* **厦门大学**

**Java上机实验**

**（第一次四个实验）**

**姓名：刘馨欣**

**学号：22920132203820**

**班级：计算机1班**

**系别：计算机**

**学院：信息科学与技术学院**

**实验时间：2015年03月20日**

**报告完成时间：2015年03月21日**

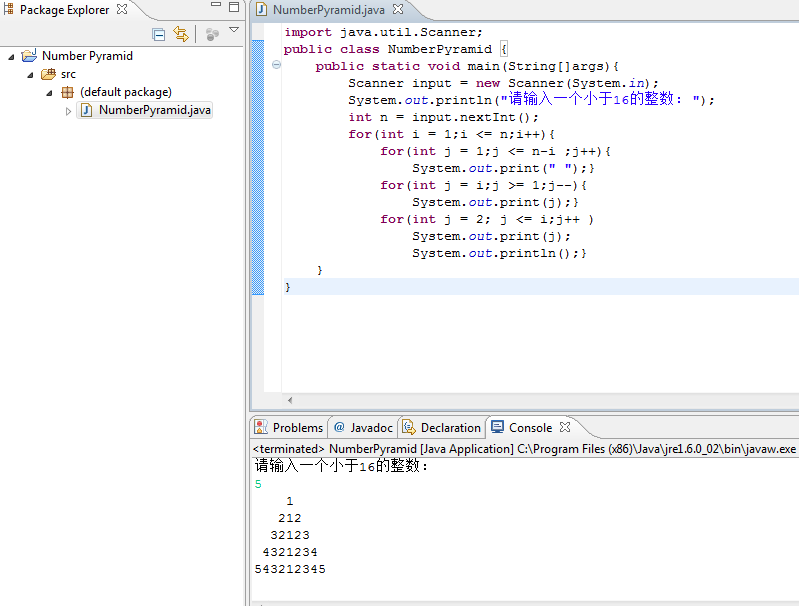
**实验一 数字金字塔**

**（Number Pyramid）**

**一、实验题目**

输入一个小于16的正整数n，输出一个数字金字塔。不考虑输入错误的情形。要求使用Scanner作为输入，System.out.print作为输出。

**二、实验结果**



**三、实验设计思路**

为完成数字金字塔的搭建，以“空格+左侧数字+右侧数字”的方式进行程序设计，在此过程中简化对核心数字及右侧空格的考虑。

**四、实验收获**

1.可将n具体化，更便于进行程序设计；

2.本程序的核心在于for循环中i与j关系的确定，建立数学规律与程序设计间的联系；

3.熟悉了Java语言。

**五、源代码**

import java.util.Scanner;

public class NumberPyramid {

public static void main(String[]args){

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

System.*out*.println("请输入一个小于16的整数： ");

int n = input.nextInt();

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

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

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

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

System.*out*.print(j);}

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

System.*out*.print(j);

System.*out*.println();}

}

}

**实验二 PI的近似值**

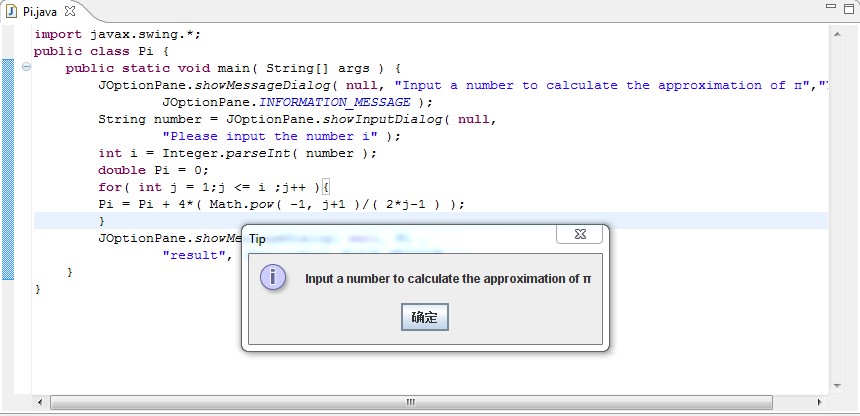
**（Approximation of π）**

**一、实验题目**

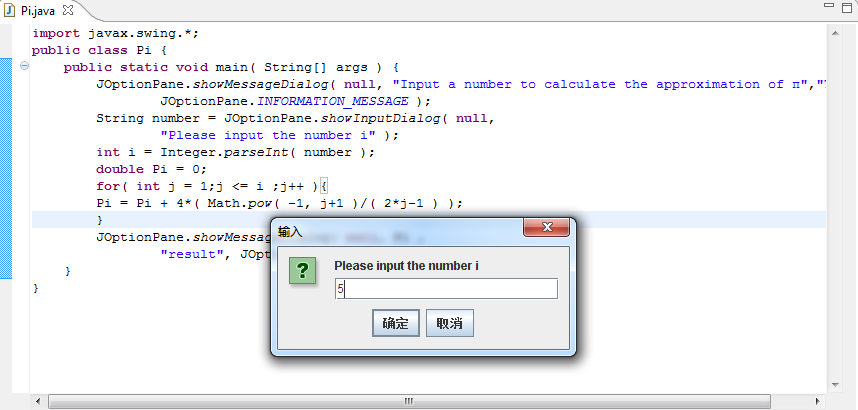
使用给定算式计算PI的近似值并显示，其中i的值由用户输入。要求使用JOptionPane.showInputDialog作为输入，JOptionPane.showMessageDialog作为输出。

**二、实验结果**

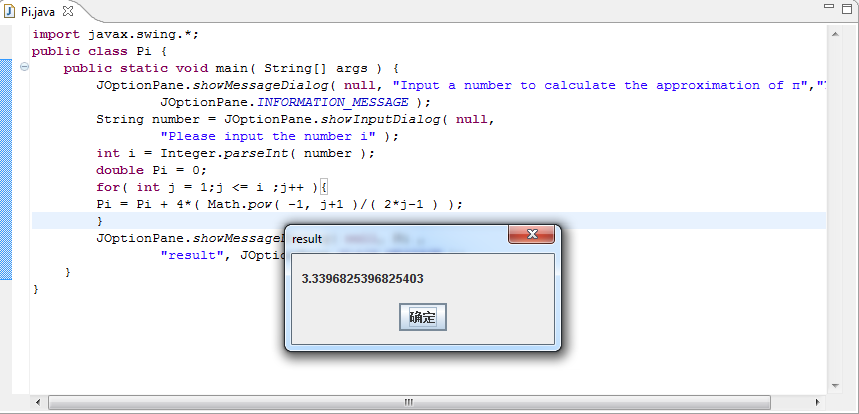
1.输入i=5时实验结果实例

****

（图一）

****

（图二）

****

（图三）

2.间隔取值结果记录（保留至小数点后6位）

i=1时，Pi=4.000000

i=5时，Pi=3.339682

i=20时，Pi=3.091624

i=60时，Pi=3.124927

i=100时，Pi=3.131593

i=1000时，Pi=3.140593

**三、实验设计思路**

本实验在程序设计方面以一个简单的for循环及Pi的自身叠加思想便可完成。实验关键在于运用Java弹出对话框（具有提示、输入等功能）的程序设计。

**四、实验收获**

1.学会用Java编程实现对话框弹出的效果；

2.了解Java在程序输入过程中左侧边框的红叉和灯泡提示有可能在点击运行后自动消失……不要跟它死磕……

**五、源代码**

import javax.swing.\*;

public class Pi {

public static void main( String[] args ) {

JOptionPane.*showMessageDialog*( null, "Input a number to calculate the approximation of π","Tip",

JOptionPane.*INFORMATION\_MESSAGE* );

String number = JOptionPane.*showInputDialog*( null,

"Please input the number i" );

int i = Integer.*parseInt*( number );

double Pi = 0;

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

Pi = Pi + 4\*( Math.*pow*( -1, j+1 )/( 2\*j-1 ) );

}

JOptionPane.*showMessageDialog*( null, Pi ,

"result", JOptionPane.*PLAIN\_MESSAGE* );

}

}

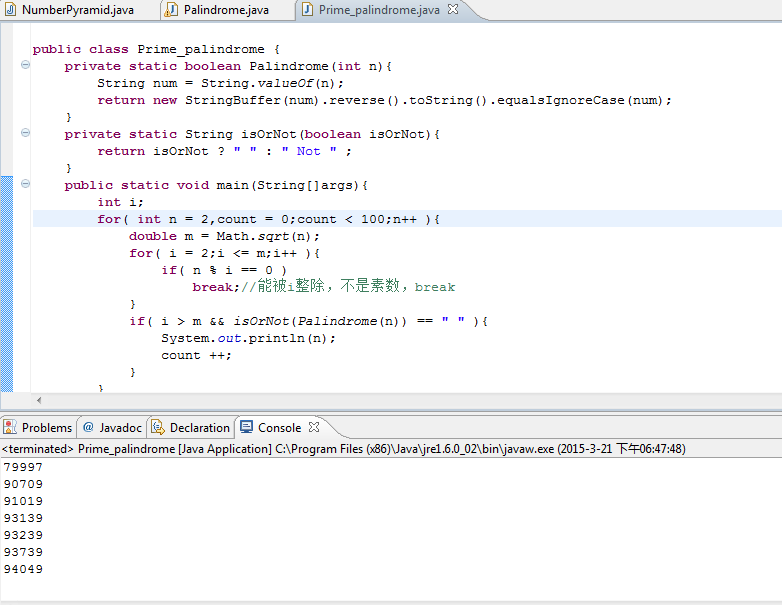
**实验三 回文素数**

**（Prime palindromes）**

**一、实验题目**

回文素数指某一个数既是回文数又是素数，例如2,3,5,7,11,101,131……编程找出前100个回文素数，并且要求按照每行10个的格式输出。输出方式任选。

**二、实验结果**

****

**三、实验设计思路**

1.首先了解回文数和素数的定义；

2.设计输出前100个素数的函数（以遍历到根号n为最简）；

3.设计输出前100个回文数的函数（字符型比取余法要好）；

4.合并程序。

**四、实验收获**

1.学习Java语言中函数的使用方法；

2.学习boolean类型的用法及如何判断素数、回文数；

3.直观体会了程序算法选择不同其复杂程度不同的实例。

**五、源代码**

public class Prime\_palindrome {

private static boolean Palindrome(int n){

String num = String.*valueOf*(n);

return new StringBuffer(num).reverse().toString().equalsIgnoreCase(num);

}

private static String isOrNot(boolean isOrNot){

return isOrNot ? " " : " Not " ;

}

public static void main(String[]args){

int i;

for( int n = 2,count = 0;count < 100;n++ ){

double m = Math.*sqrt*(n);

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

if( n % i == 0 )

break;//能被i整除，不是素数，break

}

if( i > m && *isOrNot*(*Palindrome*(n)) == " " ){

System.*out*.println(n);

count ++;

}

}

}

}

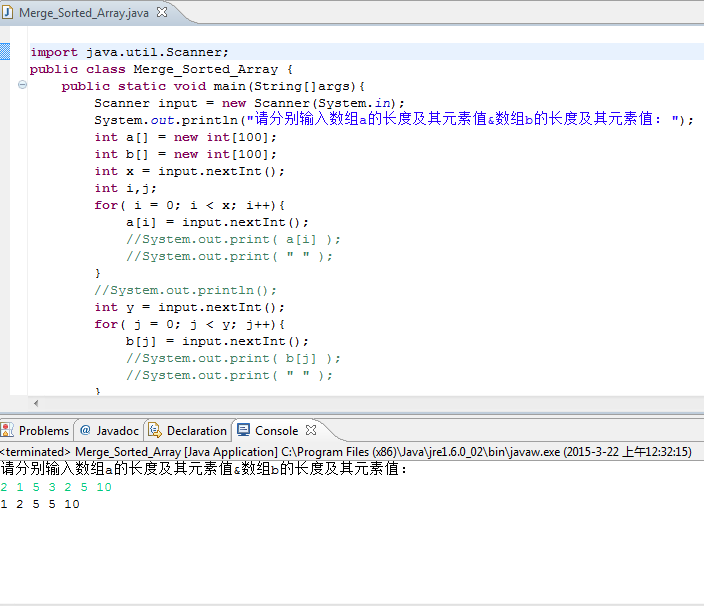
**实验四 数组合并**

**（Merge Sorted Array）**

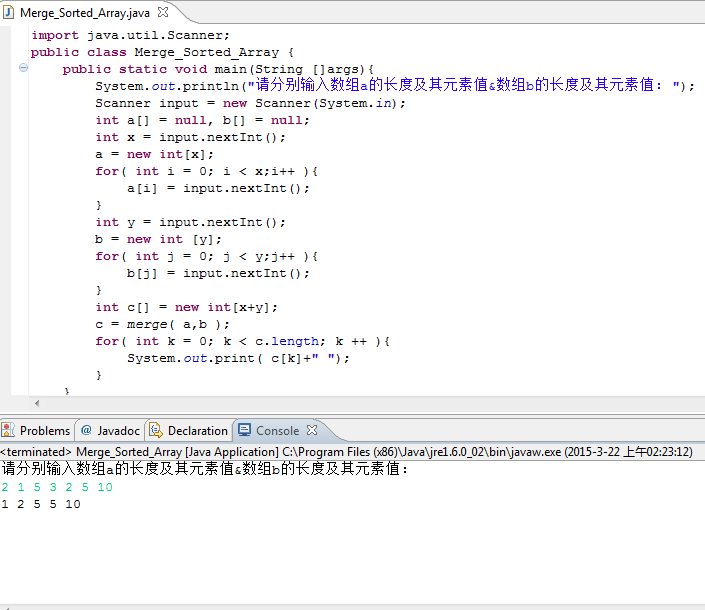
**一、实验题目**

写一个合并数组的方法，将两个已排序的数组合并成一个有序的大数组。方法原型如下：public static int[]merge(int[]list1,int[]list2)。要求提供一个测试的main函数，让用户输入两个数组的大小及元素值，最后调用上述方法并输出合并结果。输入输出方式任选。

**二、实验结果**

****

（图一 未使用函数调用）



（图二 使用函数调用）

**三、实验设计思路**

1.设计由Java语言创建动态数组并输入数组元素数值；

2.运用比较算法进行程序设计；

3.在Java中实现函数调用。

**四、实验收获**

1.复习比较算法；

2.学习Java中函数调用；

3.了解到Java中“数组名.length”的用法，很实用。

**五、源代码**

import java.util.Scanner;

public class Merge\_Sorted\_Array {

public static void main(String []args){

System.*out*.println("请分别输入数组a的长度及其元素值&数组b的长度及其元素值： ");

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

int a[] = null, b[] = null;

int x = input.nextInt();

a = new int[x];

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

a[i] = input.nextInt();

}

int y = input.nextInt();

b = new int [y];

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

b[j] = input.nextInt();

}

int c[] = new int[x+y];

c = *merge*( a,b );

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

System.*out*.print( c[k]+" ");

}

}

public static int[]merge(int []a,int []b){

int i=0,j=0;

int c[] = new int[a.length+b.length];

while( i < a.length && j < b.length ){

//System.out.println(j);

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

c[i+j] = a[i];

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

i++;

//System.out.println(i);

}

else {

c[i+j] = b[j];

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

j++;

}

}

for( ; i < a.length ; i++ ){

c[i+j] = a[i];

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

}

for( ; j < b.length ; j++ ){

c[i+j] = b[j];

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

}

return c;

}

}