## 1、基本定义

|  |
| --- |
| **import** **java.util.\***;  **import** **java.io.\***;  **public** **class** **Main**  {  **public** **static** void main(String[] args)  {  Scanner cin1 = **new** Scanner(System.in);  Scanner cin2 = **new** Scanner(**new** BufferedInputStream(System.in));  }  } |

使用cin2进行输入的时候可能会比cin1快一些。

## 2、输入具体数据

* （1）输入一个整数：int n = cin.nextInt();
* （2）输入一个字符串：String s = cin.next();
* （3）输入一个浮点数：double f = cin.nextDouble();
* （4）读入一整行：String s = cin.nextLine();

判断是否有下一个输入，可以用cin.hasNext()或cin.hasNextInt()或cin.hasNextDouble()等进行判断。

## 3、基本输出

（1）System.out.print(); //类似于cout<<…….;

（2）System.out.println(); //类似于cout<<……<<endl;

（3）System.out.printf(); //类似于C中printf的功能

|  |
| --- |
| import java.io.\*;  import java.math.\*;  import java.util.\*;  import java.text.\*;  public class Main {  public static void main(String[] args) {  Scanner cin = new Scanner(new BufferedInputStream(System.in));  int a;  double b;  a = 12345;  b = 1.234567;  System.out.println(a + " " + b);  System.out.printf("%d %10.5f\n", a, b);  // 输入b为字宽为10，右对齐，保留小数点后5位，四舍五入.  }  } |

## 4、要求具体精度的输出

（1）可以使用上面介绍的System.out.printf();

（2）对于输出浮点数要保留几位小数的问题，可以使用DecimalFormat类解决

|  |
| --- |
| import java.util.\*;  import java.text.\*;  public class Main {  public static void main(String[] args) {  DecimalFormat f = new DecimalFormat("#.00#");  DecimalFormat g = new DecimalFormat("0.000");  // 这里的0指一位数字，#指除0以外的数字  double a = 123.456789, b = 0.123456;  System.out.println(f.format(a));  System.out.println(f.format(b));  System.out.println(g.format(a));  System.out.println(g.format(b));  }  } |

输出结果：

123.457

.123

123.457

0.123

## 5、字符串的处理

（1）String

Java中字符串String是不可以修改的，要修改只能转换为字符数组。

|  |
| --- |
| String st = "abcdefg";  char[] ch;  ch = st.toCharArray(); // 字符串转换为字符数组. |

## 6、高精度问题

## 7、大数问题

Java中有两个类BigDecimal（表示浮点数）和BigInteger（表示整数）   
使用这两个类的时候需要加上import java.math.\*;

Ⅰ基本函数：

（1）valueOf(parament); 将参数转换为指定类型

|  |
| --- |
| int a = 3;  BigInteger b = BigInteger.valueOf(a); |

（2）add(); //大数加法

例如：

|  |
| --- |
| BigInteger a = new BigInteger(“11”);  BigInteger b = new BigInteger(“22”);  a.add(b); |

（3）substract(); //减法

（4）multiply(); //乘法

（5）divided(); //相除取整

（6）remainder(); //取余

（7）pow(); //a.pow(b) = a ^ b

（8）gcd(); //最大公约数

（9）abs(); //绝对值

（10）negate(); //取反数

（11）mod(); //a.mod(b) = a % b = a.remainder(b)

（12）max(); min();

（13）public int compareTo(); //比较

（14）boolean equals(); //比较是否相等

（15）BigIntergerde 构造函数

一般用到以下两种：

BigInteger(String val);

将指定字符串转换为十进制表示形式；

BigInteger(String val,int radix);

将指定基数的 BigInteger 的字符串表示形式转换为 BigInteger

例如：

BigInteger b = new BigInteger(“1010”,2);

System.out.println(b);

输出：10

* Ⅱ.基本常量：

|  |
| --- |
| A=BigInteger.ONE //=1  B=BigInteger.TEN //=10  C=BigInteger.ZERO //=0 |

Ⅲ.基本操作   
1.读入：

|  |
| --- |
| while(cin.hasNext()) //等同于!=EOF  {  int n;  BigInteger m;  n=cin.nextInt(); //读入一个int;  m=cin.BigInteger();//读入一个BigInteger;  System.out.print(m.toString());  System.out.print(m);  } |

|  |
| --- |
| import java.util.Scanner;  import java.math.\*;  import java.text.\*;  public class Main {  public static void main(String args[]) {  Scanner cin = new Scanner(System.in);  BigInteger a, b;  int c;  char op;  String s;  while (cin.hasNext()) {  a = cin.nextBigInteger();  s = cin.next();  op = s.charAt(0);  if (op == '+') {  b = cin.nextBigInteger();  System.out.println(a.add(b));  } else if (op == '-') {  b = cin.nextBigInteger();  System.out.println(a.subtract(b));  } else if (op == '\*') {  b = cin.nextBigInteger();  System.out.println(a.multiply(b));  } else {  BigDecimal a1, b1, eps;  // 浮点数  String s1, s2, temp;  s1 = a.toString();  a1 = new BigDecimal(s1);  b = cin.nextBigInteger();  s2 = b.toString();  b1 = new BigDecimal(s2);  c = cin.nextInt();  // 接收精度控制，即保留几位小数的问题  eps = a1.divide(b1, c, 4);  if (c != 0) {  temp = "0.";  for (int i = 0; i < c; i++)  temp += "0";  DecimalFormat gd = new DecimalFormat(temp);  System.out.println(gd.format(eps));  } else  System.out.println(eps);  }  }  }  } |

|  |
| --- |
| string s;  BigDecimal x;  s.charAt(i) //代表查看字符串s的第i个字符  x.stripTrailingZeros().toPlainString(); //代表x数转化成无后导零的字符串  x.toString(); //代表转化成科学计数的字符串  s.substring(1,s.lenth()-1) //代表截取下标的字符串 |

|  |
| --- |
| import java.math.BigDecimal;  import java.util.Scanner;  public class N {  public static void main(String[] args) {  Scanner in = new Scanner(System.in);  BigDecimal x;  String z = "0";  int n;  while (in.hasNextBigDecimal()) {  x = in.nextBigDecimal();  n = in.nextInt();  x = x.pow(n);  String s = x.stripTrailingZeros().toPlainString();  if(s.charAt(0) == '0') {  s = s.substring(1,s.length());  }  System.out.println(s);  }  }  } |