

java第一次作业

1.第一题

1.1题目

有1、2、3、4四个数字，它们能组成多少个互不相同且无重复数字的三位数？每行打印四个三位数。

1.2代码

思路：全排列，剔除有相同和重复数字的情况

```
public class Test1{
    public static void main(String[] args){
        int cnt=0;
        for(int i=1;i<5;i++){
            for(int j=1;j<5;j++){
                for(int k=1;k<5;k++){
                    if(i!=j&&i!=k&&j!=k){
                        int num=i*100+j*10+k;
                        cnt++;
                        System.out.print(num+"\t");
                        if(cnt==4){
                            System.out.print("\n");
                            cnt=0;
                        }
                    }
                }
            }
        }
    }
}
```

3.运行结果

```
123 124 132 134
142 143 213 214
231 234 241 243
312 314 321 324
341 342 412 413
421 423 431 432
[Finished in 1.1s]
```

2.第二题

2.1题目

用递归方法打印如下形式的数字塔：

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

2.2代码

```
public class Test2 {
    public static void main(String[] args) {
        int max = 5;
        new Test2().p(max,max);
    }
    void p(int i,int m){
        if(i>0){
            p(i-1,m);
            String s = "";
            for(int k=0; k<i; k++){
                s+=k+1+" ";
            }
            System.out.println(s);
        }
    }
}
```

2.3运行结果

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
[Finished in 599ms]
```

3.第三题

3.1题目

用筛选法求100之内的素数。要求：

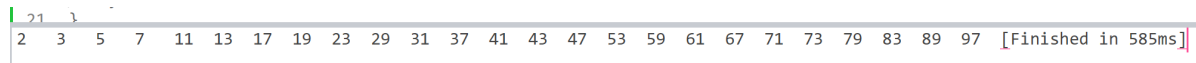
- (1) 用一个一维数组存储所有将被筛选的数；
- (2) 每行输出10个素数；
- (3) 写出主函数；

(4) 实现带参数的构造函数，其参数表示被筛选数的整数数量(从整数1开始筛选);

3.2代码

```
public class Test3{
    public static void main(String[] args){
        int[] arr=new int[100];
        for(int i=1;i<=100;i++){
            arr[i-1]=i;
            //System.out.print(arr[i-1]);
        }
        arr[0]=0;
        for(int j=2;j<=10;j++){
            int k=2;
            while(k*j<=100){
                arr[k*j-1]=0;
                k++;
            }
        }
        for(int i=0;i<=arr.length-1;i++){
            if(arr[i]!=0)
                System.out.print(arr[i]+"\\t");
        }
    }
}
```

3.3运行结果



```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 [Finished in 585ms]
```

4.第四题

4.1题目

编写一个求一元三次方程 $ax^3+bx^2+cx+d=0$ 近似根类，完成以下功能：

- (1) 用非静态成员变量存储方程的系数;
- (2) 一个输入方程系数的函数：方程系数从键盘输入，直到输入的系数满足函数单调为止;
- (3) 一个非静态成员方法用于二分法求方程的近似根，且满足：
 - ① 求近似根的区间 $[x_1, x_2]$ 从键盘输入;
 - ② 所求得的近似根 x 值直到满足 $f(x) < 10^{-5}$ 为止;
- (4) 一个非静态成员方法用于牛顿迭代法求方程的近似根，且满足：
 - ① 初始测试根值 x_0 从键盘输入;
 - ② 两次所求得的近似根 x_2 和 x_1 满足 $|x_2 - x_1| < 10^{-5}$ 为止;
- (5) 一个判断函数单调性的函数，用于测试函数单调性的 x_1 和 x_2 值从键盘输入;
- (6) 若函数单调则允许使用二分法和牛顿迭代法求方法的近似根，否则应;
- (7) 编写一个测试两种方法求近似根的主函数;

4.2代码

```
import java.util.Scanner;

public class Test4 {
    public static void main(String[] args){
        // 输入系数
        System.out.print("请依次输入一元三次方程 $ax^3+bx^2+cx+d=0$ 的系数a,b,c,d:");
        Scanner sc = new Scanner(System.in);
        double a = sc.nextDouble();
        double b = sc.nextDouble();
        double c = sc.nextDouble();
        double d = sc.nextDouble();
        Test4 t=new Test4();
        if(IsDangDiao(a,b,c,d)){
            System.out.print("使用二分法求根输入1,使用牛顿迭代法输入2,输入其他退出:");
            int mode =sc.nextInt();
            if(mode==1){
                t.ErFeng(a,b,c,d);
            }
            else if(mode==2){
                t.NewTon(a,b,c,d);
            }
        }
    }

    public static boolean IsDangDiao(double a, double b, double c, double d){
        System.out.print("请输入用于判断单调性的x1,x2:");
        Scanner sc2 = new Scanner(System.in);
        double x1 = sc2.nextDouble();
        double x2 = sc2.nextDouble();
        if(f(x1,a,b,c,d)*f(x2,a,b,c,d)<0){
            System.out.print("该函数的根存在\n");
            return true;
        }
        else{
            System.out.print("该函数的根可能不存在\n");
            return false;
        }
    }

    public static double f(double x, double a, double b, double c, double d){
        return a*x*x*x + b*x*x + c*x + d;
    }

    public static double fi(double x, double a, double b, double c, double d){
        return 3*a*x*x + 2*b*x + c;
    }

    public double ErFeng(double a,double b,double c,double d){
        System.out.print("输入近似根区间x1,x2:");
        Scanner sc2 = new Scanner(System.in);
        double x1 = sc2.nextDouble();
        double x2 = sc2.nextDouble();
        while (Math.abs(f(x1,a,b,c,d))>1e-5) {
            double xx = (x1 + x2) / 2;
```

```

        if (f(xx, a, b, c, d) * f(x1, a, b, c, d) < 0) {
            x2 = xx;
        } else {
            x1 = xx;
        }
    }
    // x1是根
    System.out.print("用二分法求出的近似根x1为"+x1+"\n");
    return 0;
}

public double NewTon(double a,double b,double c,double d){
    System.out.print("输入初始测试根值x0:");
    Scanner sc3 = new Scanner(System.in);
    double x1=sc3.nextDouble();
    double q=fi(x1,a,b,c,d);
    double p=f(x1,a,b,c,d)-fi(x1,a,b,c,d)*x1;
    double x2=-p/q;
    while(Math.abs(x2-x1)>1e-5){
        x1=x2;
        q=fi(x1,a,b,c,d);
        p=f(x1,a,b,c,d)-fi(x1,a,b,c,d)*x1;
        x2=-p/q;
    }
    System.out.print("用牛顿迭代法求出的近似根x1为"+x1+"\n");
    return 0;
}
}

```

4.3运行结果

测试 $x^3-5x+2=0$,其根为 $-1-\sqrt{2}$, $-1+\sqrt{2}$, 2

```

C:\Windows\system32\cmd.exe
请依次输入一元三次方程ax3+bx2+cx+d=0的系数a, b, c, d:1 0 -5 2
请输入用于判断单调性的x1, x2:-100 100
该函数的根存在
使用二分法求根输入1, 使用牛顿迭代法输入2, 输入其他退出:1
输入近似根区间x1, x2:-5 0
用二分法求出的近似根x1为-2. 414214015007019
请按任意键继续. . .

```

```

C:\Windows\system32\cmd.exe
请依次输入一元三次方程ax3+bx2+cx+d=0的系数a, b, c, d:1 0 -5 2
请输入用于判断单调性的x1, x2:-100 100
该函数的根存在
使用二分法求根输入1, 使用牛顿迭代法输入2, 输入其他退出:2
输入初始测试根值x0:5
用牛顿迭代法求出的近似根x1为2. 0000002805511965
请按任意键继续. . .

```

