|  |
| --- |
| **一、实验目的：**  1、掌握程序基本结构  2、掌握键盘输入Scanner类  3、对象生成 |
| **二、实验环境：**  有Java环境的计算机若干台 |
| **三、实验内容和要求：**  1、定义一个类，该类只有一个main主函数，在主函数内采用Scanner输入一个整数n，然后输入n个浮点数，求出其中最大的数，同时求输入n个数的和，并输出。    思路：使用Scanner类输入n，然后采用循环认为输入的第一个数为最大，存在max变量中，逐个输入并与max比较，如果比max大，更改max。求和则定义sum=0，逐个相加即可。  *import* java.util.Scanner;  *public class* Main {   *public static void* main(String[] args) {   Scanner scanner = *new* Scanner(System.*in*);  System.*out*.println("输入一个整数n");  *int* n = scanner.nextInt();   *float* max = 0, min = 0, sum = 0;  *for* (*int* i = 0; i < n; i++) {  *float* x = scanner.nextFloat();  sum += x;  *if* (i == 0) {  min = max = x;  }  *if* (x < min) {  min = x;  } *else* max = x;  }  System.*out*.println("sum = " + sum + "; min = " + min + "; max = " + max);  } }  2、定义一个类，该类中有两个方法，第一个方法为判断一个数是否为素数，如果是则返回true，否则返回false，第二个方法，使用第一个方法求出m到n之间的所有素数，在定义一个主类，在主函数内输入两个数，验证第一个类中定义的求出这两个数之间的所有素数。  思路：第一个类框架  class first{  boolean prime（int n）{//该方法为判断n是否为素数}  void allprime（int m，int n）{//利用前一个方法求出 m到n之间的所有素数}  }  第二个类框架：  class second{  public static void main（String args[]）  { 使用Scanner输入一个数，采用前一个类的第一方法判断输入的数是否是素数，在输入两个数，采用前一个类的第二个方法，求出这两个数之间的所有素数  }  *public class* Q2 {  *public static void* main(String[] args) {  first f = *new* first();  f.allprime(1,10);  } }  *class* first {  *boolean* prime(*int* n) {*//该方法为判断n是否为素数  boolean* flag = *true*;   *if* (n < 2) {*// 素数不小于2  return false*;  } *else* {   *for* (*int* i = 2; i <= Math.*sqrt*(n); i++) {   *if* (n % i == 0) {*// 若能被整除，则说明不是素数，返回false* flag = *false*;  *break*;*// 跳出循环* }  }  }  *return* flag;   }   *void* allprime(*int* m, *int* n) {  *for* (*int* i = m; i < n; i++) {  *if* (prime(i)) System.*out*.println(i);  }  }*//利用前一个方法求出 m到n之间的所有素数}* }    该题目的主要目的在于理解类内的函数可以自由调用，类间函数要生成对象调用 |
| **3、145=1！+4！+5！这样的数称为强大数，求1-1000000之间的所有强大数**  *public class* Q3 {  *public static void* main(String[] args) {  A3 a = *new* A3();  *for* (*int* i = 1; i < 1000000; i++) {  *if* (a.isStrongNum(i)) System.*out*.println(i);  }  } }  *class* A3 {  *public boolean* isStrongNum(*int* i) {  *int* origin = i;  *int* temp, r = 0;   *while* (i != 0) {  temp = i % 10;  i /= 10;  r += factorial(temp);  }  *if* (origin == r) *return true*;  *return false*;  }   *public int* factorial(*int* i) {  *int* r = 1;  *for* (*int* t = 1; t <= i; t++) {  r \*= t;  }  *return* r;  } }    **4、景山动物园猴山有30步楼梯，一直无聊的猴子在做一个实验，用1步或者3步爬猴山，次序无关，问一共有几种爬法，该猴子累的精疲力尽，也没有统计结果，请你帮帮忙。**    *public class* Q4 {  *public static void* main(String[] args) {  *int* n = 30;  *int*[] f = *new int*[n + 1];  f[1] = 1;  f[2] = 1;  f[3] = 2;   *for* (*int* i = 4; i < n + 1; i++) {  f[i] = f[i - 1] + f[i - 3];  }  System.*out*.println(f[n]);  } }  **如果可以用1，2，3 又有几个结果？**    **最后两个题目用于熟练程序结构。** |
| **四、实验步骤：** |
| **五、实验结果与分析（**含程序、数据记录及分析和实验总结等**）：**  **动态规划**  **每一阶楼梯都是上一阶上来的，也就是说 比如 第3阶是第2阶或直接到第3的，那么只要把前面的所有情况之和就是 到第三阶的所有方法** |
| **六：思考题：** |