

# 计算圆周率 $\pi$ 到小数点后 1000 位

Abel

2024 年 12 月 5 日

计算到圆周率  $\pi$  到小数点后 1000 位，可以使用欧拉的公式

$$\arctan(t) = \frac{t}{1+t^2} \sum_{n=0}^{\infty} \frac{(2n)!!}{(2n+1)!!} \left(\frac{t^2}{1+t^2}\right)^n$$

把  $t = 1$  代入，可以得到

$$\frac{\pi}{4} = \frac{1}{2} \sum_{n=0}^{\infty} \frac{(2n)!!}{(2n+1)!!} \left(\frac{1}{2}\right)^n$$

即

$$\pi = 2 \sum_{n=0}^{\infty} \frac{n!}{(2n+1)!!}$$

展开可得

$$\pi = 2 \left( 1 + \frac{1}{3} + \frac{1 \cdot 2}{3 \cdot 5} + \frac{1 \cdot 2 \cdot 3}{3 \cdot 5 \cdot 7} + \frac{1 \cdot 2 \cdot 3 \cdot 4}{3 \cdot 5 \cdot 7 \cdot 9} \cdots \right)$$

写成递归的形式

$$\pi = 2 \left( 1 + \frac{1}{3} \left( 1 + \frac{2}{5} \left( 1 + \frac{3}{7} \left( 1 + \frac{4}{9} (1 + \cdots) \right) \right) \right) \right)$$

这样就可以编程了

```
1 import java.math.BigInteger;
2
3 public class IntPi{
4     public static BigInteger CalIntPi(int n){
5         BigInteger a = new BigInteger("10");
6         a = a.pow(n);
7         a = a.multiply(new BigInteger("2"));
8
9         BigInteger re = new BigInteger("0");
10        final BigInteger zero = new BigInteger("0");
11        final BigInteger one = new BigInteger("1");
12        final BigInteger two = new BigInteger("2");
13        BigInteger x = new BigInteger("1");
14        BigInteger y = new BigInteger("3");
15
16        while (a.compareTo(zero) >0){
17            re = re.add(a);
18            a = a.multiply(x);
19            a = a.divide(y);
20            x = x.add(one);
21            y = y.add(two);
22        }
23        return re;
24    }
25 }
26
27 public static void main(String[] args){
28     BigInteger pi = CalIntPi(1024);
29     System.out.println(pi);
30 }
31 }
32
```

## 拉赞助

如果你觉得本文对你有所启发，可以随意赞助一点

