BigInteger类测试报告

1. 引言
   1. 编写目的

本次测试对象为BigInteger类中的如下方法：

* + 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)
  1. 测试方法  
     对上述四个方法分别采用判定覆盖、基本路径覆盖、判定覆盖、条件组合覆盖方法进行白盒测试，具体测试用例设计见第三部分（即测试概要）。
  2. 测试环境
     1. IDEA 2019
     2. Java 8
     3. JUnit 4
     4. Windows 10 Pro

1. 测试概要
   1. public BigInteger subtract(BigInteger val)

测试用例设计策略：判定覆盖

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

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

* 1. private static BigInteger multiplyByInt(int[] x, int y, int sign

测试用例设计策略：路径覆盖

路径图：



判定条件表

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

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 |

* 1. private BigInteger square()

测试用例设计策略：判定覆盖；

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

* 1. private static BigInteger multiplyKaratsuba(BigInteger x, BigInteger y)

测试用例设计策略：条件组合覆盖

判定T0：x.signum != y.signum

条件C0：x.signum != y.signum

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 |

具体测试用代码见BigIntegerTest. test\_multiplyKaratsuba()

1. 测试结论

经过上述方法进行的白盒测试，目标方法均未发现异常，本次测试完成。