

Java基础(Java SE)

大数值



Number类

▷ Number类

- ▷ Java为每一个内置数据类型提供了对应的包装类
- ▷ 所有的包装类(Integer,Long,Byte,Double,Float,Short)都是抽象类Number的子类
- ▷ 这种由编译器特别支持的包装称为装箱，所以当内置数据类型被当作对象使用时，编译器会把内置类型装箱为包装类。
- ▷ 相似地，编译器也可以把一个对象拆箱为内置类型。

Math类

▷ Math类

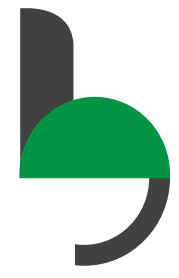
▷ Math类包含了用于执行基本数学运算的属性和方法。

▷ Math类的方法都被定义为static形式，通过Math类可以在主函数中直接调用。

大数值

- ▷ BigInteger和BigDecimal，这两个类可以处理包含任意长度的数字序列的数值
- ▷ 使用static类型的valueOf()可以将普通的数值转换成大数值
- ▷ 不能使用算术运算符如+,-处理大数值，需要使用大数值类中的特定方法。

```
BigInteger c=a.add(b);  
BigInteger d=c.multiply(b.add(BigInteger.valueOf(2)));
```



精度运算

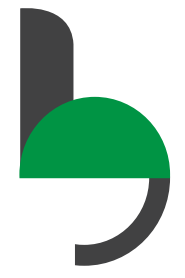
- ▷ BigInteger和BigDecimal, BigInteger实现了任意精度的整数运算, BigDecimal实现了任意精度的浮点数运算, 并且这两个可以运算任意长度的数值 (这是Integer和Decimal等做不到的, 他们都有长度限制, 例如Integer就是10亿)
- ▷ 那为什么要使用任意精度?
- ▷ 因为有的时候我们的计算对精度的要求比较高, 而浮点等运算事达不到的, 并且由于浮点数的存储及运算规上的规定, 总是会出一些奇怪的问题, 假设我们运行以下的代码:

```
import java.math.BigDecimal;

public class Test06 {
    public static void main(String[] args) {
        System.out.println(2.0 - 1.1);
    }
}
```

- ▷ 你猜结果等于多少? 是不是0.9? 但是, 真实的结果是:

```
0.8999999999999999
```



大数值解决精确计算

▷ 那如果我们使用大数值来运算

```
import java.math.BigDecimal;

public class Test06 {
    public static void main(String[] args) {
        BigDecimal result = BigDecimal.valueOf(2.0).subtract(BigDecimal.valueOf(1.1));
        System.out.println(result);
    }
}
```

▷ 结果就成了0.9

▷ 细心的你会发现，为什么不能直接“—”？这是因为Java语言的设计者没有给大数值类提供+—*%的重载，所以只能调用add/subtract/mulitiply/divide等方法进行加减乘除的运算

THE END