**BINGO for COP2250 – A Visualization**

**int[][] BingoCardArray = new int[5][5];  
  
Use a nested for-loop, to populate the 2-D array representing 5 columns for B – I – N – G – O, where   
row 1 =   
 col 1 = random # 1- 15  
 col 2 = random # 16 – 30  
 col 3 = random # 31 – 45  
 col 4 = random # 46 – 60  
 col 5 = random # 61 – 75**

The constructor of the **BingoCard** class will generate random numbers to fill in the 2-D array representing the BINGO card.

**row 2 =   
 col 1 = random # 1- 15  
 col 2 = random # 16 – 30  
 col 3 = random # 31 – 45  
 col 4 = random # 46 – 60  
 col 5 = random # 61 - 75**

**row 3 =   
 col 1 = random # 1- 15  
 col 2 = random # 16 – 30  
 col 3 = random # 31 – 45  
 col 4 = random # 46 – 60  
 col 5 = random # 61 – 75**

**row 4 =   
 col 1 = random # 1- 15  
 col 2 = random # 16 – 30  
 col 3 = random # 31 – 45  
 col 4 = random # 46 – 60  
 col 5 = random # 61 - 75**

**row 5 =   
 col 1 = random # 1- 15  
 col 2 = random # 16 – 30  
 col 3 = random # 31 – 45  
 col 4 = random # 46 – 60  
 col 5 = random # 61 – 75**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**The playGame() method in the BingoGame (Driver) class will loop to generate 50 random numbers in the range of 1 to 75. This simulates drawing 50 numbers in a BINGO game. The method will pass the number to a method, checkBingo() in the domain class*, BingoCard*, which will check if the number received as a parameter is in the B range (1 – 15), I range (16 – 30), N range (31 – 45), G range (46 -60) or O range (61 – 75). According to the range it is in, the method will check each of the cells in designated column (col 0 = B; col 1 = I; col 2 = N; col 3 = G; col 4 = O). Example: randomNum 27 was randomly generated, which belongs to the I-range, column 1, checking rows 0 to 4.**

**for (int row = 0; row <= 4; row++)  
{   
 //hard-code column 1 since checking I-column:  
 if (randomNum == BingoCardArray[row][1])**

**{  
 BingoCardArray[row][1] = 0;  
 }  
}**

**etc. //the 0 being moved to the number simulates putting a chip on the Bingo card, marking that the  
 // number was called.**

**The gotBingo() method in the Bingo (Domain) class will return TRUE if 5 numbers were set to 0 either horizontally, vertically, or diagonally. Otherwise, the method will return FALSE.**

**Each horizontal check is a set of if-else if statements**

**for (int row = 0; row < 5; row++)  
{  
 int rowTotal = 0;**

**for (int col = 0; col < 5; col++)  
 {**

**//add up all the column values in that row, and see if they add up to 0.**

**….**

**}**

**}**

**Each vertical check is a set of if-else if statements**

**for (int col = 0; col < 5; col++)  
{  
 int colTotal = 0;**

**for (int row = 0; row < 5; row++)  
 {**

**//add up all the row values in that column, and see if they add up to 0.**

**….**

**}**

**}**

**//for diagonals check….  
for (int row = 0; row < 5; row++)  
{  
 int diagonal1Total = 0;**

**int diagonal2Total = 0;**

**for (int col = 0; col < 5; col++)  
 {**

**//figure out if it is a diagonal going one way**

**//figure out if it is a diagonal going the other way**

**….**

**}**

**}**

**When gotBingo() ends, it will have either returned a TRUE or a FALSE.**

**In the driver class, the totalGamesWon is a global variable for keeping track of how many games the user wins instead of the computer. Before the**

**The determineWinner() method in the Bingo (Driver) class will call the gotBIngo() method in the domain class, and if it returns true, it will add 1 to the totalGamesWon, otherwise it will leave the totalGamesWon alone.**

**Finally, the main method will contain a do-while loop that will execute at least once, and do the following:**

1. **Instantiate the BingoCard class into an object**
2. **Call the playGame() method in the driver class**
3. **The playGame() method will call the checkBingo(int aNum) in the domain class**
4. **Call the determineWinner() method in the driver class, which calls the gotBingo() method in the domain class, and add 1 to totalGamesWom whenever gotBingo() returns true.**
5. **Ask user if he/she wants to repeat and play BINGO again.**