1. **Math类**

math类主要功能是进行数学计算的操作类，提供有基础的计算公式，这个类的构造方法被私有化了但不是单例，而且这些方法都是static方法，即: 这些方法可以直接用类名调用。

|  |
| --- |
| package com.company.数字操作类;  public class MathClass {  public static void main(String[] args) {  System.out.println(Math.abs(-20.2));  //取绝对值  System.out.println(Math.max(10.1,20.1));  //取最大值  System.out.println(Math.round(-2.4));  //四舍五入取整  System.out.println(Math.log(2.718999999));  //以e为底的对数  System.out.println(Math.log10(100));  //以10为底的对数  }  } |

虽然Math中提供有四舍五入的方法，但是他是将小数点后的所有数字都四舍五入，这样我们不能保留指定位数的小数。

如:18.6885想要保留小数点后两位:18.69。

范例:自定义的四舍五入的程序功能

|  |
| --- |
| package com.company.数字操作类;  class MathUtil {  private MathUtil(){}  /\*\*  \* 保留指定小数位数的四舍五入  \* @param num 操作的数字  \* @param scale 保留的小数  \* @return 处理后的数字  \*/  public static double round(double num , int scale){  return Math.round(num \* Math.pow(10,scale))/Math.pow(10,scale) ;  }  }  public class RoundClass {  public static void main(String[] args) {  System.out.println(MathUtil.round(18.6885, 2));  }  } |

Math类中都是我们一些基础的数学公式，需要自己重新整合。

1. Random类

Random类的主要功能是产生随机数，这个类主要是依靠内部提供的方法完成。

·产生一个不大于边界的随机正整数:public int NextInt(int bound);

范例:产生随机数;

|  |
| --- |
| package com.company.数字操作类;  import java.util.Random;  public class RandomClass {  public static void main(String[] args) {  Random random = new Random();  for (int i = 0; i < 10; i++) {  System.out.println(random.nextInt(10));  //0-9之间的随机数  }  }  } |

1. 大数字处理类

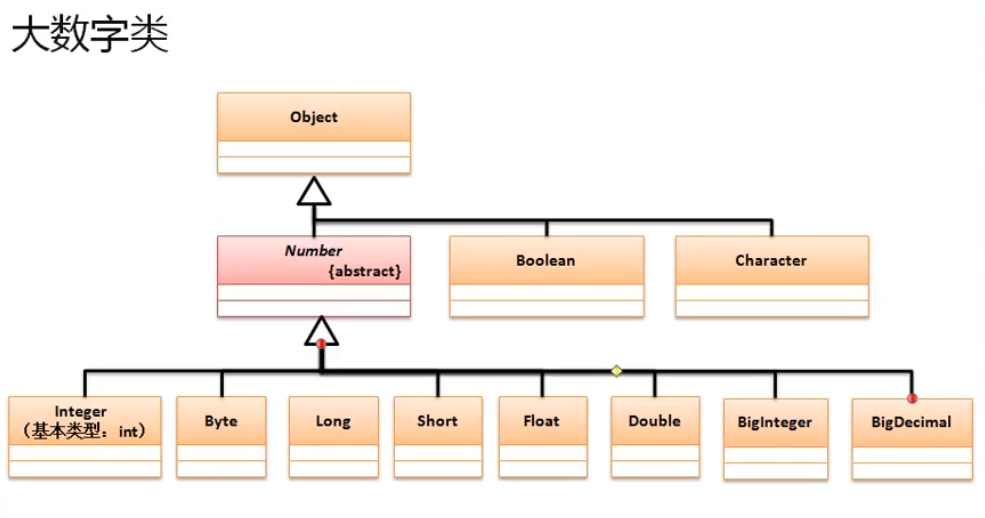
可以实现海量数字计算(同样能提供的也是基础操作)

124124234234234212412423423423421241242342342342

124124234234234212412423423423412412423423423421241242342342342124124234234234212412423423423421241242342342342124124234234234212412423423423412412423423423421241242342342342

想上面这样的数字，最早的时候只能通过String类进行保存。而String类进行数学运算需要逐位拆分，每一位自己计算，而后自己独立k控制进位处理，那么这样的开发难度非常高，所以为了解决这样的问题，提供了两个大数字操作类:BigInteger,BigDecimal。

大数字类是Number的子类:



Number类中有六个方法Integer、Byte、Long、Short、Float、Double。现在将大数据加进来又多了两个方法，BigInteger、BigDecimal。需要注意的是，虽然提供有大数字类操作类，但是整体的操作之中还是要考虑到一个性能问题。(数字越大结果出来的越慢。)

|  |
| --- |
|  |

此时的运算结果是非常缓慢的，所以任何的电脑都是有极限的。所以在实际开发中如果没有超过Integer所包含的范围，强烈不建议使用大数字操作，因为这种计算性能是很差的。