```java

package Game;

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

import java.util.HashSet;
import java.util.Set;

import static org.junit.jupiter.api.Assertions.*;

class DeckTest {

    private Deck deck;

    @BeforeEach
    void setUp() {
        deck = new Deck();
    }

    @Test
    void testInitialDeckSize() {
        int count = 0;
        while (!deck.isEmpty()) {
            deck.draw();
            count++;
        }
        assertEquals(52, count, "Deck should have 52 cards initially");
    }

    @Test
    void testDrawReducesSize() {
        int draws = 5;
        for (int i = 0; i < draws; i++) {
            assertNotNull(deck.draw(), "Draw should return a card");
        }

        int count = 0;
        while (!deck.isEmpty()) {
            deck.draw();
            count++;
        }

        assertEquals(52 - draws, count, "Deck should have 52 - draws cards left");
    }

    @Test
    void testIsEmptyAfterAllDraws() {
        for (int i = 0; i < 52; i++) {
            assertFalse(deck.isEmpty(), "Deck should not be empty before 52 draws");
            assertNotNull(deck.draw(), "Card should not be null before deck is empty");
        }
        assertTrue(deck.isEmpty(), "Deck should be empty after 52 draws");
        assertNull(deck.draw(), "Draw from empty deck should return null");
    }

    @Test
    void testAllCardsAreUnique() {
        Set<String> uniqueCards = new HashSet<>();
        while (!deck.isEmpty()) {
            Card card = deck.draw();
            assertNotNull(card, "Card should not be null");
            uniqueCards.add(card.toString());
        }
        assertEquals(52, uniqueCards.size(), "All cards in the deck should be unique");
    }
}

```