1. Code the **domain class called BingoCard**:
   1. 2-D array of int (instance variable), with 5 rows and 5 cols.
   2. Code the Constructor:
      1. Nested loop, with random numbers moved to 2-d array
      2. Be sure to put the random numbers according to their column

Example: Col 0 = B in BINGO, should only have   
random numbers between 1 – 15

* 1. Code checkBingo(int num) method (will be called by driver 50 times)
     1. Receive num, and check to see if it exists anywhere in 2-D array.
     2. If it finds it, a 0 will be put in the number’s place
  2. Code boolean gotBingo() method (will called by driver 1 time)
     1. Loops to check every column, to see if it finds 5 consecutive 0’s
     2. Loops to check every row, to see if it finds 5 consecutive 0’s
     3. Loops to check the 2 diagonals, to see if it finds 5 consecutive 0’s
     4. If any of these loops find 5 zeroes, the method will return TRUE;  
        otherwise, the method will return FALSE
  3. Code toString() method that will be indirectly referenced each time you print the BingoCard:
     1. Use /t and /n to concatenate one large String containing the entire BINGO card, with rows and columns, and the “BINGO” title above that.

1. Code the Driver Class, BingoGame:
   1. In main:
      1. ***Loop*** while the user wants to keep playing BINGO
         1. Instantiate a new BingoCard object, ***myCard***
         2. Call a method, playGame() in Driver:
            1. Loop 50 times, to do the following:

Generate a random # between 1 – 75

Call the checkBingo(..) method (in domain)  
Example: ***myCard.checkBingo(ranNum)***

* + - * 1. Call a method, determineWinner() in Driver:

Call gotBingo() method in the domain

If gotBingo() returns a TRUE,   
add 1 to totalWins, and say “You Won!”

Else if gotBingo() returns a FALSE,

Say “Sorry you did not win this time.”

Ask if you wish to play again. The user’s answer

Will cause the outermost loop in main to continue

Or to stop.

ii. Display the total number of games won.