.分析以下需求，并用代码实现(每个小需求都要封装成一个方法)：

(1)键盘录入两个数据,求两个数据之和(整数和小数)

(2)键盘录入两个数据,判断两个数据是否相等(整数和小数)

(3)键盘录入三个数据,获取两个数中较大的值(整数和小数)

(4)打印m行n列的星形矩形

(5)打印nn乘法表

(6)计算长方形和圆形的面积

import java.util.Scanner;

import java.lang.Math;

public class MyDayFiveHomeWork\_1{

public static void main(String[] args){

Scanner SC = new Scanner(System.in);

System.out.println("请问需要什么服务"+"\n"+"1.求两个数之和(两个整数或者两个小数)"

+"\n"+"2.判断两个数据是否相等(两个整数或者两个小数)"+"\n"

+"3.获取两个数中较大的值(两个整数或者两个小数)"+"\n"

+"4.打印m行n列的星形矩阵"+"\n"+"5.打印nn乘法表"+"\n"

+"6.计算长方形和圆形的面积"+"\n");

String function = SC.next();

switch(function){

case "1":

System.out.println("输入整数请按1，输入小数请按2");

int selector1 = SC.nextInt();

if(selector1==1){

System.out.println("请输入第一个整数: ");

int a1 = SC.nextInt();

System.out.println("请输入第二个整数: ");

int b1 = SC.nextInt();

System.out.println("两个数之和: "+plus(a1,b1));

}else if(selector1==2){

System.out.println("请输入第一个小数: ");

double a1 = SC.nextDouble();

System.out.println("请输入第二个小数: ");

double b1 = SC.nextDouble();

System.out.println("两数之和: "+plus(a1,b1));

}else{

System.out.println("乱敲啥！");

}

break;

case "2":

System.out.println("输入整数请按1，输入小数请按2");

int selector2 = SC.nextInt();

if(selector2==2){

System.out.println("请输入第一个整数: ");

int a2 = SC.nextInt();

System.out.println("请输入第二个整数:");

int b2 = SC.nextInt();

if(equals(a2,b2)){

System.out.println("两数相等");

}else {

System.out.println("两数不相等");

}

}else if(selector2==2){

System.out.println("请输入第一个小数: ");

double a2 = SC.nextDouble();

System.out.println("请输入第二个小数: ");

double b2 = SC.nextDouble();

if(equals(a2,b2)){

System.out.println("两数相等");

}else {

System.out.println("两数不相等");

}

}else {

System.out.println("乱敲啥！");

}

break;

case "3":

System.out.println("输入整数请按1,输入小数请按2");

int selector = SC.nextInt();

if(selector==1){

System.out.println("请输入第一个整数: ");

int a3 = SC.nextInt();

System.out.println("请输入第二个整数: ");

int b3 = SC.nextInt();

System.out.println("最大值为: "+getMax(a3,b3));

}else if(selector==2){

System.out.println("请输入第一个小数: ");

double a3 = SC.nextDouble();

System.out.println("请输入第二个小数: ");

double b3 = SC.nextDouble();

System.out.println("最大值为: "+getMax(a3,b3));

}else{

System.out.println("乱敲啥！");

}

break;

case "4":

System.out.println("请输入行数(必须整数): ");

int n = SC.nextInt();

System.out.println("请输入列数(必须整数): ");

int m = SC.nextInt();

printTheStar(n,m);

break;

case "5":

System.out.println("请输入行数(必须整数): ");

int n1 = SC.nextInt();

printTheNM(n1);

break;

case "6":

System.out.println("计算长方形面积请按1，计算圆形的面积请按2");

int selector3 = SC.nextInt();

if(selector3==1){

System.out.println("请输入长方形的长: ");

int a4 = SC.nextInt();

System.out.println("请输入长方形的宽: ");

int b4 = SC.nextInt();

System.out.println("长方形的面积为: "+getRectangleArea(a4,b4));

}else if(selector3==2){

System.out.println("请输入圆形的半径: ");

double r = SC.nextInt();

System.out.println("圆形的面积为: "+getCircularArea(r));

}else{

System.out.println("乱敲啥！");

}

break;

default :

System.out.println("无此服务,谢谢惠顾！");

break;

}

}

public static int plus(int a,int b){

return a+b;

}

public static double plus(double a,double b){

return a+b;

}

public static int getMax(int a,int b){

int max = (a>b)?a:b;

return max;

}

public static double getMax(double a,double b){

double max = (a>b)?a:b;

return max;

}

public static boolean equals(int a,int b){

return a==b;

}

public static boolean equals(double a,double b){

return a==b;

}

public static void printTheStar(int n,int m){

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

for(int b=1;b<=m;b++){

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

}

System.out.println();

}

}

public static void printTheNM(int n){

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

for(int b=1;b<=a;b++){

System.out.print(a+"X"+b+"="+(a\*b)+" ");

}

System.out.println();

}

}

public static int getRectangleArea(int a,int b){

return a\*b;

}

public static double getCircularArea(double r){

return Math.PI\*r\*r;

}

}

分析以下需求，并用代码实现：(封装成方法)

(1)打印1-100之间的所有素数及个数

(2)每行输出5个满足条件的数，之间用空格分隔

注意：大于1的能被1和其本身整除的数叫素数。

public class MyDayFiveHomeWork{

public static void main(String[] args){

int count = 0;

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

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

if(i%j==0){

continue;

}

}

count++;

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

if(count%5==0){

System.out.println();

}

}

System.out.println("\n"+"1到100之间的所有素数为: "+count);

}

}

//基本方法二

public class MyDayFiveHomeWork {

public static void main(String[] args) {

int count = 0;//记录素数的个数

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

int x = 2; //因为每个数都能被1整除，所以从2开始判断整除

for(; x<i; x++){ //从2到小于本身的数依次遍历判断

if(i%x == 0) //如果遇到能被某数整除的数，就结束跳出循环，进行后面的判断

break;

}

if(x == i){ //说明这个数只能被1和其本身整除，是素数

count++;//是素数个数加1

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

if(count % 5 == 0) //每输出5个数进行换行

System.out.println();

}

}

System.out.println();

System.out.println("100以内的素数为："+ count);

}

}

封装一个方法，接收一个参数n，打印nn乘法表。

import java.util.Scanner;

public class MyDayFiveHomeWork{

public static void main(String[] args){

Scanner SC = new Scanner(System.in);

System.out.println("请输入一个数：");

int num = SC.nextInt(); //自己输入一个数n

print(num); //调用函数（方法）print（打印乘法表）

}

//输出语句已经出现在方法函数中，输出的值就是方法函数的打印出的循环出来的值，

//所以无需再主函数内出现输出语句，直接调用整个方法函数。

//函数方法，打印乘法表 输出两个数相乘的值

public static void print(int m){

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

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

System.out.print(i+"\*"+j+"="+(i\*j)+" ");//内循环 输出相乘值

}

System.out.println();//外循环 打印乘法表 一乘法表的形式

}

}

}

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？

方法一：

//运用基本的for循环嵌套的方法写出

/\*

public class MyDayFiveHomeWork{

public static void main(String[] args){

int count = 0; //初始化计数器count值为0

//三个for循环得出从1,2,3,4，四个数中取出数

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

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

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

if(i!=j&&j!=k&&i!=k){ //if 语句判断从四个数中取出的三个数是否相等

count++; //满足if条件后，计数器总计从四个数中取出三个数且互不重复

//三位数的计算方法：

System.out.println(i\*100+j\*10+k);//@将所取出的三个数组成一个三位数@

}

}

}

}

System.out.println("总个数为"+count);

}

}

\*/

//方法二： 此方法目前无法理解 需要以后的知识

//用数组的方法，且运用方法函数的调用

/\*

public class HundredNumber{

private int[] a={1, 2, 3, 4};//定义数组并赋值

private int num=0;//定义组成三位数的个数

private int hundredNum=0;//定义组成的三位数

//hundred-百位, tens-十位, units-个位

public int hundNumber(int hundred, int tens, int units){

return 100\*hundred+10\*tens+1\*units;

}

public static void main(String[] args){

HundredNumber hn=new HundredNumber();

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

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

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

if(hn.a[i]!=hn.a[j]&&hn.a[j]!=hn.a[m]&&hn.a[m]!=hn.a[i]){

hn.hundredNum=hn.hundNumber(hn.a[i], hn.a[j],hn.a[m]);

hn.num++;

System.out.println(hn.hundredNum);

}

}

}

}

System.out.println("Total: "+hn.num);

}

\*/

//方法三： 程序没有运行出来 出现类的定义错误

/\*

package mypackage;

public class circle{

public static void main(String[] args){

circle o = new circle();

o.method1();

}

public static void method1(){

int[] a = new int[method2()];

int count = 0;

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

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

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

if(i==j&&j==t){

break;

}else{

a[count]=i\*100+j\*10+t;

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

count++;

}

}

}

}

System.out.println(count);

}

public static int method2(){

int count=0;

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

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

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

if(i==j&&j==t){

break;

}else{

count++;

}

}

}

return count;

}

}

}

.分析以下需求，并用代码实现：

(1)定义一个int类型的二维数组

(2)获取该二维数组最小元素，并输出

public class Demo{

public static void main(String[] args){

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

int min=0; //定义一个最小值用来接收二维数组里的最小值

for (int i = 0; i < array.length; i++) { //外层循环array 值为int类型数组

for (int j = 0; j < array.length; j++) { //内层循环 array中每一个数组

if(i==0&&j==0){ //min赋值为第一个结果数组中的第0个下标的值

min = array[0][0];

}

//如果min大于正在循环的数值

if(min>array[i][j]){

min = array[i][j]; //把小的一方赋予给min

}

}

}

System.out.println(min); //输出min

}

}