**BlackJackTester.java**

//\*\*\* 1) Create a constant of datatype int, named "CARDS\_IN\_DECK",  
//\*\*\* having a value of 52.  
//\*\*\* 2) Create a constant of datatype int, named "FIRST\_CARD\_IN\_HAND",  
//\*\*\* having a value of 0.  
//\*\*\* 3) Create a constant of datatype int, named "SECOND\_CARD\_IN\_HAND",  
//\*\*\* having a value of 1.  
//\*\*\* 4) Create a constant of datatype String, named "CHOICE\_HIT",  
//\*\*\* having a value of "H".  
//\*\*\* 5) Create a constant of datatype String, named "CHOICE\_STAY",  
//\*\*\* having a value of "S".  
//\*\*\* 6) Create a constant of datatype int, named "NUMBER\_OF\_SHUFFLES",  
//\*\*\* having a value of 5.  
//\*\*\* 7) Declare a variable of datatype boolean named "playerHit".  
//\*\*\* 8) Declare a variable of datatype boolean named "validChoice".  
//\*\*\* 9) Declare a variable of datatype String named "choice".  
//\*\*\* 10) Declare a variable of datatype Scanner named "input", and  
//\*\*\* assign a new Scanner instance from the console to it.

//\*\*\* 11) Create and assign values to three reference variables  
//\*\*\* per the following:  
//\*\*\* a) First reference variable:  
//\*\*\* A) Datatype is "BlackJack" and named "game1".  
//\*\*\* B) Using the one-arg constructor, create a new "BlackJack"  
//\*\*\* instance and assign to "game1".  
//\*\*\* C) Pass the named constant "CARDS\_IN\_DECK" as the one  
//\*\*\* argument to the constructor.  
//\*\*\* b) Second reference variable:  
//\*\*\* A) Datatype is "BlackJackPlayer" and named "player".  
//\*\*\* NOTE: "BlackJackPlayer" is an inner class so you will  
//\*\*\* have to reference it through the outer class.  
//\*\*\* B) Using the getter and the appropriate existing reference  
//\*\*\* variable, assign the contents of property  
//\*\*\* "player" (which is contained in class "BlackJack"), to  
//\*\*\* reference variable "player".  
//\*\*\* b) Third reference variable:  
//\*\*\* A) Datatype is "BlackJackPlayer" and named "dealer".  
//\*\*\* NOTE: "BlackJackPlayer" is an inner class so you will  
//\*\*\* have to reference it through the outer class.  
//\*\*\* B) Using the getter and the appropriate existing reference  
//\*\*\* variable, assign the contents of property  
//\*\*\* "dealer" (which is contained in class "BlackJack"), to  
//\*\*\* reference variable "dealer".  
//\*\*\* 12) Write a for-loop that iterates named constant  
//\*\*\* "NUMBER\_OF\_SHUFFLES" number of times and executing  
//\*\*\* the following line of code:  
//\*\*\* a) Invoke the "shuffleDeck" method on the appropriate existing  
//\*\*\* reference variable.  
//\*\*\* 13) Invoke the "dealCard" method on the appropriate existing  
//\*\*\* reference variable passing reference variable "player" as  
//\*\*\* the one argument.  
//\*\*\* 14) Invoke the "dealCard" method on the appropriate existing  
//\*\*\* reference variable passing reference variable "dealer" as  
//\*\*\* the one argument.

//\*\*\* 15) Invoke the "getCard" method on the reference variable  
//\*\*\* "dealer" passing the named constant "FIRST\_CARD\_IN\_HAND"  
//\*\*\* as the one argument, and send the return value to the  
//\*\*\* console (i.e. println).

//\*\*\* 16) Repeat Step 13.  
//\*\*\* 17) Repeat Step 14.  
//\*\*\* 18) Invoke the "showCurrentHand" method passing the reference  
//\*\*\* variable "player" as the one argument.  
//\*\*\* NOTE: You will not use a reference variable for invocation  
//\*\*\* because this method will be contained in this class.  
//\*\*\* You have not defined this method yet. You will at the  
//\*\*\* bottom of this class.  
//\*\*\* 19) Invoke the "showCurrentScore" method passing the reference  
//\*\*\* variable "player" as the one argument.  
//\*\*\* NOTE: You will not use a reference variable for invocation  
//\*\*\* because this method will be contained in this class.  
//\*\*\* You have not defined this method yet. You will at the  
//\*\*\* bottom of this class.  
//\*\*\* 20) Write an if-statement using the following code as the  
//\*\*\* boolean condition:  
//\*\*\* a) Invoke the "checkBust" method on reference variable "player".  
//\*\*\* 21) Execute the following line of code if the boolean  
//\*\*\* condition evaluates to true:  
//\*\*\* a) Assign the value of "false" to the boolean variable  
//\*\*\* "playerHit".  
//\*\*\* 22) Execute the following line of code if the boolean  
//\*\*\* condition evaluates to false:  
//\*\*\* NOTE: This is the "else" statement associated with the  
//\*\*\* if-statement in Step 20.  
//\*\*\* a) Assign the value of "true" to the boolean variable  
//\*\*\* "playerHit".  
//\*\*\* 23) Write a while-loop using the boolean variable "playerHit"  
//\*\*\* as the boolean condition.  
//\*\*\* 24) Inside this while-loop will be the following code:  
//\*\*\* a) Write a do-while loop using the negation of the boolean  
//\*\*\* variable "validChoice" as the boolean condition.  
//\*\*\* b) Inside this do-while loop will be the following code:  
//\*\*\* A) Prompt the user with the following line of code:  
//\*\*\* 1) System.out.print("\nWould you like to Hit(H) or Stay(S): ");  
//\*\*\* B) Invoke the "nextLine" on the variable "input" and assign the  
//\*\*\* return value to variable "choice".  
//\*\*\* C) Write an if-statement to test if variable "choice" is  
//\*\*\* not equal to the named constant "CHOICE\_HIT or the named  
//\*\*\* constant "CHOICE\_STAY".  
//\*\*\* HINT: You will need to make the comparison  
//\*\*\* case-insensitive (i.e. case does not matter).  
//\*\*\* NOTE: This is a test for invalid data. If the boolean  
//\*\*\* condition in Step C evaluates to true, then the  
//\*\*\* user input data is invalid. You will need to use  
//\*\*\* a compound (two) boolean expression joined with  
//\*\*\* a logical operator.  
//\*\*\* D) Execute the following line of code if the boolean  
//\*\*\* condition in Step C evaluates to true:  
//\*\*\* 1) Assign the value of "false" to the boolean variable  
//\*\*\* "validChoice".  
//\*\*\* E) Execute the following line of code if the boolean  
//\*\*\* condition in Step C evaluates to false:  
//\*\*\* NOTE: This is the "else" statement associated with the  
//\*\*\* if-statement in Step C.  
//\*\*\* 1) Assign the value of "true" to the boolean variable  
//\*\*\* "validChoice".  
//\*\*\* c) Write an if-statement to test if variable "choice" is equal  
//\*\*\* to the named constant "CHOICE\_HIT".  
//\*\*\* HINT: You will need to make the comparison  
//\*\*\* case-insensitive (i.e. case should not matter).  
//\*\*\* d) Execute the following block of code if the boolean  
//\*\*\* condition in Step c evaluates to true:  
//\*\*\* A) Invoke the "dealCardWithHit" method on the appropriate  
//\*\*\* existing reference variable passing reference variable  
//\*\*\* "player" as the one argument.  
//\*\*\* B) Invoke the "showCurrentHand" method passing the reference  
//\*\*\* variable "player" as the one argument.  
//\*\*\* NOTE: You will not use a reference variable for invocation  
//\*\*\* because this method will be contained in this class.  
//\*\*\* You have not defined this method yet. You will at  
//\*\*\* the bottom of this class.  
//\*\*\* C) Invoke the "showCurrentScore" method passing the reference  
//\*\*\* variable "player" as the one argument.  
//\*\*\* NOTE: You will not use a reference variable for invocation  
//\*\*\* because this method will be contained in this class.  
//\*\*\* You have not defined this method yet. You will at  
//\*\*\* the bottom of this class.  
//\*\*\* D) Write an if-statement using the following code as the  
//\*\*\* boolean condition:  
//\*\*\* 1) Invoke the "checkBust" method on reference variable "player".  
//\*\*\* E) Execute the following line of code if the boolean  
//\*\*\* condition evaluates to true:  
//\*\*\* 1) Assign the value of "false" to the boolean variable  
//\*\*\* "playerHit".  
//\*\*\* F) Execute the following line of code if the boolean  
//\*\*\* condition evaluates to false:  
//\*\*\* NOTE: This is the "else" statement associated with the  
//\*\*\* if-statement in Step D.  
//\*\*\* 1) Assign the value of "true" to the boolean variable  
//\*\*\* "playerHit".  
//\*\*\* e) Execute the following line of code if the boolean condition  
//\*\*\* evaluates to false:  
//\*\*\* NOTE: This is the "else" statement associated with the  
//\*\*\* if-statement in Step c above concerning testing  
//\*\*\* variable "choice" against the named constant  
//\*\*\* "CHOICE\_HIT".  
//\*\*\* A) Assign the value of "false" to the boolean variable  
//\*\*\* "playerHit".  
//\*\*\* 25) Write an if-statement using the following code as the  
//\*\*\* boolean condition:  
//\*\*\* a) The negation of the invocation of the "isBust" method on  
//\*\*\* the reference variable "player".  
//\*\*\* 26) Execute the following block of code if the boolean condition  
//\*\*\* evaluates to true:  
//\*\*\* a) System.out.println("\nThe " + dealer.getName() + "'s second card is: ");  
//\*\*\* b) Invoke the "getCard" method on the reference variable  
//\*\*\* "dealer" passing the named constant "SECOND\_CARD\_IN\_HAND"  
//\*\*\* as the one argument, and send the return value to the  
//\*\*\* console (i.e. println).  
//\*\*\* c) Invoke the "showCurrentScore" method passing the reference  
//\*\*\* variable "dealer" as the one argument.  
//\*\*\* NOTE: You will not use a reference variable for invocation  
//\*\*\* because this method will be contained in this class.  
//\*\*\* You have not defined this method yet. You will at  
//\*\*\* the bottom of this class.  
//\*\*\* d) Write a while-loop using the following code as the  
//\*\*\* boolean condition:  
//\*\*\* A) Retrieve the current score for the dealer by invoking  
//\*\*\* the getter for property "currentScore" on reference  
//\*\*\* variable "dealer". Is this value less-than-or-equal-to  
//\*\*\* the value of the named constant "DEALER\_HIT\_STAY\_THRESHOLD?  
//\*\*\* NOTE: The named constant "DEALER\_HIT\_STAY\_THRESHOLD is  
//\*\*\* in class "BlackJack" so you will need to use the  
//\*\*\* appropriate existing reference variable on this  
//\*\*\* named constant.  
//\*\*\* B) Execute the following line of code if the boolean  
//\*\*\* condition evaluates to true:  
//\*\*\* 1) Invoke the "dealCardWithHit" method on the appropriate  
//\*\*\* existing reference variable passing reference variable  
//\*\*\* "dealer" as the one argument.  
//\*\*\* 27) Invoke the "checkBust" method on reference variable "dealer".  
//\*\*\* 28) Display the number of hits by the dealer using the following  
//\*\*\* line of code:  
//\*\*\* a) System.out.print("\nAfter 'Hitting' " + dealer.getNumberHits() + " time(s),");  
//\*\*\* 29) Invoke the "showCurrentHand" method passing the reference  
//\*\*\* variable "dealer" as the one argument.  
//\*\*\* NOTE: You will not use a reference variable for invocation  
//\*\*\* because this method will be contained in this class.  
//\*\*\* You have not defined this method yet. You will at the  
//\*\*\* bottom of this class.  
//\*\*\* 30) Invoke the "showCurrentScore" method passing the reference  
//\*\*\* variable "dealer" as the one argument.  
//\*\*\* NOTE: You will not use a reference variable for invocation  
//\*\*\* because this method will be contained in this class.  
//\*\*\* You have not defined this method yet. You will at the  
//\*\*\* bottom of this class.  
//\*\*\* 31) Invoke the "determineOutcome" method on the appropriate existing  
//\*\*\* reference variable passing reference variable "player" as the  
//\*\*\* first argument and reference variable "dealer" as the second  
//\*\*\* argument.

//\*\*\* 32) Outside of the "main" method but inside class "BlackJackTester", create  
//\*\*\* the two methods as follows:  
//\*\*\* a) First method:  
//\*\*\* A) A static method named "showCurrentHand" that takes one  
//\*\*\* argument and returns no value.  
//\*\*\* B) The one argument is of datatype "BlackJackPlayer" and named "p".  
//\*\*\* NOTE: "BlackJackPlayer" is an inner class so you will  
//\*\*\* have to reference it through the outer class.  
//\*\*\* C) The method body consists of two lines of code as follows:  
//\*\*\* 1) System.out.println("\n" + p.getName() + "'s current hand is:");  
//\*\*\* 2) Invoke the "displayFormattedHand" method on reference variable "p".  
//\*\*\* b) Second method:  
//\*\*\* A) A static method named "showCurrentScore" that takes one  
//\*\*\* argument and returns no value.  
//\*\*\* B) The one argument is of datatype "BlackJackPlayer" and named "p".  
//\*\*\* NOTE: "BlackJackPlayer" is an inner class so you will  
//\*\*\* have to reference it through the outer class.  
//\*\*\* C) The method body consists of two lines of code as follows:  
//\*\*\* 1) System.out.print("\n" + p.getName() + "'s current score is: ");  
//\*\*\* 2) Retrieve the current score by invoking the getter for  
//\*\*\* property "currentScore" on reference variable "p", and  
//\*\*\* send the return value to the console (i.e. println).  
//\*\*\*