**算法大全**

## 一、经典算法大全

【程序1】 题目：古典问题：有一对兔子，从出生后第3个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子总数为多少？

1.程序分析：兔子的规律为数列1,1,2,3,5,8,13,21....

具体分析如下：

f(1) = 1(第1个月有一对兔子）

f(2) = 1(第2个月还是一对兔子）

f(3) = 2(原来有一对兔子，第3个开始，每个月生一对兔子）

f(4) = 3(原来有两对兔子，有一对可以生育）

f(5) = 5(原来有3对兔子，第3个月出生的那对兔子也可以生育了，那么现在有两对兔子可以生育）

f(6) = 8(原来有5对兔子，第4个月出生的那对兔子也可以生育了，那么现在有3对兔子可以生育）

..............

由以上可以看出，第n个月兔子的对数为

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

f(n-1)是上个月的兔子数量，是原来有的。

f(n-2)是可以生育的兔子数，即多出来的数量。第n-2个月开始后的第3个月是第n个月，此时第n-2个月时的兔子都可以生育了。

public class Demo01 {

public static void main(String args[]) {

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

System.out.println(f(i));

}

public static int f(int x) {

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

return 1;

else

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

}

}

或

public class Demo01 {

public static void main(String args[]) {

math mymath = new math();

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

System.out.println(mymath.f(i));

}

}

class math {

public int f(int x) {

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

return 1;

else

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

}

}

【程序2】 题目：判断101-200之间有多少个素数，并输出所有素数。

1.程序分析：判断素数的方法：用一个数分别去除2到sqrt(这个数)，如果能被整除，则表明此数不是素数，反之是素数。

public class Demo02{

public static void main(String[] args){

for(int i=2;i<=200;i++){

boolean flag=true;

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

if(i%j==0){

flag=false;

break;

}

}

if(flag==true){

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

}

}

}

}

【程序3】 题目：打印出所有的 水仙花数 ，所谓 水仙花数 是指一个三位数，其各位数字立方和等于该数本身。例如：153是一个 水仙花数 ，因为153=1的三次方＋5的三次方＋3的三次方。

1.程序分析：利用for循环控制100-999个数，每个数分解出个位，十位，百位。

public class Demo03 {

public static void main(String args[]) {

math mymath = new math();

for (int i = 100; i <= 999; i++)

if (mymath.shuixianhua(i) == true)

System.out.println(i);

}

}

class math {

public boolean shuixianhua(int x) {

int i = 0, j = 0, k = 0;

i = x/100;

j = (x%100)/10;

k = x%10;

if (x == i\*i\*i + j\*j\*j + k\*k\*k)

return true;

else

return false;

}

}

【程序4】 题目：将一个正整数分解质因数。例如：输入90,打印出90=2\*3\*3\*5。

1.程序分析：对n进行分解质因数，应先找到一个最小的质数i，然后按下述步骤完成：

(1)如果这个质数恰等于n，则说明分解质因数的过程已经结束，打印出即可。

(2)如果n > i，但n能被i整除，则应打印出i的值，并用n除以i的商,作为新的正整数你,重复执行第一步。

(3)如果n不能被i整除，则用i+1作为i的值,重复执行第一步。

import java.util.Scanner;

public class Demo04 {

public Demo04() {

super();

}

public void fenjie(int n) {

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

if (n % i == 0) {

System.out.print(i);

if(n!=i){

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

}

fenjie(n/i);

}

}

System.exit(0); //退出程序

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("请输入N的值：");

int N = in.nextInt();

System.out.print( "分解质因数：" + N +"=");

new Demo04().fenjie(N);

}

}

【程序5】 题目：利用条件运算符的嵌套来完成此题：学习成绩=90分的同学用A表示，60-89分之间的用B表示，60分以下的用C表示。

1.程序分析：(a>b)?a:b这是条件运算符的基本例子。

import java.util.Scanner;

public class Demo05 {

public static void main(String[] args) {

System.out.println("请输入N的值：");

Scanner in = new Scanner(System.in);

int N = in.nextInt();

System.out.println(N >= 90 ?"A": (N >= 60 ? "B":"C"));

}

}

【程序6】 题目：输入两个正整数m和n，求其最大公约数和最小公倍数。

1.程序分析：利用辗除法。

import java.util.Scanner;

public class Demo06 {

public static void main(String[] args){

int a,b,m,n;

Scanner in=new Scanner(System.in);

System.out.println("请输入一个正整数：");

a=in.nextInt();

System.out.println("再输入一个正整数：");

b=in.nextInt();

commonDivisor use=new commonDivisor();

m=use.commonDivisor(a,b);

n=a\*b/m;

System.out.println("最大公约数："+m);

System.out.println("最小公倍数："+n);

}

}

class commonDivisor{

public int commonDivisor(int x,int y){

if(x<y){

int t=x;

x=y;

y=t;

}

while(y!=0){

if(x==y)return x;

else{

int k=x%y;

x=y;

y=k;

}

}

return x;

}

}

【程序7】 题目：输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数。

1.程序分析：利用for循环语句,if条件语句。

import java.util.Scanner;

public class Demo07 {

public static void main(String[] args){

System.out.println("请输入一个字符串;");

Scanner in=new Scanner(System.in);

String str=in.nextLine();

char[] ch=str.toCharArray();

count use=new count();

use.count(ch);

}

}

class count{

int digital,character,blank,other;

public void count(char[] arr){

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

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

digital++;

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

character++;

}else if(arr[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);

}

}

【程序8】 题目：求s = a + aa + aaa + aaaa + aa...a的值，其中a是一个数字。例如2 + 22 + 222 + 2222 + 22222(此时共有5个数相加)，几个数相加有键盘控制。

1.程序分析：关键是计算出每一项的值。

import java.util.Scanner;

public class Demo08 {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println(请输入a的值);

int a = in.nextInt();

System.out.println(请输入n个数);

int n = in.nextInt();

int s = 0,t=0;

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

t += a;

a = a\*10;

s += t;

}

System.out.println(s);

}

}

【程序9】 题目：一个数如果恰好等于它的因子之和，这个数就称为"完数"。例如6=1＋2＋3。编程找出1000以内的所有完数。

public class Demo09 {

public static void main(String[] args) {

int s;

for (int i = 1; i <= 1000; i++) {

s = 0;

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

if (i % j == 0)

s = s + j;

if (s == i)

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

}

System.out.println();

}

}

或

public class Demo09{

public static void main(String[] args) {

int i,j,sum;

for(i=1;i<1000;i++)

{

sum = 0;

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

{

if(i%j==0)

{

sum+=j;

}

}

if(sum==i)

{

System.out.print(i+" its factors are: ");

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

{

if(i%j==0)

System.out.print(j+", ");

}

System.out.println();

}

}

}

}

【程序10】 题目：一球从100米高度自由落下，每次落地后反跳回原高度的一半；再落下，求它在第10次落地时，共经过多少米？第10次反弹多高？

public class Demo10 {

public static void main(String[] args) {

double s = 0;

double h = 100;

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

s += h;

h = h/2;

s += h;

}

System.out.println("经过路程："+s);

System.out.println("反弹高度："+h);

}

}

【程序11】 题目：有1、2、3、4个数字，能组成多少个互不相同且无重复数字的三位数？都是多少？

1.程序分析：可填在百位、十位、个位的数字都是1、2、3、4。组成所有的排列后再去掉不满足条件的排列。

public class Demo11 {

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++)

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

count += 1;

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

}

System.out.println("共" + count + "个三位数");

}

}

【程序12】 题目：企业发放的奖金根据利润提成。利润(I)低于或等于10万元时，奖金可提10%；利润高于10万元，低于20万元时，低于10万元的部分按10%提成，高于10万元的部分，可提成7.5%；20万到40万之间时，高于20万元的部分，可提成5%；40万到60万之间时高于40万元的部分，可提成3%；60万到100万之间时，高于60万元的部分，可提成1.5%，高于100万元时，超过100万元的部分按1%提成，从键盘输入当月利润lirun，求应发放奖金总数sum？

1.程序分析：请利用数轴来分界，定位。注意定义时需把奖金定义成长整型。

import java.util.Scanner;

public class Demo12 {

public static void main(String[] args) {

double sum;

System.out.println("输入当月利润：(万元)");

Scanner in = new Scanner(System.in);

double lirun = in.nextDouble();

if (lirun <= 10) {

sum = lirun \* 0.1;

} else if (lirun <= 20) {

sum = 10\*0.1 + (lirun - 10) \* 0.075;

} else if (lirun <= 40) {

sum = 10\*0.1 + 10\*0.075 + (lirun - 20) \* 0.05;

} else if (lirun <= 60) {

sum = 10\*0.1 + 10\*0.075 + 10\*0.05 + (lirun - 40) \* 0.03;

} else if (lirun <= 100) {

sum = 10\*0.1 + 10\*0.075 + 10\*0.05 + 10\*0.03 + (lirun - 60) \* 0.015;

} else {

sum = 10\*0.1 + 10\*0.075 + 10\*0.05 + 10\*0.03 + 10\*0.015 + (lirun - 100) \* 0.01;

}

System.out.println("应发的奖金是："+sum+"(万元)");

}

}

【程序13】 题目：一个整数，它加上100后是一个完全平方数，加上168又是一个完全平方数，请问该数是多少？

1.程序分析：在10万以内判断，先将该数加上100后再开方，再将该数加上168后再开方，如果开方后的结果满足如下条件，即是结果。请看具体分析：

public class Demo13 {

public static void main(String[] args) {

for(int x=1;x<100000;x++){

if(Math.sqrt(x+100)%1==0)

if(Math.sqrt(x+100+168)%1==0)

System.out.println(x+"加上100后是一个完全平方数，加上168又是一个完全平方数");

}

}

}

【程序14】 题目：输入某年某月某日，判断这一天是这一年的第几天？

1.程序分析：以3月5日为例，应该先把前两个月的加起来，然后再加上5天即本月的第几天，特殊情况，闰年且输入月份大于3时需考虑多加一天。

import java.util.Calendar;

import java.util.Scanner;

public class Demo14 {

public static void main(String[] args) {

System.out.println("请输入年,月,日：");

Scanner in = new Scanner(System.in);

int year = in.nextInt();

int month = in.nextInt();

int day = in.nextInt();

Calendar cal = Calendar.getInstance();

cal.set(year, month - 1, day);

int sum = cal.get(Calendar.DAY\_OF\_YEAR);

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

}

}

或

import java.util.\*;

public class Demo14 {

public static void main(String[] args){

int year,month,day,sum=0;

Scanner in=new Scanner(System.in);

System.out.println("输入年：");

year=in.nextInt();

System.out.println("输入月：");

month=in.nextInt();

System.out.println("输入日：");

day=in.nextInt();

switch(month){

case 1:

sum=0;

break;

case 2:

sum=31;

break;

case 3:

sum=59;

break;

case 4:

sum=90;

break;

case 5:

sum=120;

break;

case 6:

sum=151;

break;

case 7:

sum=181;

break;

case 8:

sum=212;

break;

case 9:

sum=243;

break;

case 10:

sum=273;

break;

case 11:

sum=304;

break;

case 12:

sum=334;

break;

default:

System.out.println("wrong input!");

return;

}

sum=sum+day;

boolean leap;

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

leap=true;

}else {

leap=false;

}

if(leap&&month>2){

sum++;

}

System.out.println("It is the "+sum+"th day.");

}

}

或

import java.util.Scanner;

public class Demo14 {

public static void main(String[] args){

System.out.println("请输入年 月 日：");

Scanner in=new Scanner(System.in);

int year=in.nextInt();

int month=in.nextInt();

int day=in.nextInt();

System.out.println("是该年的第"+count(year,month,day)+"天");

}

public static int count(int year,int month,int day){

int sum=0;

int days=0;

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

switch(i){

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

days=31;

break;

case 4:

case 6:

case 9:

case 11:

days=30;

break;

case 2:

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

days=29;

}else{

days=28;

}

break;

}

sum+=days;

}

sum+=day;

return sum;

}

}

【程序15】 题目：输入三个整数x,y,z，请把这三个数由小到大输出。

1.程序分析：我们想办法把最小的数放到x上，先将x与y进行比较，如果x>y则将x与y的值进行交换，然后再用x与z进行比较，如果x>z则将x与z的值进行交换，这样能使x最小。

import java.util.Arrays;

import java.util.Scanner;

public class Demo15 {

public static void main(String[] args) {

System.out.print("请输入三个数:");

Scanner in = new Scanner(System.in);

int[] arr = new int[3];

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

arr[i] = in.nextInt();

}

Arrays.sort(arr);

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

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

}

}

}

或

if(x > y) { int t = x; x = y; y = t; } if(x > z) { int t = x; x = z; z = t; } if(y > z) { int t = y; y = z; z = t; }

【程序16】 题目：输出9\*9口诀乘法表。

1.程序分析：分行与列考虑，共9行9列，i控制行，j控制列。

出现重复的乘积（全矩形）

public class Demo16 {

public static void main(String[] args) {

for (int i = 1; i <= 9; i++) {

for (int j = 1; j <= 9; j++)

System.out.print(i + "\*" + j + "=" + (i\*j) + "\t");

System.out.println();

}

}

}

不现重复的乘积(下三角)

public class Demo16 {

public static void main(String[] args) {

for (int i = 1; i <= 9; i++) {

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

System.out.print(i + "\*" + j + "=" + (i\*j) + "\t");

System.out.println();

}

}

}

【程序17】 题目：猴子吃桃问题：猴子第一天摘下若干个桃子，当即吃了一半，还不瘾，又多吃了一个第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩

下的一半零一个。到第10天早上想再吃时，见只剩下一个桃子了。求第一天共摘了多少。

1.程序分析：采取逆向思维的方法，从后往前推断。

public class Demo17 {

public static void main(String[] args) {

int sum = 1;

for (int i = 0; i < 9; i++) {

sum = (sum + 1) \* 2;

}

System.out.println("第一天共摘"+sum);

}

}

【程序18】 题目：两个乒乓球队进行比赛，各出三人。甲队为a,b,c三人，乙队为x,y,z三人。已抽签决定比赛名单。有人向队员打听比赛的名单。a说他不和x比，c说他不和x,z比，请编程

序找出三队赛手的名单。

public class Demo18 {

static char[] m = { 'a', 'b', 'c' };

static char[] n = { 'x', 'y', 'z' };

public static void main(String[] args) {

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

for (int j = 0; j < n.length; j++) {

if (m[i] == 'a' && n[j] == 'x') {

continue;

} else if (m[i] == 'a' && n[j] == 'y') {

continue;

} else if ((m[i] == 'c' && n[j] == 'x')

|| (m[i] == 'c' && n[j] == 'z')) {

continue;

} else if ((m[i] == 'b' && n[j] == 'z')

|| (m[i] == 'b' && n[j] == 'y')) {

continue;

} else

System.out.println(m[i] + " vs " + n[j]);

}

}

}

}

或

public class Demo18 {

public String a, b, c;

public Demo18(String a, String b, String c) {

this.a = a;

this.b = b;

this.c = c;

}

public static void main(String[] args) {

Demo18 arr\_a = new Demo18("a", "b", "c");

String[] b = { "x", "y", "z" };

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

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

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

Demo18 arr\_b = new Demo18(b[i], b[j], b[k]);

if (!arr\_b.a.equals(arr\_b.b) & !arr\_b.b.equals(arr\_b.c)

& !arr\_b.c.equals(arr\_b.a) & !arr\_b.a.equals("x")

& !arr\_b.c.equals("x") & !arr\_b.c.equals("z")) {

System.out.println(arr\_a.a + "--" + arr\_b.a);

System.out.println(arr\_a.b + "--" + arr\_b.b);

System.out.println(arr\_a.c + "--" + arr\_b.c);

}

}

}

}

}

}

【程序19】 题目：打印出如下图案（菱形）

1.程序分析：先把图形分成两部分来看待，前四行一个规律，后三行一个规律，利用双重for循环，第一层控制行，第二层控制列。

三角形：

\*

\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*

\*

public class Demo19 {

public static void main(String[] args) {

int i=0;

int j=0;

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

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

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

System.out.println();

}

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

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

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

System.out.println();

}

}

}

菱形：

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

public class Demo19 {

public static void main(String[] args) {

int i = 0;

int j = 0;

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

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

System.out.print( " " );

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

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

System.out.println();

}

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

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

System.out.print( " " );

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

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

System.out.println();

}

}

}

【程序20】 题目：有一分数序列：2/1，3/2，5/3，8/5，13/8，21/13...求出这个数列的前20项之和。

1.程序分析：请抓住分子与分母的变化规律。

public class Demo20 {

public static void main(String[] args) {

float fm = 1.0f;

float fz = 1.0f;

float temp;

float sum = 0f;

for (int i = 0; i < 20; i++) {

temp = fm;

fm = fz;

fz = fz + temp;

System.out.println((int) fz + "/" + (int) fm);

sum += fz / fm;

}

System.out.println(sum);

}

}

【程序21】 题目：求1+2!+3!+...+20!的和。

1.程序分析：此程序只是把累加变成了累乘。

public class Demo21 {

public static void main(String[] args) {

long sum = 0;

long fac = 1;

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

fac = fac \* i;

sum += fac;

}

System.out.println(sum);

}

}

【程序22】 题目：利用递归方法求5!。

1.程序分析：递归公式：f(n)=f(n-1)\*4!

import java.util.Scanner;

public class Demo22 {

public static long fac(int n) {

long value = 0;

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

value = 1;

} else if (n > 1) {

value = n \* fac(n - 1);

}

return value;

}

public static void main(String[] args) {

System.out.println("请输入一个数：");

Scanner in = new Scanner(System.in);

int n = in.nextInt();

System.out.println(n + "的阶乘为：" + fac(n));

}

}

【程序23】 题目：有5个人坐在一起，问第五个人多少岁？他说比第4个人大2岁。问第4个人岁数，他说比第3个人大2岁。问第三个人，又说比第2人大两岁。问第2个人，说比第一个人大两

岁。最后问第一个人，他说是10岁。请问第五个人多大？

1.程序分析：利用递归的方法，递归分为回推和递推两个阶段。要想知道第五个人岁数，需知道第四人的岁数，依次类推，推到第一人（10岁），再往回推。

直接求解：

public class Demo23 {

public static void main(String[] args) {

int n = 10;

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

n = n + 2;

}

System.out.println("第五个人" + n + "岁");

}

}

递归求解：

public class Demo23 {

public static int getAge(int n) {

if (n == 1) {

return 10;

}

return 2 + getAge(n - 1);

}

public static void main(String[] args) {

System.out.println("第五个的年龄为" + getAge(5));

}

}

【程序24】 题目：给一个不多于5位的正整数，要求：一、求它是几位数，二、逆序打印出各位数字。

本题原方法：

import java.util.Scanner;

public class Demo24 {

public static void main(String[] args) {

Demo24 use = new Demo24();

System.out.println("请输入：");

Scanner in = new Scanner(System.in);

long a = in.nextLong();

if (a < 0 || a >= 100000) {

System.out.println("Error Input, please run this program Again!");

System.exit(0);

}

if (a >= 0 && a <= 9) {

System.out.println(a + "是一位数");

System.out.println("按逆序输出是:" + a);

} else if (a >= 10 && a <= 99) {

System.out.println(a + "是二位数");

System.out.println("按逆序输出是:");

use.converse(a);

} else if (a >= 100 && a <= 999) {

System.out.println(a + "是三位数");

System.out.println("按逆序输出是:");

use.converse(a);

} else if (a >= 1000 && a <= 9999) {

System.out.println(a + "是四位数");

System.out.println("按逆序输出是:");

use.converse(a);

} else if (a >= 10000 && a <= 99999) {

System.out.println(a + "是五位数");

System.out.println("按逆序输出是:");

use.converse(a);

}

}

public void converse(long l) {

String s = Long.toString(l);

char[] ch = s.toCharArray();

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

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

}

}

}

个人版方法：

import java.util.Scanner;

public class Demo24 {

public static void main(String[] args) {

System.out.println("请输入：");

Scanner in = new Scanner(System.in);

String str = in.next();

if (str.matches("\\d+")) { //正则表达式

System.out.println("输入的是" + str.length() + "位数");

StringBuffer buf = new StringBuffer(str);

System.out.println(buf.reverse());//字符串反转

}

}

}

【程序25】 题目：一个5位数，判断它是不是回文数。即12321是回文数，个位与万位相同，十位与千位相同。

原方法：

import java.util.Scanner;

public class Demo25 {

static int[] a = new int[5];

static int[] b = new int[5];

public static void main(String[] args) {

boolean is = false;

System.out.println("Please input：");

Scanner in = new Scanner(System.in);

long l = in.nextLong();

if (l > 99999 || l < 10000) {

System.out.println("Input error, please input again!");

l = in.nextLong();

}

for (int i = 4; i >= 0; i--) {

a[i] = (int) (l / (long) Math.pow(10, i));

l = (l % (long) Math.pow(10, i));

}

System.out.println();

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

b[j] = a[i];

}

for (int i = 0, j = 4; i < 5; i++, j--) {

if (a[i] != b[j]) {

is = false;

break;

} else {

is = true;

}

}

if (is == false) {

System.out.println("is not a Palindrom!");

} else if (is == true) {

System.out.println("is a Palindrom!");

}

}

}

个人版：

import java.util.Scanner;

public class Demo25 {

public static void main(String[] args) {

System.out.println("请输入：");

Scanner in = new Scanner(System.in);

String str = in.next();

int l = Integer.parseInt(str);//转换成整数

if (l < 10000 || l > 99999) {

System.out.println("输入错误！");

System.exit(0);

}

boolean is=false;

char[] ch = str.toCharArray();

for(int i=0;i<ch.length/2;i++){

if(ch[i]!=ch[ch.length-i-1]){

is=false;

}else{

is=true;

}

}

if(is){

System.out.println("这是一个回文!");

}else{

System.out.println("不是一个回文!");

}

}

}

【程序26】 题目：请输入星期几的第一个字母来判断一下是星期几，如果第一个字母一样，则继续判断第二个字母。

1.程序分析：用情况语句比较好，如果第一个字母一样，则判断用情况语句或if语句判断第二个字母。

import java.util.Scanner;

public class Demo26 {

public static void main(String[] args) {

char weekSecond;//保存第二字母

Scanner in = new Scanner(System.in);//接收用户输入

System.out.println("请输入星期的第一个字母：");

String letter = in.next();

if (letter.length() == 1) {//判断用户控制台输入字符串长度是否是一个字母

char weekFirst = letter.charAt(0);//取第一个字符

switch (weekFirst) {

case 'm':

case 'M':

System.out.println("星期一(Monday)");

break;

case 't':

case 'T':

System.out.print("由于星期二(Tuesday)与星期四(Thursday)均以字母T开头，故需输入第二个字母才能正确判断：");

letter = in.next();

if (letter.length() == 1) {

weekSecond = letter.charAt(0);

if (weekSecond == 'U' || weekSecond == 'u') {

System.out.println("星期二(Tuesday)");

break;

} else if (weekSecond == 'H' || weekSecond == 'h') {

System.out.println("星期四(Thursday)");

break;

} else {

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

break;

}

} else {

System.out.println("输入错误，只能输入一个字母，程序结束！");

break;

}

case 'w':

case 'W':

System.out.println("星期三(Wednesday)");

break;

case 'f':

case 'F':

System.out.println("星期五(Friday)");

break;

case 's':

case 'S':

System.out.print("由于星期六(Saturday)与星期日(Sunday)均以字母S开头，故需输入第二个字母才能正确判断：");

letter = in.next();

if (letter.length() == 1) {

weekSecond = letter.charAt(0);

if (weekSecond == 'A' || weekSecond == 'a') {

System.out.println("星期六(Saturday)");

break;

} else if (weekSecond == 'U' || weekSecond == 'u') {

System.out.println("星期日(Sunday)");

break;

} else {

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

break;

}

} else {

System.out.println("输入错误，只能输入一个字母，程序结束！");

break;

}

default:

System.out.println("输入错误，不能识别的星期值第一个字母，程序结束！");

break;

}

} else {

System.out.println("输入错误，只能输入一个字母，程序结束！");

}

}

}

【程序27】 题目：求100之内的素数

public class Demo27 {

public static void main(String args[]) {

int sum, i;

for (sum = 2; sum <= 100; sum++) {

for (i = 2; i <= sum / 2; i++) {

if (sum % i == 0)

break;

}

if (i > sum / 2)

System.out.println(sum + "是素数");

}

}

}

或

public class Demo27{

public static void main(String args[]){

int w=1;

for(int i=2;i<=100;i++){

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

w=i%j;

if(w==0)break;

}

if(w!=0)

System.out.println(i+"是素数");

}

}

}

【程序28】 题目：对10个数进行排序。

1.程序分析：可以利用选择法，即从后9个比较过程中，选择一个最小的与第一个元素交换，下次类推，即用第二个元素与后8个进行比较，并进行交换。

本例代码为生成随机10个数排序，并输入1个数，插入重排序输出：

import java.util.Arrays;

import java.util.Random;

import java.util.Scanner;

public class Demo28 {

public static void main(String[] args) {

int arr[] = new int[11];

Random r = new Random();

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

arr[i] = r.nextInt(100) + 1; //得到10个100以内的整数

}

Arrays.sort(arr);

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

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

}

System.out.print("\nPlease Input a int number:" );

Scanner in = new Scanner(System.in);

arr[10] = in.nextInt();

Arrays.sort(arr);

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

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

}

}

}

个人代码：

import java.util.Arrays;

import java.util.Scanner;

public class Demo28 {

public static void main(String[] args) {

System.out.println("请输入10个数：");

Scanner in = new Scanner(System.in);

int[] arr = new int[10];

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

arr[i] = in.nextInt();

}

System.out.println("原数组为：");

for (int x : arr) {//foreach遍历

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

}

Arrays.sort(arr);

System.out.println();

System.out.println("排序后为：");

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

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

}

}

}

【程序29】 题目：求一个3\*3矩阵主对角线元素之和。

1.程序分析：利用双重for循环控制输入二维数组，再将a[i][i]累加后输出。

public class Demo29 {

public static void main(String[] args) {

double sum = 0;

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

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

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

if (i == j)

sum = sum + array[i][j];

}

System.out.println(sum);

}

}

主负对角线：

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

for(j=0;j<n;j++)

{

if(i==j) sum1+=a[i][j];

if(i+j==n-1) sum2+=a[i][j];

}

【程序30】 题目：有一个已经排好序的数组。现输入一个数，要求按原来的规律将它插入数组中。

1.程序分析：首先判断此数是否大于最后一个数，然后再考虑插入中间的数的情况，插入后此元素之后的数，依次后移一个位置。

import java.util.Random;

public class Demo30 {

public static void main(String[] args) {

int temp = 0;

int arr[] = new int[12];

Random r = new Random();

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

arr[i] = r.nextInt(1000);

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

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

for (int i = 0; i <= 9; i++)

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

if (arr[i] > arr[k]) {

temp = arr[i];

arr[i] = arr[k];

arr[k] = temp;

}

System.out.println();

for (int k = 0; k <= 10; k++)

System.out.print(arr[k] + "\t");

arr[11] = r.nextInt(1000);

for (int k = 0; k <= 10; k++)

if (arr[k] > arr[11]) {

temp = arr[11];

for (int j = 11; j >= k + 1; j--)

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

arr[k] = temp;

}

System.out.println();

for (int k = 0; k <= 11; k++)

System.out.print(arr[k] + "\t");

}

}

【程序31】 题目：将一个数组逆序输出。

程序分析：用第一个与最后一个交换。

用逆序循环控制变量输出：

public class Demo31 {

public static void main(String[] args) {

int[] a = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 };

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

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

}

}

}

【程序32】 题目：取一个整数a从右端开始的第4～7位数字。

import java.util.\*;

public class Demo32 {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.print("请输入一个7位以上的正整数：");

long l = in.nextLong();

String str = Long.toString(l);

char[] ch = str.toCharArray();

int j=ch.length;

if (j<7){System.out.println("输入错误！");

} else {

System.out.println("截取从右端开始的4～7位是："+ch[j-7]+ch[j-6]+ch[j-5]+ch[j-4]);

}

}

}

或

import java.util.Scanner;

public class Demo32{

public static void main(String[] args) {

int a = 0;

Scanner s = new Scanner(System.in);

long b = s.nextLong();

a = (int) (b % 10000000 / 1000);

System.out.println(a);

}

}

【程序33】 题目：打印出杨辉三角形（要求打印出10行如下图）

1.程序分析：

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

public class Demo33 {

public static void main(String args[]) {

int i, j;

int a[][];

int n = 10;

a = new int[n][n];

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

a[i][i] = 1;

a[i][0] = 1;

}

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

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

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

}

}

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

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

System.out.printf(a[i][j] + "\t");

}

System.out.println();

}

}

}

【程序34】 题目：输入3个数a,b,c，按大小顺序输出。

（也可互相比较交换排序）

import java.util.Arrays;

public class Demo34 {

public static void main(String[] args) {

int[] arrays = { 800, 56, 500 };

Arrays.sort(arrays);

for (int n = 0; n < arrays.length; n++)

System.out.println(arrays[n]);

}

}

或

if(x > y) { int t = x; x = y; y = t; } if(x > z) { int t = x; x = z; z = t; } if(y > z) { int t = y; y = z; z = t; }

【程序35】 题目：输入数组，最大的与第一个元素交换，最小的与最后一个元素交换，输出数组。

import java.util.\*;

public class Demo35 {

public static void main(String[] args) {

int i, min=0, max=0, n, temp1, temp2;

int a[];

System.out.println("定义数组的长度:");

Scanner in = new Scanner(System.in);

n = in.nextInt();

a = new int[n];

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

System.out.print("输入第" + (i + 1) + "个数据:");

a[i] = in.nextInt();

}

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

if (a[i] > a[max])

max = i;

if (a[i] < a[min])

min = i;

}

temp1 = a[0];

a[0] = a[max];

a[max] = temp1;

temp2 = a[min];

if (min != 0) { // 如果最小值不是a[0]，执行下面

a[min] = a[n - 1];

a[n - 1] = temp2;

} else { //如果最小值是a[0],执行下面

a[max] = a[n - 1];

a[n - 1] = temp1;

}

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

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

}

}

}

【程序36】 题目：有n个整数，使其前面各数顺序向后移m个位置，最后m个数变成最前面的m个数

import java.util.LinkedList;

import java.util.List;

import java.util.Scanner;

public class Demo36 {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("输入数字个数n：");

int n = in.nextInt();

System.out.println("输入后移位数m：");

int m = in.nextInt();

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

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

System.out.println("请输入第"+(i+1)+"个数:");

list.add(in.nextInt());

}

System.out.println("原数据排序为：");

for (int t : list) {

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

}

System.out.println();

List<Integer> temp1 = list.subList(list.size() - m, list.size());

List<Integer> temp2 = list.subList(0, list.size() - m);

temp2.addAll(0, temp1);

System.out.println("移动后排序为;");

for (int t : temp2) {

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

}

}

}

或

import java.util.\*;

public class Demo36{

public static void main(String[] args){

Scanner in=new Scanner(System.in);

System.out.println("请定义数组的长度：");

int n=in.nextInt();

System.out.println("请输入移动的位数：");

int m=in.nextInt();

int [] arr=new int [n];

int [] brr=new int [n];

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

System.out.println("请输入第"+(i+1)+"个数：");

arr[i]=in.nextInt();

}

System.out.println("排序前：");

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

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

}

System.out.println();

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

brr[i]=arr[n-m+i];

}

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

arr[m+i]=arr[i];

}

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

arr[i]=brr[i];

}

System.out.println("排序后：");

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

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

}

}

}

【程序37】 题目：有n个人围成一圈，顺序排号。从第一个人开始报数（从1到3报数），凡报到3的人退出圈子，问最后留下的是原来第几号的那位。

（约瑟夫环问题，百度百科有时间复杂度最简单的数学方法）

原例代码：

import java.util.Scanner;

public class Demo37 {

public static void main(String[] args) {

System.out.println("请输人数n：");

Scanner in = new Scanner(System.in);

int n = in.nextInt();

boolean[] arr = new boolean[n];

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

arr[i] = true; //下标为TRUE时说明还在圈里

}

int leftCount = n;

int countNum = 0;

int index = 0;

while (leftCount > 1) {

if (arr[index] == true) { //当在圈里时

countNum++; //报数递加

if (countNum == 3) { //报数为3时

countNum = 0; //从零开始继续报数

arr[index] = false; //此人退出圈子

leftCount--; //剩余人数减一

}

}

index++; //每报一次数，下标加一

if (index == n) { //是循环数数，当下标大于n时，说明已经数了一圈，

index = 0; //将下标设为零重新开始。

}

}

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

if (arr[i] == true) {

System.out.println(i);

}

}

}

}

个人代码1：

import java.util.Scanner;

public class Demo37 {

public static void main(String[] args) {

System.out.println("请输入人数：");

Scanner in = new Scanner(System.in);

int[] a = new int[in.nextInt()];

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

a[i] = 1;

}

int left = a.length;

int j = 0;

int num = 0;

while (left > 1) {

if (a[j] == 1) {

num++;

}

if (num == 3) {

a[j] = 0;

num = 0;

left--;

}

j++;

if (j == a.length) {

j = 0;

}

}

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

if (a[i] == 1) {

System.out.println("最后留下的人是"+ (i + 1) + "号");

break;

}

}

}

}

个人代码2：

import java.util.LinkedList;

import java.util.Scanner;

public class Demo37 {

public static void main(String[] args) {

LinkedList<Integer> l = new LinkedList<Integer>();

System.out.println("请输入人数：");

Scanner in = new Scanner(System.in);

int len = in.nextInt();

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

l.add(i + 1);

}

int sum = 0;

int temp = 0;

for (int i = 0; sum != len - 1;) {

if (l.get(i) != 0) {

temp++;

}

if (temp == 3) {

l.remove(i);

l.add(i, 0);

temp = 0;

sum++;

}

i++;

if (i == l.size()) {

i = 0;

}

}

for (int t : l) {

if (t != 0) {

System.out.println("最后留下的人是" + t + "号");

}

}

}

}

【程序38】 题目:写一个函数，求一个字符串的长度，在main函数中输入字符串，并输出其长度。

import java.util.Scanner;

public class Demo38 {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("请输入一个字符串：");

String mys = in.next();

System.out.println(str\_len(mys));

}

public static int str\_len(String x) {

return x.length();

}

}

或

import java.util.Scanner;

public class Demo38 {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("请输入一个字符串：");

String mys = in.next();

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

}

}

【程序39】 题目：编写一个函数，输入n为偶数时，调用函数求1/2+1/4+...+1/n,当输入n为奇数时，调用函数1/1+1/3+...+1/n

import java.util.Scanner;

public class Demo39 {

public static double ouShu(int n) {

double result = 0;

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

result += 1 / (double) i;

}

return result;

}

public static double jiShu(int n) {

double result = 0;

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

result += 1 / (double) i;

}

return result;

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("输入n的值：");

int n = in.nextInt();

if (n % 2 == 0) { //偶数，1/2+1/4+...+1/n

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

} else { //奇数，1/1+1/3+...+1/n

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

}

}

}

【程序40】 题目：字符串排序。

（利用容器类中的sort方法）

import java.util.\*;

public class Demo40 {

public static void main(String[] args) {

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

list.add("010102");

list.add("010003");

list.add("010201");

Collections.sort(list);

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

System.out.println(list.get(i));

}

}

}

或

import java.util.\*;

public class Demo40 {

public static void main(String[] args){

Scanner in=new Scanner(System.in);

System.out.println("请定义字符串的个数：");

int n=in.nextInt();

String[] str=new String[n];

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

System.out.println("请输入第"+(i+1)+"字符串：");

str[i]=in.next();

}

strSort(n,str);

System.out.println("字符串排序后：");

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

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

}

}

public static void strSort(int n,String[] arr){

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

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

if(compare(arr[i], arr[j]) == false) {

String temp = arr[i]; arr[i] = arr[j]; arr[j] = temp;

}

}

}

}

static boolean compare(String s1, String s2) {

boolean result = true;

for(int i=0; i<s1.length() && i<s2.length(); i++) {

if(s1.charAt(i) > s2.charAt(i)) {

result = false;

break;

} else if(s1.charAt(i) <s2.charAt(i)) {

result = true;

break;

} else {

if(s1.length() < s2.length()) {

result = true;

} else {

result = false;

}

}

}

return result;

}

}

【程序41】 题目：海滩上有一堆桃子，五只猴子来分。第一只猴子把这堆桃子平均分为五份，多了一个，这只猴子把多的一个扔入海中，拿走了一份。第二只猴子把剩下的桃子又平均分成五份，又多了一个，它同样把多的一个扔入海中，拿走了一份，第三、第四、第五只猴子都是这样做的，问海滩上原来最少有多少个桃子？

本题源码：

public class Demo41 {

static int ts = 0;// 桃子总数

int fs = 1;// 记录分的次数

static int hs = 5;// 猴子数

int tsscope = 5000;// 桃子数的取值范围，太大容易溢出。

public int fT(int t) {

if (t == tsscope) {

// 当桃子数到了最大的取值范围时取消递归

System.out.println("结束");

return 0;

} else {

if ((t - 1) % hs == 0 && fs <= hs) {

if (fs == hs) {

System.out.println("桃子数=" + ts + "时满足分桃条件");

}

fs += 1;

return fT((t - 1) / 5 \* 4);// 返回猴子拿走一份后的剩下的总数

} else {

// 没满足条件

fs = 1;// 分的次数重置为1

return fT(ts += 1);// 桃子数加+1

}

}

}

public static void main(String[] args) {

new Demo41().fT(0);

}

}

个人修改：

public class Demo41 {

public static void main(String[] args) {

int sum = 0;

for (int i = 6;; i++) {// 最少6个分最后一次

sum = i;// 桃子数

for (int j = 0; j < 5; j++) {// 分的次数循环

if ((sum - 1) % 5 == 0 && j < 5) {// 如果扔一个后能均分5份，继续分

sum = (sum - 1) / 5 \* 4;// 每分一次剩余桃子数

if (j == 4) {// 如果已分5次，且仍能除尽，输出，退出程序

System.out.println(i);

System.exit(0);

}

}

}

}

}

}

【程序42】 题目：809\*??=800\*??+9\*??+1。其中??代表的两位数,8\*??的结果为两位数，9\*??的结果为3位数。求??代表的两位数，及809\*??后的结果。

（本题为无解，去掉1有解）

public class Demo42 {

public static void main(String[] args) {

for (int i = 10; i < 100; i++) {

if (809 \* i == (800 \* i + 9 \* i + 1) && 8 \* i >= 10 && 8 \* i < 100

&& 9 \* i >= 100 && 9 \* i < 1000) {

System.out.println("?? =" + i);

System.out.println("809\*??="+ 809 \* i);

System.exit(0);

}

}

}

}

【程序43】 题目：求0—7所能组成的奇数个数。

暴力算法：

public class Demo43 {

public static boolean isJiShu(int n) {

if (n % 2 != 0) {

return true;

} else {

return false;

}

}

public static boolean fun(char c) {

if (c >= '0' && c <= '7') {

return true;

} else {

return false;

}

}

public static void main(String[] args) {

int count = 0;

String s;

for (int i = 0; i < 100000000; i++) {

s = "" + i;

boolean flag = true;

char[] c = s.toCharArray();

for (int j = 0; j < c.length; j++) {

if (!fun(c[j])) {

flag = false;

break;

}

}

if (flag && isJiShu(i)) {

count++;

}

s = "";

}

System.out.println("共" + count + "个。");

}

}

数学算法：

public class Demo43 {

public static void main(String[] args) {

// 因为是奇数，所以个位只能是1，3，5，7共4种，前面可随便排列

int count = 4;// 个位的4种

// 2位时，十位有8种，个位4种，8×4

// 3位时，8×8×4……

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

count = 8 \* count;

System.out.println("count:" + count);

}

}

}

个人算法：

//组成1位数是4个。

//组成2位数是7\*4个。

//组成3位数是7\*8\*4个。

//组成4位数是7\*8\*8\*4个。

//......

public class Demo43 {

public static void main (String[] args) {

int sum=4;

int j;

System.out.println("组成1位数是 "+sum+" 个");

sum=sum\*7;

System.out.println("组成2位数是 "+sum+" 个");

for(j=3;j<=9;j++){

sum=sum\*8;

System.out.println("组成"+j+"位数是 "+sum+" 个");

}

}

}

【程序44】 题目：一个偶数总能表示为两个素数之和。（注：哥德巴赫猜想是想证明对任何大于6的自然数n之内的所有偶数可以表示为两个素数之和）

public class Demo44 {

public static boolean isSuShu(int x) {

if (x == 0 || x == 1) {

return false;

}

for (int i = 2; i <= Math.sqrt(x); i++) {

if (x % i == 0) {

return false;

}

}

return true;

}

public static void main(String[] args) {

// 求了下100以内的情况

for (int i = 0; i < 100; i = i + 2) {

for (int j = 0; j <= (i + 1) / 2; j++) {

if (isSuShu(j) && isSuShu(i - j)) {

System.out.println(i + "=" + j + "+" + (i - j));

}

}

}

}

}

或

public class Demo44{

public static void main(String[] args){

for (int i=6;i<=100 ;i+=2 ){

for (int j=2;j<100 ;j++ ){

if(!isPrime(j)||!isPrime(i-j)||j>=i)

continue;

System.out.println(i+"="+j+"+"+(i-j));

break;

}

}

}

public static boolean isPrime(int n){

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

if(n%i==0)return false;

}

return true;

}

}

【程序45】 题目：（1）判断几个9能被一个素数整除。（2）判断一个整数能被几个9整除。（原题：一个素数能被几个9整除）

（一）

public class Demo45 {

public static boolean isSuShu(int x) {

if (x == 0 || x == 1) {

return false;

}

for (int i = 2; i <= Math.sqrt(x); i++) {

if (x % i == 0) {

return false;

}

}

return true;

}

public static void main(String[] args) {

int[] a = new int[100];

int n = 0;

int num = 0;

// 长度100的素数数组

while (n < 100) {

if (isSuShu(num)) {

a[n] = num;

n++;

num++;

} else {

num++;

}

}

/\* for (int t : a) {

System.out.println(t);

}\*/

String s = "9";

int index = 0;

while (s.length() < 9) {

if (new Integer(s).intValue() % a[index] == 0) {

System.out.println(s + "%" + a[index] + "=0");

if (index < 100 - 1) {

index++;

} else {

index = 0;

s = s + "9";

}

// System.exit(0);

} else {

if (index < 100 - 1) {

index++;

} else {

index = 0;

s = s + "9";

}

}

}

}

}

（二）

import java.util.\*; public class Demo45 {

public static void main (String[] args) {

Scanner in = new Scanner(System.in);

System.out.print("请输入一个整数：");

int num = in.nextInt();

int tmp = num;

int count = 0;

for(int i = 0 ; tmp%9 == 0 ;){

tmp = tmp/9;

count ++;

}

System.out.println(num+" 能够被 "+count+" 个9 整除。");

}

}

【程序46】 题目：两个字符串连接程序。

import java.util.Scanner;

public class Demo46 {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("输入第一个字符串：");

String s1 = in.next();

System.out.println("输入第一个字符串：");

String s2 = in.next();

System.out.println("连接后：\n" + s1 + s2);

}

}

或

import java.util.\*;

public class Demo46 {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.print("请输入一个字符串：");

String str1 = in.nextLine();

System.out.print("请再输入一个字符串：");

String str2 = in.nextLine();

String str = str1+str2;

System.out.println("连接后的字符串是："+str);

}

}

【程序47】 题目：读取7个数（1—50）的整数值，每读取一个值，程序打印出该值个数的。

import java.util.\*;

public class Demo47 {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n=1,num;

while(n<=7){

do{

System.out.print("请输入一个1--50 之间的整数：");

num= s.nextInt();

}while(num<1||num>50);

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

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

}

System.out.println();

n ++;

}

}

}

或

import java.util.Scanner;

public class Demo47 {

public static void print(int n) {

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

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

}

System.out.println();

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

for (int i = 0; i < 7; i++) {

int temp = in.nextInt();

print(temp);

}

}

}

【程序48】 题目：某个公司采用公用电话传递数据，数据是四位的整数，在传递过程中是加密的，加密规则如下：每位数字都加上5，然后用和除以10的余数代替该数字，再将第一位和第四位交换，第二位和第三位交换。

import java.util.Scanner;

public class Demo48{

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("请输入一个4位数字：");

String str = in.next();

if (!((str).matches("\\d{4}"))) {

System.out.println("输入的不是4位数字！");

System.exit(0);

}

char[] c = str.toCharArray();

int[] a = new int[4];

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

a[i] = ((int) (c[i] - '0') + 5) % 10;

}

int t;

t = a[0];

a[0] = a[3];

a[3] = t;

t = a[1];

a[1] = a[2];

a[2] = t;

System.out.println("结果是：" + a[0] + a[1] + a[2] + a[3]);

}

}

或

import java.util.\*;

public class Demo48 {

public static void main(String args[]) {

Scanner s = new Scanner(System.in);

int num=0,temp;

do{

System.out.print("请输入一个4位正整数：");

num = s.nextInt();

}while (num<1000||num>9999);

int a[]=new int[4];

a[0] = num/1000; //取千位的数字

a[1] = (num/100)%10; //取百位的数字

a[2] = (num/10)%10; //取十位的数字

a[3] = num%10; //取个位的数字

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

a[j]+=5; a[j]%=10;

}

for(int j=0;j<=1;j++) {

temp = a[j]; a[j] = a[3-j]; a[3-j] =temp;

}

System.out.print("加密后的数字为：");

for(int j=0;j<4;j++) System.out.print(a[j]);

}

}

【程序49】 题目：计算字符串中子串出现的次数。

import java.util.Scanner;

public class Demo49 {

public static void main(String[] args) {

Scanner in=new Scanner(System.in);

System.out.println("请输入主串：");

String str1 = in.nextLine();

System.out.println("请输入子串：");

String str2 = in.nextLine();

// 生成子串长度的N个字符串数组

String[] sa = new String[str1.length() - str2.length() + 1];

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

sa[i] = str1.substring(i, i + str2.length());

}

int sum = 0;

// 子串与N个拆开的子串比对

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

if (sa[i].equals(str2)) {

// 成功配对，计数器+1；

sum++;

// 因为不计算重叠的子串，所以跳过配对之后的部分拆分子串

i = i + str2.length();

}

}

System.out.println("主串中共包含" + sum + "个字串");

}

}

【程序50】 题目：有五个学生，每个学生有3门课的成绩，从键盘输入以上数据（包括学生号，姓名，三门课成绩），计算出平均成绩，把原有的数据和计算出的平均分数存放在磁盘文import java.io.File;

import java.io.FileWriter;

import java.util.Scanner;

class Student {

private int number = 0;

private String name = "";

private double[] a = new double[3];

public double getAve() {

return (a[0] + a[1] + a[2]) / 3;

}

public Student(int number, String name, double[] a) {

super();

this.number = number;

this.name = name;

this.a = a;

}

@Override

public String toString() {

return "学号：" + this.number + "\t姓名：" + this.name + "\r\n各科成绩：\r\n" + a[0] + "\t" + a[1] + "\t" + a[2] + "\r\n平均成绩：\r\n"

+ this.getAve();

}

}

public class Demo50 {

public static Student input() {

Scanner s = new Scanner(System.in);

System.out.println("请输入学号：");

int num = s.nextInt();

System.out.println("请输入姓名：");

String name = s.next();

System.out.println("请分别输入3门成绩：");

double[] a = new double[3];

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

a[i] = s.nextDouble();

}

return new Student(num, name, a);

}

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

Student[] st = new Student[2];

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

st[i] = input();

}

File f = new File("d:" + File.separator + "123.txt");

FileWriter output = new FileWriter(f);

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

output.write(st[i].toString() + "\r\n");

output.write("\r\n");

}

output.close();

}

}

【程序51】 Excel创建、解析、编辑

package com.umt.czj;

public class Book

{

private Integer id;

private String name;

private Double price;

private String authorName;

public String getAuthorName()

{

return authorName;

}

public void setAuthorName(String authorName)

{

this.authorName = authorName;

}

public Integer getId()

{

return id;

}

public void setId(Integer id)

{

this.id = id;

}

public Book(Integer id, String name, Double price, String authorName)

{

this.id = id;

this.name = name;

this.price = price;

this.authorName = authorName;

}

public Book()

{

}

public String getName()

{

return name;

}

public void setName(String name)

{

this.name = name;

}

public Double getPrice()

{

return price;

}

public void setPrice(Double price)

{

this.price = price;

}

}

package com.umt.czj;

import java.io.File;

import java.util.ArrayList;

import java.util.List;

import jxl.write.Number;

import jxl.Sheet;

import jxl.Workbook;

import jxl.write.Label;

import jxl.write.WritableSheet;

import jxl.write.WritableWorkbook;

/\*\*

\* 利用java技术对Excel文件进行创建、编辑、解析

\* @author caozhijun

\* 1.创建一个新的excel文件

\* 2.创建之后解析循环输出里面的内容

\* 3.修改文件的内容

\*/

public class DoExcel

{

public static void main(String[] args)

{

//创建新的excel文件

DoExcel excel = new DoExcel();

List<Book> data = new ArrayList<Book>();

Book book1 = new Book(455, "JAVA se", 103.4, "czj");

Book book2 = new Book(4009, "Ajax", 66.0, "james");

Book book3 = new Book(3, "三国志", 77.4, "科比");

Book book4 = new Book(44, "水浒传", 93.4, "韦德");

Book book5 = new Book(765, "中国交通地图册", 67.4, "麦蒂");

data.add(book1);

data.add(book2);

data.add(book3);

data.add(book4);

data.add(book5);

excel.writeExcelFile(data);

System.out.println("Write file successfully！");

//输出解析后的文件内容

List<Book> list = excel.parseExcelFile();

for (Book book : list)

{

System.out.println(book.getId() + ";" + book.getName() + ";"

+ book.getPrice() + ";" + book.getAuthorName());

}

//编辑文件

excel.editExcelFile();

}

/\*\*

\* 读取文件

\* @return

\*/

public List<Book> parseExcelFile()

{

try

{

List<Book> books = new ArrayList<Book>();

//创建要读取文件对象

Workbook book = Workbook.getWorkbook(new File("f://book.xls"));

//获得第一页

Sheet sheet1 = book.getSheet(0);

for (int j = 1; j < sheet1.getRows(); j++)

{

Book bookTemp = new Book();

//对象属性设值

for (int i = 0; i < sheet1.getColumns(); i++)

{

if (0 == i)

{

//sheet的getCell(i,j)方法获得第i+1列，第j+1行的单元格

bookTemp.setId(new Integer(sheet1.getCell(i, j)

.getContents()));

}

else if (1 == i)

{

bookTemp.setName(sheet1.getCell(i, j).getContents());

}

else if (2 == i)

{

bookTemp.setPrice(new Double(sheet1.getCell(i, j)

.getContents()));

}

else

{

bookTemp.setAuthorName(sheet1.getCell(i, j)

.getContents());

}

}

//把对象加入到集合里面

books.add(bookTemp);

}

return books;

}

catch (Exception e)

{

e.printStackTrace();

}

return null;

}

/\*\*

\* 编辑文件

\*/

public void editExcelFile()

{

try

{

//获得要编辑的文件

Workbook book2 = Workbook.getWorkbook(new File("f://book.xls"));

//利用book2得到该文件的副本

WritableWorkbook bookWri = Workbook.createWorkbook(new File(

"f://book.xls"), book2);

WritableSheet sheet1 = bookWri.getSheet(0);

//此项操作将会覆盖单元格{0，0}的内容，完成编辑的功能

sheet1.addCell(new Label(0, 0, "编号22"));

//在原有基础上增加第三页

WritableSheet sheet2 = bookWri.createSheet("第三页", 2);

sheet2.addCell(new Label(0, 0, "Wade"));

bookWri.write();

bookWri.close();

}

catch (Exception e)

{

e.printStackTrace();

}

}

/\*\*

\* 写入文件

\* @param list

\*/

public void writeExcelFile(List<Book> list)

{

try

{

//获得可写的文件对象，

WritableWorkbook excel = Workbook.createWorkbook(new File(

"f://book.xls"));

//创建第一页，名字自由设置

WritableSheet sheet1 = excel.createSheet("购书清单", 0);

//设置第一行的第一、第二、第三列单元格的内容

sheet1.addCell(new Label(0, 0, "编号"));

sheet1.addCell(new Label(1, 0, "书名"));

sheet1.addCell(new Label(2, 0, "价格"));

sheet1.addCell(new Label(3, 0, "作者"));

//把集合中的book对象迭代到文件中

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

{

Book book = list.get(i);

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

{

if (0 == j)

{

//new Number(i,j,44.4)数字类型的表达方式

sheet1.addCell(new Number(j, i + 1, book.getId()));

}

else if (1 == j)

{

//new Label(i,j,"ll")字符串类型的表达方式

sheet1.addCell(new Label(j, i + 1, book.getName()));

}

else if (2 == j)

{

sheet1.addCell(new Number(j, i + 1, book.getPrice()));

}

else

{

sheet1.addCell(new Label(j, i + 1, book.getAuthorName()));

}

}

}

WritableSheet sheet2 = excel.createSheet("第二页", 1);

sheet2.addCell(new Label(0, 0, "McGrady"));

//写入操作

excel.write();

excel.close();

}

catch (Exception e)

{

e.printStackTrace();

}

}

}

【程序52】 java操作zip、jar文件

package com.umt.czj;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.util.ArrayList;

import java.util.List;

import java.util.jar.JarOutputStream;

import java.util.zip.ZipEntry;

import java.util.zip.ZipOutputStream;

public class DoZipAndJar

{

public static void main(String[] args)

{

DoZipAndJar doZip = new DoZipAndJar();

//把f://book1.xls、f://book2.xls文件压缩到f://java/zip 中的book.zip

try

{

List<File> sourceFileZip = new ArrayList<File>();

File sourceFileZip1 = new File("f://book.xls");

File sourceFileZip2 = new File("f://book2.xls");

sourceFileZip.add(sourceFileZip1);

sourceFileZip.add(sourceFileZip2);

File resultFileZip = new File("f://java/zip/Book.zip");

//如果目录不存在则创建新的目录

if (!new File("f://java/zip").exists())

{

new File("f://java/zip").mkdirs();

}

doZip.createZipFile(sourceFileZip, resultFileZip);

}

catch (Exception ex)

{

ex.printStackTrace();

System.out.println("Sorry! create zip file failed!");

}

System.out.println("Congratulations! create zip file successfully!");

//把f://DoExcel.class、f://bus.service.xml文件打包到f://java/jar 中的bus\_cc\_r3.jar

try

{

List<File> sourceFileJar = new ArrayList<File>();

File sourceFileJar1 = new File("f://DoExcel.class");

File sourceFileJar2 = new File("f://bus.service.xml");

sourceFileJar.add(sourceFileJar1);

sourceFileJar.add(sourceFileJar2);

File resultFileJar = new File("f://java/jar/bus\_cc\_r3.jar");

if (!new File("f://java/jar").exists())

{

new File("f://java/jar").mkdirs();

}

doZip.createJarFile(sourceFileJar, resultFileJar);

}

catch (Exception ex)

{

ex.printStackTrace();

System.out.println("Sorry! create jar file failed!");

}

System.out.println("Congratulations! create jar file successfully!");

}

/\*\*

\* 把f://book.xls文件压缩到f://java/zip 中的book.zip

\* @param sourceFile

\* @param resultFile

\*/

public void createZipFile(List<File> sourceFileList, File resultFile)

{

FileInputStream in = null;

try

{

//实例化Zip文件输出流

ZipOutputStream zipOut = new ZipOutputStream(new FileOutputStream(

resultFile));

//实例化所要压缩文件集合输入流

for (File sourceFile : sourceFileList)

{

in = new FileInputStream(sourceFile);

//Zip文件输出流装入zip条目

zipOut.putNextEntry(new ZipEntry(sourceFile.getName()));

int i = 0;

//压缩文件到指定目录

while ((i = in.read()) != -1)

{

zipOut.write(i);

}

}

in.close();

zipOut.close();

}

catch (Exception e)

{

e.printStackTrace();

}

}

/\*\*

\* 把f://DoExcel.class文件压缩到f://java/jar 中的book.jar

\* @param sourceFile

\* @param resultFile

\*/

public void createJarFile(List<File> sourceFileList, File resultFile)

{

FileInputStream in = null;

try

{

//实例化jar文件输出流

JarOutputStream jarOut = new JarOutputStream(new FileOutputStream(

resultFile));

for (File sourceFile : sourceFileList)

{

//实例化所要压缩文件集合输入流

in = new FileInputStream(sourceFile);

//jar文件输出流装入jar条目

jarOut.putNextEntry(new ZipEntry(sourceFile.getName()));

int i = 0;

//把文件放到到指定目录

while ((i = in.read()) != -1)

{

jarOut.write(i);

}

}

in.close();

jarOut.close();

}

catch (Exception e)

{

e.printStackTrace();

}

}

}

【程序53】 java发送手机短信

JAVA发送手机短信,流传有几种方法:

(1)使用webservice接口发送手机短信,这个可以使用sina提供的webservice进行发送,但是需要进行注册;

(2)使用短信mao的方式进行短信的发送,这种方式应该是比较的常用,前提是需要购买硬件设备,呵呵

(3)使用中国网建提供的SMS短信平台（申请账号地址：http://sms.webchinese.cn/default.shtml）

package com.text;

import org.apache.commons.httpclient.Header;

import org.apache.commons.httpclient.HttpClient;

import org.apache.commons.httpclient.NameValuePair;

import org.apache.commons.httpclient.methods.PostMethod;

public class SendMsg\_webchinese {

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

HttpClient client = new HttpClient();

PostMethod post = new PostMethod("http://sms.webchinese.cn/web\_api/");

post.addRequestHeader("Content-Type",

"application/x-www-form-urlencoded;charset=gbk");// 在头文件中设置转码

NameValuePair[] data = { new NameValuePair("Uid", "cshxxxxxxxx"), // 注册的用户名

new NameValuePair("Key", "53295058d1c46710666a"),

// 注册成功后,登录网站使用的密钥，这个密钥要登录到国建网然后有一个API接口，点进去就有一个key，可以改，那个才是密钥

new NameValuePair("smsMob", "187xxxxxxx"), // 手机号码

new NameValuePair("smsText", "通知") };//设置短信内容

post.setRequestBody(data);

client.executeMethod(post);

Header[] headers = post.getResponseHeaders();

int statusCode = post.getStatusCode();

System.out.println("statusCode:" + statusCode);

for (Header h : headers) {

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

}

String result = new String(post.getResponseBodyAsString().getBytes(

"gbk"));

System.out.println(result);

post.releaseConnection();

}

运行本程序首先的代入三个jar包：

commons-codec-1.4

commons-httpclient-3.1

commons-logging-1.1.1

GBK编码发送接口地址：

http://gbk.sms.webchinese.cn/?Uid=本站用户名&Key=接口安全密码&smsMob=手机号码&smsText=短信内容

UTF-8编码发送接口地址：

http://utf8.sms.webchinese.cn/?Uid=本站用户名&Key=接口安全密码&smsMob=手机号码&smsText=短信内容

获取短信数量接口地址(UTF8)：

http://sms.webchinese.cn/web\_api/SMS/?Action=SMS\_Num&Uid=本站用户名&Key=接口安全密码

获取短信数量接口地址(GBK)：

http://sms.webchinese.cn/web\_api/SMS/GBK/?Action=SMS\_Num&Uid=本站用户名&Key=接口安全密码

短信发送后返回值 说　明

-1 没有该用户账户

-2 密钥不正确（不是用户密码）

-3 短信数量不足

-11 该用户被禁用

-14 短信内容出现非法字符

-41 手机号码为空

-42 短信内容为空

大于0 短信发送数量

【程序54】 String方法集合

package com.umt.czj;

public class StringTest

{

/\*\*

\* String 的最常见方法集合

\* 也可以叫做String API大全

\*/

public static void main(String[] args)

{

String str = "wade399|rrrr";

//返回索引为2的char值

char charAt = str.charAt(2);

System.out.println(charAt);

//返回索引为2的Unicode

int codePointAt = str.codePointAt(2);

System.out.println(codePointAt);

/\*

\* 与另一个字符窜按照字典顺序比较

\* 等于0：两个字符串完全相等

\* 大于0：前面的在后面的之前

\* 小于0：前面的在后面的之后

\*/

int compareTo = str.compareTo("wade399");

System.out.println(compareTo);

System.out.println(str.compareTo("Wade399"));

System.out.println(str.compareTo("3ade399"));

System.out.println(str.compareTo("yade399"));

System.out.println(str.compareToIgnoreCase("WADE399"));

//在字符串结尾处加上字符串

System.out.println(str.concat("10"));

//与一个StringBuffer比较是否相等

System.out.println(str.contentEquals(new StringBuffer("waDe399")));

//测试该字符串是否以"64"结尾

System.out.println(str.endsWith("64"));

//测试该字符串是否与另一字符串内容相等

System.out.println(str.equals("wade388"));

//测试该字符串是否与另一字符串地址相等

System.out.println(str=="wade399");

//不计较大小写，是否与另一字符串内容相等

System.out.println(str.equalsIgnoreCase("WADE399"));

//使用默认字符集将此字符串编码为byte序列并将结果存储到一个新的byte数组中

byte getBytes[] = str.getBytes();

System.out.println(getBytes[0]+";"+getBytes[1]+";"+getBytes[2]);

//返回次字符串的哈希吗

int hashCode = str.hashCode();

System.out.println(hashCode);

/\*

\* 返回指定字符在此字符或字符串中第一次出现的索引，没有出现为-1

\* str.indexOf("r",1)表示从指定的索引开始收索

\*/

int indexOf = str.indexOf("r");

System.out.println(indexOf);

System.out.println(str.indexOf('w'));

System.out.println(str.indexOf("r",1));

//返回长度

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

//此字符串是否匹配该正则表达式，正则表达式不必出现/^$/等字样

System.out.println(str.matches("[url=file://\\d]\\d[/url]+"));

/\*

\* 通过用newChar替换此字符串中出现的所有oldChar得到的新字符串

\* replaceFirst表示替换首次出现的旧字符串，并非所有的

\*/

System.out.println(str.replace("399", "444"));

System.out.println(str.replace("9", "8"));

System.out.println(str.replaceFirst("9", "8"));

/\*

\* 根据给定正则表达式的匹配拆分此字符串

\* split("[url=file://\\+|\\-|\\\*|\\/]\\+|\\-|\\\*|\\/[/url]"):按照"+"或"-"或"\*"或"/"把字符串切成块状

\*/

String []split = str.split("\\|");

System.out.println(split[0]+";"+split[1]);

System.out.println(str.split("[url=file://\\+|\\-|\\\*|\\/]\\+|\\-|\\\*|\\/[/url]"));

//测试该字符串是否以"r"开始

System.out.println(str.startsWith("r"));

/\*

\* 按照规则截取字符串

\* substring(1)：从索引为1处开始到结束

\* substring(1,3)：从索引为1处到3处组成的新字符串

\*/

System.out.println(str.substring(1));

System.out.println(str.substring(1,3));

//返回char类型的数组

char toCharArray[] = str.toCharArray();

System.out.println(toCharArray);

//返回String形式

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

//返回字符串的小写形式

System.out.println(str.toLowerCase());

//返回字符串的大写形式

System.out.println(str.toUpperCase());

//去掉前后空格

System.out.println(str.trim());

/\*

\* 返回基本数据类型的String类型

\* String.valueOf(44)效果相当于：new Integer(44).toString()

\* 基本数据类型包括int，long，float，double，short，char，byte

\*/

int i = 99;

String valueOf = String.valueOf(i);

System.out.println(valueOf);

System.out.println(String.valueOf(true));

System.out.println(String.valueOf(55.7F));

System.out.println(String.valueOf(444.6));

System.out.println(String.valueOf(8888L));

}

}

【程序55】 String与Date相互转化

DateFormat formatTime = DateFormat.getDateTimeInstance();

DateFormat formatDate = DateFormat.getDateInstance();

1.带时间的date类型

String timeStr = "2010-11-12 12:33:21";

Date time = formatTime.parse(timeStr);

2.不带时间的date类型，默认为00:00:00

String dateStr = "2010-11-12";

Date date = formatDate.parse(dateStr);

3.格式化time 与date

String str5 = formatDate.format(date);

String str6 = formatTime.format(time);

4.任意格式去格式化time

String format1 = "yyyy-MM-dd hh:mm:ss";

String format2 = "yyyyMMddhhmmss";

String format3 = "yyyy/MM/dd hh:mm:ss";

String format4 = "yyyy\_MM\_dd\_hh\_mm\_ss";

String time1 = FastDateFormat.getInstance(format1).format(time);

String time2 = FastDateFormat.getInstance(format2).format(time);

String time3 = FastDateFormat.getInstance(format3).format(time);

String time4 = FastDateFormat.getInstance(format4).format(time);

【程序56】 从控制台中获取字符串

从控制台中获取字符串，如

“hksdfjJHHuhuBJH88...&\*$$%56HGGFGH77778JJK”,包含字母、数字、特殊字符等，打印出该字符串中出现次数最多的字符和出现的次数，如果有多个，则打印出多个；打印出该字符串中出现次数最少的字符和出现的次数，如果也有多个，则打印出多个；区分大小写！

package com.umt.string;

import java.util.ArrayList;

import java.util.Collections;

import java.util.HashMap;

import java.util.Iterator;

import java.util.List;

import java.util.Map;

import java.util.Scanner;

public class StringShowingMostChar

{

public static void main(String h[])

{

Scanner scanner = new Scanner(System.in);

String waitProcessString = scanner.next();

System.out.println("你好，你输入的字符串为: " + waitProcessString);

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

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

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

{

list\_waitProcessString.add(waitProcessString.substring(i, i + 1));

}

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

{

String temp = list\_waitProcessString.get(i);

if (!list\_noRepeat.contains(temp))

{

list\_noRepeat.add(temp);

}

}

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

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

{

int count = 0;

for (int j = 0; j < list\_waitProcessString.size(); j++)

{

if (list\_waitProcessString.get(j).equals(list\_noRepeat.get(i)))

{

count += 1;

}

}

map.put(list\_noRepeat.get(i), count);

}

List<Integer> list\_values = new ArrayList<Integer>();

Iterator iterator\_values = map.values().iterator();

while (iterator\_values.hasNext())

{

Integer value = (Integer) iterator\_values.next();

list\_values.add(value);

}

Collections.sort(list\_values);

Iterator iterator\_most = list\_noRepeat.iterator();

while (iterator\_most.hasNext())

{

String singerChar = (String) iterator\_most.next();

if (list\_values.get((list\_values.size() - 1)) == map.get(singerChar))

{

System.out.println("出现次数最多的字符为: " + singerChar + "；出现最多的次数为："+ list\_values.get((list\_values.size() - 1)));

}

}

Iterator iterator\_least = list\_noRepeat.iterator();

while (iterator\_least.hasNext())

{

String singerChar = (String) iterator\_least.next();

if (list\_values.get(0) == map.get(singerChar))

{

System.out.println("出现次数最少的字符为: " + singerChar+ "；出现最少的次数为：" + list\_values.get(0));

}

}

}

}

执行结果：

dhfgdjgjfdhjdfh....\_-\*&^%##$%^73457467777547657idfgduHGJhuhhdsfguuufuuduugdgj

你好，你输入的字符串为: dhfgdjgjfdhjdfh....\_-\*&^%##$%^73457467777547657idfgduHGJhuhhdsfguuufuuduugdgj

出现次数最多的字符为: d ；出现最多的次数为： 9

出现次数最多的字符为: u ；出现最多的次数为： 9

出现次数最少的字符为: - ；出现最少的次数为：1

出现次数最少的字符为: \_ ；出现最少的次数为：1

出现次数最少的字符为: \* ；出现最少的次数为：1

出现次数最少的字符为: # ；出现最少的次数为：1

出现次数最少的字符为: 3 ；出现最少的次数为：1

出现次数最少的字符为: i ；出现最少的次数为：1

出现次数最少的字符为:H ；出现最少的次数为：1

出现次数最少的字符为:G；出现最少的次数为：1

出现次数最少的字符为: J ；出现最少的次数为：1

出现次数最少的字符为: s ；出现最少的次数为：1

【程序57】解压缩zip文件

解压缩zip文件：

import org.apache.tools.ant.Project;

import org.apache.tools.ant.taskdefs.Expand;

public void unzip(String sourceZip, String destDir) throws Exception

{

try

{

Project p = new Project();

Expand e = new Expand();

e.setProject(p);

e.setSrc(new File(sourceZip));

e.setOverwrite(false);

e.setDest(new File(destDir));

e.setEncoding("gbk");

e.execute();

} catch (Exception e)

{

throw e;

}

}

【程序58】 十进制转化为二进制、八进制、十六进制

从控制台输入数字，数字范围不限制，把该数字转化为二进制、八进制、十六进制。。

package com.umt.czj;

import java.math.BigInteger;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.Scanner;

public class NumberTransfer

{

// 二进制参数

private static final Long TWO\_NUMBER\_STEP = 2L;

// 八进制参数

private static final Long EIGHT\_NUMBER\_STEP = 8L;

// 十六进制参数

private static final Long SIXTEEN\_NUMBER\_STEP = 16L;

// 表达式结果长度

private static final int NUMBER\_STEPS = 32;

// 十六进制大于等于10用字母代替

private static final int NUMBER\_TEN = 10;

// 十六进制小于等于15用字母代替

private static final int NUMBER\_FIFTEEN = 15;

// 待处理数字超过二进制范围报错

private static final String EXCEED\_RANG\_BINARY = "The number exceeding rang for binary!";

// 待处理数字超过八二进制范围报错

private static final String EXCEED\_RANG\_OCTAL = "The number exceeding rang for octal!";

// 待处理数字超过十六进制范围报错

private static final String EXCEED\_RANG\_HEX = "The number exceeding rang for hex!";

// 十六进制数字子母映射表

private static Map<Long, String> hexMap = new HashMap<Long, String>();

public static void main(String[] args)

{

try

{

Scanner scan = new Scanner(System.in);

System.out.println("请输入十进制数字：");

BigInteger number = scan.nextBigInteger();

System.err.println(number + "的 二进制表示形式为: "+ getNumberByStep(number, TWO\_NUMBER\_STEP));

System.err.println(number + "的 八进制表示形式为: "+ getNumberByStep(number, EIGHT\_NUMBER\_STEP));

System.err.println(number + "的十六进制表示形式为: "+ getNumberByStep(number, SIXTEEN\_NUMBER\_STEP));

}

catch (Throwable th)

{

th.printStackTrace();

throw new RuntimeException(th);

}

}

/\*\*

\* 根据数字来源和进制参数得到结果的处理逻辑

\* @param number

\* @param step

\* @return

\*/

public static String getNumberByStep(BigInteger number, Long step)

{

hexMap.put(10L, "A");

hexMap.put(11L, "B");

hexMap.put(12L, "C");

hexMap.put(13L, "D");

hexMap.put(14L, "E");

hexMap.put(15L, "F");

BigInteger result = new BigInteger(step.toString());

String numberModel = "00000000000000000000000000000000";

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

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

{

numberList.add(numberModel.substring(i, i + 1));

}

// 进制的阶乘和结果映射

Map<Integer, BigInteger> map\_step = new HashMap<Integer, BigInteger>();

for (int i = 0; i < NUMBER\_STEPS; i++)

{

if (0 == i)

{

map\_step.put(i, new BigInteger("1"));

}

else if (1 == i)

{

map\_step.put(i, new BigInteger(step.toString()));

}

else

{

result = result.multiply(new BigInteger(step.toString()));

map\_step.put(i, result);

}

}

// 进制转换逻辑

for (int i = NUMBER\_STEPS - 1; i >= 0; i--)

{

BigInteger value = map\_step.get(i);

int index = 0;

// 待处理数字比映射结果大

if (number.min(value).equals(value))

{

index = i;

// 相除的结果

BigInteger divideResult = number.divide(value);

long divideResultInteger = divideResult.longValue();

String divideResultString = divideResult.toString();

//余数

BigInteger remainderResult = number.remainder(value);

// 若为十六进制且相除的结果在10~15之间则取映射结果

if (step == SIXTEEN\_NUMBER\_STEP

&& divideResultInteger >= NUMBER\_TEN

&& divideResultInteger <= NUMBER\_FIFTEEN)

{

divideResultString = hexMap.get(divideResultInteger);

numberList.set(NUMBER\_STEPS - index - 1, divideResultString);

}

else

{

// 二进制超过范围

if (TWO\_NUMBER\_STEP == step

&& (TWO\_NUMBER\_STEP - 1 < divideResultInteger || 0 > divideResultInteger))

{

return EXCEED\_RANG\_BINARY;

}

// 八进制超过范围

if (EIGHT\_NUMBER\_STEP == step

&& ( EIGHT\_NUMBER\_STEP - 1 < divideResultInteger || 0 > divideResultInteger))

{

return EXCEED\_RANG\_OCTAL;

}

// 十六进制超过范围

if (SIXTEEN\_NUMBER\_STEP == step

&& ( SIXTEEN\_NUMBER\_STEP - 1 <divideResultInteger|| 0 > divideResultInteger ))

{

return EXCEED\_RANG\_HEX;

}

// 数字模板集合设值

numberList.set(NUMBER\_STEPS - index - 1,

divideResult.toString());

}

// 把余数赋值给待处理数字，进行下一轮循环，直到最后一轮得到结果

number = remainderResult;

}

}

// 返回计算结果

String resultData = "";

for (int j = 0; j < numberList.size(); j++)

{

String tempNumber = numberList.get(j);

resultData += tempNumber;

}

return resultData;

}

}