

```
1
2 package Game;
3
4 import org.junit.jupiter.api.BeforeEach;
5 import org.junit.jupiter.api.Test;
6
7 import java.util.HashSet;
8 import java.util.Set;
9
10 import static org.junit.jupiter.api.Assertions.*;
11
12 class DeckTest {
13
14     private Deck deck;
15
16     @BeforeEach
17     void setUp() {
18         deck = new Deck();
19     }
20
21     @Test
22     void testInitialDeckSize() {
23         int count = 0;
24         while (!deck.isEmpty()) {
25             deck.draw();
26             count++;
27         }
28         assertEquals(52, count, "Deck should have 52 cards initially");
29     }
30
31     @Test
32     void testDrawReducesSize() {
33         int draws = 5;
34         for (int i = 0; i < draws; i++) {
35             assertNotNull(deck.draw(), "Draw should return a card");
36         }
37
38         int count = 0;
39         while (!deck.isEmpty()) {
40             deck.draw();
41             count++;
42         }
43
44         assertEquals(52 - draws, count, "Deck should have 52 - draws cards left");
45     }
46
47     @Test
48     void testIsEmptyAfterAllDraws() {
49         for (int i = 0; i < 52; i++) {
50             assertFalse(deck.isEmpty(), "Deck should not be empty before 52 draws");
51             assertNotNull(deck.draw(), "Card should not be null before deck is empty");
52         }
53         assertTrue(deck.isEmpty(), "Deck should be empty after 52 draws");
54         assertNull(deck.draw(), "Draw from empty deck should return null");
55     }
56
57     @Test
58     void testAllCardsAreUnique() {
59         Set<String> uniqueCards = new HashSet<>();
60         while (!deck.isEmpty()) {
61             Card card = deck.draw();
62             assertNotNull(card, "Card should not be null");
63             uniqueCards.add(card.toString());
64         }
65         assertEquals(52, uniqueCards.size(), "All cards in the deck should be unique");
66     }
67 }
68
```