Cards

What's wrong with `buildDeckOfCards()`?

```java
1  import java.util.*;
2
3
4  public class Cards {
5
6      enum Suit {CLUB, DIAMOND, HEART, SPADE}
7
8      enum Rank {
9          ACE, DEUCE, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT,
10         NINE, TEN, JACK, QUEEN, KING
11     }
12
13     static class Card {
14         private final Suit suit;
15         private final Rank rank;
16
17         public Card(Suit suit, Rank rank) {
18             this.suit = suit;
19             this.rank = rank;
20         }
21     }
22
23     static Collection<Suit> suits = List.of(Suit.values());
24     static Collection<Rank> ranks = List.of(Rank.values());
25
26     public List<Card> buildDeckOfCards() {
27         List<Card> deck = new ArrayList<>();
28         for (Iterator<Suit> i = suits.iterator(); i.hasNext(); )
29             for (Iterator<Rank> j = ranks.iterator(); j.hasNext(); )
30                 deck.add(new Card(i.next(), j.next()));
31         return deck;
32     }
33 }
```

Click here to expand...

Bug: On line 30, the `i.next()` will move the iterator along which isn't intended

Solution: Iterate through the suits and ranks independently using nested for-each loops:

```java
public List<Card> buildDeckOfCards() {
    List<Card> deck = new ArrayList<>();
    for (Suit suit : suits) {
        for (Rank rank : ranks) {
            deck.add(new Card(suit, rank));
        }
    }
    return deck;
}
```