Class BigInteger内部方法

测试用例设计

华南理工大学

16-5班游程卉项目组

编写 蓝艺

二 0 19年 4月

修订记录

| 日期 | 修订 版本 | Defect ID CR号 | 修改 章节 | 修改描述 | 作者 |
| --- | --- | --- | --- | --- | --- |
| 2019-04-27 | 1.00 |  |  | 初稿完成 | 蓝艺 |
|  |  |  |  |  |  |
|  |  |  |

目录

[1 简介 4](#_Toc408917965)

[1.1 编写目的 4](#_Toc408917966)

[1.2 参考资料 4](#_Toc408917967)

[1.3 范围 4](#_Toc408917968)

[2 测试用例设计 4](#_Toc408917969)

[2.1 测试对象 4](#_Toc408917970)

[2.2 测试分析与用例设计 4](#_Toc408917971)

[2.2.1 public BigInteger substract(BigInteger val) 5](#_Toc408917972)

[2.2.2 private static BigInteger multiplyByInt(int[] x, int y int sign): 5](#_Toc408917973)

[2.2.3 private BigInteger square(): 5](#_Toc408917974)

[2.2.4 private static BigInteger multiplyKaratsuba(BigInteger x, BigInteger y) 6](#_Toc408917975)

# 简介

## 编写目的

本文档用于指导开发人员和测试人员共同完成单元测试的实施，本文档针对的测试对象为BigInteger类中的下列4个方法：

1. public BigInteger subtract(BigInteger val)
2. private static BigInteger multiplyByInt(int[] x, int y, int sign)
3. private BigInteger square()
4. private static BigInteger multiplyKaratsuba(BigInteger x, BigInteger y)

## 参考资料

IDEA 2019 Offcial Tutorial

JUnit 4 Offcial Tutorial

实验指导手册

## 范围

4 methods of class BigInteger:

public BigInteger subtract(BigInteger val)

private static BigInteger multiplyByInt(int[] x, int y, int sign)

private BigInteger square()

private static BigInteger multiplyKaratsuba(BigInteger x, BigInteger y)

# 测试用例设计

## 测试对象

4 methods of class BigInteger:

public BigInteger subtract(BigInteger val)

private static BigInteger multiplyByInt(int[] x, int y, int sign)

private BigInteger square()

private static BigInteger multiplyKaratsuba(BigInteger x, BigInteger y)

## 测试分析与用例设计

对上述四个方法依次采用判定覆盖、基本路径覆盖、判定覆盖、条件组合覆盖方法进行白盒测试。

### public BigInteger substract(BigInteger val)

1. 测试用例设计策略：判定覆盖
2. 判定条件表

|  |  |  |
| --- | --- | --- |
| T0 | C0 | val.signum==0 |
| T1 | C1 | this.signum==0 |
| T2 | C2 | val.signum!=this.signum |
| T3 | C3 | cmp==0 |
| T4 | C4 | cmp>0 |
| T5 | C5 | cmp==signum |

由于判定表与条件表表述相同，故而使用TC统一指代，

1. 测试用例表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 用例ID | this.val | val | TC0 | TC1 | TC2 | TC3 | TC4 | TC5 | 输出 |
| U0 | 2.0E+800 | 0 | T | - | - | - | - | - | 2.0E+800 |
| U1 | 0 | 2.0E+800 | F | T | - | - | - | - | -2.0E+800 |
| U2 | 2.0E+800 | -2.0E+800 | F | F | T | - | - | - | 4.0E+800 |
| U3 | 2.0E+800 | 2.0E+800 | F | F | F | T | - | - | 0 |
| U4 | 4.0E+800 | 2.0E+800 | F | F | F | F | T | T | 2.0E+800 |
| U5 | 2.0E+800 | 4.0E+800 | F | F | F | F | F | F | -2.0E+800 |

### private static BigInteger multiplyByInt(int[] x, int y, int sign)

1. 测试用例设计策略：基本路径覆盖
2. 路径图：  
   
3. 判定条件表：

|  |  |  |
| --- | --- | --- |
| T0 | C0 | Integer.bitCount(y) == 1 |
| T1 | C1 | carry == 0L |

1. 测试用例表：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例ID | X | Y | TC0 | TC1 | 路径 | 输出 |
| U0 | 4.0 | 4.0E+200 | F | F | abcdedfijm | 1.6E+201 |
| U1 | 400 | 400 | F | T | abcdfikm | 1.6E+5 |
| U2 | 400 | 4 | T | - | ablm | 1.6E+3 |

### private BigInteger square():

1. 测试用例设计策略：判定覆盖法
2. 判定条件表：

|  |  |  |
| --- | --- | --- |
| T0 | C0 | signum==0 |
| T1 | C1 | mag.length < KARATSUBA\_SQUARE\_THRESHOLD |
| T2 | C2 | len < TOOM\_COOK\_SQUARE\_THRESHOLD |

注：signum=0且BigInteger对象长度不为0的测试用例无法构建而导致无法调用此函数，故而TC0在本次测试中将维持False的状态；

用例表

1. 测试用例表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | X | TC1 | TC2 | 输出 |
| U0 | 2.0E+500 | T | - | 4.0E+1000 |
| U1 | 2.0E+1280 | F | T | 4.0E+2560 |
| U2 | 2.0E+2160 | F | F | 4.0E+4320 |

### private static BigInteger multiplyKaratsuba(BigInteger x, BigInteger y)

1. 测试用例设计策略：条件组合覆盖
2. 判定条件表：

|  |  |  |
| --- | --- | --- |
| T0 | C0 | x.signum != y.signum |

1. 测试用例表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例ID | X | Y | T0 | C0 | 输出 |
| U0 | 2.0E+800 | 2.0E+800 | F | F | 4.0E+1600 |
| U1 | 2.0E+800 | -2.0E+800 | T | T | -4.0E+1600 |