```
//*********************************************************
//
//  BlackJack.Java              Authors: Lewis, Chase, and Coleman
//
//  The BlackJack class provides an implementation of a single
//  deck blackjack game.  It makes use of the Hand class to
//  represent a player's hand and the Deck class to represent
//  the deck of cards for the game.
//
//*********************************************************
import jss2.exceptions.*;
import java.util.*;

public class Blackjack
{
  Hand dealer;    //to hold the dealer's cards
  Hand player;    //to hold the player's cards
  Deck newdeck;   //a set of cards

  public Blackjack(Hand dlr, Hand plr)
  {
    dealer = dlr;
    player = plr;
    newdeck = new Deck();
  }//Blackjack constructor

  /*********************************************************
    deal method - deals the intitial cards to each player
  *********************************************************/
  public void dealInitialCards()
  {
    dealer.newCard(newdeck);
    dealer.newCard(newdeck);
    player.newCard(newdeck);
    player.newCard(newdeck);

  }//end deal method

  /*********************************************************
    hit method - adds the next random card from the deck to
    the given player's hand
  *********************************************************/
  public Card hit(Hand whohit)
  {
    Card result = whohit.newCard(newdeck);

    return result;

  }//end hit method

  /*********************************************************
    handValue method - returns the value of the given player's
    hand
  *********************************************************/
  public int handValue(Hand whohand)
  {
    int result = whohand.getHandValue();

    return result;

  }// end handValue method

  /*********************************************************
    discard method - discards a given card from the given
    player's hand or throws an exception if the card is not
    in the hand
  *********************************************************/
  public void discard(Hand whodis, Card discrd) throws ElementNotFoundException
  {
```

```
  Card card=null;
  boolean found = false;
  Iterator<Card> scan = whodis.iterator();
  while (scan.hasNext() && !found)
  {
      card = scan.next();
      if(discrd.equals(card))
      {
        whodis.remove(card);
        found = true;
      }
  }
  if(!found)
      throw new ElementNotFoundException("BlackJack");

}//end discard

/*********************************************************
  blackj method - tests to see if the player's hand has
  a value of 21
*********************************************************/
public boolean blackj()
{
  boolean result = false;

  if(player.getHandValue() == 21)
    result = true;

  return result;

}//end blackj

/*********************************************************
  bust method - tests a given player's hand to see if they
  have gone over 21
*********************************************************/
public boolean bust(Hand whobust)
{
  boolean result = false;

  if(whobust.getHandValue() > 21)
    result = true;

  return result;

}//end bust

/*********************************************************
   dealerPlays method - adds cards to the dealer's hand
   until the value is >= 16
*********************************************************/
public Hand dealerPlays()
{
  Hand result = dealer;

  while(dealer.getHandValue() <= 16)
  {
    dealer.newCard(newdeck);
  }

  return result;

}//end dealerPlays


/*********************************************************
  winner method - determines the winner of the game
*********************************************************/
public String winner()
```

```
  {
    String result = "";
    if((player.getHandValue() < dealer.getHandValue()) &&
                dealer.getHandValue() <= 21 )
      result = "Lose";
    else if ((player.getHandValue() == dealer.getHandValue()) &&
                dealer.getHandValue() <= 21 )
      result = "Push";
    else
      result = "Win";

    return result;

  }//end winner



}//end Blackjack
```