

Rappels sur les tests

Test d'une classe d'entier

Question 1 : Quels sont les éléments du code qu'il est important de tester en premier ?

Le constructeur et la méthode `equals` sont les éléments les plus important à tester en premier. Il vont nous servir pour tous les tests (besoin de créer des objets et/ou de les comparer pour tous les tests). Les constantes peuvent ensuite être testées.

Question 2 : Quel devrait-être le nom de la classe de test qui va tester la classe `BigInteger` ?

Par convention, on nomme une classe de test avec le nom de la classe testée suivi de `Test`. La classe de test devrait donc s'appeler `BigIntegerTest`.

Question 3 : Donner le code de tests unitaires pour les différents éléments de la classe (constantes, constructeurs et méthodes).

```
import org.junit.jupiter.api.Test;

import java.math.BigInteger;

import static org.assertj.core.api.Assertions.*;

public class BigIntegerTest {
    @Test
    void testEquals(){
        BigInteger b1 = new BigInteger("+1");
        BigInteger b2 = new BigInteger("1");
        BigInteger b3 = new BigInteger("-1");
        BigInteger b4 = new BigInteger("0");
        BigInteger b5 = new BigInteger("0");
        assertThat(b1).isEqualTo(b1).isEqualTo(b2).isNotEqualTo(b3)
            .isNotEqualTo(b4).isNotEqualTo("+1");
        assertThat(b5).isEqualTo(b5).isEqualTo(b4).isNotEqualTo(b1);
    }
}
```



```

        .isEqualTo(BigInteger.ZERO);
    assertThat(new BigInteger("5").subtract(new BigInteger("-5")))
        .isEqualTo(BigInteger.TEN);
}

@Test
void testMultiply(){
    assertThat(BigInteger.ONE.multiply(BigInteger.ONE))
        .isEqualTo(BigInteger.ONE);
    assertThat(new BigInteger("-5").multiply(BigInteger.TWO))
        .isEqualTo(new BigInteger("-10"));
}

@Test
void testDivide(){
    assertThat(BigInteger.ONE.divide(BigInteger.ONE))
        .isEqualTo(BigInteger.ONE);
    assertThat(BigInteger.TEN.divide(BigInteger.TWO))
        .isEqualTo(new BigInteger("5"));
}

@Test
void testAbs(){
    assertThat(BigInteger.ONE.abs()).isEqualTo(BigInteger.ONE);
    assertThat(new BigInteger("-10").abs()).isEqualTo(BigInteger.TEN);
}

@Test
void testGcd(){
    assertThat(BigInteger.ONE.gcd(BigInteger.ONE))
        .isEqualTo(BigInteger.ONE);
    assertThat(BigInteger.TEN.gcd(BigInteger.TWO))
        .isEqualTo(BigInteger.TWO);
    assertThat(new BigInteger("-250").gcd(new BigInteger("-75")))
        .isEqualTo(new BigInteger("25"));
}
}

```