# 单元一：

## 1.

public class Test101{

public static void main(String[] args){

System.out.println("Hello world!");

}

}

## 2.

public class Test102 {

public static void main(String[] args){

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

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

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

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

}

}

# 单元二：

1.

import java.util.Scanner;

public class Test201 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.println("请输入您需要转换的字母：");

char c = input.next().charAt(0);

//如果输入的是大写，+32即可得到小写

if(c>='A' && c<='Z'){

c+=32;

System.out.println("您输入的大写字母"+(char)(c-32)+"被转换成了"+c);

}else if(c>='a' && c<='z'){ //如果输入的是小写，-32即可得大小写

c-=32;

System.out.println("您输入的小写字母"+(char)(c+32)+"被转换成了"+c);

}else{

System.out.println("输入的字符有误！！");

}

}

}

2.

public class Test202 {

public static void main(String[] args) {

int int\_max=java.lang.Integer.MAX\_VALUE;

System.out.println("int的最大是为："+int\_max);

}

}

3.

public class Test203 {

public static void main(String[] args) {

float float\_max=java.lang.Float.MAX\_VALUE;

float float\_min=java.lang.Float.MIN\_VALUE;

double double\_max=java.lang.Double.MAX\_VALUE;

double double\_min=java.lang.Double.MIN\_VALUE;

System.out.println("单精度浮点数的最大值为："+float\_max);

System.out.println("单精度浮点数的最小值为："+float\_min);

System.out.println("双精度浮点数的最大值为："+double\_max);

System.out.println("双精度浮点数的最小值为："+double\_min);

}

}

5.

import java.util.Scanner;

public class Test205{

public static void main(String[] args) {

Scanner scan =new Scanner(System.in);

System.out.println("输入一个整数:");

int a = scan.nextInt();

if(a%2 != 0)

System.out.printf("%d是奇数", a);

else

System.out.printf("%d是偶数", a);

}

}

6.

public class Test206 {

public static void main(String[] args) {

int n;

int sum=0;

for(int i=1;i<=100;i++){

sum+=i;

}

System.out.println("1至100相加的和为："+sum);

}

}

7.

ublic class Test207 {

public static void main(String[] args) {

int count = 0;

for(int i = 101;i<201;i++) {

boolean flag = true;

for(int k =2;k<i;k++) {

if(i%k == 0) {

flag = false;

break;

}

}

if(flag) {

++count;

System.out.println(i+"是素数");

}

}

System.out.println("101-200间共有"+count+"个素数");

}

}

8.

public class Test208 {

public static void main(String[] args) {

int b, s, g;

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

{

g = i % 10; //g存个位数

s = ( i / 10 ) % 10; //s存十位数

b = i / 100; //b存百位数

//如果它的个十百位数字的3次幂之和等于它本身那么就是水仙花数

if( i == ( (g\*g\*g) + (s\*s\*s) + (b\*b\*b) ) )

{

System.out.println(i + "是水仙花数");

}

}

}

}

9.

import java.util.Scanner;

public class Test209 {

public static void main(String[] args) {

int score;

char dengji;

Scanner sc=new Scanner(System.in);

System.out.println("请输入一个整数成绩:");

score=sc.nextInt();

dengji=score>=90?'A':(score>=60?'B':'C');

System.out.println("考试成绩等级为："+dengji);

}

}

10.

public class Test210 {

public static void main(String[] args) {

System.out.println("12的阶乘值是:" + factorial(12));

}

public static int factorial(int i){

if(i < 1)

return 1;

return factorial(i-1) \* i;

}

}

11.

import java.util.Scanner;

public class Test211 {

public static void main(String[] args) {

System.out.println("请输入一个年份：");

int year;

Scanner scan=new Scanner(System.in);

year=scan.nextInt();

if((year%4==0 && year%100!=0)||(year%400==0))//判断闰年成立的条件

System.out.println(year+"是闰年！");

else

System.out.println(year+"不是闰年！");

}

}

12.

import java.util.Scanner;

public class Test212 {

public static void main(String[] args) {

int a,b;

Scanner scan=new Scanner(System.in);

System.out.println("请输入第一个非零整数：");

a=scan.nextInt();

System.out.println("请输入第二个非零整数：");

b=scan.nextInt();

if(a%b==0)

System.out.println("第一个整数是第二个整数的倍数");

else

System.out.println("第一个整数不是第二个整数的倍数");

}

}

13.

import java.util.Scanner;

public class Test213 {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

System.out.print("请输入需要查询的年份：");

int year=input.nextInt();

System.out.print("请输入需要查询的月份：");

int month=input.nextInt();

switch(month){

case 1:case 3:case 5:case 7:case 8:case 10:case 12:

System.out.println(year+"年"+month+"月有31天。");

break;//每一个case后需要添加break跳出switch语句

case 2:

if(year%4==0&&year%100!=0||year%400==0){

System.out.println(year+"年"+month+"月有29天。");

}else{

System.out.println(year+"年"+month+"月有28天。");

}

break;//此处break如果省略，将继续向下执行，一直到遇到break为止或者是switch语句结束

case 4:case 6:case 9:case 11:

System.out.println(year+"年"+month+"月有30天。");

break;

default:

System.out.println("ERROR！");

//break;//default可以省略、此处的break也可以省略

}

}

}

14.

public class Test214 {

public static void main(String[] args) {

int i=1;

int sum=0;

while(i<=100){

if(i%2!=0){

sum+=i;

}

i++;

}

System.out.println("1至100内所有奇数的和为："+sum);

}

}

15.

public class Test215 {

public static void main(String[] args) {

int a=0,sum=0;

for(int n=1;n<=5;n++)

{

a=(a\*10)+2;

sum+=a;

}

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

}

}

16.

import java.util.Scanner;

public class Test216{

public static void main(String[] args) {

System.out.println("请输入任意自然数以验证“角谷猜想”：");

int n;

Scanner scan=new Scanner(System.in);

n=scan.nextInt();//将键盘输入的自然数赋值给n

int count=0;//保存计算次数

while(n!=1){

{

if(n%2==0)

n=n/2;

else

n=n\*3+1;

}

count++;

}

System.out.println("该自然数经过"+count+"次计算，最终得到1！");

}

}

17.

public class Test217 {

public static void main(String[] args) {

for(int n=0;n<500;n++){

if(n%3==0&&n%6==0){

System.out.println(n);

}

}

}

}

18. public class Test218{

public static void main(String[] args) {

int n = 6;

printStar(n);

}

//打印星星

private static void printStar(int n){

//打印上半部分

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

for(int j=0;j<2\*n;j++){

if(j<n-i)

System.out.print(" ");

if(j>=n-i && j<=n+i)

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

}

System.out.println();

}

//打印下半部分

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

System.out.print(" ");

for(int j=0;j<2\*n-i;j++){

if(j<i)

System.out.print(" ");

if(j>=i && j<2\*n-i-1)

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

}

System.out.println();

}

}

}

19.

import java.util.\*;

public class Test219 {

public static void main(String[] args) {

int digital = 0;

int character = 0;

int other = 0;

int blank = 0;

char[] ch=null;

System.out.println("请输入一行字符，比如123 ABC！@#：");

Scanner sc = new Scanner(System.in);

String s = sc.nextLine();

ch = s.toCharArray();

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

if(ch[i]>='0'&&ch[i]<='9') {

digital ++;

} else if((ch[i]>='a'&&ch[i]<='z')||ch[i]>='A'&&ch[i]<='Z') {

character ++;

} else if(ch[i]==' ') {

blank ++;

} else {

other++;

}

}

System.out.println("数字个数: " + digital);

System.out.println("英文字母个数: " + character);

System.out.println("空格个数: " + blank);

System.out.println("其他字符个数:" + other );

}

}

20.

public class Test220 {

public static void main(String[] args) {

for (int i=1; i<500; i++) {

int sum=0;

for (int j=1;j<i/2+1;j++) {

if (i%j==0) {

sum+=j;

if (sum==i) {

System.out.print(i+" ");

}

}

}

}

}

}

21

public class Test221 {

public static void main(String[] args) {

int count=0;

for(int i=1;i<=4;i++){

for(int j=1;j<=4;j++){

for(int k=1;k<=4;k++){

//相互都不相等

if(i!=j&&i!=k&&j!=k){

count++;

System.out.println(100\*i+10\*j+k);

}

}

}

}

System.out.println("一共有"+count+"个数！");

}

}

22

import java.util.Scanner;

public class Test222 {

public static void main(String[] args) {

double height=100;//初始高度

int n;//第n次落地

double sum=0;//共经过的米数

System.out.println("请输入落地次数：");

Scanner sc=new Scanner(System.in);

n=sc.nextInt();

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

sum+=height+height/2;

height=height/2;//反弹高度

}

System.out.println("第"+n+"次落地时，共经过"+sum+"米，"+"反弹"+height+"米");

}

}

23.

/\*

分析：第10天剩下1个桃子，其实是第9天它吃过了第8天剩下的一半又吃了一个后剩下的

那么，倒过来算，第9天还没吃之前应该是(1+1)\*2，当天还没吃之前的桃子数是吃过之后剩下的

桃子数加上一再乘以二。

\*/

public class Test223 {

public static void main(String[] args) {

// 定义一个变量来记录总桃子数，第9天晚上剩下1个

int sum = 1;

// 逆向计算，一直算出第1天还没吃之前的桃子数

for (int i = 9; i >= 1; i--) {

sum = (sum + 1) \* 2;

}

System.out.println("该猴子第一天共摘了" + sum + "个桃子。");

}

}

24.

public class Test224 {

public static void main(String[] args) {

int sum = 0;

for (int i = 1; i <= 20; i++) {

sum += factorial(i);

}

System.out.println("1!+2!+3!+...+20!的和为：" + sum);

}

public static int factorial(int i) {

if (i < 1)

return 1;

return factorial(i - 1) \* i;

}

}

25.

public class Test225 {

public static void main(String[] args) {

// 定义一个变量来记录第一个人的年龄

int age = 10;

for (int i = 1; i < 5; i++) {

age += 2;// 每一个人都比前一个人大2岁

}

System.out.println("第五个人的年龄是" + age + "岁。");

}

}

26.

import java.util.GregorianCalendar;

import java.util.Scanner;

public class Test226 {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

System.out.print("输入年份：");

int year = scan.nextInt();

System.out.print("输入月份：");

int month = scan.nextInt();

System.out.print("输入日期：");

int day = scan.nextInt();

//判断是否是闰年

//GregorianCalendar:判断年份是否是闰年的方法

GregorianCalendar gre = new GregorianCalendar();

boolean isLeapYear=gre.isLeapYear(year);//返回true:是闰年，false：不是闰年

int ap=isLeapYear?29:28;//判断2月份的天数

int days=0;

switch (month) {

case 1:

days=day;

break;

case 2:

days=31+day;

break;

case 3:

days=31+ap+day;

break;

case 4:

days=31+ap+31+day;

break;

case 5:

days=31+ap+31+30+day;

break;

case 6:

days=31+ap+31+30+31+day;

break;

case 7:

days=31+ap+31+30+31+30+day;

break;

case 8:

days=31+ap+31+30+31+30+31+day;

break;

case 9:

days=31+ap+31+30+31+30+31+31+day;

break;

case 10:

days=31+ap+31+30+31+30+31+31+30+day;

break;

case 11:

days=31+ap+31+30+31+30+31+31+30+31+day;

break;

case 12:

days=31+ap+31+30+31+30+31+31+30+31+30+day;

break;

default:

System.out.println("月份输入错误");

break;

}

System.out.println("这一天是这一年的第"+days+"天");

}

}

27.

public class Test227 {

public static void main(String[] args) {

int num1, num2, num3;

System.out.println("请输入3个整数，每个数之间按回车键：");

Scanner sc = new Scanner(System.in);

num1 = sc.nextInt();

num2 = sc.nextInt();

num3 = sc.nextInt();

int j;

if (num1 > num2) {

j = num2;

num2 = num1;

num1 = j;

}

if (num1 > num3) {

j = num3;

num3 = num1;

num1 = j;

}

if (num2 > num3) {

j = num3;

num3 = num2;

num2 = j;

}

System.out.println("这三个数按照从小到大的顺序排列为：" + num1 + " " + num2 + " " + num3);

}

}

28.

public class Test228 {

public static void main(String[] args) {

double fenmu = 1, fenzi = 2;

double sum = 0;

for (int i = 1; i <= 20; i++) {

sum += (double) fenzi / fenmu;// 将分子的类型强制转换成double

fenzi = fenmu + fenzi;

fenmu = fenzi - fenmu;

}

System.out.println(sum);

}

}

29.

import java.util.Scanner;

public class Test229 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("请输入星期的首字母（不区分大小写）：");

String str1 = scanner.nextLine().toUpperCase();

switch (str1) {

case "M":

System.out.println("Monday");

break;

case "T":

System.out.println("请输入星期的次字母（不区分大小写）：");

String str2 = scanner.nextLine().toUpperCase();

if (str2.equals("U")) {

System.out.println("Tuesday");

} else if (str2.equals("H")) {

System.out.println("Thursday");

} else {

System.out.println("error！");

}

break;

case "W":

System.out.println("Wednesday");

break;

case "F":

System.out.println("Friday");

break;

case "S":

System.out.println("请输入星期的次字母（不区分大小写）：");

String str3 = scanner.nextLine().toUpperCase();

if (str3.equals("U")) {

System.out.println("Sunday");

} else if (str3.equals("A")) {

System.out.println("Saturday");

} else {

System.out.println("error！");

}

break;

default:

System.out.println("error！");

}

}

}

30.

import java.util.Scanner;

public class Test230 {

public static void main(String[] args) {

int n;

System.out.println("请输入一个正整数：");

Scanner scan=new Scanner(System.in);

n=scan.nextInt();

if(n%2==0){

System.out.println(getOushu(n));

}else{

System.out.println(getJishu(n));

}

}

// 偶数递归

public static double getOushu(int n){

if(n <= 2){

return (double)1/2;

}

return getOushu(n - 2) + (double)1/n;

}

// 奇数递归

public static double getJishu(int n){

if(n <= 1){

return 1;

}

return getJishu(n - 2) + (double)1/n;

}

}

31.

import java.util.Scanner;

public class Test231 {

public static void main(String[] args) {

System.out.println("请输入2个整数，每个数之间按回车键：");

Scanner in=new Scanner(System.in);

int n1=in.nextInt();

int n2=in.nextInt();

System.out.println("互换前第一个数为："+n1+",第二个数为："+n2);

{

n1=n1^n2;

n2=n2^n1;

n1=n1^n2;

}

System.out.println("交换后第一个数："+n1+",第二个数："+n2);

}

}

32.

import java.util.Scanner;

public class Test232 {

public static void main(String[] args){

Scanner scan=new Scanner(System.in);

System.out.println("请输入QQ等级：");

int i=scan.nextInt();

if(i==1)

System.out.println("需要5天活跃天数");

else if(i==2)

System.out.println("需要12天活跃天数");

else if(i==3)

System.out.println("需要21天活跃天数");

else if(i==4)

System.out.println("需要32天活跃天数");

else if(i==5)

System.out.println("需要45天活跃天数");

else if(i==6)

System.out.println("需要60天活跃天数");

else if(i==7)

System.out.println("需要77天活跃天数");

else if(i==8)

System.out.println("需要96天活跃天数");

else if(i==12)

System.out.println("需要192天活跃天数");

else if(i==16)

System.out.println("需要320天活跃天数");

else if(i==32)

System.out.println("需要1152天活跃天数");

else if(i==48)

System.out.println("需要2496天活跃天数");

else

System.out.println("对不起，暂时未知！");

}

}

33.

import java.util.Scanner;

public class Test233 {

public static void main(String[] args) {

System.out.println("请输入您的年终奖：");

double bonus;// 年终奖金额

Scanner scan = new Scanner(System.in);

bonus = scan.nextDouble();

double tax; // 年终奖应缴税

// 累计应缴税额(累计应缴税额 \* 税率 - 速算扣除数)

if (bonus <= 36000.0) {

tax = bonus \* 0.03;

} else if (bonus <= 144000.0) {

tax = bonus \* 0.1 - 210;

} else if (bonus <= 300000.0) {

tax = bonus \* 0.2 - 1410;

} else if (bonus <= 420000.0) {

tax = bonus \* 0.25 - 2660;

} else if (bonus <= 660000.0) {

tax = bonus \* 0.30 - 4410;

} else if (bonus <= 960000.0) {

tax = bonus \* 0.35 - 7160;

} else {

tax = bonus \* 0.45 - 15160;

}

// 税后年终奖(税后年终奖 = 税前年终奖 - 年终奖缴税)

System.out.println("应缴纳税额为：" + tax + ",实发金额为：" + (bonus - tax));

}

}

34.

import java.util.Scanner;

public class Test234 {

public static void main(String[] args) {

System.out.println("请输入本次购物金额：");

double amount;

Scanner scan=new Scanner(System.in);

amount=scan.nextDouble();

if(amount<200){

System.out.println("不享受折扣！");

}else if(amount<400){

System.out.println("95折");

}else if(amount<600){

System.out.println("9折");

}else if(amount<800){

System.out.println("85折");

}else if(amount<1000){

System.out.println("83折");

}else if(amount<1200){

System.out.println("8折");

}else if(amount<1400){

System.out.println("78折");

}else if(amount<1600){

System.out.println("75折");

}else if(amount<1800){

System.out.println("73折");

}else if(amount<2000){

System.out.println("7折");

}else if(amount<2200){

System.out.println("65折");

}else{

System.out.println("6折");

}

}

}

35.

public class Test235 {

public static void main(String[] args) {

// 空心菱形

for (int i = 1; i < 5; i++) {

// 打印上半部分空格

for (int j = 1; j <= 4 - i; j++) {

System.out.print(" ");

}

// 打印原本实心的\*部分

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

// 仅在一行的开头和末尾打印\*

if (j == 1 || j == 2 \* i - 1) {

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

} else {

System.out.print(" ");

}

}

System.out.println(" ");

}

for (int i = 3; i >= 1; i--) {

// 打印下半部分空格

for (int j = 1; j <= 4 - i; j++) {

System.out.print(" ");

}

// 打印原本实心的\*部分

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

// 仅在一行的开头和末尾打印\*

if (j == 1 || j == 2 \* i - 1) {

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

} else {

System.out.print(" ");

}

}

System.out.println(" ");

}

}

}

# 单元三：

1.

面向对象中的类就是对某一类事物的总称，具有相同特征和行为的一类事务就是类，比如手机、学生等。

2.

**public** **class** TestRectangle {

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

Rectangle r=**new** Rectangle(3,2);

System.***out***.println("矩形面积为"+r.getArea()+"\n矩形周长为："+r.getPerimeter());

}

}

**class** Rectangle {

**public** **int** width;

**public** **int** length;

**public** Rectangle() { //定义无参构造方法

}

**public** Rectangle(**int** w,**int** l) { //定义有参构造方法

width=w;

length=l;

}

**public** **int** getPerimeter() {

**return** 2\*(width+length);

}

**public** **int** getArea() {

**return** width\*length;

}

}

3.

**import** java.util.Scanner;

**public** **class** TestRectangle {

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

**int** width,length;

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

System.***out***.println("请输入矩形的长：");

length=sc.nextInt();

System.***out***.println("请输入矩形的宽：");

width=sc.nextInt();

Rectangle r=**new** Rectangle(width,length);

System.***out***.println("矩形面积为"+r.getArea()+"\n矩形周长为："+r.getPerimeter()+"\n矩形的个数为："+Rectangle.*getRectangleCount*());

System.***out***.println("请输入矩形的长：");

length=sc.nextInt();

System.***out***.println("请输入矩形的宽：");

width=sc.nextInt();

r=**new** Rectangle(width,length);

System.***out***.println("矩形面积为"+r.getArea()+"\n矩形周长为："+r.getPerimeter()+"\n矩形的个数为："+Rectangle.*getRectangleCount*());

}

}

**class** Rectangle {

**public** **int** width;

**public** **int** length;

**public** **static** **int** *count*;

**public** Rectangle() { //定义无参构造方法

*count*++;

}

**public** Rectangle(**int** w,**int** l) { //定义有参构造方法

width=w;

length=l;

*count*++;

}

**public** **static** **int** getRectangleCount() { //定义静态成员方法

**return** *count*;

}

**public** **int** getPerimeter() {

**return** 2\*(width+length);

}

**public** **int** getArea() {

**return** width\*length;

}

}

4.

**public** **class** TestEmployee {

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

Employee e=**new** Employee();

e.employeeid=1;

e.name="张三";

e.age=25;

System.***out***.println("员工编号为："+e.employeeid+"\n员工姓名为："+e.name+"\n员工年龄为:"+e.age);

}

}

**class** Employee{

**public** **int** employeeid;

**public** String name;

**public** **int** age;

}

5.

**public** **class** TestEmployee {

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

Employee e=**new** Employee();

e.setEmployeeid(1);

e.setName("张三");

e.setAge(25);

e.display();

}

}

**class** Employee{

**private** **int** employeeid;

**private** String name;

**private** **int** age;

**public** **int** getEmployeeid() {

**return** employeeid;

}

**public** **void** setEmployeeid(**int** employeeid) {

**this**.employeeid = employeeid;

}

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** Employee(){

}

**public** **void** display(){

System.***out***.println("员工编号为："+employeeid+"\n员工姓名为："+ name+"\n员工年龄为:"+ age);

}

}

6. 创建person.java包，在此包中定义Teacher类，创建test.java包，在此包中定义一个TeacherTest类，并在TeacherTest类中定义和创建Teacher类的对象。

Teacher类包含在包person.java中，类前面加package person.java

**package** person.java;

**public** **class** Teacher {

}

TeacherTest类在包test.java中，类前面加package test.java,此类若要用类Teacher,需要用语句import person.java.Teacher导入Teahcer类所在的包

**package** test.java;

**import** person.java.Teacher;

**public** **class** TestTeacher {

Teacher t=**new** Teacher();

}

# 单元四：

1.D

2.D

3. A

4.C

5.A

6. Father类的输出

Son类的输出

7.

无参构造

有参构造,int类型第一个参数10,第二个参数208.抽象类与接口有什么区别？

# 单元五：

1.

public class Test501 {

public static void main(String[] args) {

String[] a = new String[3];

a[0] = "A";

a[1] = "B";

a[2] = "C";

for (int i = 0; i < 3; i++) {

System.out.println(a[i]);

}

}

}

2.

public class Test502 {

public static void main(String[] args) {

String[] a = new String[3];

a[0] = "我";

a[1] = "爱";

a[2] = "java";

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

System.out.print(a[i]);

}

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

for (int c = a.length - 1; c >= 0; c--) {

System.out.print(a[c]);

}

}

}

3.

import java.util.Scanner;

public class Test503 {

public static void main(String[] args) {

int a[]=new int[5];

int b[]=new int[5];

System.out.println("请输入5个整数,用回车隔开：");

Scanner scan=new Scanner(System.in);

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

a[i]=scan.nextInt();

}

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

b[i]=a[i];

System.out.println("b["+i+"]"+"="+b[i]);

}

}

}

4.

import java.util.Arrays;

import java.util.Scanner;

public class Test504 {

public static void main(String[] args) {

int[] arr = new int[5];

int sum = 0;

Scanner sc = new Scanner(System.in);

System.out.println("请输入5个成绩：");

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

arr[i] = sc.nextInt();

sum = sum + arr[i];

}

Arrays.sort(arr);// 调用排序方法

int max = arr[4];

int min = arr[0];

System.out.println("这5个学生的总分为：" + sum);

System.out.println("这5个学生的平均成绩为：" + sum / 5);

System.out.println("这5个学生的最高分为：" + max);

System.out.println("这5个学生的最低分为：" + min);

}

}

5.

import java.util.Scanner;

public class Test505 {

public static void main(String[] args) {

// 从控制台获取行数

Scanner sc = new Scanner(System.in);

System.out.print("打印杨辉三角形的行数：");

int row = sc.nextInt();

// 根据行数定义好二维数组，由于每一行的元素个数不同，所以不定义每一行的个数

int[][] arr = new int[row][];

// 遍历二维数组

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

// 初始化每一行的这个一维数组

arr[i] = new int[i + 1];

for (int j = 1; j <= row - i; j++) {

System.out.print(" ");

}

// 遍历这个一维数组，添加元素

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

// 每一列的开头和结尾元素为1，开头的时候，j=0，结尾的时候，j=i

if (j == 0 || j == i) {

arr[i][j] = 1;

} else {

// 每一个元素是它上一行的元素和斜对角元素之和

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

}

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

}

System.out.println();

}

}

}

6.

import java.util.Scanner;

public class Test506 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("请输入你要插入的数：");

int a = sc.nextInt();

int[] arr = {0,3,8,11,12,78,82,90,95,100,103};

insert(arr, a); //调用insert方法

}

public static void insert(int[] arr,int a) {

int[] b = new int[arr.length+1];

if (a < arr[arr.length-1]) {

int count = 0;

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

if (a < arr[i]) {

b[i] = a;

}else {

b[i] = arr[i];

count++;

}

}

int sum = count;

for (int i = 1; i < b.length-sum; i++) {

b[count+1] = arr[count];

count++;

}

}else {

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

b[i]= arr[i];

}

b[arr.length] = a;

}

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

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

}

}

}

7.

import java.util.Scanner;

public class Test507 {

public static void main(String[] args) {

int key;

boolean found = false;

int[] arr = { 12, 78, 0, 3, 11, 3, 8, 11, 82, 90 };

System.out.println("请输入一个整数：");

Scanner scan = new Scanner(System.in);

key = scan.nextInt();

for (int i = 0; i < arr.length - 1; i++) {

if (arr[i] == key) {

found = true;

System.out.println("存在下标为" + i + "的数和输入的数相同");

}

}

if (found == false) {

System.out.println("此数不存在！");

}

}

}

8.

public class Test508 {

public static void main(String[] args) {

char[] c=new char[3];

c[0]='我';

c[1]='爱';

c[2]='你';

for(int i=c.length-1;i>=0;i--){

System.out.print(c[i]);

}

}

}

9.

import java.util.Arrays;

public class Test509 {

public static void main(String[] args) {

int min;

int max;

int[] a={98,86,1,37,100,71,50};

Arrays.sort(a);

min=a[0];

max=a[a.length-1];

System.out.println("最大值为："+max);

System.out.println("最小值为："+min);

}

}

10.

public class Test510 {

public static void main(String[] args) {

int a[][] = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };

int b[][] = new int[3][3];

System.out.println("转置前的矩阵为： ");

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

b[j][i] = a[i][j];// 进行转置

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

}

System.out.println();

}

System.out.println("转置后的矩阵为： ");

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

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

}

System.out.println();

}

}

}

11.

import java.util.\*;

public class Test511 {

public static void main(String[] args) {

int digital = 0;

int character = 0;

int other = 0;

int blank = 0;

char[] ch=null;

System.out.println("请输入一行字符，比如123 ABC！@#：");

Scanner sc = new Scanner(System.in);

String s = sc.nextLine();

ch = s.toCharArray();

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

if(ch[i]>='0'&&ch[i]<='9') {

digital ++;

} else if((ch[i]>='a'&&ch[i]<='z')||ch[i]>='A'&&ch[i]<='Z') {

character ++;

} else if(ch[i]==' ') {

blank ++;

} else {

other++;

}

}

System.out.println("数字个数: " + digital);

System.out.println("英文字母个数: " + character);

System.out.println("空格个数: " + blank);

System.out.println("其他字符个数:" + other );

}

}

12.

public class Test512 {

public static void main(String[] args) {

//加法运算符

String str1="Hello";

String str2=",";

String str3="Java";

String str4=str1+str2+str3;

System.out.println("使用加法运算符输出："+str4);

//append()

StringBuffer sb=new StringBuffer();

sb.append(str1).append(str2).append(str3);

System.out.println("使用append()方法输出："+sb.toString());

//append()方法的另外一种形式

StringBuilder sb1=new StringBuilder();

sb1.append(str1).append(str2).append(str3);

System.out.println("append()方法的另一种形式输出："+sb1.toString());

}

}

13.

import java.text.NumberFormat;

import java.util.Locale;

import java.util.Scanner;

public class Test513 {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

System.out.println("请输入一个数字：");

double number = scan.nextDouble();// 获取用户输入数字

System.out.println("该数字用Locale类的常量作为格式化对象的构造参数，将获得不同的货币格式：");

// 创建格式化对象

NumberFormat format = NumberFormat.getCurrencyInstance(Locale.CHINA);

// 输出格式化货币格式

System.out.println("Locale.CHINA：" + format.format(number));

format = NumberFormat.getCurrencyInstance(Locale.US);

System.out.println("Locale.US：" + format.format(number));

format = NumberFormat.getCurrencyInstance(Locale.ENGLISH);

System.out.println("Locale.ENGLISH：" + format.format(number));

}

}

14.

import java.util.Scanner;

public class Test514 {

public static void main(String[] args) {

System.out.println("输入:");

Scanner in = new Scanner(System.in);

String a = in.nextLine(); // 获取String值

StringBuilder stringBuilder = new StringBuilder(a); // 建立StringBuilder实例

for (int i = 0; i < stringBuilder.length(); i++) { // for循环语句，每次对比输入的字符串的每一个字符

if (stringBuilder.charAt(i) == ' ') { // 如果字符等于空格

stringBuilder.deleteCharAt(i); // 删掉空格部分

i--; // i自减1

} else {

stringBuilder.charAt(i); // 输出有字符的部分

}

}

System.out.println("输出：" + stringBuilder.toString());// 输出结果

}

}

15.

import java.util.Scanner;

public class Test515 {

public static void main(String[] args) {

System.out.println("请输入一个字符串：");

String s = new String();

Scanner scan = new Scanner(System.in);

s = scan.nextLine();

StringBuffer sb = new StringBuffer(s);

System.out.println("逆序输出：");

System.out.println(sb.reverse().toString());

}

}

16.

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import java.util.regex.PatternSyntaxException;

public class Test516{

/\*\*

\* ^ 匹配输入字符串开始的位置

\* \d 匹配一个或多个数字，其中 \ 要转义，所以是 \\d

\* $ 匹配输入字符串结尾的位置

\*/

public static final String CHINA\_REGEX\_EXP = "^((13[0-9])|(14[5,7,9])|(15[0-3,5-9])|(166)|(17[0-9])|(18[0-9])|(19[1,8,9]))\\d{8}$";

public static final String HK\_REGEX\_EXP = "^(5|6|8|9)\\d{7}$";

public static void main(String[] args) {

String phoneNum = "1637865679";

System.out.println(isPhoneNum(phoneNum));

}

/\*\*

\* 校验是否为大陆号码或香港号码

\* @param str

\* @return 符合规则返回true

\* @throws PatternSyntaxException

\*/

public static boolean isPhoneNum(String str) throws PatternSyntaxException {

return isChinaPhoneNum(str) || isHkPhoneNum(str);

}

/\*\*

\* 大陆手机号码11位数，匹配格式：前三位固定格式+后8位任意数

\* 此方法中前三位格式有：

\* 13+任意数

\* 145,147,149

\* 15+除4的任意数(不要写^4，这样的话字母也会被认为是正确的)

\* 166

\* 17+任意数

\* 18+任意数

\* 198,199

\* @param str

\* @return 正确返回true

\* @throws PatternSyntaxException

\*/

public static boolean isChinaPhoneNum(String str) throws PatternSyntaxException {

// ^ 匹配输入字符串开始的位置

// \d 匹配一个或多个数字，其中 \ 要转义，所以是 \\d

// $ 匹配输入字符串结尾的位置

Pattern p = Pattern.compile(CHINA\_REGEX\_EXP);

Matcher m = p.matcher(str);

return m.matches();

}

/\*\*

\* 香港手机号码8位数，5|6|8|9开头+7位任意数

\*

\* @param str

\* @return 正确返回true

\* @throws PatternSyntaxException

\*/

public static boolean isHkPhoneNum(String str) throws PatternSyntaxException {

Pattern p = Pattern.compile(HK\_REGEX\_EXP);

Matcher m = p.matcher(str);

return m.matches();

}

}

17.

public class Test517 {

public static void main(String[] args) {

String ip = "255.255.255.256";

System.out.println(ipCheck(ip));

}

public static boolean ipCheck(String ip) {

if (ip != null && !ip.isEmpty()) {

// 定义正则表达式

String regex = "^(1\\d{2}|2[0-4]\\d|25[0-5]|[1-9]\\d|[1-9])\\."

+ "(1\\d{2}|2[0-4]\\d|25[0-5]|[1-9]\\d|\\d)\\." + "(1\\d{2}|2[0-4]\\d|25[0-5]|[1-9]\\d|\\d)\\."

+ "(1\\d{2}|2[0-4]\\d|25[0-5]|[1-9]\\d|\\d)$";

// 判断ip地址是否与正则表达式匹配

if (ip.matches(regex)) {

// 返回判断信息

return true;

} else {

// 返回判断信息

return false;

}

}

return false;

}

}

18.

import java.util.Scanner;

public class Test518 {

public static void main(String[] args) {

System.out.println("请输入一个字符串：");

String s = new String();

Scanner scan = new Scanner(System.in);

s = scan.nextLine();

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

}

}

19.

public class Test519 {

public static void main(String[] args) {

String str="abcdefg1234!";

System.out.println(str.charAt(3));

}

}

20.

public class Test520 {

public static void main(String[] args) {

String str="abcdefg1234!";

System.out.println(str.substring(2, 6));

}

}

21.

import java.text.DateFormat;

import java.util.Calendar;

import java.util.Date;

public class Test521 {

public static void main(String[] args) {

System.out.print("从今天起，150天以后的日期为：");

Calendar calendar = Calendar.getInstance();

calendar.add(Calendar.DATE, 150);

Date date = calendar.getTime();

DateFormat format = DateFormat.getDateInstance(DateFormat.FULL);

String string = format.format(date);

System.out.println(string);

}

}

22.

import java.util.Random;

public class Test522 {

public static void main(String[] args) {

int[] array = randomNum(16, 29);// 调用randomNum()方法

System.out.print("生成的随机数为：");

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

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

}

System.out.println();

// 调用冒泡排序方法

sort(array);

System.out.print("使用冒泡排序算法后：");

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

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

}

}

// randomNum()方法

public static int[] randomNum(int start, int end) {

Random r = new Random();

int[] array = new int[5];

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

array[i] = r.nextInt(end) % (end - start + 1) + start;

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

if (array[i] == array[j]) {// 检测是否重复

i = -1; // 如果重复则重新开始

break;

}

}

}

return array;

}

// 冒泡排序

public static void sort(int[] array) {

for (int i = 0; i < array.length - 1; i++) {

// 当前值当作最小值

int min = array[i];

for (int j = i + 1; j < array.length; j++) {

if (min > array[j]) {

// 如果后面有比min值还小的就交换

min = array[j];

array[j] = array[i];

array[i] = min;

}

}

}

}

}

23.

import java.util.Scanner;

public class Test523 {

public static void main(String[] args) {

double a;

double b;

System.out.println("请输入一个实数：");

Scanner scan=new Scanner(System.in);

a=scan.nextDouble();

b=Math.sqrt(a);

System.out.println(a+"的平方根是："+b);

}

}

24.

import java.util.ArrayList;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

public class Test524 {

public static void main(String[] args) {

Set<String> set = new HashSet<>();

set.add("a");

set.add("A");

set.add("c");

set.add("C");

set.add("a");

set.add("c");

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

list.add("a");

list.add("A");

list.add("c");

list.add("C");

list.add("a");

list.add("c");

System.out.println(set);

System.out.println(list);

}

}

25.

public class Emp {

private String e\_id;

private String e\_name;

public Emp(String e\_id, String e\_name) {

super();

this.e\_id = e\_id;

this.e\_name = e\_name;

}

public String getE\_id() {

return e\_id;

}

public void setE\_id(String e\_id) {

this.e\_id = e\_id;

}

public String getE\_name() {

return e\_name;

}

public void setE\_name(String e\_name) {

this.e\_name = e\_name;

}

}

import java.util.HashMap;

import java.util.Iterator;

import java.util.Map;

import java.util.Set;

import java.util.TreeMap;

public class Test525 {

public static void main(String[] args) {

Map<String,String> map=new HashMap();

Emp emp=new Emp("001","国庆节");

Emp emp1=new Emp("005","星期一");

Emp emp2=new Emp("003","劳动节");

Emp emp3=new Emp("008","春节");

map.put(emp.getE\_id(), emp.getE\_name());

map.put(emp1.getE\_id(), emp1.getE\_name());

map.put(emp2.getE\_id(), emp2.getE\_name());

map.put(emp3.getE\_id(), emp3.getE\_name());

Set<String> keys=map.keySet();

System.out.println("遍历集合map：");

Iterator<String> it=keys.iterator();

while(it.hasNext())

{

String key=it.next();

System.out.println(key+" "+map.get(key));

}

System.out.println("移除的是"+map.remove("005"));

Map<String,String>treemap=new TreeMap();

treemap.putAll(map);

Set set=treemap.keySet();

System.out.println("遍历集合treemap：");

Iterator<String> ss=set.iterator();

while(ss.hasNext())

{

String str=ss.next();

System.out.println(str+" "+treemap.get(str));

}

}

}

26.

import java.util.ArrayList;

import java.util.Random;

public class Test526 {

public static void main(String[] args) {

ArrayList<Object> array = new ArrayList<Object>();

Random rand = new Random();

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

int choice = rand.nextInt(3);

switch (choice) {

case 0:

int num = rand.nextInt(9);

array.add(num);

break;

case 1:

int num2 = rand.nextInt(25) + 65;

char low = (char) num2;

array.add(low);

break;

case 2:

int num3 = rand.nextInt(25) + 97;

char high = (char) num3;

array.add(high);

break;

default:

break;

}

}

System.out.println("验证码是：");

for (Object object : array) {

System.out.print(object);

}

}

}

27.

import java.util.Random;

public class Test527 {

public static void main(String[] args) {

int[] array = randomNum(1, 30);

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

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

}

}

public static int[] randomNum(int start, int end) {

Random r = new Random();

int[] array = new int[9];

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

array[i] = r.nextInt(end) % (end - start + 1) + start;

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

if (array[i] == array[j]) {// 检测是否重复

i = -1; // 如果重复则重新开始

break;

}

}

}

return array;

}

}

28.

import java.util.ArrayList;

public class Test528 {

public static void main(String[] args) {

ArrayList<Integer> list = new ArrayList<Integer>();

list.add(1);

list.add(2);

list.add(3);

list.add(4);

list.add(5);

for(Object o:list) {

System.out.println(o);

}

}

}

29.

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class Sort {

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

Scanner input=new Scanner(System.in);

boolean flag=true;

String temp=null;

public void write() {

do {

temp=input.next();

if(temp.equals("0")) {

flag=false;

}else {

list.add(temp);

}

}while(flag);

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

for (int j = 0; j <list.size()-1-i; j++) {

if(list.get(j).toUpperCase().compareTo(list.get(j+1).toUpperCase())>0) {

temp=list.get(j);

list.set(j,list.get(j+1));

list.set(j+1, temp);

}

}

}

System.out.println("以下是从小到大的单词排序");

for (String str : list) {

System.out.print(str+"\t");

}

}

}

public class Test529 {

public static void main(String[] args) {

System.out.println("请输入单词（输入0结束）：");

Sort sl=new Sort();

sl.write();

}

}

30.

public class Customer {

private int ID;

public Customer() {

super();

// TODO Auto-generated constructor stub

}

public Customer(int iD, String name, double balance) {

super();

ID = iD;

this.name = name;

this.balance = balance;

}

@Override

public String toString() {

return "Customer [ID=" + ID + ", name=" + name + ", balance=" + balance + "]";

}

@Override

public int hashCode() {

final int prime = 31;

int result = 1;

result = prime \* result + ID;

long temp;

temp = Double.doubleToLongBits(balance);

result = prime \* result + (int) (temp ^ (temp >>> 32));

result = prime \* result + ((name == null) ? 0 : name.hashCode());

return result;

}

@Override

public boolean equals(Object obj) {

if (this == obj)

return true;

if (obj == null)

return false;

if (getClass() != obj.getClass())

return false;

Customer other = (Customer) obj;

if (ID != other.ID)

return false;

if (Double.doubleToLongBits(balance) != Double.doubleToLongBits(other.balance))

return false;

if (name == null) {

if (other.name != null)

return false;

} else if (!name.equals(other.name))

return false;

return true;

}

private String name;

private double balance;

public int getID() {

return ID;

}

public void setID(int iD) {

ID = iD;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public double getBalance() {

return balance;

}

public void setBalance(double balance) {

this.balance = balance;

}

}

import java.util.HashMap;

public class Test530 {

public static void main(String[] args) {

HashMap<Integer, Customer> cus=new HashMap<>();

Customer cus1=new Customer(123, "张嘻嘻", 500);

Customer cus2=new Customer(124, "李呵呵", 300);

Customer cus3=new Customer(125, "王哈哈", 700);

cus.put(cus1.getID(), cus1);

cus.put(cus2.getID(), cus2);

cus.put(cus3.getID(), cus3);

Customer customer = cus.get(124);

System.out.println(customer);

}

}

# 单元六：

4. throw语句后面不能有其他语句，没有办法执行

Exception in thread "main" java.lang.Error: Unresolved compilation problem:

Unreachable code

5.请写出以下代码段的运行结果（代码调整）。

**public** **class** Test {

**public** **static** **void** A(**int** x) **throws** MyException{

**if**(x<0)

{

**throw** **new** MyException( x+"不能小于0");

}

}

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

**try**{

**int** x=-5;

*A*(x);

}**catch**(MyException e) {

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

}

}

}

**class** MyException **extends** Exception{

**public** MyException(String s) {

**super**(s);

}

}

运行结果：MyException: -5不能小于0

6.

**import** java.util.Scanner;

**public** **class** Test {

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

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

Object x[] = **new** String[3];

**int** i=sc.nextInt();

x[0] = **new** Integer(i);

}

}

运行程序，任意输入一个数6，则会抛出：

6

Exception in thread "main" java.lang.ArrayStoreException: java.lang.Integer

at Test.main(Test2.java:8)

7.

**import** java.util.Scanner;

**public** **class** Test2 {

**public** **static** **void** TestAge(**int** age) **throws** IllegalAgeException{

**if**(age<18 || age>60)

{

**throw** **new** IllegalAgeException(age);

}

**else** System.***out***.print("age="+age+" 年龄符合要求，在18岁到60岁之间");

}

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

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

**try**{

**int** age=sc.nextInt();

*TestAge*(age);

}**catch**(IllegalAgeException e) {

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

}

}

}

**class** IllegalAgeException **extends** Exception{

**public** IllegalAgeException(**int** age) {

**super**("age="+age+" 年龄小于18岁或者大于60岁不符合要求");

}

}

8.

**import** java.util.Scanner;

**public** **class** Test {

**public** **static** **void** testIdentifier(String identifier) **throws** IllegalIdentifierException{

**char** ch=identifier.charAt(0);

**if**((ch>='a'&& ch<='z')||(ch>='A'&& ch<='Z')|| ch=='\_')

{

System.***out***.print("identifier="+identifier+"标识符以字母或者下划线开始");

}

**else** **throw** **new** IllegalIdentifierException(identifier);

}

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

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

**try**{

String identifier=sc.next();

*testIdentifier*(identifier);

}**catch**(IllegalIdentifierException e) {

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

}

}

}

**class** IllegalIdentifierException **extends** Exception{

**public** IllegalIdentifierException(String identifier) {

**super**("identifier="+identifier+"标识符没有以字母或者下划线开始，不符合要求");

}

}

# 单元七：

1.不能 ，把t.run()改为t.start()

2.

**public** **class** TestRunnable **implements** Runnable{

**private** **int** apple=10;

**public** **void** run() {

**for**(**int** i=1;i<=10;i++)

{ **synchronized**(**this**){

**if**(apple>0) {

System.***out***.println(Thread.*currentThread*().getName()+"正在吃苹果："+apple);

apple--;

}

}

}

}

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

TestRunnable tt=**new** TestRunnable();

Thread t1=**new** Thread(tt,"青青");

Thread t2=**new** Thread(tt,"丽丽");

t1.start();

t2.start();

}

}

3.

**public** **class** TestPriority **extends** Thread{

**public** TestPriority(String s){

**super**(s);

}

**public** **void** run() {

System.***out***.println("正在运行的线程名字："+Thread.*currentThread*().getName());

}

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

TestPriority t1=**new** TestPriority ("线程1");

TestPriority t2=**new** TestPriority ("线程2");

TestPriority t3=**new** TestPriority ("线程3");

t1.setPriority(1);

t2.setPriority(5);

t3.setPriority(10);

t1.start();

t2.start();

t3.start();

}

}

4.

不能，应该回到就绪状态。

# 单元八：

2.

当调用字节流的read()方法时，读取一个字节的值，有时候会读到11111111，如果返回的是byte类型，这个值在计算机内部表示-1， 就会提前结束流操作；而如果返回的是int类型，int为4个字节，那么读到的11111111则会表示成00000000 00000000 00000000 11111111，这个值表示的是255，就不会和输入流定义的结束标志冲突了。

为了能够保证输入流的正确运行，read()方法返回得并不是字节值，而是返回int，这样做的目的是为了正确判断读到文件结尾 。

字符流Reader的方法read()的返回值为char，字符流的read()方法一次读取一个字符，类型char的取值范围是0到65535，这个范围内的所有字符，都有可能在读取的数据中出现，所以没有办法表示读取到流末尾，-1不在char的范围内出现，用来表示流末尾，为了返回-1，就用int来表示返回值，而不是char。

3.

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**public** **class** FileCopy {

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

// **TODO** Auto-generated method stub

FileCopy f= **new** FileCopy();

**try** {

f.copy("d:/File/1.txt","d:/File/copy1.txt");

} **catch** (Exception e) {

e.printStackTrace();

}

}

**public** **void** copy(String f1,String f2) **throws** Exception{

FileInputStream fis = **new** FileInputStream(f1); //首先构建输入流的对象,指定需要读取的文件路径

FileOutputStream fos = **new** FileOutputStream(f2,**false**); //构建文件输出流的对象，即将文件复制在哪里去,后面的true代表每次写入时不清空当前文件内容

**int** value = fis.read(); //一个字节一个字节的读取文件的内容

**while**(value!=-1){

fos.write(value);

fos.flush();

value = fis.read();

}

fis.close();

fos.close();

}

}

4.编写程序，用字节流实现视频文件的复制（字符流复制音频文件后比源文件小，打不开）。

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

**public** **class** FileMp3Copy {

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

File f = **new** File("D:\\file\\1.avi"); // 要操作的文件

FileInputStream fis=**new** FileInputStream(f);

BufferedInputStream bis=**new** BufferedInputStream(fis);

File fcopy = **new** File("D:\\file\\副本1.avi"); // 要操作的文件

FileOutputStream fos=**new** FileOutputStream(fcopy);

BufferedOutputStream bos=**new** BufferedOutputStream(fos);

**byte**[] b=**new** **byte**[32];

**int** len=0;

**while**((len=bis.read(b))!=-1) {

bos.write(b,0,len);

}

bis.close();

bos.close();

fis.close();

fos.close();

}

}

5.

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

**public** **class** DeleteBmp {

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

File directory=**new** File("..\\Picture");

*deleteDir*(directory);

System.***out***.println(directory.getAbsolutePath());

}

**public** **static** **void** deleteDir(File dir) {

**if** (!dir.exists()) {

//如果文件夹不存在，直接返回，函数结束

**return**;

}

**if** (dir.isFile()) {

//如果dir是文件，调用delete()方法删除文件

String filename=dir.getName();

**if**(filename.substring(filename.lastIndexOf(".")).equals(".bmp"))

dir.delete();

} **else** **if** (dir.isDirectory()) {//dir是文件夹

File[] files = dir.listFiles();//获取文件夹的所有文件和子文件夹

**for** (File myfile : files)

{//对数组files的每一个文件或文件夹，递归调用deleteDir(File dir)方法

*deleteDir*(**new** File(myfile.getAbsolutePath()));

}

}

}

}

# 单元九：

1.B 2.C 3.D 4.A 5.D 6.D 7.D 8.D 9.D 10.A 11.C 12.B 13.B 14.C 15.C

16.B 17.D 18.C