Domaine Sciences et Technologies
LICENCE INFORMATIQUE

Année 2018-19

Programmation 2 : Corrigé du TD 5
Code UE : SIN3U02

Tests unitaires

1 Test d'une classe d'entier

```
public class BigIntegerTest {
    BigInteger twelve;

    @BeforeAll
    void initialize(){
        twelve = BigInteger(12);
    }

    @Test
    void testEquals() {
        assertEquals(BigInteger.ONE, BigInteger.ONE);
        assertNotEquals(BigInteger.TEN, BigInteger.ONE);
        assertNotEquals(BigInteger.TEN, new String("10"));
    }

    @Test
    void testDivideByZero() {
        assertThrows(ArithmeticException.class, () -> {
            BigInteger.ONE.divide(BigInteger.ZERO);
        });
    }

    @Test
    void testAdd(){
        BigInteger a = new BigInteger("-12");
        BigInteger b = new BigInteger("123");
        BigInteger sum = new BigInteger("111");
        assertEquals(sum, a.add(b));
        assertEquals(sum, b.add(a));
    }

    @Test
    void testNegate(){
        BigInteger a = new BigInteger("12");
        BigInteger b = new BigInteger("-12");
        assertEquals(a, b.negate());
    }

    @Test
    void testMultiply(){
        BigInteger a = new BigInteger("-12");
        BigInteger b = new BigInteger("12");
        BigInteger product = new BigInteger("-144");
        assertEquals(product, a.multiply(b));
    }

    @Test
    void testSubtract(){
        Random random = new Random(0);
        BigInteger[] valuesA = new  BigInteger[100];
        BigInteger[] valuesSub = new BigInteger[100];
        for(int i = 0; i<100; i++){
            int a = random.nextInt();
            int b = random.nextInt();
            long sub = ((long)a)-b;
            valuesA[i] = new BigInteger(Integer.toString(a)).
                    subtract(new BigInteger(Integer.toString(b)));
            valuesSub[i] = new BigInteger(Long.toString(sub));
        }
        assertArrayEquals(valuesSub, valuesA);
    }

    @Test
    void testDivideAndRemainder(){
        Random random = new Random(0);
        BigInteger[][] computedValues = new BigInteger[100][];
        BigInteger[][] expectedValues = new BigInteger[100][2];
        for(int i = 0; i<100; i++){
            int a = random.nextInt(100);
            int b = random.nextInt(100)+1;
            int quotient = a/b;
            int remainder = a%b;

            String stringA = Integer.toString(a);
            String stringB = Integer.toString(b)

            computedValues[i] = new BigInteger(stringA).
                    divideAndRemainder(new BigInteger(stringB));

            expectedValues[i][0] = new BigInteger(Long.toString(quotient));
            expectedValues[i][1] = new BigInteger(Long.toString(remainder));
        }
        assertArrayEquals(expectedValues, computedValues);
    }
}
```