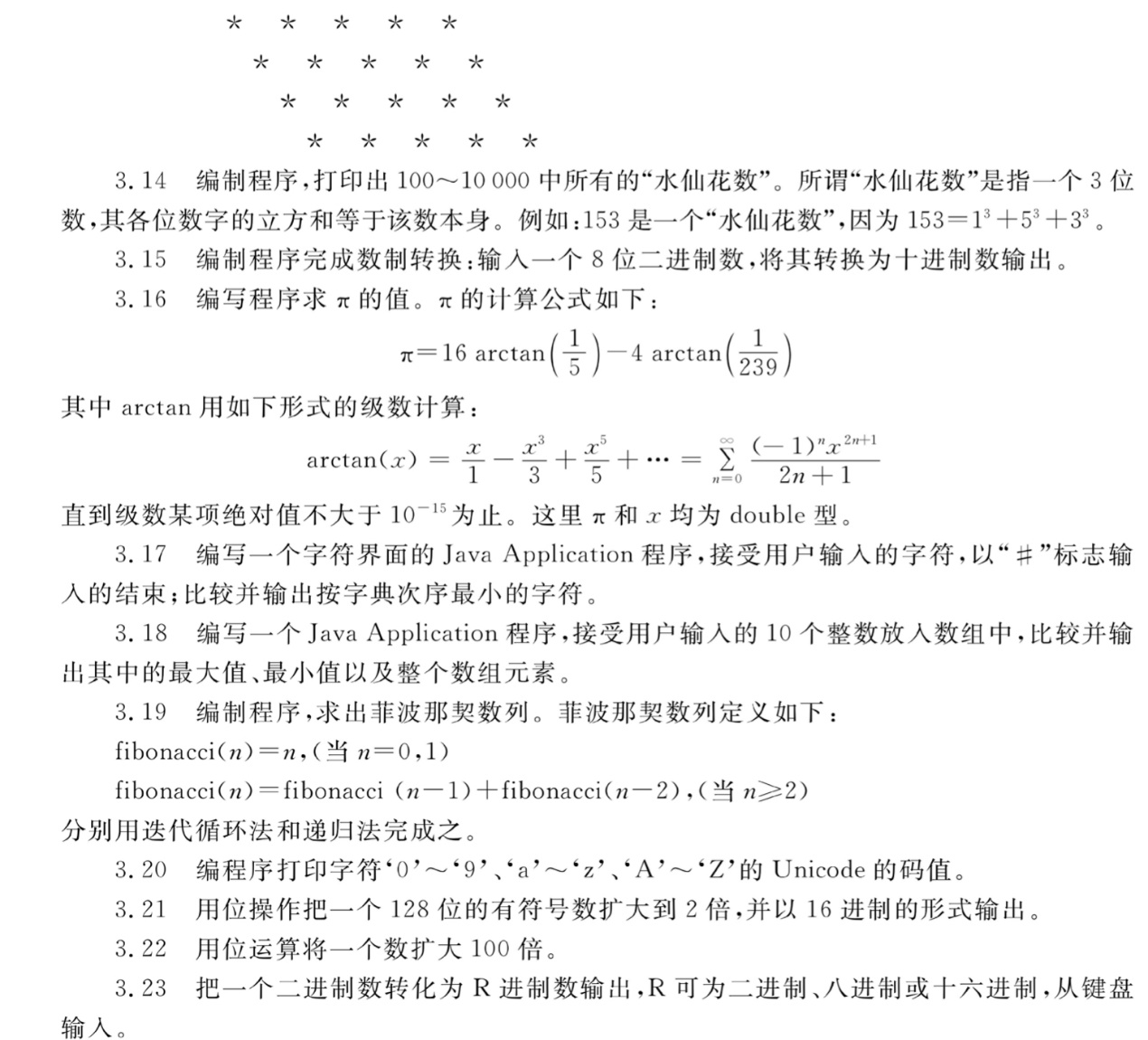
}



# 第三章

## 题目截图





## 解答

### 3.1

**boolean  
int[2^31-1,-2^31]  
long  
float  
char  
double  
short  
byte[127,-128]**

### 3.2

**合法**:(1)(2)(6) **非法**:(3)(4)(5)

### 3.3

Unicode编码;每个Unicode码占用16bit;  
**'\n'  
'\t'  
'\" '  
'\''  
'\\'**

### 3.4

--a%b++=2 (a>=1&&a<=12?a:b)=2 f^(a>b)=false (--a)<<a=2 (a+=a),a=1 a\*=2+3,a=3 (a!=b)&&(3==2+1)||(4<2+5)=true

class Test4{

public static void main(String[] args){

int a=3,b=-5;

boolean f=true;

System.out.println("--a%b++="+(--a%b++));

System.out.println("(a>=1&&a<=12?a:b)="+(a>=1&&a<=12?a:b));

System.out.println("f^(a>b)="+(f^(a>b)));

System.out.println("(--a)<<a="+((--a)<<a));

System.out.println("(a+=a),a="+a);

a=3;

System.out.println("a\*=2+3,a="+a);

a=3;

System.out.println("(a!=b)&&(3==2+1)||(4<2+5)="+((a!=b)&&(3==2+1)||(4<2+5)));

}

}



### 3.5

x+a%b\*(int)(x+y)%2=3.5 (float)(a+b)/2+(int)x%(int)y=7.0

class Test5{

public static void main(String[] args){

double x=2.5;

int a=7,b=3;

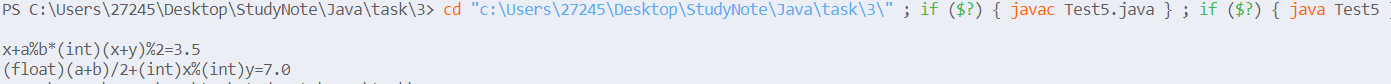
float y=4.7f;

System.out.println("x+a%b\*(int)(x+y)%2="+(x+a%b\*(int)(x+y)%2));

System.out.println("(float)(a+b)/2+(int)x%(int)y="+((float)(a+b)/2+(int)x%(int)y));

}

}



### 3.6

double类型

### 3.7

class MyClass{

public static void main(String[] args)

{

int b=128;

char c='a';

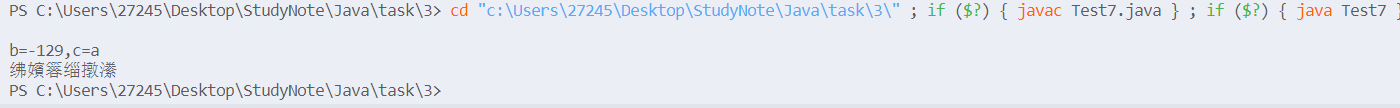
b=(int)(-129);

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

System.out.println("程序结束");

}

}



主类具有main方法; 有五处错误;改正如上; MyClass.java; MyClass.class;

### 3.8

import java.util.Scanner;

class Test8{

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

int x,sum1=0,sum2=0;

x=input.nextInt();

while(x!=0)

{

if(x>0)

sum1++;

else sum2++;

x=input.nextInt();

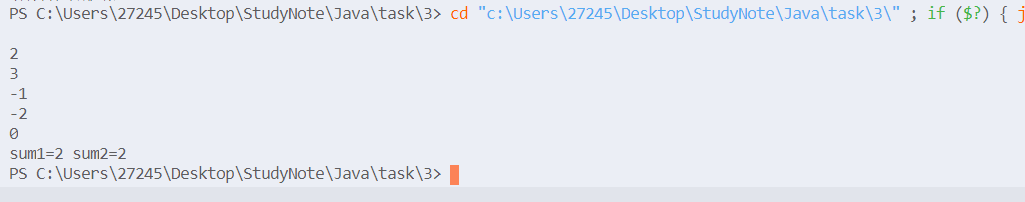
}

System.out.println("sum1="+sum1+" sum2="+sum2);

}

}

1 -1 2 -3 5 0  
sum1=3 sum2=2



### 3.9

13 Error! 12 December

import java.util.Scanner;

class Test9{

public static void main(String[] args)

{

int x;

Scanner input=new Scanner(System.in);

x=input.nextInt();

while(x<=0||x>12)

{

System.out.println("Error!");

x=input.nextInt();

}

switch(x)

{

case 1:{

System.out.println("January");

break;

}

case 2:{

System.out.println("February");

break;

}

case 3:{

System.out.println("March");

break;

}

case 4:{

System.out.println("April");

break;

}

case 5:{

System.out.println("May");

break;

}

case 6:{

System.out.println("June");

break;

}

case 7:{

System.out.println("July");

break;

}

case 8:{

System.out.println("August");

break;

}

case 9:{

System.out.println("September");

break;

}

case 10:{

System.out.println("October");

break;

}

case 11:{

System.out.println("November");

break;

}

case 12:{

System.out.println("December");

break;

}

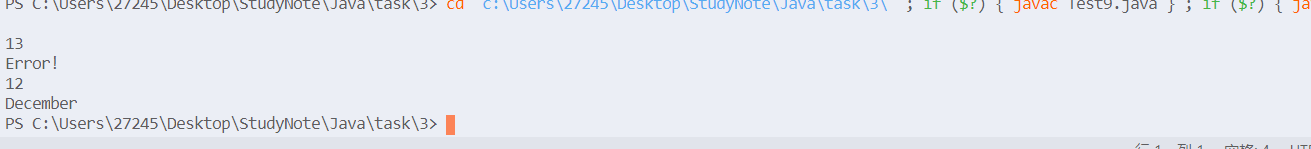
default:

break;

}

}

}



### 3.10

**import** java.text.DecimalFormat;

**import** javax.swing.JOptionPane;

**public** **class** Test10 {

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

String number;

**double** number1=0;

number=JOptionPane.*showInputDialog*(" 输入第一个浮点数");

number1=Double.*parseDouble*(number);

**int** int1=(**int**)(number1);

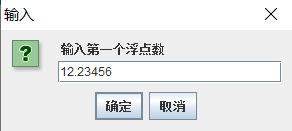
**double** double1=number1-int1;

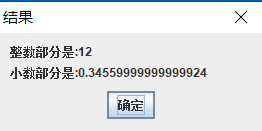
JOptionPane.*showMessageDialog*(**null**, "整数部分是:"+int1+"\n小数部分是:"+double1,"结果",JOptionPane.***PLAIN\_MESSAGE***);

System.*exit*(0);

}

}





### 3.11

顺序结构:声明语句,赋值语句,方法调用语句;  
选择结构:if语句,switch语句;  
循环结构:while语句,do-while语句,for语句;

### 3.12

break:直接从该循环中跳出; continue:跳过本次循环,判断是否进入下次循环;

### 3.13

class Test13{

public static void main(String[] args){

for(int i=0;i<4;i++)

{

for(int j=0;j<i;j++)

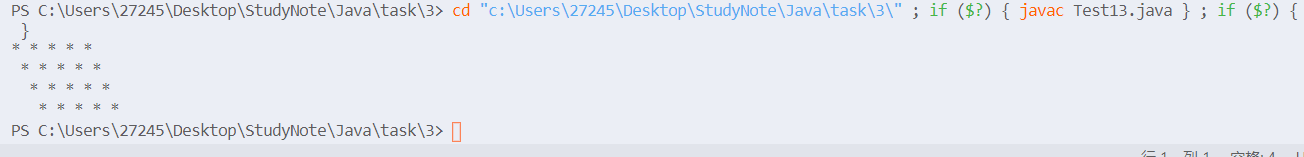
System.out.print(" ");

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

}

}

}



### 3.14

153 370 371 407

class Test14{

public static void main(String[] args)

{

int a,b,c;

for(int i=100;i<1000;i++)

{

a=i/100;

b=(i/10)%10;

c=i%10;

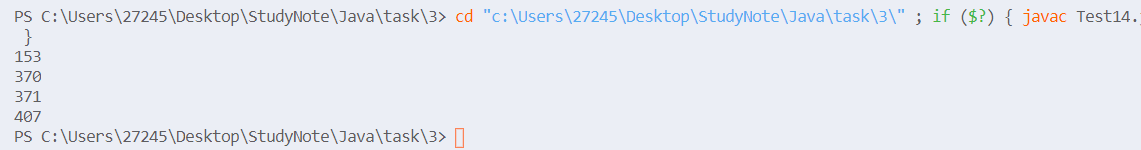
if(a\*a\*a+b\*b\*b+c\*c\*c==i)

System.out.println(i);

}

}

}



### 3.15

import java.util.Scanner;

class Test15{

public static void main(String[] args)

{

Scanner input=new Scanner(System.in);

String s=input.next();

int x=128,sum=0;

for(int i=0;i<8;i++)

{

sum+=(s.charAt(i)-'0')\*x;

x/=2;

}

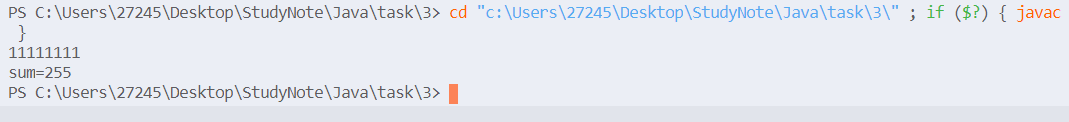
System.out.println("sum="+sum);

}

}

11111111

sum=255



### 3.16

class Test16{

public static void main(String[] args){

double a=16\*arctan(1/5.0),b=4\*arctan(1/239.0);

System.out.println("pi="+(a-b));

}

static double arctan(double x)

{

double sum=0;

double v=x;

int t=1;

double s=t\*v/1;

for(int i=1;Math.abs(s)>1e-15;i++)

{

sum+=s;

v\*=x\*x;

t\*=-1;

s=(t\*v)/(2\*i+1);

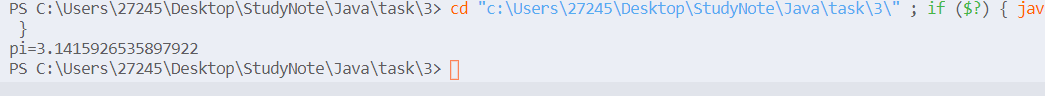
}

return sum;

}

}

pi=3.1415926535897922



### 3.17

**import** java.util.Scanner;

**class** **Test17**{

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

    {

        Scanner input=**new** **Scanner**(System**.**in);

        String s;

        int i=0;

        char t,c;

        s=input**.next**();

        t=s**.charAt**(0);

        c=s**.charAt**(i++);

**while**(c**!=**'#')

        {

**if**(c**>=**'a'**&&**c**<=**'z')

            c+='A'-'a';

**if**(c**<**t)

            t=c;

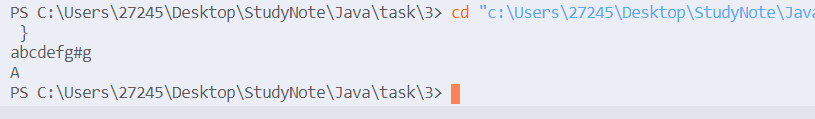
            c=s**.charAt**(i++);

        }

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

    }

}



### 3.19

import java.util.Scanner;

class Test19{

public static void main(String[] args)

{

Scanner input=new Scanner(System.in);

int n;

n=input.nextInt();

int f[]=new int[n+2];

f[0]=1;

f[1]=1;

for(int i=2;i<=n;i++)

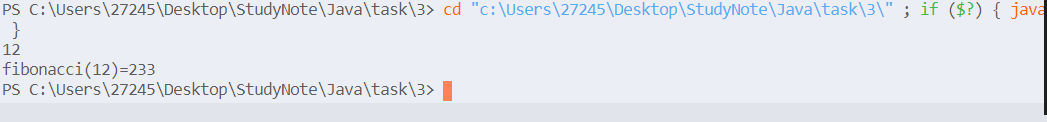
f[i]=f[i-1]+f[i-2];

System.out.println("fibonacci("+n+")="+f[n]);

}

}

5 fibonacci(5)=8



import java.util.Scanner;

class Test19\_1{

public static void main(String[] args)

{

Scanner input=new Scanner(System.in);

int n;

n=input.nextInt();

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

}

static int f(int n)

{

if(n==0||n==1)

return 1;

else {

return f(n-1)+f(n-2);

}

}

}

4 fibonacci(4)=5 6 fibonacci(6)=13



### 3.20

class Test20{

public static void main(String[] args)

{

for(char c='0';c<='9';c++)

System.out.print((int)c+" ");

System.out.println();

for(char c='a';c<='z';c++)

System.out.print((int)c+" ");

System.out.println();

for(char c='A';c<='Z';c++)

System.out.print((int)c+" ");

System.out.println();

}

}

48 49 50 51 52 53 54 55 56 57 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90



### 3.21

public class Test21{

public static void main(String[] args)

{

long a,b,s;

a=1;

b=-1;

System.out.printf("128位的原数: %016x%016x\n",a,b);

s=(b>>>63)&1;

a=(a<<1)+b;

b=b<<1;

System.out.prinf("扩大2倍后的结果:%016x%016x",a,b);

}

}

### 3.22

**import** javax.swing.JOptionPane;

**public** **class** Tested2 {

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

String input =JOptionPane.*showInputDialog*(**null**,"Enter an input:","Mutiply 100",JOptionPane.***QUESTION\_MESSAGE***);

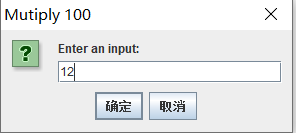
**int** a=Integer.*parseInt*(input),b;

b=(a<<5)+(a<<6)+(a<<2);

JOptionPane.*showMessageDialog*(**null**, a+"\*100="+b);

}

}



### 3.23

**import** javax.swing.JOptionPane;

**public** **class** Test23 {

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

**int** R,number=0;

String s=JOptionPane.*showInputDialog*(**null**,"Enter an input","please enter a binary number",javax.swing.JOptionPane.***QUESTION\_MESSAGE***);

R=Integer.*valueOf*(JOptionPane.*showInputDialog*(**null**,"Enter an input","转换成的目标进制数",JOptionPane.***QUESTION\_MESSAGE***));

**int** len=s.trim().length();

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

{

number =number\*2+(s.charAt(i)-'0');

}

JOptionPane.*showConfirmDialog*(**null**, "二进制数:"+s+"\n对应"+R+"进制数:"+*convert*(number,R),"结果",JOptionPane.***PLAIN\_MESSAGE***);

}

**public** **static** String convert(**int** number,**int** R)

{

String s="";

**int** c;

**char** s1[]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','K','L','M','N'};

**while**(number>0)

{

s=s1[number%R]+s;

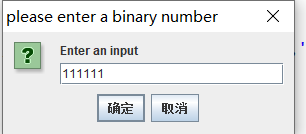
number=number/R;

}

**return** s;

}

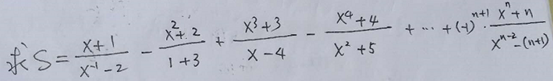
}





## 附加题

### 一、输入正整数n、实数x，求S。



**package** Test3\_2;

**import** java.util.Scanner;

**public** **class** Test3\_2\_1{

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

**int** n;

**double** x,s=0,t1,t2;

**int** p=1;

Scanner input = **new** Scanner(System.***in***);

n=input.nextInt();

x=input.nextDouble();

t1=1;

t2=1/x/x;

**for**(**int** i=1;i<=n;i++)

{

t1\*=x;

t2\*=x;

s+=(t1+i)/(t2-(i+1)\*p)\*p;

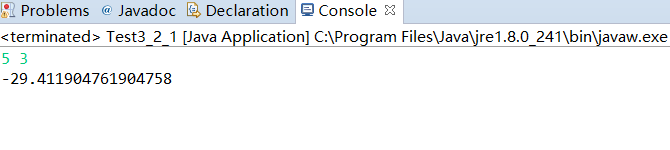
p\*=(-1);

}

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

}

}



### 二、输入正整数N，显示出下列形状的杨辉三角形。(N<=20,注：一定要是下列形状)

**1**

**1 1**

**1 2 1**

**1 3 3 1**

**1 4 6 4 1**

**1 5 10 10 5 1**

**......**

**1 ...... 1**

**package** Test3\_2;

**import** java.util.Scanner;

**public** **class** Test3\_2\_2 {

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

**int** N;

Scanner input=**new** Scanner(System.***in***);

N=input.nextInt();

**int** s[][]=**new** **int**[N+5][N+5];

System.***out***.println(1);

s[1][1]=1;

s[2][1]=1;

s[2][2]=1;

System.***out***.println(1+" "+1);

**for**(**int** i=3;i<=N;i++)

{

s[i][1]=1;

System.***out***.print(1+" ");

**for**(**int** j=2;j<=i-1;j++)

{

s[i][j]=s[i-1][j-1]+s[i-1][j];

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

}

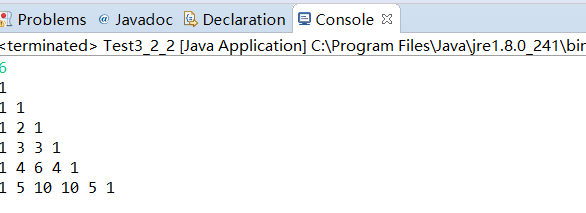
s[i][i]=1;

System.***out***.println(1+" ");

}

}

}



### 三、将3.8题改为：输入一系列整数，统计出其中（1）正整数、负整数的个数；（2）正整数的最大值、最小值；（3）负整数的最大值、最小值；（4）所有数的均方差S,S=(Xi-△x)2,其中△x为平均数，Xi为第i个整数；（5）如输入的是0，则停止输入，然后显示出上述求解的结果。

**package** Test3\_2;

**import** java.util.Scanner;

**class** Test3\_2\_3{

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

{

Scanner input = **new** Scanner(System.***in***);

**int** x,sum1=0,sum2=0;

**int** sum[]=**new** **int**[10000];

**int** i=0,t=0;

**int** max1=0,max2=0,min1=0,min2=0;

x=input.nextInt();

**while**(x!=0)

{

**if**(x>0)

{

sum[i++]=x;

sum1++;

t+=x;

**if**(sum1==1)

{

max1=x;

min1=x;

}

**else**{

**if**(max1<x)max1=x;

**if**(min1>x)min1=x;

}

}

**else**{

sum2++;

sum[i++]=x;

t+=x;

**if**(sum2==1)

{

max2=x;

min2=x;

}

**else**{

**if**(max2<x)max2=x;

**if**(min2>x)min2=x;

}

}

x=input.nextInt();

}

**double** f=0;

**for**(i-=1;i>=0;i--)

{

f+=(sum[i]-t/(sum1+sum2))\*(sum[i]-t/(sum1+sum2));

}

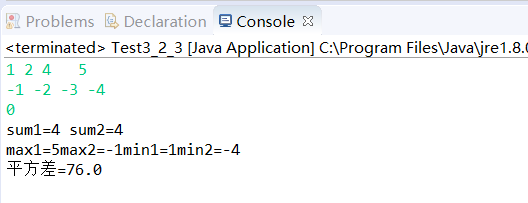
System.***out***.println("sum1="+sum1+" sum2="+sum2);

System.***out***.println("max1="+max1+"max2="+max2+"min1="+min1+"min2="+min2);

System.***out***.println("平方差="+f);

}

}



### 四、将3.13题改为：显示出下列图案

**AB**

**ABCD**

**ABCDEF**

**ABCDEFGH**

**654321**

**4321**

**21**

**package** Test3\_2;

**public** **class** Test3\_2\_4 {

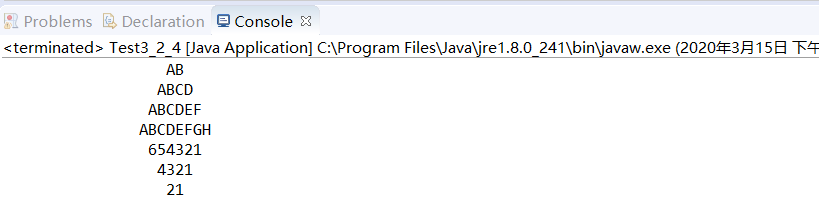
**public** **static** **void** main(String[] args)

{

System.***out***.printf(" AB\n ABCD\n ABCDEF\n ABCDEFGH\n 654321\n 4321\n 21");

}

}



### 五、请写出完成下面功能的程序。假定某班有10个学生，本学期上了3门课，请依次输入 姓名、第1门课成绩、第2门课成绩、第3门课成绩。 即首先输入第一个同学的姓名、第1门课成绩、第2门课成绩、第3门课成绩； 再输入第二个同学的姓名、第1门课成绩、第2门课成绩、第3门课成绩； 再输入第三个同学的姓名、第1门课成绩、第2门课成绩、第3门课成绩；......， 直至输入第10个同学的姓名、第1门课成绩、第2门课成绩、第3门课成绩。 然后求出每个同学的3门课的总分，再按总分对数组排序，最后按总分由高到低的顺序输出排序后的信息：    名次   姓名  第一门   第二门    第三门    总分

**package** Test3\_2;

**import** java.util.Scanner;

**public** **class** Test3\_2\_5 {

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

{

Student p[]=**new** Student[10];

Scanner input=**new** Scanner(System.***in***);

**for**(**int** i=0;i<10;i++){

String name=input.next();

**double** a=input.nextDouble();

**double** b=input.nextDouble();

**double** c=input.nextDouble();

p[i]=**new** Student(name,a,b,c);

p[i].sum();

}

java.util.Arrays.*sort*(p);

**for**(Student a:p){

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

}

}

}

**class** Student{

String name;

**double** a,b,c;

**double** total;

**public** Student(String name,**double** a,**double** b,**double** c)

{

**this**.name=name;

**this**.a=a;

**this**.b=b;

**this**.c=c;

}

**public** **void** sum()

{

**this**.total=**this**.a+**this**.b+**this**.c;

}

**public** **int** compareTo(Student o){

**if**(**this**.total>o.total)

**return** -1;

**else** **return** 1;

}

}

第四章

4-1.修饰类的访问控制有哪些?修饰类中成员的访问控制符有哪些?各有什么含义作用?请举例说明.

修饰类的访问控制:public,protected;

修饰类中成员的访问控制符:private,package protected ,public;

4-2.构造方法特殊在哪里?构造方法什么时候执行?被谁调用?

1.这能在创建对象用new命令调用;  
2.构造方法是在创建对象时被自动调用的;  
3.构造方法名必须与类名相同,没有返回值,可以有参量,并且可以被重载;

4-3.关键字static可以修饰类的那些组成部分?实例变量和类变量的区别是什么?

1.类的属性和方法;  
2.实例变量是非static的成员变量,而类变量是所有对象共有的变量;

4-4静态属性有什么特点?类的对象可以访问或修改静态属性吗?

1.是类的属性,不属于某个对象;

2.可以访问或修改;

4-5构造方法重载的特点是什么?有什么作用?

1.方法名不变,形式参数会有区别;  
2.用于创建对象传递给对象的参数可以多样化

4-6.什么是包?如何创建包?包物理上对应什么操作系统中的实体?

1.包是相关一组类的集合;

2.package语句;

3.文件夹;

4-7.试写出创建一个名为MyPackage包的语句,这个语句应该放在程序的什么位置?

1.package MyPackage;

2.第一行;

4-8.试写出引用MyPackage包中所有类的语句,而引用MyPackage包中的一个MyClass1的语句;

1.import MyPackage.\*;

2.import Mypackage.MyClass1;

4-9阅读下面的程序,说明他们的输出.

class MyClass{

int data;

MyClass(int d)

{

data=d;

}

int getData()

{

return data;

}

void setData(int d)

{

data=d;

}

}

public class Test9{

public static void main(String[] args)

{

MyClass myobj,myref;

myobj = new MyClass(-1);

myref = myobj;

System.out.println("the original data is:"+myobj.getData());

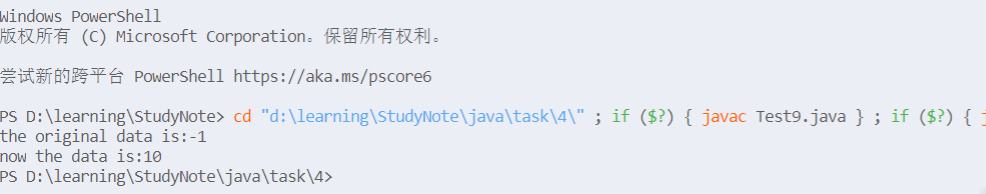
myref.setData(10);

System.out.println("now the data is:"+myobj.getData());

}

}

the original data is:-1 now the data is:10



4-10.编写一个类实现秒表的功能.要求实现开始计时,停止计时,分针秒针归零,获得分针示数,获得秒针示数,获得总时间的毫秒数等功能.

public class Test10{

int minute;//分针示数

double seconds;//秒针示数

void start();//开始计时

void stop();//停止计时

void reZero();//分针秒针归零

int getMinute();//获得分针示数

double getSeconds();//获得秒针示数

int getToatal();//获得总时间的毫秒数

}

4-11.编写一个类实现复数的运算,要求至少实现复数相加,复数相减,复数相乘等功能;

import java.util.Scanner;

public class Test11{

static void add(double a,double b,double c,double d)

{

System.out.println((a+c)+"i+"+(b+d));

}

static void subtract(double a,double b,double c,double d)

{

System.out.println((a-c)+"i+"+(b-d));

}

static void mutiply(double a,double b,double c,double d)

{

System.out.println((a\*d+b\*c)+"i+"+(b\*d-a\*c));

}

public static void main(String args[])

{

double a,b,c,d;

Scanner input =new Scanner(System.in);

a=input.nextDouble();

b=input.nextDouble();

c=input.nextDouble();

d=input.nextDouble();

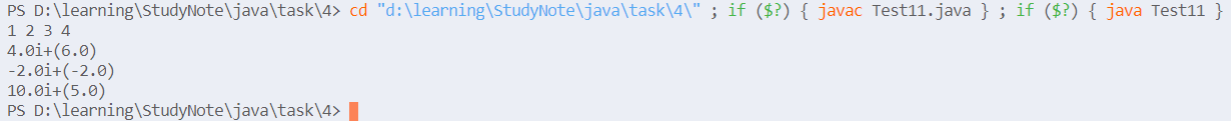
add(a,b,c,d);

subtract(a,b,c,d);

mutiply(a,b,c,d);

}

}



4-12.编程创建一个Box类.要求:定义三个实例变量分别表示立方体的长宽高,定义一个构造方法对这三个变量进行初始化,然后定义一个方法求立方体的体积;创建一个对象,求给定尺寸的立方体的体积;

public class Test12{

public static void main(String[] args)

{

Box a= new Box(1,2,3);

System.out.println(a.getSize());

}

}

class Box{

double length,width,height;

public Box(double l,double w,double h)

{

length = l;

width = w;

height = h;

}

public double getSize()

{

double s=length\*(width\*height);

return s;

}

}



4-13.学生类的创建与使用.

**1.创建一个student类,包括的域有学号,班号,姓名,性别,年龄等,都是private类型;  
2.声明一个构造方法,以初始化对象的所有的域;  
3.声明分别获得个属性的各个Public方法;  
4.声明修改个属性的各个public方法;  
5.声明一个public型的toString()方法,把该类的所有域信息组合成一个字符串;  
6.在类中声明统计班级总人数的私有域count得到班级总人数的public方法(可在构造方法中进行Student对象个数的增加);  
7.将类student放在子包Student中;  
8.在子 包student外,创建测试类student的主类;在主类中,使用student类创建两个student对象,输出对象的所有域信息;修改对象的姓名与年龄,修改后显示各对象的姓名和年龄;你叫两个student对象的年龄大小,输出年龄较大的Student对象;**

import student.Student;

class Test13{

public static void main(String args[]){

Student a=new Student(1,3,"Lucre","boy",20);

Student b=new Student(23,5,"Kiwis","boy",20);

Student c=new Student(24,5,"XiaoWu","boy",20);

System.out.println(a.getId()+",”+a.getClassId()+','+a.getName()+','+a.getSex()+','+a.getAge());

a.setAge(19);

a.setName("Panda");

a.setSex("girl");

System.out.println(a.toString());

if(a.getAge()>b.getAge())

System.out.println(a);

else System.out.println(b);

}

}

package student;

public class Student{

private int id;

private int classId;

private String name;

private String sex;

private int age;

private static int count=0;

public Student(int id,int classId,String name,String sex,int age)

{

this.id=id;

this.classId=classId;

this.name=name;

this.sex=sex;

this.age=age;

count++;

}

public int getId()

{

return id;

}

public int getClassId(){

return classId;

}

public String getName()

{

return name;

}

public String getSex()

{

return sex;

}

public int getAge()

{

return age;

}

public void setId(int x)

{

this.id=x;

}

public void setClassId(int x)

{

this.classId=x;

}

public void setName(String x)

{

this.name=x;

}

public void setSex(String x)

{

this.sex=x;

}

public void setAge(int x)

{

this.age=x;

}

public String toString()

{

return "id="+id+",classId="+classId+",name="+name+",sex="+sex+",age="+age;

}

public int getCount()

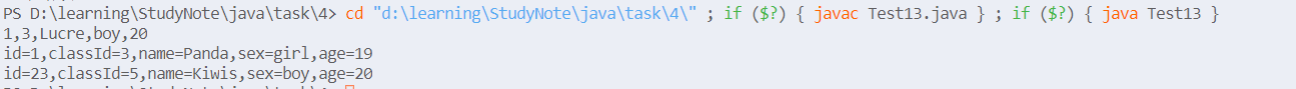
{

return count;

}

}

1,3,Lucre,boy,20  
id=1,classId=3,name=Panda,sex=girl,age=19  
id=23,classId=5,name=Kiwis,sex=boy,age=20



4-14 定义一个学生类和一个课程类,要求能够表示现实生活中一个学生可以选择多门课程的情况.编写程序进行测试;

class Test14{

public static void main(String args[])

{

Student a=new Student("Chengfen");

a.select("math");

a.select("English");

}

}

public class Student{

Curriculumn t[]=new Curriculumn[20];

String name;

int num;

public Student(String s)

{

name=s;

num=0;

}

public void select(String s)

{

t[num++]=new Curriculumn(s);

}

}

public class Curriculumn{

String name;

public Curriculumn(String s)

{

name=s;

}

}

4-15 定义一个方程类,一求方程的两个实根;请将测试主类和方程定义在不同的包中;

import equation.\*;

class Test15{

public static void main(String args[])

{

Equation a= new Equation(4,5,1);

System.out.println("two results are:"+a.solve1()+" "+a.solve2());

}

}

package equation;

import java.lang.Math;

public class Equation{

private int a,b,c;

public Equation(int a,int b,int c)

{

this.a=a;

this.b=b;

this.c=c;

}

public double solve1()

{

return (-b+Math.sqrt(b\*b-4\*a\*c))/2\*a;

}

public double solve2()

{

return (-b-Math.sqrt(b\*b-4\*a\*c))/2\*a;

}

}



第五章

## 5-1 试描述继承下的父类和子类的概念;父类和子类有何关系?

1.父类是子类的一般化,而子类是父类的特殊化;  
2.子类是父类的公共性基础上的延伸与扩展;

## 5-2 什么是单重继承?什么是多重继承?Java采用什么继承

1.一个父类只有一个子类是单继承;  
2.一个父类可以有多个子类是多继承;  
3.多继承;

## 5-3 Java中如何定义继承关系?写出定义类库中的类java.awt.Frame的子类MyFrame的类头的语句.

1.extend是语句  
2.import java.awt.Frame;  
class MyFrame extends Frame

## 5-4 关键字protected的作用是什么?什么情况下比较好?

1.父类定义的proteceted成员在子类和同一包内可以直接使用;  
2.如果确实需要在子类中修改超类的成员;

## 5-5 什么叫方法的重新定义?

在子类中重新定义父类的同名方法;如果子类重新定义了同名方法,我们则可以:  
super.function();调用父类的同名方法;

## 5-6 试解释构造方法重载作用.一个构造方法如何调用同类的其他方法函数?如何调用父类的构造方法?

1.构造方法的重载,针对不同的参数个数或类型创建类的对象;  
2.this(); 3.super();

## 5-7 阅读下面的程序并写出程序的执行结果,并说明为什么.

class S1{

public static void main(String args[])

{

new S2();

}

S1()

{

System.out.println("S1");

}

}

class S2 extends S1{

S2()

{

System.out.println("S2");

}

}

S1 S2

## 5-8 定义一个类MyRectangle代表矩形,为矩形定义getLength方法(获得矩形的长度),getWidth方法(获得矩形的宽度),setLength(设置长度),setWidth方法(设置宽度),getArea方法(求矩形的面积),toString方法(显示矩形的格式),为矩形派生出一个子类MySquare代表正方形,并对getArea和toString进行重写.并测试;

class Test8{

public static void main(String args[])

{

MySquare a =new MySquare(3.5);

System.out.println("Area is :"+a.getArea());

System.out.println(a);

MyRectangle b = new MyRectangle(12,3);

System.out.println("Area is :"+b.getArea());

System.out.println(b);

b.setWidth(5);

System.out.println(b);

}

}

class MySquare extends MyRectangle{

public MySquare(double s)

{

super(s,s);

}

public double getArea()

{

return super.getArea();

}

public String toString()

{

return "边长是:"+super.getWidth();

}

}

public class MyRectangle{

private double width;

private double length;

public MyRectangle(double a,double b)

{

width=a;

length=b;

}

public double getLength()

{

return length;

}

public double getWidth()

{

return width;

}

public void setLength(double a)

{

length=a;

}

public void setWidth(double b)

{

width=b;

}

public double getArea()

{

return width\*length;

}

public String toString()

{

return "常为:"+length+",宽为:"+width;

}

} 

## 5-9 编写一个类,以实现地址的概念,包阔的属性有"国家","省份","市县","街道","门牌","单位","邮编",定义构造方法设置这些属性,并定义一个方法返回一个寄给该地址的信封的字符串.请从地址类派生出国内,国际两种模式,要求在子类中通过方法重载,返回国际或国内的地址字符串.

class Test9

{

public static void main(String args[])

{

NativeAddress a =new NativeAddress("China","Anhui","shucheng","Walk","520","PostOffice","0520");

AbroadAddress b =new AbroadAddress("America","Dex","Geroge","Upon","345","bank","5432");

System.out.println(a);

System.out.println(b);

}

}

class Address{

String country;

String province;

String city;

String street;

String doorpalte;

String unit;

String postnumber;

public Address(String a,String b,String c,String d,String e,String f,String h)

{

country=a;

province=b;

city=c;

street=d;

doorpalte=e;

unit=f;

postnumber=h;

}

public String toString()

{

return postnumber+'\n'+country+province+city+street+doorpalte+unit+'\n';

}

}

class AbroadAddress extends Address{

public AbroadAddress(String a,String b,String c,String d,String e,String f,String h)

{

super(a,b,c,d,e,f,h);

}

public String toString()

{

return super.country+'\n'+super.province+'\n'+super.city+'\n'+super.street+'\n'+super.doorpalte+'\n'+super.unit+'\n'+super.postnumber;

}

}

class NativeAddress extends Address{

public NativeAddress(String a,String b,String c,String d,String e,String f,String h)

{

super(a,b,c,d,e,f,h);

}

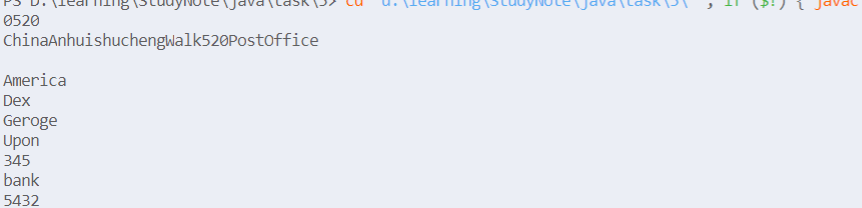
public String toString()

{

return super.toString();

}

}



# 第六章

## 6-8.

### 1.

Person.java

abstract class Person{

    protected String name;

    public Person(String name)

    {

        this**.**name = name ;

    }

    abstract public double pay();

    public String toString()

    {

        return name+"的工资支出:"+this**.**pay()+'\n';

    }

}

Teacher.java

class Teacher extends Person{

    private double baseWage ;

    private  double teachingHours;

    public Teacher(String name,double baseWage,double teachingHours)

    {

        super(name);

        this**.**baseWage = baseWage;

        this**.**teachingHours = teachingHours ;

    }

    public double pay()

    {

        return baseWage + teachingHours \* 30;

    }

}

Student.java

class Student extends Person{

    private double scholarship ;

    public Student(String name,double scholarship)

    {

        super(name);

        this**.**scholarship = scholarship;

    }

    public double pay()

    {

        return scholarship ;

    }

}

Test.java

class Test{

    public static void main(String[] args)

    {

        Teacher a = new Teacher("Andriod",5000,30);

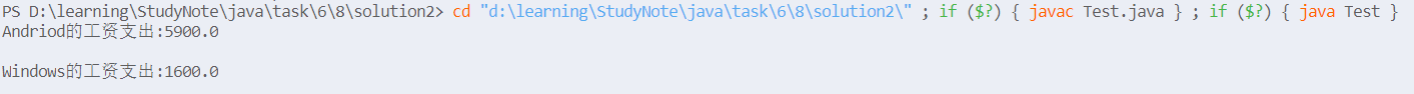
        Student b = new Student("Windows",1600);

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

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

    }

}



### 2.

PersonPay.java

public interface PersonPay

{

    public double pay();

}

Person.java

abstract class Person implements PersonPay{

    protected String name;

    public Person(String name)

    {

        this**.**name = name ;

    }

    public String toString()

    {

        return name+"的工资支出:"+this**.**pay()+'\n';

    }

}

其余与第一种方法无异;

## 6-9

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

public class Test extends JFrame{

    private String name;

    private JLabel nameLabel;

    private JTextField nameField , displayField ;

    private JButton exitButton ;

    public Test()

    {

        super("Inner Class Demonstration");

        Container container = getContentPane();

        container**.**setLayout( new FlowLayout());

        nameLabel = new JLabel("姓名");

        nameField = new JTextField(20);

        container**.**add(nameLabel);

        container**.**add(nameField);

        displayField = new JTextField(30);

        displayField**.**setEditable(false);

        container**.**add(exitButton);

        ActionEventhandler handler =  new ActionEventhandler();

        nameField**.**addActionListener(handler);

        exitButton**.**addActionListener(handler);

    }

    public void displayName()

    {

        displayField**.**setText("姓名是:"+name);

    }

    public static void main(String[] args)

    {

        Test window = new Test();

        window**.**setSize(400,150);

        window**.**setVisible(true);

    }

    private class ActionEventhandler implements ActionListener{

        public void actionPerformed(ActionEvent event)

        {

            if(event**.**getSource()==exitButton)

            System**.**exit(0);

            else if(event**.**getSource()==nameField)

            {

                name = event**.**getActionCommand();

                nameField**.**setText("");

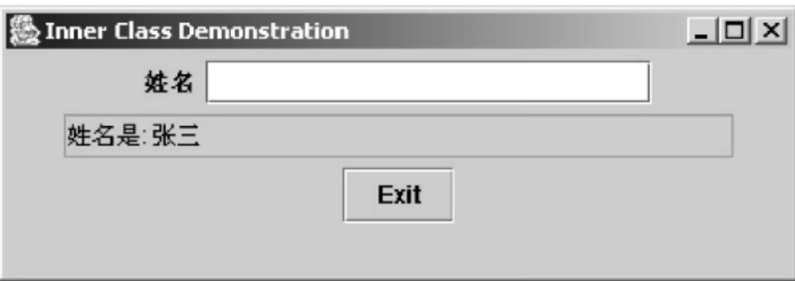
            }

            displayName();

        }

    }

}



## 6-10

Time.java

import java.text.DecimalFormat;

public class Time{

    private int hour;

    private int minute;

    private int second;

    private static DecimalFormat twoDigits = new DecimalFormat("00");

    public Time()

    {

        this(0,0,0);

    }

    public Time(int h)

    {

        this(h,0,0);

    }

    public Time(int h,int m)

    {

        this(h,m,0);

    }

    public Time(int h,int m,int s)

    {

        setTime(h,m,s);

    }

    public Time(Time time)

    {

        this(time**.**getHour(),time**.**getMinute(),time**.**getSecond());

    }

    public void setTime(int h,int m,int s)

    {

        setHour(h);

        setMinute(m);

        setSecond(s);

    }

    public void setHour(int h)

    {

        hour = h;

    }

    public void setMinute(int m)

    {

        minute = m;

    }

    public void setSecond(int s)

    {

        second = s;

    }

    public int getHour()

    {

        return hour;

    }

    public int getMinute()

    {

        return minute;

    }

    public int getSecond()

    {

        return second;

    }

    public String toString()

    {

        String s;

        s=getSecond()+":"+twoDigits**.**format(getMinute())+":"+twoDigits**.**format(getSecond())+(getHour()<12?"AM":"PM");

        return s;

    }

}

Test.java

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

public class TimeTestWindow extends JFrame{

    private Time time;

    private JLabel hourLabel,minuteLabel,secondLabel ;

    private JTextField hourField,minuteField,secondField,displayField;

    public TimeTestWindow()

    {

        super("嵌套内部类使用例子");

        time= new Time();

        createGUI();

        registerEventHandlers();

    }

    private void createGUI()

    {

        Container container = getContentPane();

        container**.**setLayout(new FlowLayout());

        hourLabel = new JLabel("设置时间:");

        hourField = new JTextField(10);

        container**.**add(hourLabel);

        container**.**add(hourField);

        minuteField = new JLabel("设置分:");

        minuteField = new JTextField(10);

        container**.**add(minuteLabel);

        container**.**add(minuteField);

        secondLabel = new JLabel("设置秒:");

        secondField = new JTextField(10);

        container**.**add(secondLabel);

        containeradd(secondField);

        displayField = new JTextField(30);

        displayField**.**setEditable(false);

        container**.**add(displayField);

    }

    private void registerEventHandlers()

    {

        ActionEventHandler handler = new ActionEventHandler();

        hourField**.**addActionListener(handler);

        minuteField**.**addActionListener(handler);

        secondField**.**addActionListener(handler);

    }

    private class ActionEventHandler implements ActionListener{

        public void actionPerformed(ActionEvent event)

        {

            if(event**.**getSource()==hourField)

            {

                time**.**setHour(Integer**.**parseInt(event**.**getActionCommand()));

            }

            else if(event**.**getSource()==minuteField)

            {

                time**.**setMinute(Integer**.**parseInt(event**.**getActionCommand()));

            }

            else if(event**.**getSource()==secondField)

            {

                time**.**setSecond(Integer**.**parseInt(event**.**getActionCommand()));

            }

            displayTime();

        }

    }

    public void displayTime()

    {

        displayField**.**setText("时间是:"+time);

    }

    public static void main(String[] args)

    {

        TimeTestWindow window = new TimeTestWindow();

        window**.**addWindowListener(

            newWindowAdapter(){

                public void windowClosing(WindowEvent event)

                {

                    System**.**exit(0);

                }

            }

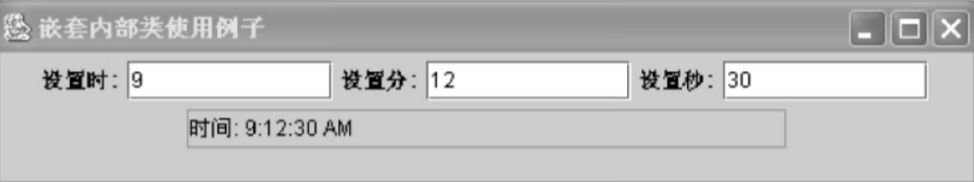
        );

        window**.**setSize(550,150);

        window**.**setVisible(true);

    }

}



# 第七章

## 7.4

可以将字符串分解为组成它的单词符号.

## 7.5

public class Test5{

public static void main(String[] args)

{

String str1 = new String("Good luck!");

String str2 = new String("Andry");

String str3 = new String(str1.concat(str2));

System.out.println(str3);

}

}



## 7.6

public class Test6{

public static void main(String args[])

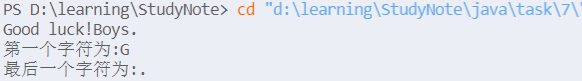
{

String str = new String("Good luck!Boys.");

System.out.println(str+'\n'+"第一个字符为:"+str.charAt(0)+'\n'+"最后一个字符为:"+str.charAt(str.length()-1));

}

}



## 7.7

import java.util.\*;//导入vector

class Test7{

public static void main(String args[])

{

Vector vec = new Vector();

vec.add("数学");

vec.add("语文");

vec.add("英语");

vec.add("政治");

vec.add("物理");

vec.add("化学");

System.out.println("课程有:");

Enumeration items = vec.elements();

while(items.hasMoreElements())

System.out.println(items.nextElement());

System.out.println("物理所放的位置在:"+vec.indexOf("物理"));

}

}



## 7.8

import java.util.\*;

class Test8{

public static void main(String args[])

{

List<Student> list =new ArrayList<Student>();

list.add(new Student("Lucre",20));

list.add(new Student("Amy",21));

Iterator iter = list.iterator();

while(iter.hasNext())

{

Student s = (Student) iter. next();

s.showMessage();

}

}

}

class Student{

private String name;

private int age;

public Student(String s,int a)

{

name = s;

age = a;

}

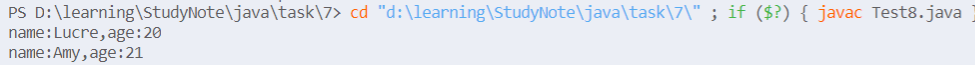
public void showMessage()

{

System.out.println("name:"+name+",age:"+age);

}

}



## 7.9

import java.util.\*;

class Test9{

    public static void count(String str){

    //*创建26个空间大小的数组，存放26个字母*

        int[] nums = new int[26];

        String loc[] = new String[26];

        for(int i=0;i<26;i++)

        loc[i] = "";

        int count=0;

        for(char i: str**.**toCharArray()){

        //*自动将char i转化成ascall码*

            if(i>=97 && i<= 122){

            //*利用数组的索引进行存储*

                nums[i-97]++;

                loc[i-97]+=count+" ";

            }

            count++;

        }

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

            if(nums[i] != 0){

                //*i加上97并且再转化为char类型就可以显示相应的字符*

                char j = (char)(i+97);

                System**.**out**.**println( j + "====" + nums[i]+",location:"+loc[i]);

            }

        }

    }

    public static void main(String[] args)

    {

        Scanner input = new Scanner(System**.**in);

        String s = input**.**nextLine();

        count(s);

        MyStringTokenizer test = new MyStringTokenizer(s);

        System**.**out**.**println("这句话总共有"+test**.**counts()+"单词,依次是:");

        while(test**.**hasMoreWords())

        {

            System**.**out**.**println(test**.**nextWord());

        }

        test**.**nextWord1();

    }

}

class MyStringTokenizer{

    String s;

    String[] words ;

    int index = 0;

    public MyStringTokenizer(String s)

    {

        this**.**s = s;

        this**.**words = s**.**split("\\n|\\t|\\r| ");//*分割单词*

    }

    public int counts()

    {

        return words**.**length;//*返回单词数*

    }

    public boolean hasMoreWords()

    {

        if(index<words**.**length)

        {

            return true;

        }

        else return false;

    }

    public String nextWord(){

        index++;

        return words[index-1];

    }

    public void nextWord1(){

        index = words**.**length-1;

        while(index>=0)

        {

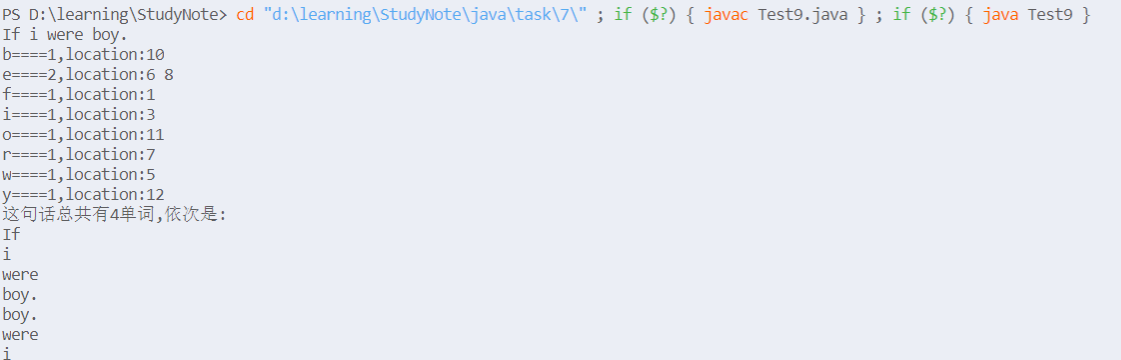
            System**.**out**.**println(words[index]);

            index--;

        }

    }

}



## 7.10

import java.util.\*;

class Test10{

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

String s = input.nextLine();

MyStringTokenizer test = new MyStringTokenizer(s);

System.out.println("这句话总共有"+test.counts()+"单词,依次是:");

while(test.hasMoreWords())

{

System.out.println(test.nextWord());

}

}

}

class MyStringTokenizer{

String s;

String[] words ;

int index = 0;

public MyStringTokenizer(String s)

{

this.s = s;

this.words = s.split("\\n|\\t|\\r| ");//分割单词

}

public int counts()

{

return words.length;//返回单词数

}

public boolean hasMoreWords()

{

if(index<this.counts())

{

return true;

}

else return false;

}

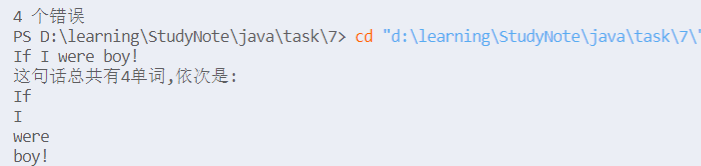
public String nextWord(){

index++;

return words[index-1];

}

}



## 附加题

import java.util.Scanner;

public class Extra{

        public static void main(String[] args)

        {

            Scanner input = new Scanner(System**.**in);

            String s ="";

            int a=0,b=0;

            for(int i=0;i<5;i++)

            {

                s = input**.**nextLine();

                MyStringTokenizer test = new MyStringTokenizer(s);

                a+=test**.**countWords();

                b+=test**.**countSentences()+1;

            }

            System**.**out**.**println("这句话总共有"+a+"单词,共有"+b+"个句子");

        }

}

class MyStringTokenizer{

    String s;

    String[] words ;

    String[] sentences;

    int index = 0;

    public MyStringTokenizer(String s)

    {

        this**.**s = s;

        this**.**words = s**.**split("\\n|\\t|\\r| ");//*分割单词*

        this**.**sentences = s**.**split(".|");

    }

    public int countWords()

    {

        return words**.**length;//*返回单词数*

    }

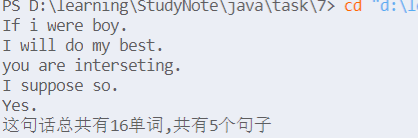
    public int countSentences()

    {

        return sentences**.**length;//*返回单词数*

    }

}



# 第八章

## 8.2

import java.awt.\*;

import javax.swing.\*;

import java.util.Date;

class Test2 extends JFrame{

public void paint(Graphics g)

{

int x1,y1,x2,y2;

x1 = (int) (Math.random() \* 500);

y1 = (int) (Math.random() \* 500);

x2 = (int) (Math.random() \* 500);

y2 = (int) (Math.random() \* 500);//随机生成两个端点

g.setColor(new Color((int)(Math.random()\*256),(int)(Math.random()\*256),(int)(Math.random()\*256)));//随机生成颜色

g.drawLine(x1,y1,x2,y2);

}

public static void main(String[] args){

Test2 win = new Test2();

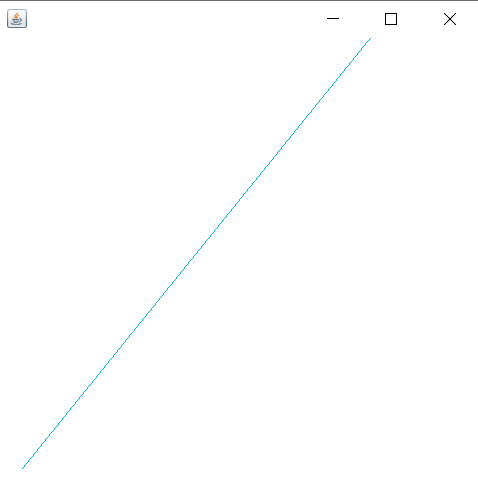
win.setSize(500,500);

win.setVisible(true);

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

}

}



## 8.3

import java.awt.\*;

import javax.swing.\*;

import java.util.Date;

class Test3 extends JFrame{

public void paint(Graphics g)

{

for(int i=1;i<=8;i++)

{

g.setColor(new Color((int)(Math.random()\*256),(int)(Math.random()\*256),(int)(Math.random()\*256)));//随机生成颜色

g.drawOval(200-i\*10,200-i\*10,i\*20,i\*20);//前两个坐标并非圆心

}

}

public static void main(String[] args){

Test3 win = new Test3();

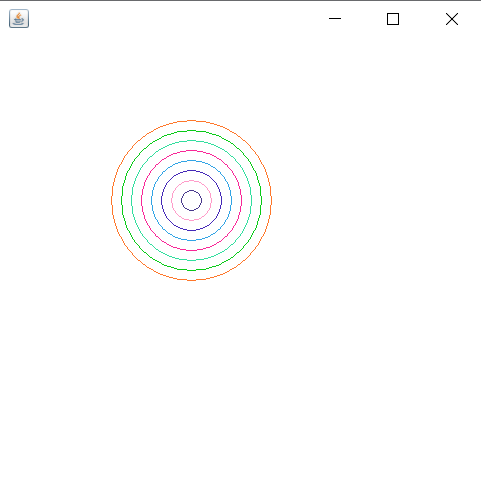
win.setSize(500,500);

win.setVisible(true);

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

}

}



## 8.4

import java.awt.\*;

import javax.swing.\*;

import java.util.Date;

class Test4 extends JFrame{

public void paint(Graphics g)

{

for(int i=1;i<=8;i++)

{

g.setColor(new Color((int)(Math.random()\*256),(int)(Math.random()\*256),(int)(Math.random()\*256)));//随机生成颜色

g.drawArc(200-i\*10,200-i\*10,i\*20,i\*20,0,360);

}

}

public static void main(String[] args){

Test4 win = new Test4();

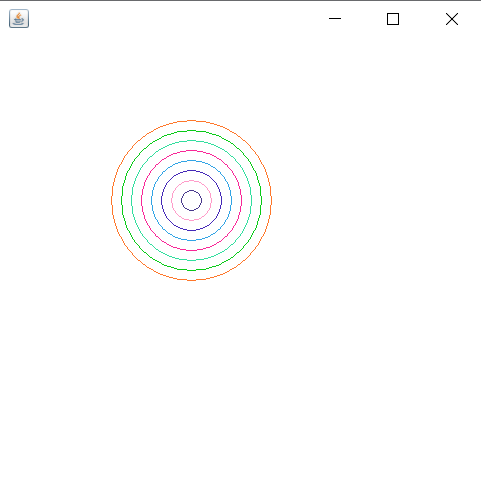
win.setSize(500,500);

win.setVisible(true);

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

}

}



## 8.5

import java.awt.\*;

import java.awt.geom.GeneralPath;

import javax.swing.\*;

public class Test5 extends JFrame{

public Test5()

{

super("Draw 2D Shapes");

getContentPane().setBackground(Color.WHITE);

setSize(200,200);

setVisible(true);

}

public void paint(Graphics g){

super.paint(g);

int x[] = {60,72,114,78,88,60,32,42,6,48};//把五角星的点坐标顺时针列出来

int y[] = {60,96,96,114,156,122,156,114,96,96};

Graphics2D g2 = (Graphics2D) g ;//转换

GeneralPath star = new GeneralPath();

star.moveTo(x[0],y[0]);

for(int count = 1;count < x.length;count++)

{

star.lineTo(x[count],y[count]);//连线

}

star.closePath();

g2.setColor(Color.RED);

g2.fill(star);

}

public static void main(String[] args)

{

Test5 win = new Test5();

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

}

}



## 8.6

import java.awt.\*;

import java.awt.geom.GeneralPath;

import javax.swing.\*;

public class Test6 extends JFrame{

    public Test6()

    {

        super("Draw 2D Shapes");

        getContentPane()**.**setBackground(Color**.**WHITE);

        setSize(300,300);

        setVisible(true);

    }

    public void paint(Graphics g){

        super**.**paint(g);

        g**.**drawRect(40, 40, 100, 100);

        g**.**drawLine(40, 40, 60, 20);

        g**.**drawLine(60, 20, 160, 20);

        g**.**drawLine(140, 40, 160, 20);

        g**.**drawLine(160, 120, 160, 20);

        g**.**drawLine(160, 120, 140, 140);

        Graphics2D g2 = (Graphics2D) g ;//*转换*

        Stroke dashed = new BasicStroke(3, BasicStroke**.**CAP\_BUTT, BasicStroke**.**JOIN\_BEVEL, 0, new float[]{9}, 0);

        g2**.**setStroke(dashed);

        g2**.**drawLine(60,120,40,140);

        g2**.**drawLine(60,120,160,120);

        g2**.**drawLine(60,20,60,120);

    }

    public static void main(String[] args)

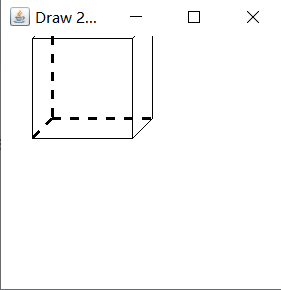
    {

        Test6 win = new Test6();

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

    }

}



## 8.7

import java.awt.Color;

import java.awt.GradientPaint;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.geom.Rectangle2D;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class Test7 extends JFrame {

FillGradientPanel fillGradientPanel = new FillGradientPanel(); // 创建面板类的实例

public static void main(String args[]) { // 主方法

Test7 frame = new Test7(); // 创建窗体类的实例

frame.setVisible(true); // 显示窗体

}

public Test7() {

super(); // 调用超类的构造方法

setTitle("为图形填充渐变色"); // 窗体标题

setBounds(100, 100, 340, 360); // 窗体的显示位置和大小

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); // 窗体关闭方式

add(fillGradientPanel); // 将面板类的实例添加到窗体容器中

}

class FillGradientPanel extends JPanel { // 创建内部面板类

public void paint(Graphics g) { // 重写paint()方法

Graphics2D g2 = (Graphics2D) g; // 获得Graphics2D对象

Rectangle2D.Float rect = new Rectangle2D.Float(20, 20, 280, 280);// 创建矩形对象

// 创建循环渐变的GraphientPaint对象

GradientPaint paint = new GradientPaint(5,40,Color.RED,15,50,Color.YELLOW,true);

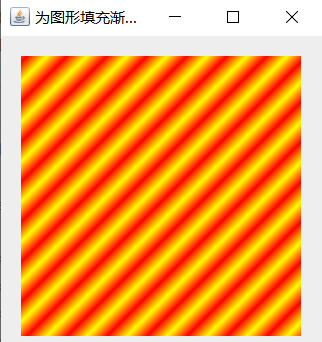
g2.setPaint(paint);// 设置渐变

g2.fill(rect);// 绘制矩形

}

}

}



## 8.8

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Graphics;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JSlider;

import javax.swing.event.ChangeEvent;

import javax.swing.event.ChangeListener;

class SineDraw extends JPanel {

private static final int SCALEFACTOR = 200;

private int cycles;

private int points;

private double[] sines;

private int[] pts;

public SineDraw() {

setCycles(5);

}

public void setCycles(int newCycles) {

cycles = newCycles;

points = SCALEFACTOR \* cycles \* 2;

sines = new double[points];

for (int i = 0; i < points; i++) {

double radians = (Math.PI / SCALEFACTOR) \* i;

sines[i] = Math.sin(radians);

}

repaint();

}

public void paintComponent(Graphics g) {

super.paintComponent(g);

int maxWidth = getWidth();

double hstep = (double) maxWidth / (double) points;

int maxHeight = getHeight();

pts = new int[points];

for (int i = 0; i < points; i++)

pts[i] = (int) (sines[i] \* maxHeight / 2 \* .95 + maxHeight / 2);

g.setColor(Color.RED);

for (int i = 1; i < points; i++) {

int x1 = (int) ((i - 1) \* hstep);

int x2 = (int) (i \* hstep);

int y1 = pts[i - 1];

int y2 = pts[i];

g.drawLine(x1, y1, x2, y2);

}

}

}

public class Test8 extends JPanel {

private SineDraw sines = new SineDraw();

private JSlider adjustCycles = new JSlider(1, 30, 5);

public Test8() {

super(new BorderLayout());

add(BorderLayout.CENTER,sines);

adjustCycles.addChangeListener(new ChangeListener() {

public void stateChanged(ChangeEvent e) {

sines.setCycles(((JSlider) e.getSource()).getValue());

}

});

add(BorderLayout.SOUTH, adjustCycles);

}

public static void main(String[] args) {

JPanel p = new Test8();

JFrame frame = new JFrame();

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

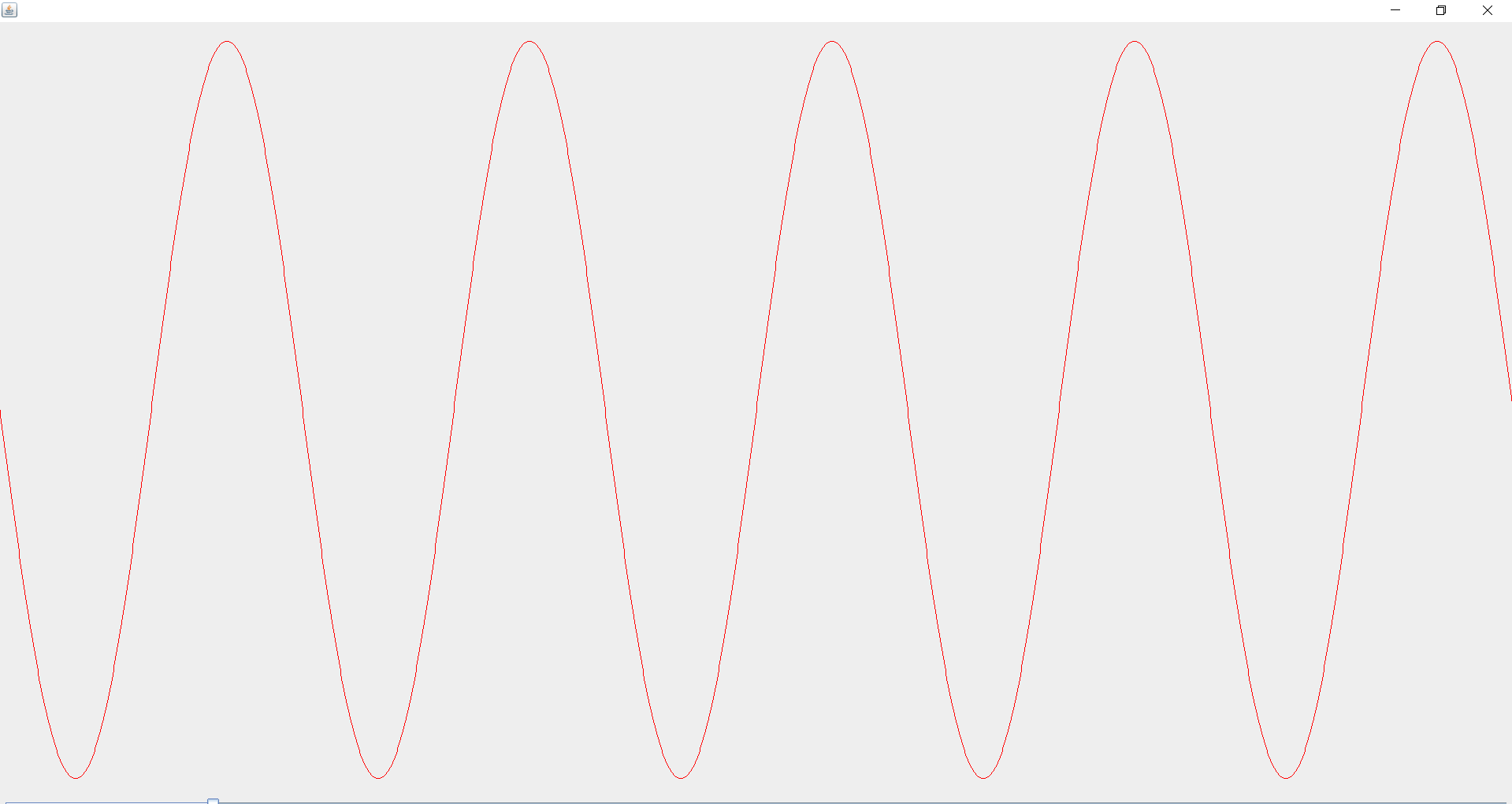
frame.add(p);

frame.setSize(500,500 );

frame.setVisible(true);

}

}



## 附加题

import java.awt.\*;

import java.awt.geom.QuadCurve2D;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class Extra extends JFrame{

    public static void main(String args[]) {

        new Extra();

    }

    public Extra() {

        Toolkit kit = Toolkit**.**getDefaultToolkit();

        Dimension screen = kit**.**getScreenSize();

        int wight = screen**.**width;

        int hight = screen**.**height;

        this**.**add(new DrawQuadCurve());

        this**.**setBounds(wight/4, hight/4, 400, 300);

        this**.**setDefaultCloseOperation(3);

        this**.**setVisible(true);

    }

}

class DrawQuadCurve extends JPanel{

    public void paint(Graphics g) {

        Graphics2D g2 = (Graphics2D)g;

        QuadCurve2D**.**Double quadCurve1 = new QuadCurve2D**.**Double(10,20,100,32,200,200);

        g2**.**setColor(Color**.**RED);

        g2**.**draw(quadCurve1);

        QuadCurve2D**.**Double quadCurve2 = new QuadCurve2D**.**Double(10,25,100,123,200,156);

        g2**.**setColor(Color**.**BLUE);

        g2**.**draw(quadCurve2);

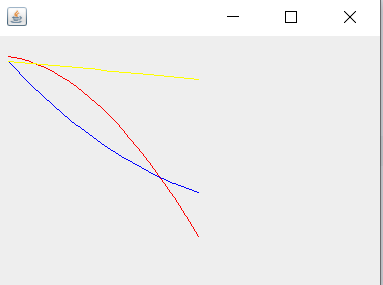
        QuadCurve2D**.**Double quadCurve3 = new QuadCurve2D**.**Double(10,25,100,33,200,43);

        g2**.**setColor(Color**.**YELLOW);

        g2**.**draw(quadCurve3);

    }

}



# 第九章

## 9.6

Caculator.java

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Caculator extends JApplet{

    JLabel aJLabel;

    JTextField aField,bField;

    JTextArea output;

    JButton action;

    public void init()

    {

        Container container = getContentPane();

        container**.**setLayout(new FlowLayout());

        aJLabel = new JLabel("请输入两个整型数据:");

        aField = new JTextField(5);

        bField = new JTextField(5);

        output = new JTextArea(5,20);

        action = new JButton("计算");

        action**.**addActionListener(new ActionListener()

        {

            public void actionPerformed(ActionEvent event)

            {

                int a,b;

                a = Integer**.**parseInt(aField**.**getText());

                b = Integer**.**parseInt(bField**.**getText());

                String s = a+"+"+b+"="+(a+b)+"\n";

                s+=a+"-"+b+"="+(a-b)+"\n";

                s+=a+"\*"+b+"="+(a\*b)+"\n";

                s+=a+"/"+b+"="+(a/b)+"\n";

                s+=a+"%"+b+"="+(a%b)+"\n";

                output**.**setText(s);

            }

        });

        output**.**setEditable(false);

        container**.**add(aJLabel);

        container**.**add(aField);

        container**.**add(bField);

        container**.**add(action);

        container**.**add(output);

    }

}

Caculator.html

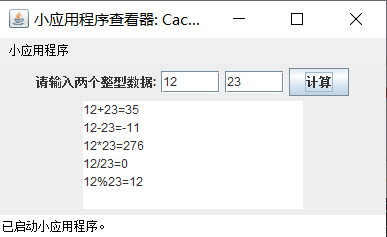
<html>

    <body>

        <Applet code = "Caculator.class" Height = 30 Width = 300></Applet>

    </body>

</html>



## 9.7

package Test7;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class ColorChange extends JFrame{

/\*\*

*\**

\*/

    private static final long serialVersionUID = 1L;

    private JButton button;

    private Color color;

    public ColorChange()

    {

        super("Change Color");

        Container container = getContentPane();

        container**.**setLayout(new FlowLayout());

        button = new JButton("Change Color");

        button**.**addActionListener(new ActionListener()

        {

            public void actionPerformed(ActionEvent event)

            {

                color = JColorChooser**.**showDialog(ColorChange**.**this, "Choose a color", color);

                if(color == null)

                {

                    color = Color**.**LIGHT\_GRAY;

                }

                container**.**setBackground(color);

            }

        });

        container**.**add(button);

        setSize(500,500);

        setVisible(true);

    }

    public static void main(String []args)

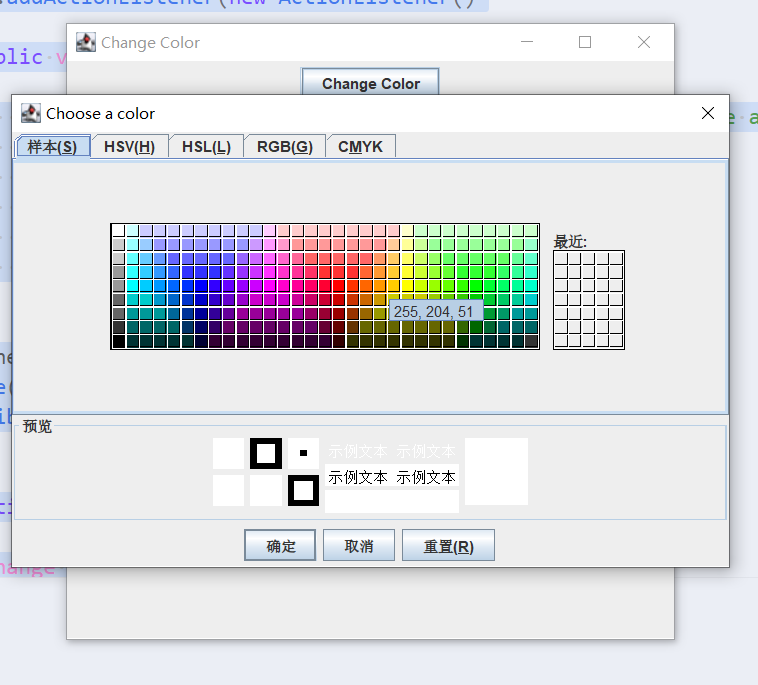
    {

        ColorChange win = new ColorChange();

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

    }

}



## 9.8

package Test8;

import javax.swing.\*;

import java.awt.event.\*;

import java.awt.\*;

public class ChangeColors extends JFrame{

/\*\*

*\**

\*/

    private static final long serialVersionUID = 1L;

   //*private JButton button;*

    private Color color[] = {Color**.**red,Color**.**yellow,Color**.**blue,Color**.**cyan};

    private Color selectedColor = Color**.**red;

    private String  colorString[] = {"Red","Yellow","Blue","Cyan"};

    JCheckBox checkBox1,checkBox2;

    JComboBox<Object> combobox;

    JButton okButton , exitButton;

    private Container container;

    public ChangeColors()

    {

        super("Color选择");

        container = getContentPane();

        container**.**setLayout(new GridLayout(3,1,5,5));

        JPanel jPanel1,jPanel2;

        jPanel1 = new JPanel();

        jPanel2 = new JPanel();

        combobox = new JComboBox<Object>(colorString);

        combobox**.**setMaximumRowCount(3);

        combobox**.**addItemListener(new ItemListener()

        {

            public void itemStateChanged(ItemEvent event)

            {

                 if(event**.**getStateChange()==ItemEvent**.**SELECTED)

                 selectedColor = color[combobox**.**getSelectedIndex()];

            }

        });

        container**.**add(combobox);

        checkBox1 = new JCheckBox("background");

        jPanel1**.**add(checkBox1);

        checkBox2 = new JCheckBox("foreground");

        jPanel1**.**add(checkBox2);

        container**.**add(jPanel1);

        okButton = new JButton("OK");

        okButton**.**addActionListener(new ActionListener()

        {

            public void actionPerformed(ActionEvent event)

            {

                if(checkBox1**.**isSelected())

                combobox**.**setBackground(selectedColor);

                if(checkBox2**.**isSelected())

                combobox**.**setForeground(selectedColor);

            }

        });

        exitButton = new JButton("Exit");

        exitButton**.**addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent event)

            {

                System**.**exit(0);

            }

        });

        jPanel2**.**add(okButton);

        jPanel2**.**add(exitButton);

        container**.**add(jPanel2);

        container**.**setSize(400,150);

        setVisible(true);

    }

    public static void main(String []args)

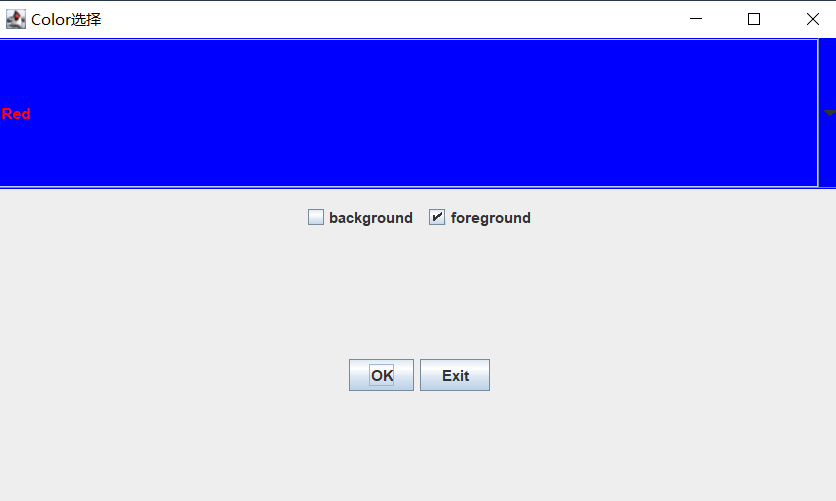
    {

        ChangeColors win = new ChangeColors();

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

    }

}



## 9.9

package Test9;

import javax.swing.\*;

import java.awt.\*;

public class Test9 extends JFrame{

/\*\*

*\**

\*/

    private static final long serialVersionUID = 1L;

    private GridBagLayout layout;

    private GridBagConstraints constraints;

    private Container container;

    public Test9()

    {

        super("Test9");

        container = getContentPane();

        layout = new GridBagLayout();

        container**.**setLayout(layout);

        constraints = new GridBagConstraints();

        String stu[] ={"大学生","中学生","小学生"};

        JComboBox<String> comboBox = new JComboBox<String>(stu);

        JTextField textField =  new JTextField("文本框");

        String fonts[] = {"宋体","楷体"};

        JList<String> list = new JList<String>(fonts);

        String names[] = {"按钮0","按钮1","按钮2","按钮3","按钮4"};

        JButton buttons[] = new JButton[names**.**length];

        for(int count = 0;count < buttons**.**length;count++)

        {

            buttons[count] = new JButton(names[count]);

        }

        constraints**.**weightx = 1;

        constraints**.**weighty = 1;

        constraints**.**fill = GridBagConstraints**.**BOTH;

        constraints**.**gridwidth = GridBagConstraints**.**REMAINDER;

        addComponent(textField);

        constraints**.**gridwidth = 1;

        addComponent(buttons[0]);

        constraints**.**gridwidth = GridBagConstraints**.**RELATIVE;

        addComponent(buttons[1]);

        constraints**.**gridwidth = GridBagConstraints**.**REMAINDER;

        addComponent(buttons[2]);

        constraints**.**weightx = 0;

        constraints**.**gridwidth = GridBagConstraints**.**REMAINDER;

        addComponent(comboBox);

        constraints**.**weighty = 1;

        constraints**.**gridwidth = GridBagConstraints**.**REMAINDER;

        addComponent(buttons[3]);

        constraints**.**gridwidth = GridBagConstraints**.**RELATIVE;

        addComponent(buttons[4]);

        constraints**.**gridwidth = GridBagConstraints**.**REMAINDER;

        addComponent(list);

        setSize(300,300);

        setVisible(true);

    }

    public void addComponent(Component com)

    {

        layout**.**setConstraints(com, constraints);

        container**.**add(com);

    }

    public static void main(String[] args)

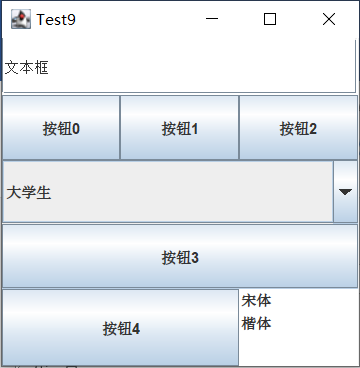
    {

        Test9 win = new Test9();

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

    }

}



## 9.10

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

import java.util.Vector;

class Student {

    private int no;

    private String name;

    private String sex;

    private int age;

    private String hobby;

    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 String getSex() {

        return sex;

    }

    public void setSex(String sex) {

        this**.**sex = sex;

    }

    public int getAge() {

        return age;

    }

    public void setAge(int age) {

        this**.**age = age;

    }

    public String getHobby() {

        return hobby;

    }

    public void setHobby(String hobby) {

        this**.**hobby = hobby;

    }

    public Student(int no,String name,String sex,int age,String hobby){

        this**.**no=no;

        this**.**name=name;

        this**.**sex=sex;

        this**.**age=age;

        this**.**hobby=hobby;

    }

    public  Student(){

    }

    public String toString(){

        return "学号："+no+"   姓名"+name+"  性别："+sex+"  年龄："+age+"  爱好:"+hobby+"\n";

    }

}

public class Main extends JFrame{

/\*\*

*\**

\*/

    private static final long serialVersionUID = 1L;

    JMenuBar menuBar;

    Container container;

    CardLayout cardLayout;

    JTextArea jTextArea;

    JPanel mainPanel;

    JPanel detailPanel;

    JPanel addPanel;

    JPanel modifyPanel;

    JTextField modifInput;

    JPanel searchPanel;

    JTextArea searchResault;

    JTextField searchInput;

    JPanel removePanel;

    JTextField removeInput;

    JTextField addNoInput;

    JTextField addNameInput;

    JTextField addAgeInput;

    JTextField addsexInput;

    JTextField addhobbyInput;

    JTextField modifyNoInput;

    JTextField modifyNameInput;

    JTextField modifyAgeInput;

    JTextField modifySexInput;

    JTextField modifyHobbyInput;

    Vector<Student> students;

    Main(){

        students = new Vector<Student>();

        container = getContentPane();

        mainPanel = new JPanel();

        cardLayout = new CardLayout();//*设置卡片布局*

        mainPanel**.**setLayout(cardLayout);

        container**.**setLayout(new FlowLayout());

        setMenu();//*设置菜单*

        setDetailPanel();//*设置详情页*

        setaddPanel();//*设置添加对象页*

        setRemovePanel();//*设置删除对象页*

        setModifyPanel();//*设置修改对象页*

        setSearchPanel();//*设置查询对象页*

        container**.**add(mainPanel);

    }

    public void setMenu(){

        JMenu menu;

        menuBar = new JMenuBar();

        this**.**setJMenuBar(menuBar);

        menu = new JMenu("编辑");

        menuBar**.**add(menu);

        JMenuItem mainItem = new JMenuItem("显示详情");

        mainItem**.**addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent actionEvent) {

                cardLayout**.**show(mainPanel,"detailPanel");//*切换卡片页面*

                showDetails();

            }

        });

        menu**.**add(mainItem);

        JMenuItem addItem = new JMenuItem("添加");

        addItem**.**addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent actionEvent) {

                cardLayout**.**show(mainPanel,"addPanel");//*切换卡片页面*

                //*清空上次输入*

                addNoInput**.**setText("");

                addAgeInput**.**setText("");

                addNameInput**.**setText("");

                addsexInput**.**setText("");

                addhobbyInput**.**setText("");

            }

        });

        menu**.**add(addItem);

        JMenuItem modifyItem = new JMenuItem("修改");

        modifyItem**.**addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent actionEvent) {

                cardLayout**.**show(mainPanel,"modifyPanel");//*切换卡片页面*

                //*清空上次输入*

                modifInput**.**setText("");

                modifyNoInput**.**setText("");

                modifyAgeInput**.**setText("");

                modifyNameInput**.**setText("");

                modifySexInput**.**setText("");

                modifyHobbyInput**.**setText("");

            }

        });

        menu**.**add(modifyItem);

        JMenuItem removeItem = new JMenuItem("删除");

        removeItem**.**addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent actionEvent) {

                removeInput**.**setText("");

                cardLayout**.**show(mainPanel,"removePanel");//*切换卡片页面*

            }

        });

        menu**.**add(removeItem);

        JMenuItem searchItem = new JMenuItem("查询");

        searchItem**.**addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent actionEvent) {

                cardLayout**.**show(mainPanel,"searchPanel");//*切换卡片页面*

                //*清空上次查询结果*

                searchResault**.**setText("");

                searchInput**.**setText("");

            }

        });

        menu**.**add(searchItem);

    }

    public void setDetailPanel(){

        detailPanel = new JPanel();

        detailPanel**.**setLayout(new FlowLayout());

        jTextArea = new JTextArea(26,45);

        jTextArea**.**setEditable(false);

        jTextArea**.**setBackground(Color**.**black);

        jTextArea**.**setForeground(Color**.**green);

        detailPanel**.**add(jTextArea);

        showDetails();

        mainPanel**.**add("detailPanel",detailPanel);//*添加进卡片*

    }

    public void showDetails(){

        jTextArea**.**setText("");

        jTextArea**.**append("已有学生信息：\n\n");

        Integer index=0;

        for (Student student:students) {

            jTextArea**.**append((++index)**.**toString()+". ");

            jTextArea**.**append(student**.**toString());

        }

        if(jTextArea**.**getLineCount()<=3){//*判断是否有对象*

            jTextArea**.**setText("无任何学生信息");

        }

    }

    public void setaddPanel(){

        addPanel = new JPanel(new FlowLayout());

        JLabel no = new JLabel("学号：");

        addNoInput = new JTextField(15);

        JLabel name = new JLabel("姓名：");

        addNameInput = new JTextField(15);

        JLabel sex = new JLabel("性别：");

        addsexInput = new JTextField(15);

        JLabel age = new JLabel("年龄：");

        addAgeInput = new JTextField(15);

        JLabel hobby = new JLabel("爱好：");

        addhobbyInput = new JTextField(15);

        //*设置布局为Box*

        Box h1 = Box**.**createHorizontalBox();

        h1**.**add(no);

        h1**.**add(addNoInput);

        Box h2 = Box**.**createHorizontalBox();

        h2**.**add(name);

        h2**.**add(addNameInput);

        Box h3 = Box**.**createHorizontalBox();

        h3**.**add(sex);

        h3**.**add(addsexInput);

        Box h4 = Box**.**createHorizontalBox();

        h4**.**add(age);

        h4**.**add(addAgeInput);

        Box h5 = Box**.**createHorizontalBox();

        h5**.**add(hobby);

        h5**.**add(addhobbyInput);

        Box v = Box**.**createVerticalBox();

        v**.**add(Box**.**createVerticalStrut(25));

        v**.**add(h1);

        v**.**add(Box**.**createVerticalStrut(25));

        v**.**add(h2);

        v**.**add(Box**.**createVerticalStrut(25));

        v**.**add(h3);

        v**.**add(Box**.**createVerticalStrut(25));

        v**.**add(h4);

        v**.**add(Box**.**createVerticalStrut(25));

        v**.**add(h5);

        v**.**add(Box**.**createVerticalStrut(25));

        JButton addButton = new JButton("添加");

        Box h6 =Box**.**createHorizontalBox();

        h6**.**add(addButton);

        v**.**add(h6);

        addPanel**.**add(v);

        mainPanel**.**add("addPanel",addPanel);//*添加进卡片*

        addButton**.**addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent actionEvent) {

                int no = Integer**.**parseInt(addNoInput**.**getText());

                String name = addNameInput**.**getText();

                String sex = addsexInput**.**getText();

                int age = Integer**.**parseInt(addAgeInput**.**getText());

                String hobby = addhobbyInput**.**getText();

                students**.**add(new Student(no,name,sex,age,hobby));

                JOptionPane**.**showMessageDialog(container, "添加成功",null,JOptionPane**.**PLAIN\_MESSAGE);

                cardLayout**.**show(mainPanel,"detailPanel");

                showDetails();

            }

        });

    }

    public void setSearchPanel() {

        JLabel jLabel = new JLabel("请输入你要查找的学号：");

        searchInput = new JTextField(15);

        JButton searchButton = new JButton("开始查找");

        searchResault = new JTextArea(20,45);

        searchResault**.**setForeground(Color**.**green);

        searchResault**.**setBackground(Color**.**black);

        Box h1 = Box**.**createHorizontalBox();

        h1**.**add(jLabel);

        h1**.**add(searchInput);

        Box h2 = Box**.**createHorizontalBox();

        h2**.**add(searchButton);

        Box h3 = Box**.**createHorizontalBox();

        h3**.**add(searchResault);

        Box v = Box**.**createVerticalBox();

        v**.**add(Box**.**createVerticalStrut(25));

        v**.**add(h1);

        v**.**add(Box**.**createVerticalStrut(20));

        v**.**add(h2);

        v**.**add(Box**.**createVerticalStrut(10));

        v**.**add(h3);

        searchPanel = new JPanel(new FlowLayout());

        searchPanel**.**add(v);

        mainPanel**.**add("searchPanel",searchPanel);//*添加进卡片*

        searchButton**.**addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent actionEvent) {

                int no = Integer**.**parseInt(searchInput**.**getText());

                searchResault**.**setText("查询到以下结果：\n");

                for (Student student:students) {

                    if(student**.**getNo()==no){

                        searchResault**.**append(student**.**toString());

                    }

                }

                if(searchResault**.**getLineCount()<=2){

                    searchResault**.**append("未找到任何结果，该学号不存在");

                }

            }

        });

    }

    public void setRemovePanel() {

        JLabel jLabel = new JLabel("请输入你要删除的学生学号：");

        removeInput = new JTextField(15);

        JButton removeButton = new JButton("确认删除");

        Box h1 = Box**.**createHorizontalBox();

        h1**.**add(jLabel);

        h1**.**add(removeInput);

        Box h2 = Box**.**createHorizontalBox();

        h2**.**add(removeButton);

        Box v = Box**.**createVerticalBox();

        v**.**add(Box**.**createVerticalStrut(150));

        v**.**add(h1);

        v**.**add(Box**.**createVerticalStrut(20));

        v**.**add(h2);

        removePanel = new JPanel(new FlowLayout());

        removePanel**.**add(v);

        mainPanel**.**add("removePanel",removePanel);//*添加进卡片*

        removeButton**.**addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent actionEvent) {

                int no = Integer**.**parseInt(removeInput**.**getText());

                for (Student student:students) {

                    if(student**.**getNo()==no){

                        students**.**remove(student);

                        break;

                    }

                }

                int res = JOptionPane**.**showConfirmDialog(container, "删除成功，是否继续删除？",null,JOptionPane**.**YES\_NO\_OPTION);

                if(res ==JOptionPane**.**NO\_OPTION) {

                    cardLayout**.**show(mainPanel, "detailPanel");

                    showDetails();

                }

            }

        });

    }

    public void setModifyPanel(){

        JLabel input = new JLabel("请输入你要修改的学生学号：");

        modifInput = new JTextField(15);

        JLabel no = new JLabel("学号修改为：");

        modifyNoInput = new JTextField(15);

        JLabel name = new JLabel("姓名修改为：");

        modifyNameInput = new JTextField(15);

        JLabel sex = new JLabel("性别修改为：");

        modifySexInput = new JTextField(15);

        JLabel age = new JLabel("年龄修改为：");

        modifyAgeInput = new JTextField(15);

        JLabel hobby = new JLabel("爱好修改为：");

        modifyHobbyInput = new JTextField(15);

        Box h0 = Box**.**createHorizontalBox();

        h0**.**add(input);

        h0**.**add(modifInput);

        Box h1 = Box**.**createHorizontalBox();

        h1**.**add(no);

        h1**.**add(modifyNoInput);

        Box h2 = Box**.**createHorizontalBox();

        h2**.**add(name);

        h2**.**add(modifyNameInput);

        Box h3 = Box**.**createHorizontalBox();

        h3**.**add(sex);

        h3**.**add(modifySexInput);

        Box h4 = Box**.**createHorizontalBox();

        h4**.**add(age);

        h4**.**add(modifyAgeInput);

        Box h5 = Box**.**createHorizontalBox();

        h5**.**add(hobby);

        h5**.**add(modifyHobbyInput);

        Box v = Box**.**createVerticalBox();

        v**.**add(Box**.**createVerticalStrut(25));

        v**.**add(h0);

        v**.**add(Box**.**createVerticalStrut(25));

        v**.**add(h1);

        v**.**add(Box**.**createVerticalStrut(25));

        v**.**add(h2);

        v**.**add(Box**.**createVerticalStrut(25));

        v**.**add(h3);

        v**.**add(Box**.**createVerticalStrut(25));

        v**.**add(h4);

        v**.**add(Box**.**createVerticalStrut(25));

        v**.**add(h5);

        v**.**add(Box**.**createVerticalStrut(25));

        JButton modifyButton = new JButton("修改");

        Box h6 =Box**.**createHorizontalBox();

        h6**.**add(modifyButton);

        v**.**add(h6);

        modifyPanel = new JPanel(new FlowLayout());

        modifyPanel**.**add(v);

        mainPanel**.**add("modifyPanel",modifyPanel);//*添加进卡片*

        modifyButton**.**addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent actionEvent) {

                int no = Integer**.**parseInt(modifInput**.**getText());

                boolean flag = false;

                for(Student student:students){

                    if(no == student**.**getNo()){

                        flag = true;

                        int sno = Integer**.**parseInt(modifyNoInput**.**getText());

                        String name = modifyNameInput**.**getText();

                        String sex = modifySexInput**.**getText();

                        int age = Integer**.**parseInt(modifyAgeInput**.**getText());

                        String hobby = modifyHobbyInput**.**getText();

                        student**.**setNo(sno);

                        student**.**setAge(age);

                        student**.**setHobby(hobby);

                        student**.**setName(name);

                        student**.**setSex(sex);

                    }

                }

                if(flag) {

                    int res = JOptionPane**.**showConfirmDialog(container, "修改成功，是否继续修改", null, JOptionPane**.**YES\_NO\_OPTION);

                    if (res == JOptionPane**.**NO\_OPTION) {

                        cardLayout**.**show(mainPanel, "detailPanel");

                        showDetails();

                    }

                }

                else {

                    JOptionPane**.**showMessageDialog(container,"未能找到要修改的学生学号","warning",JOptionPane**.**WARNING\_MESSAGE);

                }

            }

        });

    }

    public static void main(String[] args) {

        Main app =new Main();

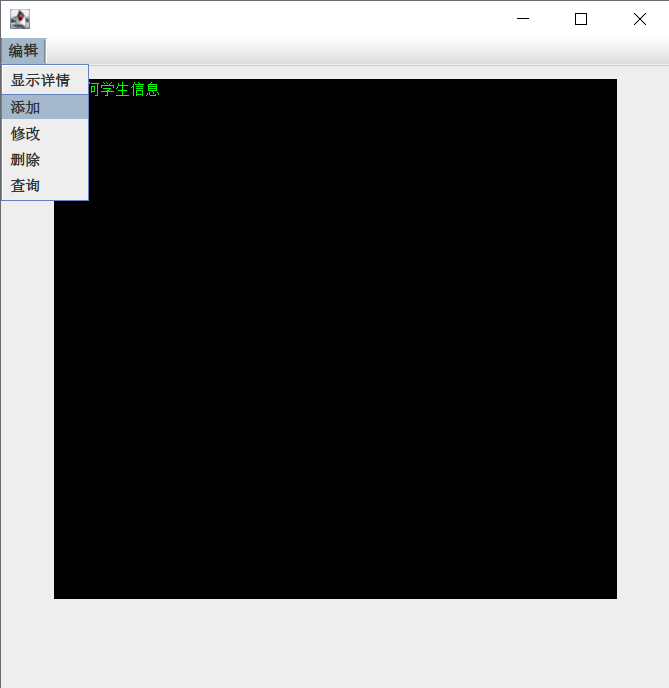
        app**.**setSize(550,560);

        app**.**setVisible(true);

        app**.**setDefaultCloseOperation(EXIT\_ON\_CLOSE);

    }

}



# 第十章

## 10.2

try{

    //监视的程序块,发生异常,由catch程序处理

}

catch(异常类型或标识符)

{

    //捕获异常,并处理

}

...

finally{

    //最终处理

}

## 10.4

public class test {

    public static void main(String args[]) {

        int i = 1, j = 1;

        try{

            i++;

            j--;

            if(i/j>1)

            i++;

        }

        catch(ArithmeticException e)

        {

            System**.**out**.**println(0);

        }

        catch(ArrayIndexOutOfBoundsException e)

        {

            System**.**out**.**println(1);

        }

        catch(Exception e)

        {

            System**.**out**.**println(2);

        }

        finally{

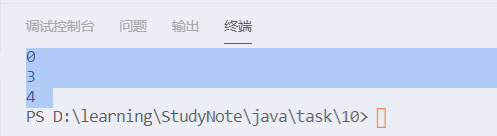
            System**.**out**.**println(3);

        }

        System**.**out**.**println(4);

    }

}



## 10.5

package Test5;

import java.util.Scanner;

public class test {

    public static void main(String[] args)

    {

        try{

            Scanner input = new Scanner(System**.**in);

            String s1 = input**.**next();

            String s2 = input**.**next();

            double a = Double**.**parseDouble(s1);

            double b = Double**.**parseDouble(s2);

            System**.**out**.**println(a+b);

            input**.**close();

        }

        catch(NumberFormatException e)

        {

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

        }

        finally{

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

        }

    }

}



## 10.6

package Test6;

class BalanceNotEnough extends Exception{

/\*\*

*\**

\*/

    private static final long serialVersionUID = 1L;

    public BalanceNotEnough()

    {}

    public String toString()

    {

        return "Balance is not Enough;";

    }

}

class BankAccount {

    private double deposit;

    public BankAccount(double s) {

        deposit = s;

    }

    public void save(double s)

    {

        deposit += s;

    }

    public void get(double s) throws BalanceNotEnough{

        if(s>deposit)

        {

            throw new BalanceNotEnough();

        }

        deposit -= s;

    }

}

public class Test {

    public static void main(String []args)

    {

        try{

            BankAccount a = new BankAccount(123);

            a**.**save(10000);

            a**.**get(22222);

        }

        catch(BalanceNotEnough e)

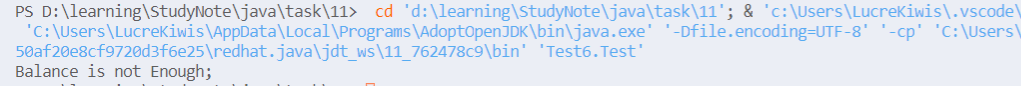
        {

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

        }

    }

}



## 附加

package Extra;

import java.util.Scanner;

class Exception1 extends Exception{

/\*\*

*\**

\*/

    private static final long serialVersionUID = -5439949687883287283L;

    private int value;

    public Exception1(int v)

    {

        value = v;

    }

    public int getValue()

    {

        return value;

    }

}

//*输入二个2位数,做加法;如不是2位数,则抛出异常,下周末与第11章的一起交*

public class Test5plus {

    public static void main(String[] args)

    {

        try{

            int a,b;

            Scanner input = new Scanner(System**.**in);

            a = input**.**nextInt();

            b = input**.**nextInt();

            input**.**close();

            if(a>=100||a<=9){

                throw new Exception1(a);

            }

            if(b>=100||b<=9){

                throw new Exception1(b);

            }

            System**.**out**.**println(a+b);

        }

        catch(Exception1 e)

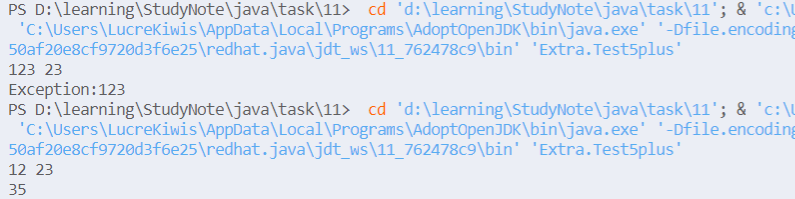
        {

            System**.**out**.**println("Exception:"+e**.**getValue());

        }

    }

}



# 第十一章

## 11.8

package src.Test8;

import java.awt.\*;

import java.awt.event.\*;

public class ThreadTest extends WindowAdapter implements ActionListener {

    Frame f;

    static ThreadTest**.**Thread3 wt1, wt2;

    public static void main(String[] args) {

        ThreadTest win = new ThreadTest();

        win**.**display();

        wt1 = win**.**new Thread3("Welcome");

        wt2 = win**.**new Thread3("How are you?");

        wt2**.**start();

        wt2**.**setButton();

    }

    public void display() {

        f = new Frame("Welcome");

        f**.**setSize(400, 250);

        f**.**setLocation(200, 140);

        f**.**setBackground(Color**.**lightGray);

        f**.**setLayout(new GridLayout());

        f**.**addWindowListener(this);

        f**.**setVisible(true);

    }

    public class Thread3 extends Thread {

        Panel p1;

        Label lb1;

        TextField tf1, tf2;

        Button b1, b2;

        int sleeptime = (int) (Math**.**random() \* 100);

        public Thread3(String s) {

            super(s);

            for (int i = 0; i < 100; i++) {

                s += " ";

            }

            tf1 = new TextField(s);

            f**.**add(tf1);

            p1 = new Panel();

            p1**.**setLayout(new FlowLayout(FlowLayout**.**LEFT));

            lb1 = new Label("Sleep");

            tf2 = new TextField("" + sleeptime);

            p1**.**add(lb1);

            p1**.**add(tf2);

            b1 = new Button("Start");

            b2 = new Button("Stop");

            p1**.**add(b1);

            p1**.**add(b2);

            b1**.**addActionListener(new ThreadTest());

            b2**.**addActionListener(new ThreadTest());

            f**.**add(p1);

            f**.**setVisible(true);

        }

        public void run()

        {

            String str;

            while(true)

            {

                try{

                    str = tf1**.**getText();

                    str = str**.**substring(1)+str**.**substring(0,1);

                    tf1**.**setText(str);

                    Thread3**.**sleep(sleeptime);

                }

                catch(InterruptedException e)

                {

                    break;

                }

            }

        }

        public void setButton() {

            if (this**.**isAlive())

                b1**.**setEnabled(false);

            if (this**.**isInterrupted())

                b2**.**setEnabled(false);

        }

    }

    public void windowClosing(WindowEvent e)

    {

        System**.**exit(0);

    }

    public void actionPerformed(ActionEvent e)

    {

        if((e**.**getSource()==wt1**.**b1)||(e**.**getSource()==wt1**.**b2))

        actionPerformed(e,wt1);

        if((e**.**getSource()==wt2**.**b1)||(e**.**getSource()==wt2**.**b2))

        actionPerformed(e,wt2);

    }

    public void actionPerformed(ActionEvent e,Thread3 w)

    {

        if(e**.**getSource()==w**.**b1)

        {

            w**.**sleeptime = Integer**.**parseInt(w**.**tf2**.**getText());

            w**.**start();

        }

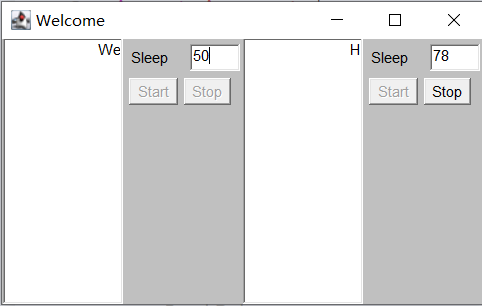
        if(e**.**getSource()==w**.**b2)

        w**.**interrupt();

        w**.**setButton();

    }

}



# 第十二章

## 12.8

package Test8;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.TreeMap;

public class Test {

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

        BufferedReader br = new BufferedReader(new FileReader("file.txt"));

        TreeMap<Character, Integer> tm = new TreeMap<>();

        int ch;

        while ((ch = br**.**read()) != -1) {

            char c = (char)ch;

            /\**if (tm.containsKey(c)) {*

*tm.put(c, 1);*

*}else{*

*tm.put(c, tm.get(c) + 1);*

*}*\*/

            tm**.**put(c, tm**.**containsKey(c) ? tm**.**get(c)+1 : 1);    //*双目运算*

        }

        br**.**close();

        BufferedWriter bw = new BufferedWriter(new FileWriter("numberOfCharacter.txt"));

        for (Character key : tm**.**keySet()) {

            switch (key) {

            case '\t':

                bw**.**write("\\t" + ":" + tm**.**get(key));  //*写出键和值*

                break;

            case '\n':

                bw**.**write("\\n" + ":" + tm**.**get(key));

                break;

            case '\r':

                bw**.**write("\\r" + ":" + tm**.**get(key));

                break;

            case ' ':

                bw**.**write("空格" + ":" + tm**.**get(key));

                break;

            default:

                bw**.**write(key + ":" + tm**.**get(key));

                break;

            }

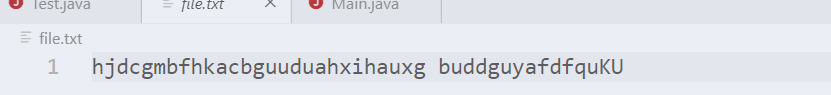
            bw**.**newLine();

        }

        bw**.**close();

    }

}





## 12.9

package Test9;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import java.util.\*;

class Employee implements Serializable {

    private int uid;

    private String name;

    private String address;

    private double salary;

    public Employee() {

    }

    public Employee(int uid, String name, String address, double salary) {

        setUid(uid);

        setName(name);

        setAddress(address);

        setSalary(salary);

    }

    @Override

    public String toString() {

        return "雇员号：" + uid + "     姓名：" + name + "     地址：" + address + "     工资：" + salary;

    }

    public int getUid() {

        return uid;

    }

    public void setUid(int uid) {

        this**.**uid = uid;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this**.**name = name;

    }

    public String getAddress() {

        return address;

    }

    public void setAddress(String address) {

        this**.**address = address;

    }

    public double getSalary() {

        return salary;

    }

    public void setSalary(double salary) {

        this**.**salary = salary;

    }

}

public class Main extends JFrame implements ActionListener {

    private Vector<Employee> employees;

    private JTextArea jTextArea;

    private JButton insertButton;

    private JButton deleteButton;

    private JButton findButton;

    private JButton modifyButton;

    private Container container;

    public Main() {

        employees = new Vector<>();

        jTextArea = new JTextArea(20, 48);

        jTextArea**.**setForeground(Color**.**green);

        jTextArea**.**setBackground(Color**.**black);

        jTextArea**.**setEditable(false);

        container = getContentPane();

        container**.**setLayout(new FlowLayout());

        insertButton = new JButton("插入");

        insertButton**.**addActionListener(this);

        deleteButton = new JButton("删除");

        deleteButton**.**addActionListener(this);

        findButton = new JButton("查找");

        findButton**.**addActionListener(this);

        modifyButton = new JButton("修改");

        modifyButton**.**addActionListener(this);

        addComponent();

        setSize(500, 400);

        read();

        setText();

        setLocationRelativeTo(null);

    }

    //*内部类，实现输入并插入雇员信息*

    class InsertWindow extends JDialog implements ActionListener {

        private JLabel uidLabel;

        private JTextField uidTextFiled;

        private JLabel nameLabel;

        private JTextField nameTextFiled;

        private JLabel addressLabel;

        private JTextField addressTextField;

        private JLabel salaryLabel;

        private JTextField salaryTextFiled;

        private Container container;

        public JButton ensure;

        public InsertWindow(Frame owner, String title, boolean modal) {

            super(owner, title, modal);

            container = getContentPane();

            container**.**setLayout(new FlowLayout());

            uidLabel = new JLabel("雇员号：");

            uidTextFiled = new JTextField(14);

            nameLabel = new JLabel("姓 名：");

            nameTextFiled = new JTextField(15);

            addressLabel = new JLabel("地 址：");

            addressTextField = new JTextField(15);

            salaryLabel = new JLabel("工 资：");

            salaryTextFiled = new JTextField(15);

            ensure = new JButton("确定插入");

            container**.**add(uidLabel);

            container**.**add(uidTextFiled);

            container**.**add(nameLabel);

            container**.**add(nameTextFiled);

            container**.**add(addressLabel);

            container**.**add(addressTextField);

            container**.**add(salaryLabel);

            container**.**add(salaryTextFiled);

            container**.**add(ensure);

            ensure**.**addActionListener(this);

            setSize(250, 180);

            setResizable(false);

            setLocationRelativeTo(owner);//*设置居中出现在父窗口上*

            setVisible(true);

        }

        public Employee getEmployee() {

            Employee employee = new Employee();

            employee**.**setUid(Integer**.**parseInt(uidTextFiled**.**getText()));

            employee**.**setName(nameTextFiled**.**getText());

            employee**.**setAddress(addressTextField**.**getText());

            employee**.**setSalary(Double**.**parseDouble(salaryTextFiled**.**getText()));

            return employee;

        }

        @Override

        public void actionPerformed(ActionEvent actionEvent) {

            if (actionEvent**.**getSource() == ensure) {

                insert(getEmployee());

                setText();

                dispose();

            }

        }

    }

    //*内部类，实现查找雇员信息*

    class FindWindow extends JDialog implements ActionListener{

        private JLabel uidLabel;

        private JTextField uidJTextField;

        private JButton ensure;

        private JTextArea jTextArea;

        Container container;

        public FindWindow(Frame owner, String title, boolean modal) {

            super(owner, title, modal);

            uidLabel = new JLabel("请输入要查找的雇员号：");

            uidJTextField = new JTextField(15);

            ensure = new JButton("确认查找");

            jTextArea = new JTextArea(12,36);

            jTextArea**.**setBackground(Color**.**lightGray);

            jTextArea**.**setForeground(Color**.**black);

            container = getContentPane();

            container**.**setLayout(new FlowLayout());

            container**.**add(uidLabel);

            container**.**add(uidJTextField);

            container**.**add(ensure);

            container**.**add(jTextArea);

            ensure**.**addActionListener(this);

            setSize(400, 275);

            setResizable(false);

            setLocationRelativeTo(this);//*设置居中出现在父窗口上*

            setVisible(true);

        }

        @Override

        public void actionPerformed(ActionEvent actionEvent) {

            if(actionEvent**.**getSource()==ensure){

                int uid = Integer**.**parseInt(uidJTextField**.**getText());

                int index = find(uid);

                if(index==-1){

                    jTextArea**.**setText("未找到uid为"+uid+"的雇员");

                }

                else {

                    jTextArea**.**setText("找到以下内容：\n"+employees**.**elementAt(index)**.**toString());

                }

            }

        }

    }

    //*内部类，实现删除雇员信息*

    class DeleteWindow extends JDialog implements ActionListener {

        private JLabel uidLabel;

        private JTextField uidJTextField;

        private JButton ensure;

        Container container;

        public DeleteWindow(Frame owner, String title, boolean modal) {

            super(owner, title, modal);

            uidLabel = new JLabel("请输入要删除的雇员号：");

            uidJTextField = new JTextField(15);

            ensure = new JButton("确认删除");

            container = getContentPane();

            container**.**setLayout(new FlowLayout());

            container**.**add(uidLabel);

            container**.**add(uidJTextField);

            container**.**add(ensure);

            ensure**.**addActionListener(this);

            setSize(200, 120);

            setResizable(false);

            setLocationRelativeTo(owner);//*设置居中出现在父窗口上*

            setVisible(true);

        }

        @Override

        public void actionPerformed(ActionEvent actionEvent) {

            if (actionEvent**.**getSource() == ensure) {

                delete(Integer**.**parseInt(uidJTextField**.**getText()));

                setText();

                dispose();

            }

        }

    }

    //*内部类，实现修改雇员信息*

    class ModifyWindow extends JDialog implements ActionListener{

        private JLabel preuidLabel;

        private JTextField preuidJTextFiled;

        private JLabel signLabel;

        private JLabel uidLabel;

        private JTextField uidTextFiled;

        private JLabel nameLabel;

        private JTextField nameTextFiled;

        private JLabel addressLabel;

        private JTextField addressTextField;

        private JLabel salaryLabel;

        private JTextField salaryTextFiled;

        private Container container;

        public JButton ensure;

        public ModifyWindow(JFrame owner,String title,boolean modal){

            super(owner,title,modal);

            container = getContentPane();

            container**.**setLayout(new FlowLayout());

            preuidLabel = new JLabel("请输入要修改的雇员号");

            preuidJTextFiled = new JTextField(15);

            signLabel = new JLabel("请输入修改后的雇员信息");

            uidLabel = new JLabel("雇员号：");

            uidTextFiled = new JTextField(14);

            nameLabel = new JLabel("姓 名：");

            nameTextFiled = new JTextField(15);

            addressLabel = new JLabel("地 址：");

            addressTextField = new JTextField(15);

            salaryLabel = new JLabel("工 资：");

            salaryTextFiled = new JTextField(15);

            ensure = new JButton("确定修改");

            container**.**add(preuidLabel);

            container**.**add(preuidJTextFiled);

            container**.**add(signLabel);

            Box h1 =Box**.**createHorizontalBox();

            h1**.**add(uidLabel);

            h1**.**add(uidTextFiled);

            container**.**add(h1);

            Box h2 =Box**.**createHorizontalBox();

            h2**.**add(nameLabel);

            h2**.**add(nameTextFiled);

            container**.**add(h2);

            Box h3 =Box**.**createHorizontalBox();

            h3**.**add(addressLabel);

            h3**.**add(addressTextField);

            container**.**add(h3);

            Box h4 =Box**.**createHorizontalBox();

            h4**.**add(salaryLabel);

            h4**.**add(salaryTextFiled);

            container**.**add(h4);

            container**.**add(ensure);

            ensure**.**addActionListener(this);

            setSize(250, 240);

            setResizable(false);

            setLocationRelativeTo(owner);//*设置居中出现在父窗口上*

            setVisible(true);

        }

        @Override

        public void actionPerformed(ActionEvent actionEvent) {

            if(actionEvent**.**getSource()==ensure){

                int index = find(Integer**.**parseInt(preuidJTextFiled**.**getText()));

                if(index==-1){

                    JOptionPane**.**showMessageDialog(container,"你要修改的雇员不存在",null,JOptionPane**.**WARNING\_MESSAGE);

                }

                else {

                    Employee employee = new Employee();

                    employee**.**setUid(Integer**.**parseInt(uidTextFiled**.**getText()));

                    employee**.**setName(nameTextFiled**.**getText());

                    employee**.**setAddress(addressTextField**.**getText());

                    employee**.**setSalary(Double**.**parseDouble(salaryTextFiled**.**getText()));

                    modify(index, employee);

                    setText();

                }

                this**.**dispose();

            }

        }

    }

    @Override

    public void actionPerformed(ActionEvent e) {

        if (e**.**getSource() == insertButton) {

            InsertWindow insertWindow = new InsertWindow(this, "插入", true);

        }

        if (e**.**getSource() == deleteButton) {

            DeleteWindow deleteWindow = new DeleteWindow(this, "删除", true);

        }

        if(e**.**getSource()==findButton){

            FindWindow findWindow = new FindWindow(this, "查找", true);

        }

        if(e**.**getSource()==modifyButton){

            ModifyWindow modifyWindow = new ModifyWindow(this, "修改", true);

        }

    }

    public void addComponent() {

        container**.**add(jTextArea);

        container**.**add(insertButton);

        container**.**add(deleteButton);

        container**.**add(findButton);

        container**.**add(modifyButton);

    }

    public void setText() {

        int sign = 1;

        jTextArea**.**setText("");

        for (Employee each : employees) {

            jTextArea**.**append(sign + ". " + each**.**toString() + "\n");

            sign++;

        }

    }

    //*写入文件*

    public void write() {

        try {

            FileOutputStream file = new FileOutputStream("./employee.dat");

            ObjectOutputStream out = new ObjectOutputStream(file);

            out**.**writeObject(employees);

            out**.**flush();

            out**.**close();

            file**.**close();

            employees**.**clear();

        } catch (Exception e) {

            System**.**err**.**println(e);

        }

    }

    //*读取文件*

    public void read() {

        try {

            FileInputStream file = new FileInputStream("./employee.dat");

            ObjectInputStream in = new ObjectInputStream(file);

            employees = (Vector<Employee>) in**.**readObject();

            in**.**close();

            file**.**close();

        } catch (Exception e) {

            System**.**err**.**println(e);

        }

    }

    //*插入雇员信息*

    public void insert(Employee employee) {

        employees**.**add(employee);

    }

    //*删除雇员信息*

    public boolean delete(int uid) {

        int index = find(uid);

        if (index == - 1) {

            return false;

        } else {

            employees**.**removeElementAt(index);

            return true;

        }

    }

    //*查找雇员信息*

    public int find(int uid) {

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

            if (employees**.**elementAt(i)**.**getUid() == uid) {

                return i;

            }

        }

        return - 1;//*未找到返回-1*

    }

    //*修改雇员信息*

    public boolean modify(int index,Employee employee) {

        if (index == - 1) {

            return false;

        } else {

            //*修改*

            employees**.**set(index,employee);

            return true;

        }

    }

    public static void main(String[] args) {

        Main app = new Main();

        app**.**setVisible(true);

        app**.**addWindowListener(new WindowAdapter() {

            @Override

            public void windowClosing(WindowEvent e) {//*关闭时保存文件*

                app**.**write();

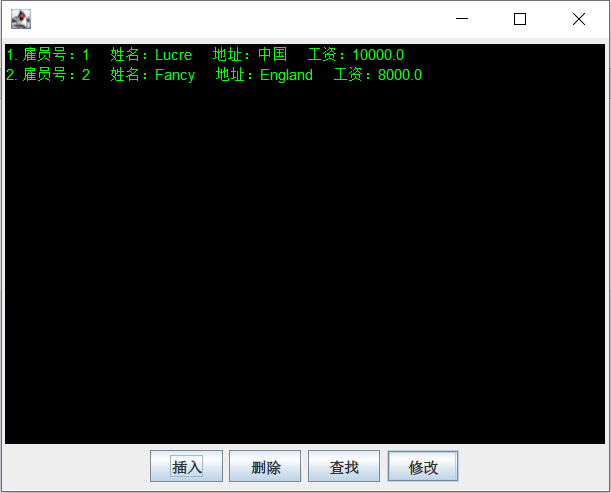
                System**.**exit(0);

            }

        });

    }

}





# 第十三章

### 13-4

import java.io.\*;

import java.net.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Test4 extends JFrame{

    private JTextField enter;

    private JTextArea display;

    private DatagramSocket socket;

    public Test4() {

        super("聊天界面");

        Container container = getContentPane();

        enter = new JTextField(20);

        enter**.**addActionListener(

                new ActionListener() {

                    public void actionPerformed(ActionEvent event)

                    {

                        try {

                            String message = event**.**getActionCommand();

                            byte data[] = message**.**getBytes();

                            DatagramPacket send1 = new DatagramPacket(data,

                                    data**.**length,InetAddress**.**getLocalHost(),6000);

                            displayMessage("\n已发送消息:"+new String(send1**.**getData(),0,send1**.**getLength()));

                            socket**.**send(send1);

                        }

                        catch(IOException ioException){

                            ioException**.**printStackTrace();

                        }

                    }

                }

        );

        JPanel jpanel1 = new JPanel();

        jpanel1**.**add(new JLabel("输入要发送的信息："));

        jpanel1**.**add(enter);

        container**.**add(jpanel1,BorderLayout**.**NORTH);

        display = new JTextArea();

        container**.**add(new JScrollPane(display),BorderLayout**.**CENTER);

        getContentPane()**.**add( new JScrollPane(display),

                BorderLayout**.**CENTER);

        setSize(400,300);

        setVisible(true);

        try {

            socket = new DatagramSocket(5000);

        }

        catch(SocketException socketException) {

            socketException**.**printStackTrace();

            System**.**exit(1);

        }

    }

    private void waitForPackets() {

        while(true) {

            try {

                byte data[] = new byte[100];

                DatagramPacket receive1 = new DatagramPacket(data,data**.**length);

                socket**.**receive(receive1);  //*等待数据包*

                //*将接收的数据包显示*

                displayMessage("\n小红对你说:"+

                        new String(receive1**.**getData(),0,receive1**.**getLength())

                );

            }

            catch(IOException ioException){

                displayMessage(ioException**.**toString() + "\n");

                ioException**.**printStackTrace();

            }

        }

    }

    private void sendPacketToClient(DatagramPacket receive1) throws IOException

    {

        DatagramPacket sendPacket = new DatagramPacket(

                receive1**.**getData(),receive1**.**getLength(),receive1**.**getAddress(),receive1**.**getPort());

        socket**.**send(sendPacket);

    }

    private void displayMessage(final String messageToDisplay)

    {

        SwingUtilities**.**invokeLater(

                new Runnable() {

                    public void run()

                    {

                        display**.**append(messageToDisplay);

                        display**.**setCaretPosition(display**.**getText()**.**length());

                    }

                }

        );

    }

    public static void main(String args[])

    {

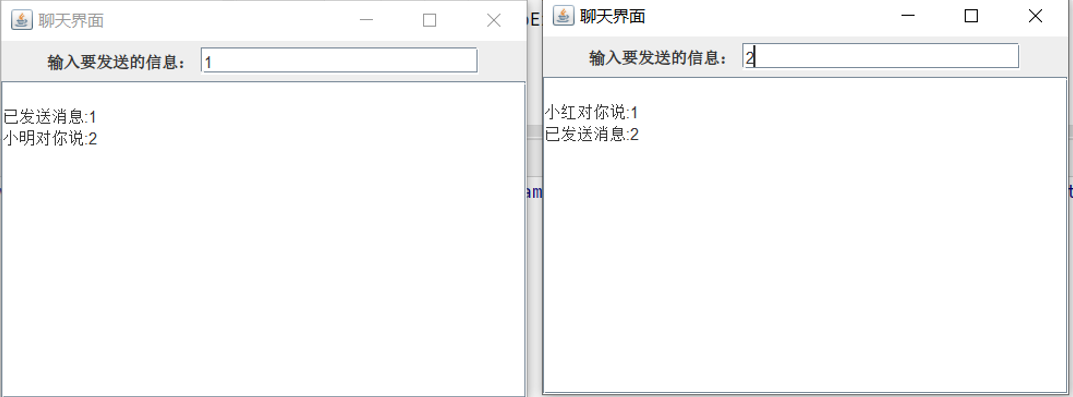
        Test4 app = new Test4();

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

        app**.**waitForPackets();

    }

}



### 13-6

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import java.net.\*;

public class Test6 extends JFrame implements MouseListener {

    private static int BASESIZE = 30;

    private static int tx = 50;

    private static int ty = 60;

    private boolean who = true;

    private boolean enable = true;

    private int[][] board;

    private ServerSocket ss;

    private InputStream input;

    private String host;

    private int port1;

    private int port2;

    public Test6(String title, String host, int port1, int port2) {

        super(title);

        this**.**host = host;

        this**.**port1 = port1;

        this**.**port2 = port2;

        board = new int[15][15];

        //*将棋盘各落子情况初始化为-1*

        for (int i = 0; i < 15; i++) {

            for (int j = 0; j < 15; j++) {

                board[i][j] = - 1;

            }

        }

        try {

            ss = new ServerSocket(port1);

        } catch (IOException e) {

            e**.**printStackTrace();

        }

        //*新建一个连接线程*

        new Thread() {

            @Override

            public void run() {

                try {

                    while (true) {//*循环等待连接*

                        Socket s = ss**.**accept();

                        input = s**.**getInputStream();

                        byte[] data = new byte[1024];

                        int length = input**.**read(data);

                        String string = new String(data, 0, length);

                        String[] point = string**.**split("-");

                        int x = Integer**.**parseInt(point[0]);

                        int y = Integer**.**parseInt(point[1]);

                        //*填充棋盘*

                        board[y][x] = who?1:0;

                        paintChess(x, y, who); //*放置对面棋子*

                        who = ! who; //*将颜色切换回来*

                        enable = true;//*允许己方放置棋子*

                    }

                } catch (IOException e) {

                    e**.**printStackTrace();

                }

            }

        }**.**start();

        setBackground(Color**.**lightGray);

        setSize(520, 520);

        setVisible(true);

        setResizable(false);

        this**.**addMouseListener(this);

    }

    @Override

    public void mouseReleased(MouseEvent mouseEvent) {

        if (! enable)

            return;

        int x = mouseEvent**.**getX();

        int y = mouseEvent**.**getY();

        x = (int) Math**.**round((1.0 \* x - tx) / BASESIZE);

        y = (int) Math**.**round((1.0 \* y - ty) / BASESIZE);

        //*越界判断*

        if (y >= 15 || x >= 15 || y < 0 || x < 0)

            return;

        if (board[y][x] == 1 || board[y][x] == 0)

            return;

        //*发送数据到对面*

        try {

            Socket s = new Socket(host, port2);

            OutputStream output = s**.**getOutputStream();

            output**.**write((x + "-" + y)**.**getBytes());

        } catch (IOException e) {

            e**.**printStackTrace();

        }

        //*填充棋盘*

        board[y][x] = who ? 1 : 0;

        //*绘制落子情况*

        paintChess(x, y, who);

        who = ! who;//*切换角色*

        enable = false;//*下完一个棋子后不再允许下，直到对面下完*

    }

    @Override

    public void paint(Graphics g) {

        Graphics2D g2d = (Graphics2D) g;

        g2d**.**setRenderingHint(RenderingHints**.**KEY\_ANTIALIASING, RenderingHints**.**VALUE\_ANTIALIAS\_ON);

        paintMap(g2d);

    }

    //*画棋谱*

    public void paintMap(Graphics g) {

        g**.**translate(tx, ty);//*转换坐标系*

        g**.**setColor(Color**.**blue);

        for (int i = 0; i < 15; i++) {

            g**.**drawLine(0, i \* BASESIZE, 14 \* BASESIZE, i \* BASESIZE);

            g**.**drawLine(i \* BASESIZE, 0, i \* BASESIZE, 14 \* BASESIZE);

        }

    }

    public void paintChess(int x, int y, boolean who) {

        Graphics g = getGraphics();

        Graphics2D g2d = (Graphics2D) g;

        g2d**.**setRenderingHint(RenderingHints**.**KEY\_ANTIALIASING, RenderingHints**.**VALUE\_ANTIALIAS\_ON);

        g2d**.**setColor(who ? Color**.**black : Color**.**white);

        g2d**.**fillOval(x \* BASESIZE + tx - BASESIZE / 2, y \* BASESIZE + ty - BASESIZE / 2, BASESIZE, BASESIZE);

        checkWiner();

    }

    public void checkWiner() {//*判断胜方*

        int black\_count = 0;

        int white\_count = 0;

        for (int i = 0; i < 15; i++) {//*横向判断*

            black\_count = 0;

            white\_count = 0;

            for (int j = 0; j < 15; j++) {

                if (board[i][j] == 1) {

                    black\_count++;

                    if (black\_count == 5) {

                        messageBox("黑棋胜利");

                        return;

                    }

                } else {

                    black\_count = 0;

                }

                if (board[i][j] == 0) {

                    white\_count++;

                    if (white\_count == 5) {

                        messageBox("白棋胜利");

                        return;

                    }

                } else {

                    white\_count = 0;

                }

            }

        }

        for (int i = 0; i < 15; i++) {//*竖向判断*

            black\_count = 0;

            white\_count = 0;

            for (int j = 0; j < 15; j++) {

                if (board[j][i] == 1) {

                    black\_count++;

                    if (black\_count == 5) {

                        messageBox("黑棋胜利");

                        return;

                    }

                } else {

                    black\_count = 0;

                }

                if (board[j][i] == 0) {

                    white\_count++;

                    if (white\_count == 5) {

                        messageBox("白棋胜利");

                        return;

                    }

                } else {

                    white\_count = 0;

                }

            }

        }

        for (int i = 0; i <= 10; i++) {//*左向右斜判断*

            for (int j = 0; j <= 10; j++) {

                black\_count = 0;

                white\_count = 0;

                for (int k = 0; k < 5; k++) {

                    if (board[i + k][j + k] == 1) {

                        black\_count++;

                        if (black\_count == 5) {

                            messageBox("黑棋胜利");

                            return;

                        }

                    } else {

                        black\_count = 0;

                    }

                    if (board[i + k][j + k] == 0) {

                        white\_count++;

                        if (white\_count == 5) {

                            messageBox("白棋胜利");

                            return;

                        }

                    } else {

                        white\_count = 0;

                    }

                }

            }

        }

        for (int i = 4; i < 15; i++) {//*右向左斜判断 11->12*

            for (int j = 0; j <= 10; j++) {

                black\_count = 0;

                white\_count = 0;

                for (int k = 0; k < 5; k++) {

                    if (board[i - k][j + k] == 1) {

                        black\_count++;

                        if (black\_count == 5) {

                            messageBox("黑棋胜利");

                            return;

                        }

                    } else {

                        black\_count = 0;

                    }

                    if (board[i - k][j + k] == 0) {

                        white\_count++;

                        if (white\_count == 5) {

                            messageBox("白棋胜利");

                            return;

                        }

                    } else {

                        white\_count = 0;

                    }

                }

            }

        }

    }

    public void messageBox(String message){

        JOptionPane**.**showMessageDialog(this,message,"游戏结束",JOptionPane**.**PLAIN\_MESSAGE);

        System**.**exit(1);

    }

    @Override

   public void mouseEntered(MouseEvent mouseEvent) {

    }

    @Override

    public void mouseExited(MouseEvent mouseEvent) {

    }

   @Override

    public void mouseClicked(MouseEvent mouseEvent) {

    }

   @Override

    public void mousePressed(MouseEvent mouseEvent) {

    }

    public static void main(String[] args) {

        Test6 app1 = new Test6("角色1", "localhost", 8888, 9999);

        app1**.**setDefaultCloseOperation(EXIT\_ON\_CLOSE);

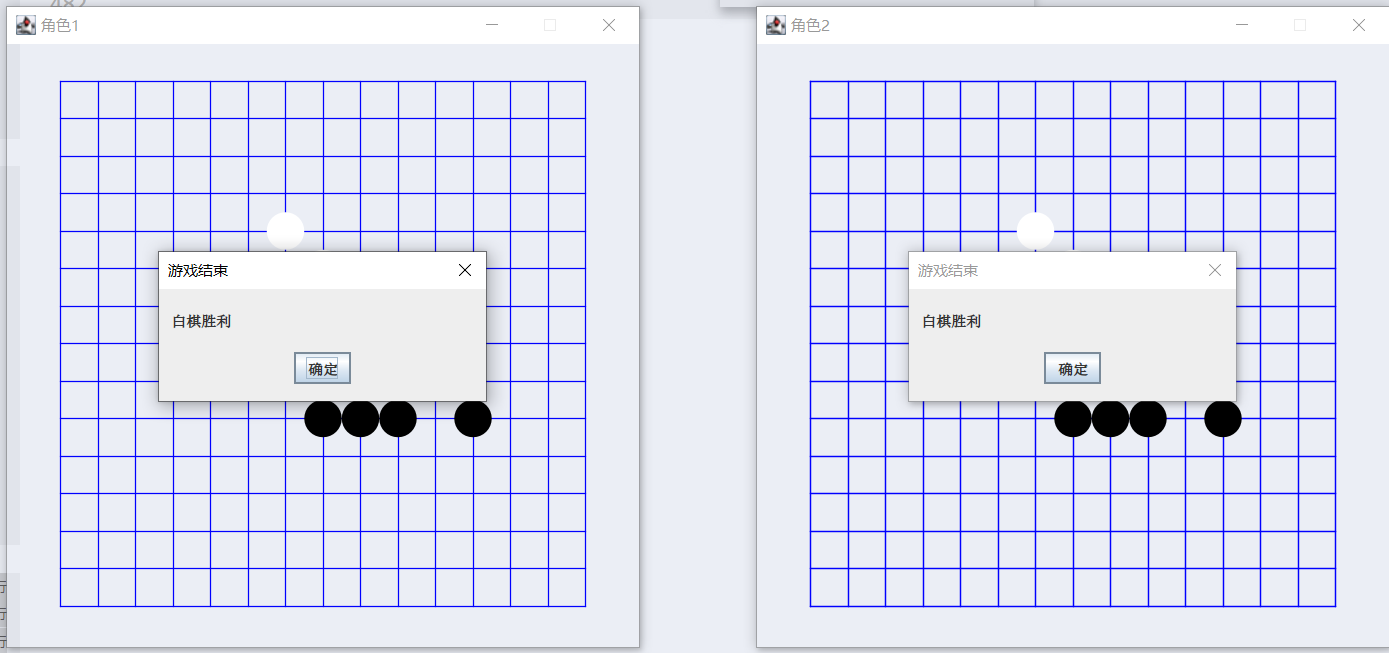
        Test6 app2 = new Test6("角色2", "localhost", 9999, 8888);

        app1**.**setLocation(200,100);

        app2**.**setLocation(800,100);

    }

}



# 第十四章

### 14.4

import java.sql.\*;

import java.util.Scanner;

public class course {

    static PreparedStatement pstmt = null;

    static Connection conn = null; //*保存数据库连接的成员变量*

    static Statement stmt = null;

    static ResultSet rs = null;

    static Statement stmt2 = null;

    public class stedent {

        int xh;

        String xm;

        int cj;

        String dj;

    }

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

        try {

            Class**.**forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

            conn = DriverManager**.**getConnection("jdbc:sqlserver://localhost\\SQLEXPRESS2017:1433;DatabaseName=Study", "sa", "123456");

            System**.**out**.**println("SQL Server Connected Success!");

        } catch (ClassNotFoundException e) {

            System**.**out**.**println("driver failure!"); //*驱动没有成功的加载时抛出的异常进行处理*

        } catch (SQLException e) {

            System**.**out**.**println("connection failure!" + e**.**toString()); //*连接失败时抛出的异常进行处理*

        }

        if (check\_username\_password() == 0) {

            for (; ; ) {

                System**.**out**.**println("Please select one function to execute:");

                System**.**out**.**println("  0--exit.");

                System**.**out**.**println("  2--创建课程表     8--修改课程记录  ");

                System**.**out**.**println("  f--显示课程记录    ");

                System**.**out**.**println("  5--添加课程记录   b--删除课程记录  ");

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

                Scanner s = new Scanner(System**.**in);

                String str = s**.**next();

                if (str**.**equals("0")) System**.**exit(0);

                if (str**.**equals("2")) create\_course\_table();

                if (str**.**equals("5")) insert\_rows\_into\_course\_table();

                if (str**.**equals("8")) current\_of\_update\_for\_course();

                if (str**.**equals("b")) current\_of\_delete\_for\_course();

                if (str**.**equals("f")) using\_cursor\_to\_list\_course();

                pause();

            }

        } else {

            System**.**out**.**println("Your name or password is error,you can not be logined in the system!");

        }

        if (rs != null) {

            rs**.**close();

        }

        if (stmt != null) {

            stmt**.**close();

        }

        if (stmt2 != null) {

            stmt2**.**close();

        }

        if (conn != null) {

            conn**.**close();

        }

    }

    public static int create\_course\_table() throws SQLException

    {

        String yn;

        rs = stmt**.**executeQuery("SELECT name FROM sysobjects WHERE xtype = 'U' and name='course'");

        if(rs**.**next())

        {

            System**.**out**.**println("ERROR:  table course already exists");

            return 0;

        }

        else

        {

            try {

                stmt = conn**.**createStatement(); //*建立Statement类对象*

                //*定义插入记录的SQL语句*

                String r1 = "CREATE TABLE course(cno char(1) NOT null primary key,cname char(10) null ,"

                        + "cpno char(1) null ,ccredit int null)";

                stmt**.**executeUpdate(r1); //*执行SQL命令*

                //*stmt.close(); // 释放Statement所连接的数据库及JDBC资源*

                System**.**out**.**println("Success to create table course!");

            } catch (SQLException e1) { //*数据库读取时产生的异常进行处理*

                System**.**out**.**println("ERROR: Failed to create table course!");

                e1**.**printStackTrace();

                return 0;

            }

            try {

                stmt = conn**.**createStatement(); //*建立Statement类对象*

                stmt**.**executeUpdate("insert into course values('1', 'C语言', '',3)");

                stmt**.**executeUpdate("insert into course values('2', '数据库', '1',4)");

                stmt**.**executeUpdate("insert into course values('3', '编译原理', '2',5)");

                stmt**.**executeUpdate("insert into course values('4', '数据结构', '3',2)");

                stmt**.**executeUpdate("insert into course values('5', '操作系统', '4',3)");

                System**.**out**.**printf("Success to insert rows to course table!\n");

                //*stmt.close(); // 释放Statement所连接的数据库及JDBC资源*

            } catch (SQLException e1) { //*数据库读取时产生的异常进行处理*

                System**.**out**.**println("ERROR: insert rows ");

                e1**.**printStackTrace();

            }

            //*con.close(); // 关闭与数据库的连接*

            return 0;

        }

    }

    private static int check\_username\_password() {

        int num = 0;

        int insertResult;

        String delTabsql = "drop table users";

        String createTabsql = "create table users(uno varchar(6) not null, uname varchar(10) not null, upassword varchar(10) default null, uclass varchar(1) default 'a');";

        String insertSql = "insert into users values('000001','admin','admin', 'Z'),('999999','guest','guest','A');";

        try {

            stmt = conn**.**createStatement();

            stmt2 = conn**.**createStatement();

            stmt**.**executeUpdate(delTabsql);

            stmt**.**executeUpdate(createTabsql);

            System**.**out**.**println("Success to create table users!");

            insertResult = stmt**.**executeUpdate(insertSql);

            if (insertResult == 2) {

                System**.**out**.**println("Success to insert datas!");

            } else {

                System**.**out**.**println("Failed to insert datas!");

            }

        } catch (SQLException e) {

            System**.**out**.**println("Failed to create table users!");

            e**.**printStackTrace();

        }

        while (num < 3) {

            System**.**out**.**println("Please input user name(eg:guest):");

            Scanner name\_scanner = new Scanner(System**.**in);

            String name = name\_scanner**.**next();

            System**.**out**.**println("Please input user password(eg:guest):");

            Scanner pwd\_scanner = new Scanner(System**.**in);

            String pwd = pwd\_scanner**.**next();

            String checkSql = "select count(\*) from users where uname = '" + name + "' and upassword = '" + pwd + "' ";

            ResultSet rs;

            try {

                rs = stmt**.**executeQuery(checkSql);

                if (rs**.**next()) {

                    if (rs**.**getInt(1) == 1) {

                        return 0;

                    }

                }

                num++;

            } catch (SQLException e) {

                e**.**printStackTrace();

            }

        }

        return -1;

    }

    public static int insert\_rows\_into\_course\_table()

    {

        String icname = "xxxxxxxxxx";

        int yn;

        while(true){

            System**.**out**.**println("Please input cno(eg:1,2,3,...):");

            Scanner s1 = new Scanner(System**.**in);

            String icno = s1**.**next();

            System**.**out**.**println("Please input course name(eg:XXXXXXXXXX):");

            Scanner s2 = new Scanner(System**.**in);

            icname = s2**.**next();

            System**.**out**.**println("Please input cpno(eg:1,2,3...):");

            Scanner s3 = new Scanner(System**.**in);

            String icpno = s3**.**next();

            System**.**out**.**println("Please input ccredit(eg:3):");

            Scanner s4 = new Scanner(System**.**in);

            int iccredit = s4**.**nextInt();

            try{//*采用预编译方式定义SQL语句，使添加的数据信息以参数的形式给出*

                //*con = onecon.getConnection();*

                String str = "insert into course(cno,cname,cpno,ccredit) values(?,?,?,?)";

                //*创建PreparedStatement对象，*

                pstmt = conn**.**prepareStatement(str);

                pstmt**.**setString(1, icno); //*给第一个参数设定值*

                pstmt**.**setString(2, icname); //*给第二个参数设定值*

                pstmt**.**setString(3, icpno);

                pstmt**.**setInt(4, iccredit);

                System**.**out**.**println(pstmt**.**toString());

                pstmt**.**executeUpdate();//*执行SQL语句*

                System**.**out**.**println("execute successfully!\n");

                //*con.close();*

            } catch (SQLException e1) {//*执行SQL语句过程中出现的异常进行处理*

                System**.**out**.**println("ERROR: execute \n" );

                e1**.**printStackTrace();

            }

            System**.**out**.**println("Insert again?(1--yes,0--no):");

            Scanner y = new Scanner(System**.**in);

            yn = y**.**nextInt();

            if (yn==1){

                continue;

            }

            else break;

        }

        return 0;

    }

    public static int current\_of\_update\_for\_course() throws SQLException

    {

        System**.**out**.**println("Please input cno to be updated(1、2...,\*\*--All):\n");

        Scanner s= new Scanner(System**.**in);

        String ccno = s**.**next();

        //*con = onecon.getConnection();*

        if (ccno**.**equals("\*\*"))

            rs = stmt**.**executeQuery("SELECT cno,cname,cpno,ccredit FROM course ");

        else

            rs = stmt**.**executeQuery("SELECT cno,cname,cpno,ccredit FROM course where cno = '" + ccno + "'");

        while(rs**.**next())

        {

            System**.**out**.**println( "cno     cname    cpno  ccredit");

            String ss=rs**.**getString(1);

            System**.**out**.**println(ss+"\t"+rs**.**getString(2)+"\t"+"  "+ rs**.**getString(3)+"\t"+rs**.**getString(4));

            System**.**out**.**println("UPDATE ?(y/n/0,y--yes,n--no,0--exit)");

            Scanner y= new Scanner(System**.**in);

            String yn = y**.**next();

            if (yn**.**equals("y") || yn**.**equals("Y"))

            {

                System**.**out**.**println("Please input new cname(eg:数据库):");

                Scanner s1= new Scanner(System**.**in);

                String icname = s1**.**next();

                System**.**out**.**println("Please input new cpno(eg:1,2,...):");

                Scanner s2= new Scanner(System**.**in);

                String icpno = s2**.**next();

                System**.**out**.**println("Please input new ccredit(eg:3):");

                Scanner s3= new Scanner(System**.**in);

                int iccredit = s3**.**nextInt();

                String str = "update course set cname='" + icname + "',cpno='" + icpno+ "',ccredit='" + iccredit + "' where cno = '" + ss + "'";

                stmt2**.**executeUpdate(str);//*执行SQL语句*

            };

            if (yn**.**equals("0")) break;

        };

        //*con.close();*

        return 0;

    }

    static int current\_of\_delete\_for\_course() throws SQLException

    {

        System**.**out**.**printf("Please input cno to be deleted(1、2...,\*\*--All):");

        Scanner s= new Scanner(System**.**in);

        String ccno = s**.**next();

        //*con = onecon.getConnection();*

        pstmt = conn**.**prepareStatement("delete from course  where cno= '" + ccno + "'");

        pstmt**.**executeUpdate();//*执行删除*

        /\*

*if (ccno.equals("\*\*"))*

*rs = stmt.executeQuery("SELECT cno,cname,cpno,ccredit FROM course ");*

*else*

*rs = stmt.executeQuery("SELECT cno,cname,cpno,ccredit FROM course cno = '" + ccno + "'");*

*while( rs.next())*

*{*

*System.out.println( "cno     cname    cpno  ccredit");*

*String ss=rs.getString(1);*

*System.out.println(ss+"\t"+rs.getString(2)+"\t"+"  "+ rs.getString(3)+"\t"+ rs.getInt(4));*

*System.out.println("delete?(y/n/0,y--yes,n--no,0--exit)");*

*Scanner y= new Scanner(System.in);*

*String yn = y.next();*

*if (yn.equals("y") || yn.equals("Y"))*

*{*

*pstmt = conn.prepareStatement("delete from course  where cno= '" + ccno + "'");*

*pstmt.executeUpdate();// 执行删除*

*//pstmt.close(); // 释放Statement所连接的数据库及JDBC资源*

*};*

*if (yn.equals("0")) break;*

*};*

\*/

        //*con.close(); // 关闭与数据库的连接*

        return 0;

    }

    public static int using\_cursor\_to\_list\_course() throws SQLException

    {

        //*con = onecon.getConnection();*

        rs = stmt**.**executeQuery("select \* from course order by cno");

        System**.**out**.**printf("cno     \tcname    \tcpno  \tccredit  \n");

      /\*

*if(!rs.next())*

*System.out.println("ERROR: open");*

*else*

*{*

*rs.previous();*

*System.out.printf("Open successfully!\n");*

*System.out.printf("cno     \tcname    \tcpno  \tccredit  \n");*

*}*\*/

        while (rs**.**next())

        {

            System**.**out**.**println(rs**.**getString(1)+"\t"+rs**.**getString(2)+"\t"+ "  "+rs**.**getString(3)+"\t"+rs**.**getString(4));

        }

        //*stmt.close();*

        //*con.close();*

        return 0;

    }

    void ErrorHandler() {

        //*display error information from SQLCA*

        System**.**out**.**printf("\nError Handler called:\n");

    }

    static void pause() {

        System**.**out**.**printf("Press any key & enter key to continue!");

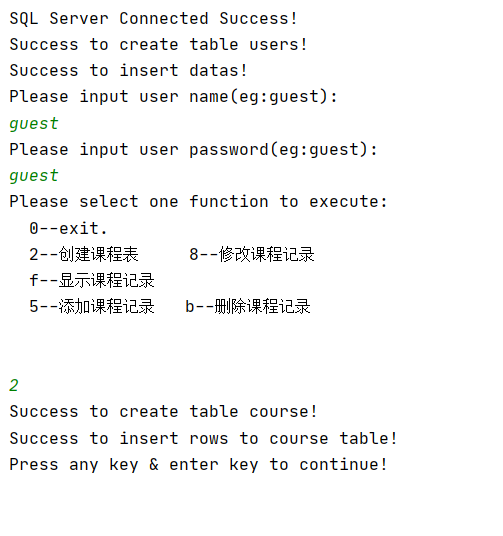
        Scanner input = new Scanner(System**.**in);

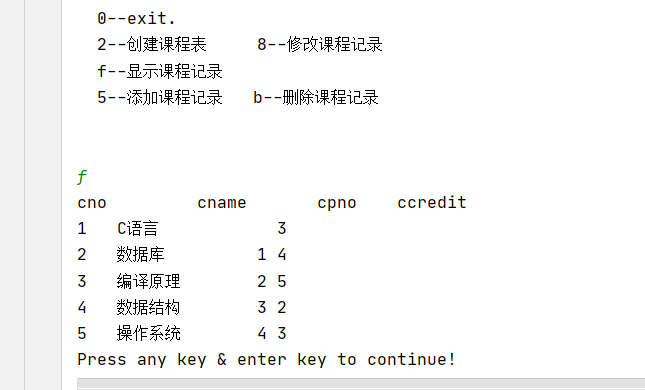
        String str = input**.**nextLine();

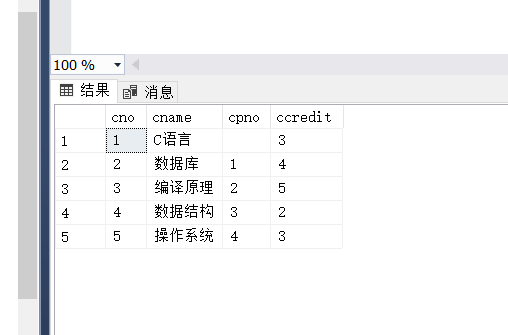
    }

}

1.创建

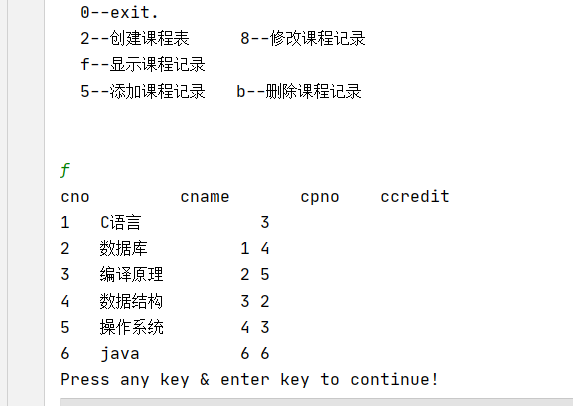




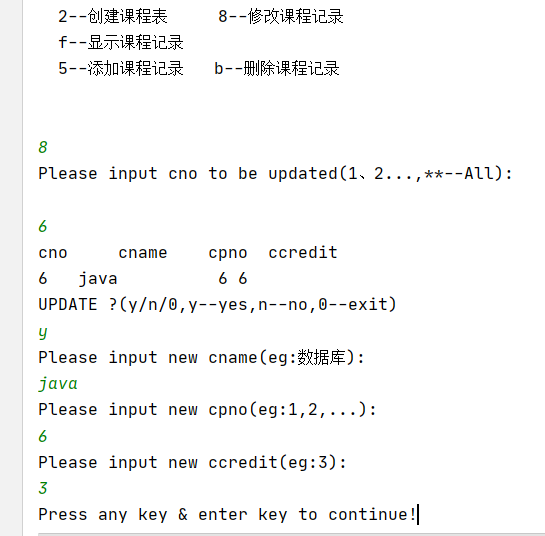


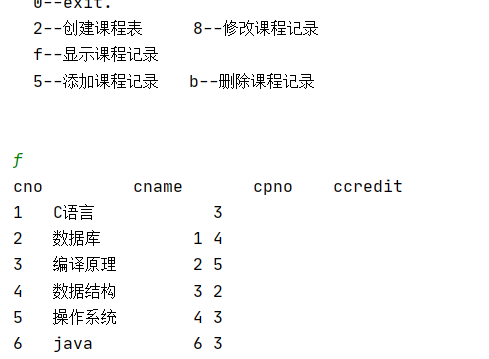
2，添加





3，修改





4，删除



### 14.5

import java.sql.\*;

import java.util.Scanner;

public class course {

    static PreparedStatement pstmt = null;

    static Connection conn = null; //*保存数据库连接的成员变量*

    static Statement stmt = null;

    static ResultSet rs = null;

    static Statement stmt2 = null;

    public class stedent {

        int xh;

        String xm;

        int cj;

        String dj;

    }

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

        try {

            Class**.**forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

            conn = DriverManager**.**getConnection("jdbc:sqlserver://localhost\\SQLEXPRESS2017:1433;DatabaseName=Study", "sa", "123456");

            System**.**out**.**println("SQL Server Connected Success!");

        } catch (ClassNotFoundException e) {

            System**.**out**.**println("driver failure!"); //*驱动没有成功的加载时抛出的异常进行处理*

        } catch (SQLException e) {

            System**.**out**.**println("connection failure!" + e**.**toString()); //*连接失败时抛出的异常进行处理*

        }

        if (check\_username\_password() == 0) {

            for (; ; ) {

                System**.**out**.**println("Please select one function to execute:");

                System**.**out**.**println("  0--exit.");

                System**.**out**.**println("  1--创建学生表      2--创建课程表      8--修改课程记录  ");

                System**.**out**.**println("  3--创建成绩表      f--显示课程记录    j--学生成绩统计表");

                System**.**out**.**println("  5--添加课程记录    b--删除课程记录     k--课程成绩统计表");

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

                Scanner s = new Scanner(System**.**in);

                String str = s**.**next();

                if (str**.**equals("0")) System**.**exit(0);

                if (str**.**equals("1")) create\_student\_table();

                if (str**.**equals("2")) create\_course\_table();

                if (str**.**equals("3")) create\_sc\_table();

                if (str**.**equals("5")) insert\_rows\_into\_course\_table();

                if (str**.**equals("8")) current\_of\_update\_for\_course();

                if (str**.**equals("b")) current\_of\_delete\_for\_course();

                if (str**.**equals("f")) using\_cursor\_to\_list\_course();

                if (str**.**equals("j")) using\_cursor\_to\_total\_s\_sc();

                if (str**.**equals("k")) using\_cursor\_to\_total\_c\_sc();

                pause();

            }

        } else {

            System**.**out**.**println("Your name or password is error,you can not be logined in the system!");

        }

        if (rs != null) {

            rs**.**close();

        }

        if (stmt != null) {

            stmt**.**close();

        }

        if (stmt2 != null) {

            stmt2**.**close();

        }

        if (conn != null) {

            conn**.**close();

        }

    }

    public static int using\_cursor\_to\_total\_c\_sc() throws SQLException

    {

        //*con = onecon.getConnection();*

        rs = stmt**.**executeQuery("select course.cno,cname,count(grade),sum(grade),avg(grade),MIN(grade),MAX(grade) from course,sc where course.cno=sc.cno group by course.cno,cname");

        System**.**out**.**printf("cno     \tcname   \tcount   \tsum     \tavg     \tmin     \tmax \n");

      /\**if(!rs.next())*

*System.out.println("ERROR: open");*

*else*

*{*

*rs.previous();*

*System.out.printf("Open successfully!\n");*

*System.out.printf("cno     \tcname   \tcount   \tsum     \tavg     \tmin     \tmax \n");*

*}*\*/

        while (rs**.**next())

        {

            System**.**out**.**println(rs**.**getString(1)+"\t"+rs**.**getString(2)+"\t"+ rs**.**getString(3)+"\t"+rs**.**getString(4)+"\t"+rs**.**getString(5)+"\t"+rs**.**getString(6)+"\t"+rs**.**getString(7));

        }

        //*stmt.close();*

        //*con.close();*

        return 0;

    }

    public static int using\_cursor\_to\_total\_s\_sc() throws SQLException

    {

        //*con = onecon.getConnection();*

        rs = stmt**.**executeQuery("select student.sno,sname,cno,grade"

                + " from student,sc "

                + "where student.sno=sc.sno group by student.sno,sname,cno,grade order by grade desc");

        System**.**out**.**printf("sno     \tsname   \tcno   \tgrade    \n");

      /\*

*if(!rs.next())*

*System.out.println("ERROR: open");*

*else*

*{*

*rs.previous();*

*System.out.printf("Open successfully!\n");*

*System.out.printf("sno     \tsname   \tcount   \tsum     \tavg     \tmin     \tmax    \n");*

*}*\*/

        while (rs**.**next())

        {

            System**.**out**.**println(rs**.**getString(1)+"\t"+rs**.**getString(2)+"\t"+rs**.**getString(3)+"\t"+rs**.**getString(4)+"\t");

        }

        //*stmt.close();*

        //*con.close();*

        return 0;

    }

    public static int create\_student\_table() throws SQLException

    {

        String yn;

        rs = stmt**.**executeQuery("SELECT name FROM sysobjects WHERE xtype = 'U' and name='student'");

        if(rs**.**next())

        {

            System**.**out**.**println("ERROR:  table student already exists");

            return 0;

        }

        else

        {

            try {

                stmt = conn**.**createStatement(); //*建立Statement类对象*

                //*定义插入记录的SQL语句*

                String r1 = "CREATE TABLE student (sno char(5) NOT null primary key,"

                        + "sname char(6) null ,ssex char(2) null ,sage int null ,sdept char(2) null)";

                stmt**.**executeUpdate(r1); //*执行SQL命令*

                //*stmt.close(); // 释放Statement所连接的数据库及JDBC资源*

                System**.**out**.**println("Success to create table student!");

            } catch (SQLException e1) { //*数据库读取时产生的异常进行处理*

                e1**.**printStackTrace();

                System**.**out**.**println("ERROR: create table student ");

            }

            try {

                stmt = conn**.**createStatement(); //*建立Statement类对象*

                stmt**.**executeUpdate("insert into student(sno,sname,ssex,sage,sdept) values('95001', '李斌', '男',16,'CS')");

                stmt**.**executeUpdate("insert into student(sno,sname,ssex,sage,sdept) values('95002', '赵霞', '女',18, 'IS')");

                stmt**.**executeUpdate("insert into student(sno,sname,ssex,sage,sdept) values('95003', '周淘', '男',17, 'CS')");

                stmt**.**executeUpdate("insert into student(sno,sname,ssex,sage,sdept) values('95004', '钱乐', '女',18, 'IS')");

                stmt**.**executeUpdate("insert into student(sno,sname,ssex,sage,sdept) values('95005', '孙力', '男',16, 'MA')");

                System**.**out**.**printf("Success to insert rows to student table!\n");

                //*stmt.close(); // 释放Statement所连接的数据库及JDBC资源*

            } catch (SQLException e1) { //*数据库读取时产生的异常进行处理*

                e1**.**printStackTrace();

                System**.**out**.**println("ERROR: insert rows");

            }

            //*con.close(); // 关闭与数据库的连接*

            return 0;

        }

    }

    public static int create\_sc\_table() throws SQLException

    {

        String yn;

        rs = stmt**.**executeQuery("SELECT name FROM sysobjects WHERE xtype = 'U' and name='sc'");

        if(rs**.**next())

        {

            System**.**out**.**println("ERROR:  table sc already exists");

            return 0;

        }

        else

        {

            try {

                stmt = conn**.**createStatement(); //*建立Statement类对象*

                //*定义插入记录的SQL语句*

                String r1 = "CREATE TABLE sc(sno char(5) NOT null ,cno char(1) NOT null ,"

                        + "grade int null ,primary key(sno,cno),foreign key(sno) references student(sno),"

                        + "foreign key(cno) references course(cno))";

                stmt**.**executeUpdate(r1); //*执行SQL命令*

                //*stmt.close(); // 释放Statement所连接的数据库及JDBC资源*

                System**.**out**.**println("Success to create table sc!");

            } catch (SQLException e1) { //*数据库读取时产生的异常进行处理*

                System**.**out**.**println("ERROR: create table sc ");

            }

            try {

                stmt = conn**.**createStatement(); //*建立Statement类对象*

                stmt**.**executeUpdate("insert into sc values('95001', '1', 66)");

                stmt**.**executeUpdate("insert into sc values('95001', '2', 66)");

                stmt**.**executeUpdate("insert into sc values('95001', '3', 66)");

                stmt**.**executeUpdate("insert into sc values('95002', '2', 0)");

                stmt**.**executeUpdate("insert into sc values('95002', '3', 97)");

                stmt**.**executeUpdate("insert into sc values('95002', '4', 0)");

                stmt**.**executeUpdate("insert into sc values('95003', '1', 88)");

                stmt**.**executeUpdate("insert into sc values('95003', '2', 68)");

                stmt**.**executeUpdate("insert into sc values('95003', '3', 88)");

                stmt**.**executeUpdate("insert into sc values('95004', '2', 76)");

                stmt**.**executeUpdate("insert into sc values('95004', '3', 76)");

                stmt**.**executeUpdate("insert into sc values('95005', '1', 87)");

                stmt**.**executeUpdate("insert into sc(sno,cno) values('95005', '2')");

                System**.**out**.**printf("Success to insert rows to sc table!\n");

                //*stmt.close(); // 释放Statement所连接的数据库及JDBC资源*

            } catch (SQLException e1) { //*数据库读取时产生的异常进行处理*

                System**.**out**.**println("ERROR: insert rows ");

            }

            //*con.close(); // 关闭与数据库的连接*

            return 0;

        }

    }

    public static int create\_course\_table() throws SQLException

    {

        String yn;

        rs = stmt**.**executeQuery("SELECT name FROM sysobjects WHERE xtype = 'U' and name='course'");

        if(rs**.**next())

        {

            System**.**out**.**println("ERROR:  table course already exists");

            return 0;

        }

        else

        {

            try {

                stmt = conn**.**createStatement(); //*建立Statement类对象*

                //*定义插入记录的SQL语句*

                String r1 = "CREATE TABLE course(cno char(1) NOT null primary key,cname char(10) null ,"

                        + "cpno char(1) null ,ccredit int null)";

                stmt**.**executeUpdate(r1); //*执行SQL命令*

                //*stmt.close(); // 释放Statement所连接的数据库及JDBC资源*

                System**.**out**.**println("Success to create table course!");

            } catch (SQLException e1) { //*数据库读取时产生的异常进行处理*

                System**.**out**.**println("ERROR: Failed to create table course!");

                e1**.**printStackTrace();

                return 0;

            }

            try {

                stmt = conn**.**createStatement(); //*建立Statement类对象*

                stmt**.**executeUpdate("insert into course values('1', 'C语言', '',3)");

                stmt**.**executeUpdate("insert into course values('2', '数据库', '1',4)");

                stmt**.**executeUpdate("insert into course values('3', '编译原理', '2',5)");

                stmt**.**executeUpdate("insert into course values('4', '数据结构', '3',2)");

                stmt**.**executeUpdate("insert into course values('5', '操作系统', '4',3)");

                System**.**out**.**printf("Success to insert rows to course table!\n");

                //*stmt.close(); // 释放Statement所连接的数据库及JDBC资源*

            } catch (SQLException e1) { //*数据库读取时产生的异常进行处理*

                System**.**out**.**println("ERROR: insert rows ");

                e1**.**printStackTrace();

            }

            //*con.close(); // 关闭与数据库的连接*

            return 0;

        }

    }

    private static int check\_username\_password() {

        int num = 0;

        int insertResult;

        String delTabsql = "drop table users";

        String createTabsql = "create table users(uno varchar(6) not null, uname varchar(10) not null, upassword varchar(10) default null, uclass varchar(1) default 'a');";

        String insertSql = "insert into users values('000001','admin','admin', 'Z'),('999999','guest','guest','A');";

        try {

            stmt = conn**.**createStatement();

            stmt2 = conn**.**createStatement();

            stmt**.**executeUpdate(delTabsql);

            stmt**.**executeUpdate(createTabsql);

            System**.**out**.**println("Success to create table users!");

            insertResult = stmt**.**executeUpdate(insertSql);

            if (insertResult == 2) {

                System**.**out**.**println("Success to insert datas!");

            } else {

                System**.**out**.**println("Failed to insert datas!");

            }

        } catch (SQLException e) {

            System**.**out**.**println("Failed to create table users!");

            e**.**printStackTrace();

        }

        while (num < 3) {

            System**.**out**.**println("Please input user name(eg:guest):");

            Scanner name\_scanner = new Scanner(System**.**in);

            String name = name\_scanner**.**next();

            System**.**out**.**println("Please input user password(eg:guest):");

            Scanner pwd\_scanner = new Scanner(System**.**in);

            String pwd = pwd\_scanner**.**next();

            String checkSql = "select count(\*) from users where uname = '" + name + "' and upassword = '" + pwd + "' ";

            ResultSet rs;

            try {

                rs = stmt**.**executeQuery(checkSql);

                if (rs**.**next()) {

                    if (rs**.**getInt(1) == 1) {

                        return 0;

                    }

                }

                num++;

            } catch (SQLException e) {

                e**.**printStackTrace();

            }

        }

        return -1;

    }

    public static int insert\_rows\_into\_course\_table()

    {

        String icname = "xxxxxxxxxx";

        int yn;

        while(true){

            System**.**out**.**println("Please input cno(eg:1,2,3,...):");

            Scanner s1 = new Scanner(System**.**in);

            String icno = s1**.**next();

            System**.**out**.**println("Please input course name(eg:XXXXXXXXXX):");

            Scanner s2 = new Scanner(System**.**in);

            icname = s2**.**next();

            System**.**out**.**println("Please input cpno(eg:1,2,3...):");

            Scanner s3 = new Scanner(System**.**in);

            String icpno = s3**.**next();

            System**.**out**.**println("Please input ccredit(eg:3):");

            Scanner s4 = new Scanner(System**.**in);

            int iccredit = s4**.**nextInt();

            try{//*采用预编译方式定义SQL语句，使添加的数据信息以参数的形式给出*

                //*con = onecon.getConnection();*

                String str = "insert into course(cno,cname,cpno,ccredit) values(?,?,?,?)";

                //*创建PreparedStatement对象，*

                pstmt = conn**.**prepareStatement(str);

                pstmt**.**setString(1, icno); //*给第一个参数设定值*

                pstmt**.**setString(2, icname); //*给第二个参数设定值*

                pstmt**.**setString(3, icpno);

                pstmt**.**setInt(4, iccredit);

                System**.**out**.**println(pstmt**.**toString());

                pstmt**.**executeUpdate();//*执行SQL语句*

                System**.**out**.**println("execute successfully!\n");

                //*con.close();*

            } catch (SQLException e1) {//*执行SQL语句过程中出现的异常进行处理*

                System**.**out**.**println("ERROR: execute \n" );

                e1**.**printStackTrace();

            }

            System**.**out**.**println("Insert again?(1--yes,0--no):");

            Scanner y = new Scanner(System**.**in);

            yn = y**.**nextInt();

            if (yn==1){

                continue;

            }

            else break;

        }

        return 0;

    }

    public static int current\_of\_update\_for\_course() throws SQLException

    {

        System**.**out**.**println("Please input cno to be updated(1、2...,\*\*--All):\n");

        Scanner s= new Scanner(System**.**in);

        String ccno = s**.**next();

        //*con = onecon.getConnection();*

        if (ccno**.**equals("\*\*"))

            rs = stmt**.**executeQuery("SELECT cno,cname,cpno,ccredit FROM course ");

        else

            rs = stmt**.**executeQuery("SELECT cno,cname,cpno,ccredit FROM course where cno = '" + ccno + "'");

        while(rs**.**next())

        {

            System**.**out**.**println( "cno     cname    cpno  ccredit");

            String ss=rs**.**getString(1);

            System**.**out**.**println(ss+"\t"+rs**.**getString(2)+"\t"+"  "+ rs**.**getString(3)+"\t"+rs**.**getString(4));

            System**.**out**.**println("UPDATE ?(y/n/0,y--yes,n--no,0--exit)");

            Scanner y= new Scanner(System**.**in);

            String yn = y**.**next();

            if (yn**.**equals("y") || yn**.**equals("Y"))

            {

                System**.**out**.**println("Please input new cname(eg:数据库):");

                Scanner s1= new Scanner(System**.**in);

                String icname = s1**.**next();

                System**.**out**.**println("Please input new cpno(eg:1,2,...):");

                Scanner s2= new Scanner(System**.**in);

                String icpno = s2**.**next();

                System**.**out**.**println("Please input new ccredit(eg:3):");

                Scanner s3= new Scanner(System**.**in);

                int iccredit = s3**.**nextInt();

                String str = "update course set cname='" + icname + "',cpno='" + icpno+ "',ccredit='" + iccredit + "' where cno = '" + ss + "'";

                stmt2**.**executeUpdate(str);//*执行SQL语句*

            };

            if (yn**.**equals("0")) break;

        };

        //*con.close();*

        return 0;

    }

    static int current\_of\_delete\_for\_course() throws SQLException

    {

        System**.**out**.**printf("Please input cno to be deleted(1、2...,\*\*--All):");

        Scanner s= new Scanner(System**.**in);

        String ccno = s**.**next();

        //*con = onecon.getConnection();*

        pstmt = conn**.**prepareStatement("delete from course  where cno= '" + ccno + "'");

        pstmt**.**executeUpdate();//*执行删除*

        /\*

*if (ccno.equals("\*\*"))*

*rs = stmt.executeQuery("SELECT cno,cname,cpno,ccredit FROM course ");*

*else*

*rs = stmt.executeQuery("SELECT cno,cname,cpno,ccredit FROM course cno = '" + ccno + "'");*

*while( rs.next())*

*{*

*System.out.println( "cno     cname    cpno  ccredit");*

*String ss=rs.getString(1);*

*System.out.println(ss+"\t"+rs.getString(2)+"\t"+"  "+ rs.getString(3)+"\t"+ rs.getInt(4));*

*System.out.println("delete?(y/n/0,y--yes,n--no,0--exit)");*

*Scanner y= new Scanner(System.in);*

*String yn = y.next();*

*if (yn.equals("y") || yn.equals("Y"))*

*{*

*pstmt = conn.prepareStatement("delete from course  where cno= '" + ccno + "'");*

*pstmt.executeUpdate();// 执行删除*

*//pstmt.close(); // 释放Statement所连接的数据库及JDBC资源*

*};*

*if (yn.equals("0")) break;*

*};*

\*/

        //*con.close(); // 关闭与数据库的连接*

        return 0;

    }

    public static int using\_cursor\_to\_list\_course() throws SQLException

    {

        //*con = onecon.getConnection();*

        rs = stmt**.**executeQuery("select \* from course order by cno");

        System**.**out**.**printf("cno     \tcname    \tcpno  \tccredit  \n");

      /\*

*if(!rs.next())*

*System.out.println("ERROR: open");*

*else*

*{*

*rs.previous();*

*System.out.printf("Open successfully!\n");*

*System.out.printf("cno     \tcname    \tcpno  \tccredit  \n");*

*}*\*/

        while (rs**.**next())

        {

            System**.**out**.**println(rs**.**getString(1)+"\t"+rs**.**getString(2)+"\t"+ "  "+rs**.**getString(3)+"\t"+rs**.**getString(4));

        }

        //*stmt.close();*

        //*con.close();*

        return 0;

    }

    void ErrorHandler() {

        //*display error information from SQLCA*

        System**.**out**.**printf("\nError Handler called:\n");

    }

    static void pause() {

        System**.**out**.**printf("Press any key & enter key to continue!");

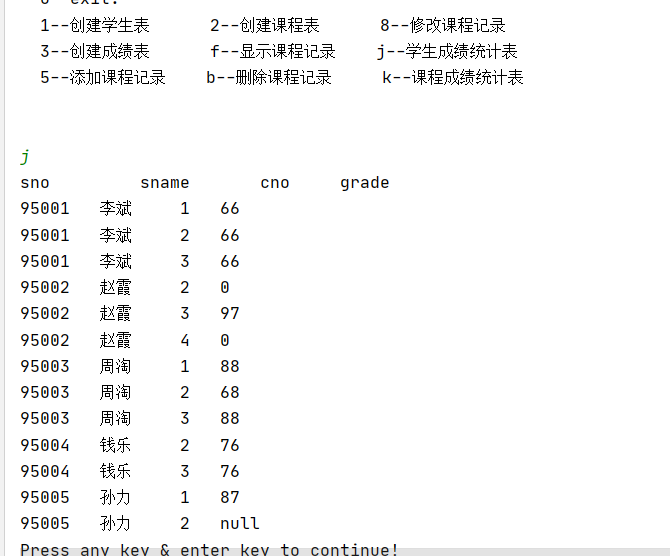
        Scanner input = new Scanner(System**.**in);

        String str = input**.**nextLine();

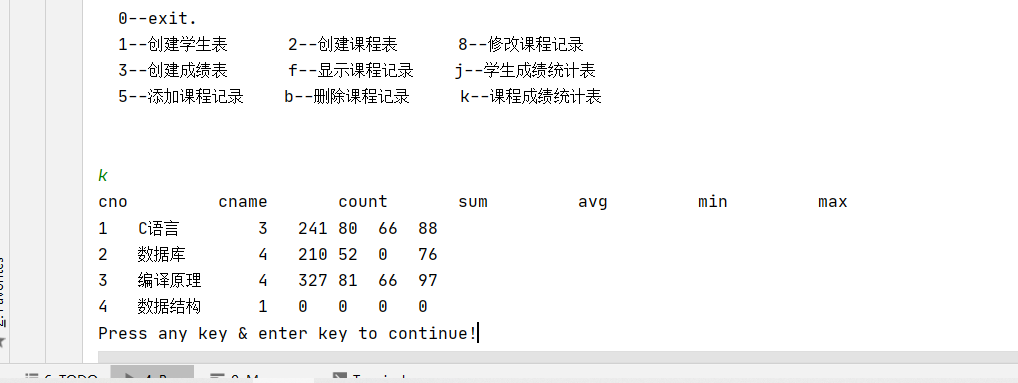
    }

}

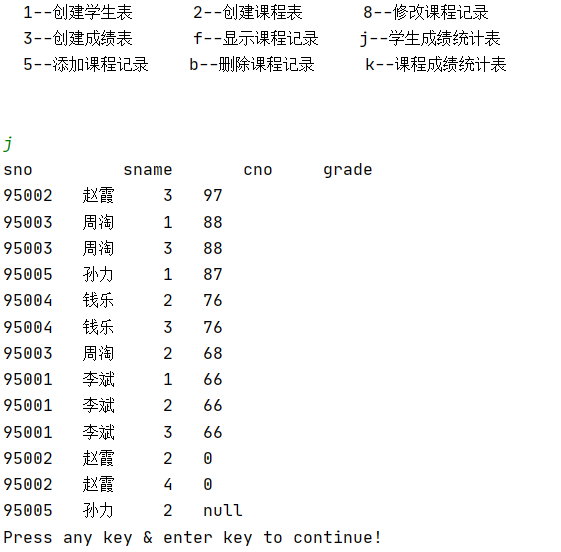
（1）



（2）

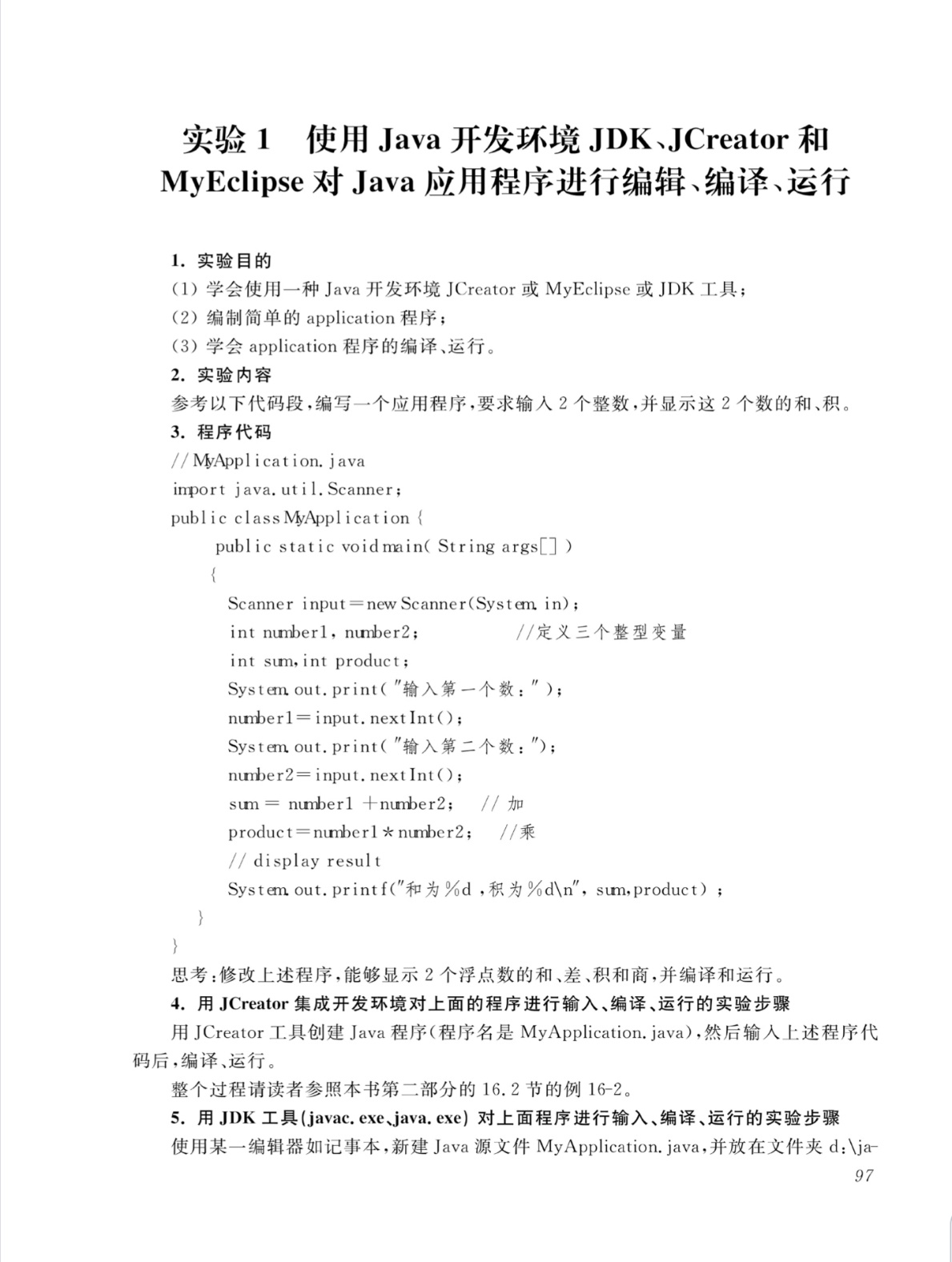


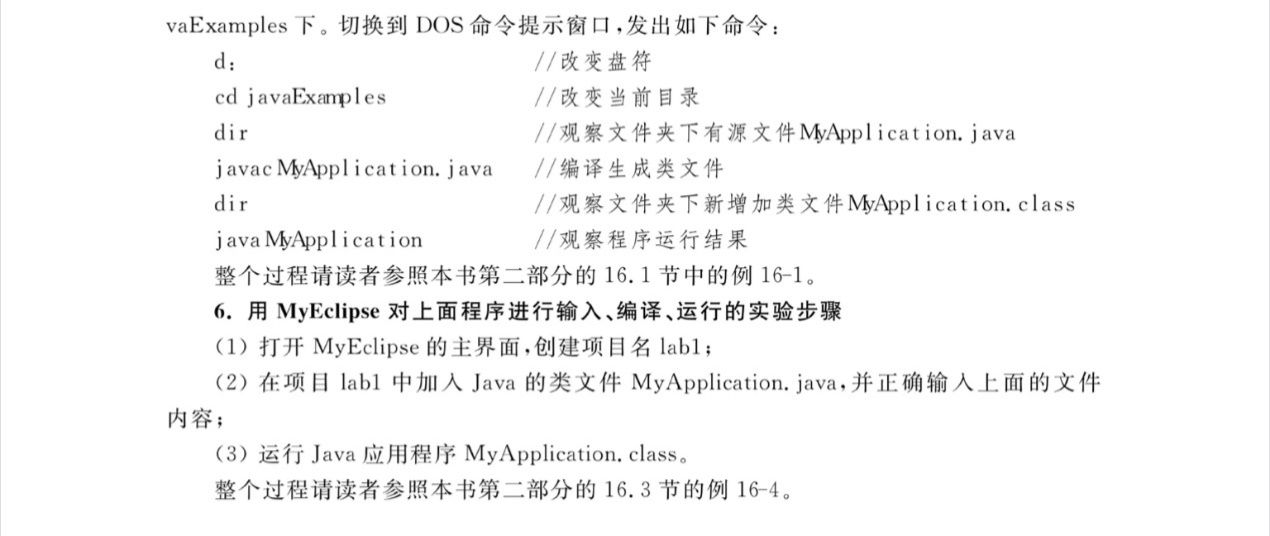
其中 14.5中加  (3) 按降序显示每门课的各个同学的成绩,包括 课程名\课程编号\学号\姓名\成绩



# 实验

## 实验一



可执行代码

**package** labl;

**import** java.util.Scanner;

**public** **class** MyApplication{

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

{

Scanner input=**new** Scanner(System.***in***);

**double** number1,number2;

**double** sum=0;

**double** product=0;

System.***out***.print("输入第一个数:");

number1=input.nextDouble();

System.***out***.print("输入第二个数:");

number2=input.nextDouble();

sum=number1+number2;

product+=number1\*number2;

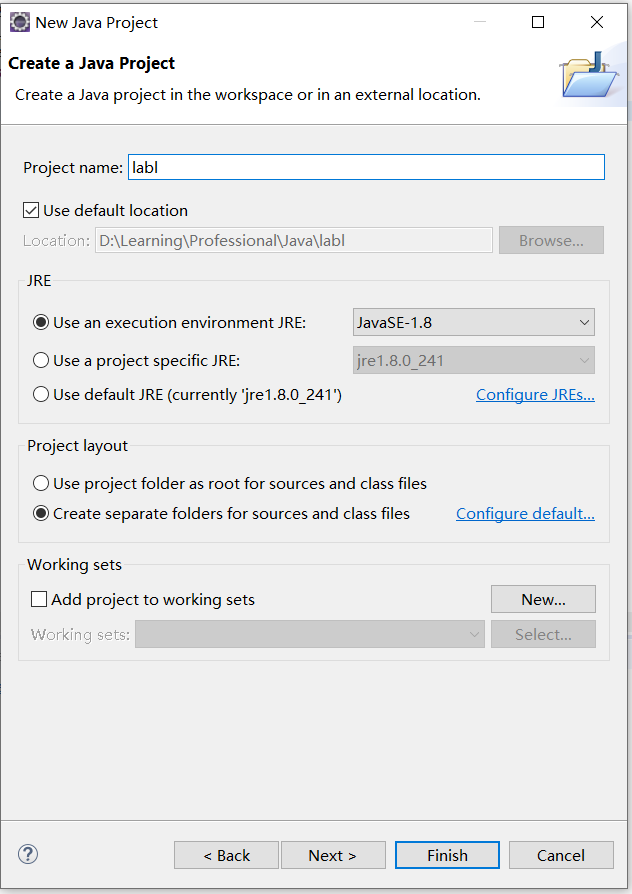
System.***out***.printf("和为%f,积为%f的\n",sum,product);

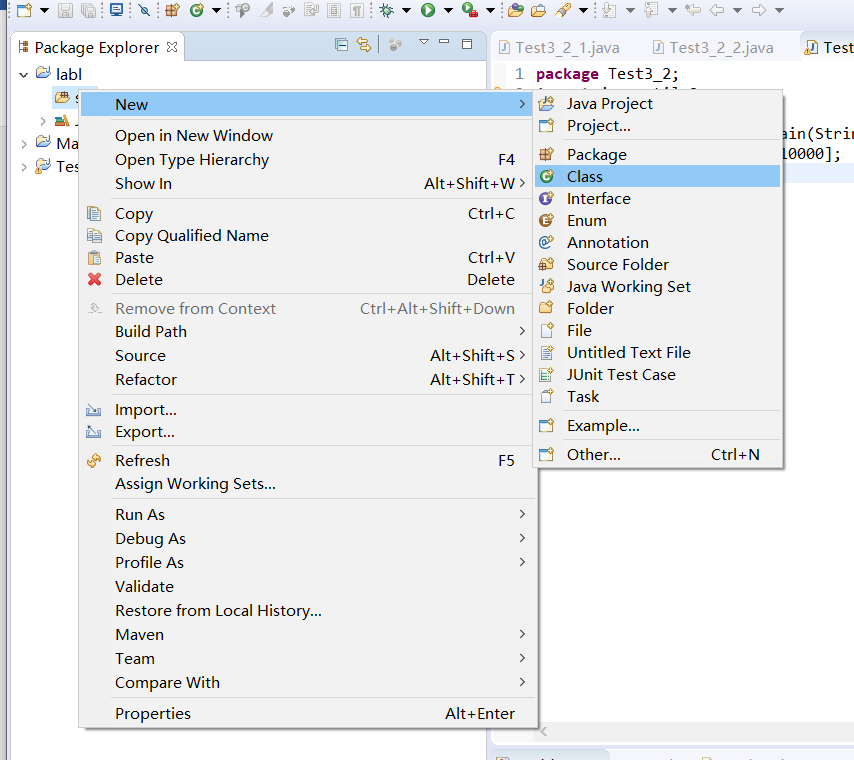
System.***out***.printf("差为%f,商为%f的\n",number1-number2,number1/number2);

}

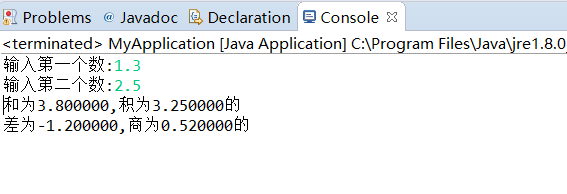
}

### Eclipse上运行:

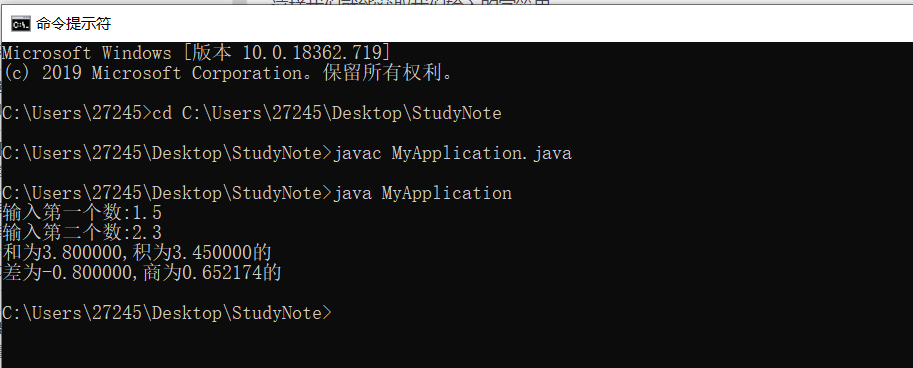




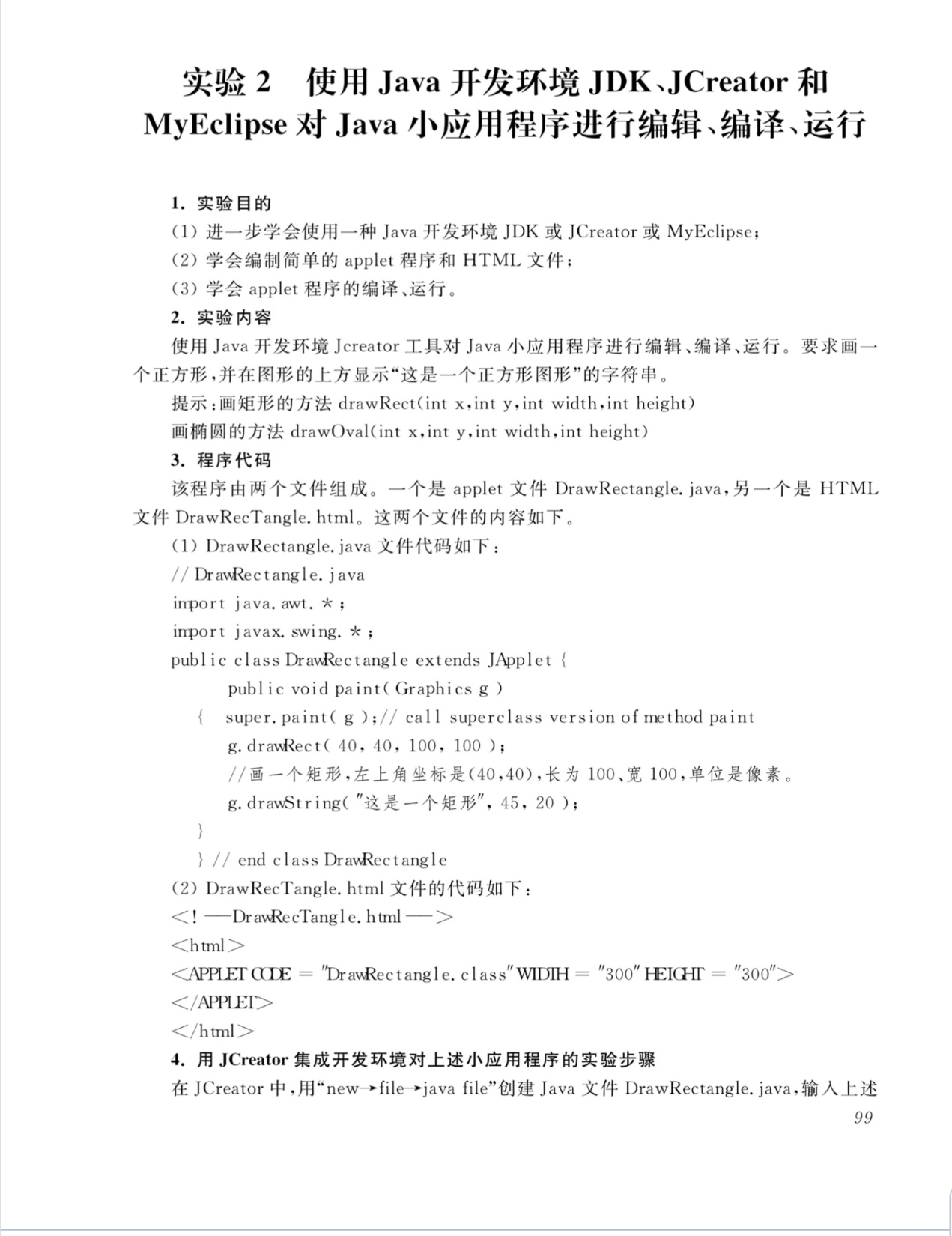


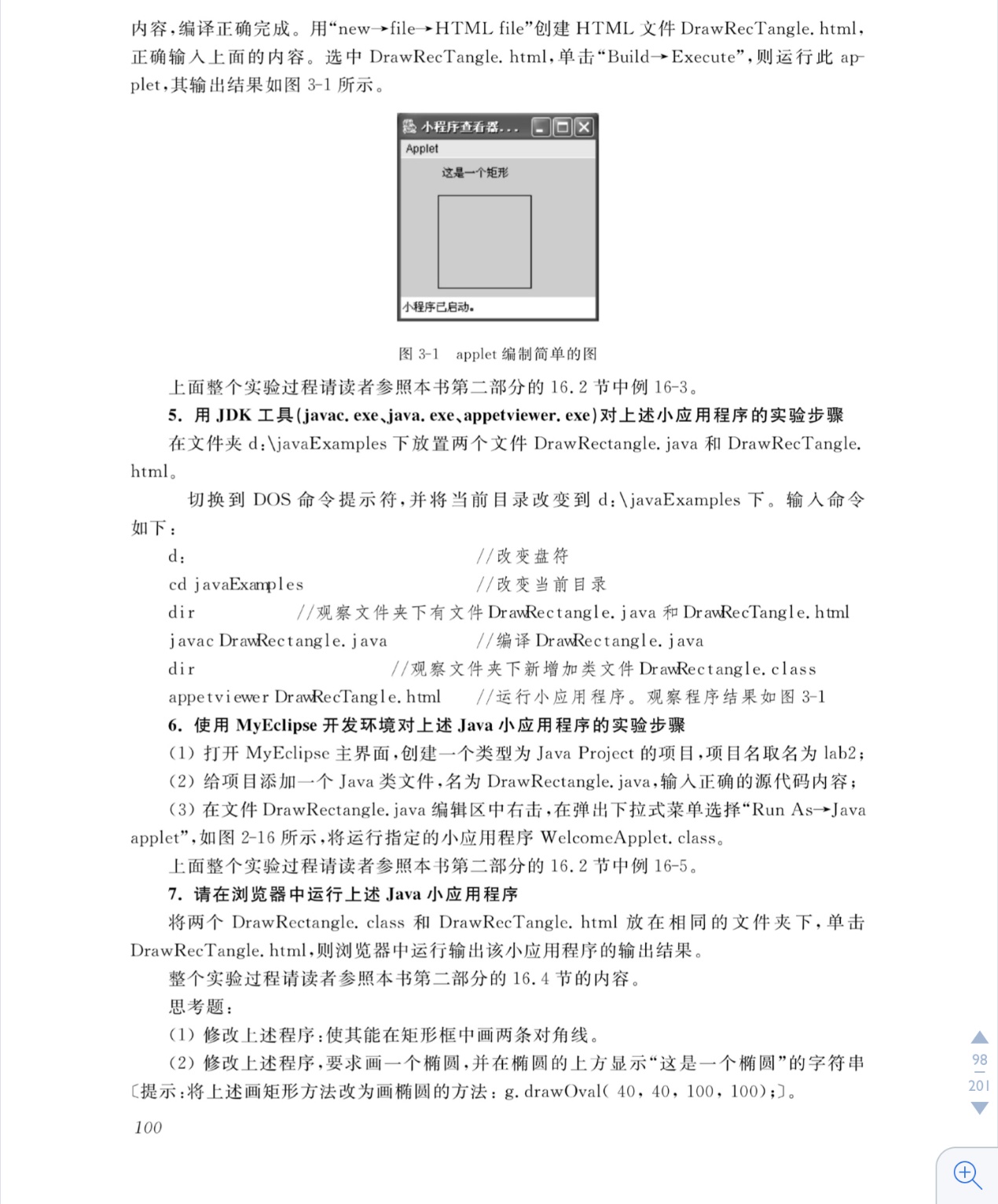


### CMD上运行:

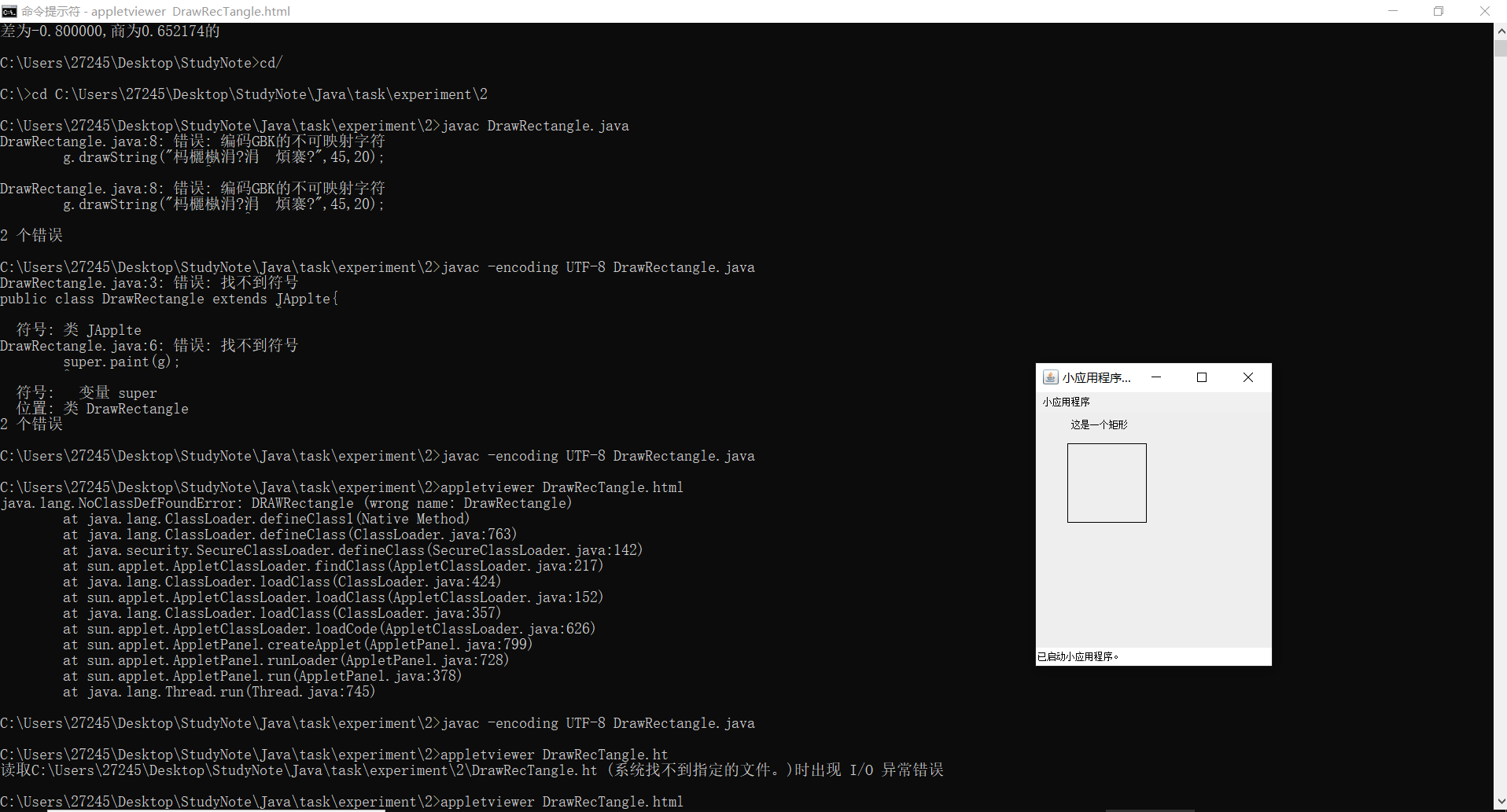


## 实验二





### CMD运行:



### Eclipse运行:

