1. JOptionPane类的showMessageDialog方法（静态方法）：消息对话框显示任意文本

Eg：import javax.swing.JOptionPane;

Public class WelcomeInMessageDialogBox{

Public static void main(String[] args){

JOptionPane.showMessageDialog(null,”welcome to Java!”);

}

}

两种调用方式:JOptionPane.showMessageDialog(null,x);

JOptionPane.showMessageDialog(null,x,y,JOptionPane.INFORMATION\_MESSAGE);

----🡪另一种表述：

JOptionPane window = new JOptionPane(System.in);

Window.showMessageDialog(null,”welcome to Java”);

1. JOptionPane类中的showInputDialog方法：

调用方法：String input = JOptionPane.showInputDialog(x);

String input = JOptionPane.showInputDialog(null,x,y,JOptionPane.

QUESTION\_MESSAGE);

1. JOptionPane类中的showConfirmDialog():

Int option = JOptionPane.showConfirmDialog(null, “continue?”);

Yes返回的值为： JOptionPane.YES\_OPTION == 0;

No 返回的值为: JOptionPane.NO\_OPTION == 1;

Cancel 返回的值：JOption.CANCEL\_OPTION == 2;

Eg： int option = JOptionPane.showConfirmDialog(null,”judge a = 4”);

int answer;

if(answer == JOptionPane.YES\_OPTION)

System.out.print(“yes”)

1. Println &Scanner方法:

Java不直接支持控制台输入，但是可以使用Scanner类来创建它的对象，以读取来自System.in的输入。

Eg：import java.util.Scanner;

Public class ComputerAreaWithConsoleInput{

Public static void main(String[] args){

Scanner input = new Scanner(System.in);

System.out.print (“Enter a number for radius:”);

double radius = input.nextDouble();

double area = radius\*radius\*3.1415926;

//display result

System.out.println(“The area for the circle of radius”+radius+”is”+area);

}

}

注：Scanner对象的方法：nextByte()\nextShort()\nextInt()\nextLong()\nextFloat()\

NextDouble()\next()-读取一个字符串，在空白符之前结束\nextLine()-读取一行文本在回车键之前结束。

格式化输出： System.out.printf(format, item1, item2…itemn);

Eg: double x = 2.0/3 ;

System.out.printf(“x is %4.2f” , x);

1. System类中的exit(0);方法。终止程序，参数0表示程序正常结束

Eg: System.out.println(“Error : invalid Status ”);

System.exit(0);

1. 将数字字符串转化为数字: Integer类中的parseInt方法 / Double类中的parseDouble方法：

Int intValue = Integer.parseInt(intString) ;

Double doubleValue = Double.parseDouble(doubleString);

1. java API 中的Math类的pow方法：Math.pow(a,b)->ab

Random()方法: Math.random()产生一个随机数，返回一个双精度的大于0小于1的浮点数。(int)(Math.random()\*10);🡺取一位整数。

取0-100的随机数: (int)(Math.random()\*101);

取50-99 :50+(int)(Math.random()\*50);

* 1. Math.round()方法：得到最接近的值

在执行窄化转换时，总是对数字进行截尾操作，用Math.Round()进行操作四舍五入。

1. Arrays类 java.util.Arrays包括各种各样的静态方法,这些方法都有对所有基本类型的重载方法：

对整个数组或部分数组进行排序

Java.util.Arrays.sort(listName);

二分查找法（数组有序）

Java.util.Arrays.binarySearch(listName, number);

采用equals方法检测两个数组是否相等，如果内容相等则数组相等。

Java.util.Arrays.equals(listName1, listName2);

使用fill方法填充整个数组或部分数组

Java.util.Arrays.fill(list1,5)//fill 5 to the whole array

Java.util.Arrays.fill(list2,1,3,8)//fill 8 to a partial array

1. Date类java.util.Date

+Date()以当前的时间构造一个Date对象

+Date(elapseTime:long)以GTM 1970.1.1 00:00:00 开始为给定时间构造一个Date对象

+toString()返回日期和时间的字符串表达式

+getTime()返回从GTM开始算起的毫秒数

1. System.currentTimeMillis()方法：返回从1970年1月1日00：00：00开始到现在当前 的时刻的毫秒数。返回的数据类型是long。系统相关
2. Random类java.util.Random

+Random()以当前时间作为种子构造Random对象

+Random(seed:long)以特定种子构造Random对象

+nextInt()返回一个随机整型值

+nextInt(n:int)返回一个在0-n之间的（不包括0，n）的随机整型值

+nextLong()

+nextDouble()返回一个0.0到1.0之间（不包括）的随机doube型值

+nextFloat()

+nextBoolean()返回一个随机布尔值

创建一个Random对象时必须指定一个种子或者使用默认种子，无参构造方法使用当前已经逝去的时间作为种子

* 1. 当有seed的时候，相同的随机变量每次产生的随机数是相同的。
  2. Java.util.Random rand = new Java.util.Random(47);

1. Org.apache.commons.logging (commons-logging-1.1.1.jar)

logFactory: