

What's a Collections Class?

It's important to understand that the Collections class is not the Collections Framework.

The framework contains many interfaces and implemented classes, as well as helper classes, which this Collections class is just one example.

At one time, Java had interfaces, but no support for static or default methods on them, so useful methods were packaged in these helper classes.

Some of these methods have since been implemented on the interfaces themselves, but there's still some functionality on the Collections class you might find useful.

I'll examine some of these in code, and compare them to the methods on the interfaces now available.

The Setup - a Quick Review

This code has Card, a record, with a nested enum, named Suit, declared in it.

Suit is either a Club, Diamond, Heart or Spade, and this enum has a helper function, getImage, that returns a printable character value for that suit.

I overrode the toString method on Card, to print that character along with what I call the face, or face value of the card, so 2 through 10, or Jack, Queen, King, or Ace.

Card
<pre>private Suit suit; private String face; private int rank;</pre>
<div>10♣(8) 2♦(0) A♥(12) Q♠(10)</div>
<pre>public static methods Card getNumericCard(Suit suit, int number) Card getFaceCard(Suit suit, char abbrev) List<Card> getStandardDeck() void printDeck(List<Card> deck) void printDeck(List<Card> deck, String description, int rows) public instance methods String toString()</pre>

<<Enumeration>>
Suit
CLUB, DIAMOND, HEART, SPADE
<pre>public instance methods char getImage()</pre>
CLUB = 9827
DIAMOND = 9830
HEART = 9829
SPADE = 9824

The Setup - a Quick Review

Each card will also have a rank, starting with 0 for the lowest card.

In a standard deck, the lowest card, a two, is 0, because an Ace is usually the highest value card, though an ace can represent a one sometimes.

The Card has static helper functions I'll be using in the code in this video, one to get an instance of a numeric card, one to get an instance of a face card.

Card
<pre>private Suit suit; private String face; private int rank; public static methods Card getNumericCard(Suit suit, int number) Card getFaceCard(Suit suit, char abbrev) List<Card> getStandardDeck() void printDeck(List<Card> deck) void printDeck(List<Card> deck, String description, int rows) public instance methods String toString()</pre>

10♣(8)
2♦(0)
A♥(12)
Q♠(10)

<<Enumeration>> Suit
CLUB, DIAMOND, HEART, SPADE
public instance methods char getImage()
CLUB = 9827
DIAMOND = 9830
HEART = 9829
SPADE = 9824

The Setup - a Quick Review

In addition, I can get a list of cards, that represents a standard deck of cards.

And there's functions to print the deck of cards.

Card
<pre>private Suit suit; private String face; private int rank;</pre>
<div>10♣(8) 2♦(0) A♥(12) Q♠(10)</div>
<div>public static methods Card getNumericCard(Suit suit, int number) Card getFaceCard(Suit suit, char abbrev) List<Card> getStandardDeck() void printDeck(List<Card> deck) void printDeck(List<Card> deck, String description, int rows) public instance methods String toString()</div>

<<Enumeration>>
Suit
CLUB, DIAMOND, HEART, SPADE
<div>public instance methods char getImage()</div>
CLUB = 9827
DIAMOND = 9830
HEART = 9829
SPADE = 9824