

Name(s) _____ Period _____ Date _____

Activity Guide - BingoCaller FRQ



BingoCaller FRQ

In a game called Bingo, players mark off numbers on cards as the numbers are called randomly by a "caller." Each bingo card has 25 spaces arranged in a square of 5 rows and 5 columns. The columns are labeled from left to right with the letters "B", "I", "N", "G", "O".

The spaces in the card are assigned random values as follows:

- Each space in the "B" column contains a number from 1 to 15, inclusive.
- Each space in the "I" column contains a number from 16 to 30, inclusive.
- Each space in the "N" column contains a number from 31 to 45, inclusive.
- Each space in the "G" column contains a number from 46 to 60, inclusive.
- Each space in the "O" column contains a number from 61 to 75, inclusive.

When a game of Bingo is played, each number can only be called once. A bingo call contains a column letter followed by a value (for example, "B7").

The **BingoCaller** class is designed to simulate a caller in a bingo game and keeps track of the numbers that have been called. The incomplete class declaration is shown below. You will be asked to implement two methods.

```
public class BingoCaller {

    /** COLUMN_LETTERS[i] represents the letter located at a particular index value i */
    private static final String[] COLUMN_LETTERS = {"B", "I", "N", "G", "O"};

    /** numbersCalled[i] represents whether or not the value i-1 has been called */
    private static boolean[] numbersCalled = new boolean[75];

    /* Returns a boolean indicating whether a value has been called.
     * Precondition: numbersCalled contains 75 boolean values
     * Postcondition: numbersCalled is unchanged
     */
    public static boolean hasBeenCalled(int num) {
        /** to be implemented in part (a) */
    }

    /* Returns a random value based on the given letter
     * Precondition: letter is not null or an empty string
     * Postcondition: random value is between 1-15 for the letter "B", 16-30 for the letter
     *                    "I", 31-45 for the letter "N", 46-60 for the letter "G", or
     *                    61-75 for the letter "O", inclusive
     */
    public static int getRandomNumber(String letter) {
        /** to be implemented in part (b) */
    }

    /* Returns a String containing the column letter and value
     * of a randomly generated bingo call.
     * Precondition: numbersCalled contains 75 boolean values
     * Postcondition: numbersCalled is updated to indicate that a value has been called
     */
    public static String makeCall() {
        /** to be implemented in part (c) */
    }

    // There may be instance variables, constructors, and methods that are not shown.
}
```

- (a) Write the `hasBeenCalled()` method, which returns a `boolean` value indicating whether a given number has been previously called. If the given number is invalid (less than `1` or greater than `75`), the method returns `false`.

The 1D array `numbersCalled` is used to keep track of whether a value was called. It contains 75 `boolean` values that are initialized to `false`. An element of the 1D array `numbersCalled` is marked `true` if the value at `index - 1` has been called.

For example, to check whether `1` has been called by the caller, the following call could be made:

```
hasBeenCalled(1);    // Checks the first element of numbersCalled
```

Complete the `hasBeenCalled()` method.

```
/** Returns a boolean indicating whether a value has been called.
 * Precondition: numbersCalled contains 75 boolean values
 * Postcondition: numbersCalled is unchanged
 */
public static boolean hasBeenCalled(int num)
```

Class information for this question

```
public class BingoCaller

private static final String[] COLUMN_LETTERS
private static boolean[] numbersCalled

public static boolean hasBeenCalled(int num)
public static int getRandomNumber(String letter)
public static String makeCall()
```

(b) Write the `getRandomNumber()` method, which returns a random number that falls within the range of possible values for a column.

- If `letter` is "B", return a random number from 1 to 15, inclusive.
- If `letter` is "I", return a random number from 16 to 30, inclusive.
- If `letter` is "N", return a random number from 31 to 45, inclusive.
- If `letter` is "G", return a random number from 46 to 60, inclusive.
- If `letter` is "O", return a random number from 61 to 75, inclusive.

Complete the `getRandomNumber()` method.

```
/** Returns a random value based on the given letter
 * Precondition: letter is not null or an empty string
 * Postcondition: random value is between 1-15 for the letter "B", 16-30 for the
 *                   letter "I", 31-45 for the letter "N", 46-60 for the letter "G",
 *                   or 61-75 for the letter "O", inclusive
 */
public static int getRandomNumber(String letter)
```

Class information for this question

```
public class BingoCaller
```

```
private static final String[] COLUMN_LETTERS
private static boolean[] numbersCalled
```

```
public static boolean hasBeenCalled(int num)
public static int getRandomNumber(String letter)
public static String makeCall()
```

- (c) Write the `makeCall()` method, which randomly selects a column letter from the `COLUMN_LETTERS` array and randomly selects a number that falls within the range of possible values for the column using the method `getRandomNumber()`.

The `makeCall()` method should check if the number has been called using the method `hasBeenCalled()`. If `hasBeenCalled()` returns `true`, return a `String` containing the text "`{randomLetter}{randomValue} has already been called`", where `randomLetter` is the random letter from `COLUMN_LETTERS` and `randomValue` is the random number returned from `getRandomNumber()`. Otherwise, return a `String` containing the bingo call and update the element in the `numbersCalled` array to indicate that it has been called.

Complete the `makeCall()` method.

Assume that `hasBeenCalled()` and `getRandomNumber()` work as specified, regardless of what you wrote in parts (a) and (b). You must use `hasBeenCalled()` and `getRandomNumber()` appropriately to receive full credit.

```
/* Returns a String containing the column letter and value
 * of a randomly generated bingo call.
 * Precondition: numbersCalled contains 75 boolean values
 * Postcondition: numbersCalled is updated to indicate that a
 * value has been called
 */
public static String makeCall()
```

Class information for this question

```
public class BingoCaller
```

```
private static final String[] COLUMN_LETTERS
private static boolean[] numbersCalled
```

```
public static boolean hasBeenCalled(int num)
public static int getRandomNumber(String letter)
public static String makeCall()
```

Reflection

Assess your solution using the Scoring Guidelines and provide brief responses (1-2 sentences each) to the following questions:

1. For which rows on the Scoring Guidelines did you get the full point?
2. What information did you highlight and/or annotate in the original FRQ that helped you get the full point for these rows?
3. Which rows on the Scoring Guidelines are you still confused about?
4. If you were giving tips to other students completing this FRQ, what advice would you give them?

Rate your experience answering this FRQ by coloring in the stars.

I understood what the FRQ was asking me to do.



I am confident I can tackle similar FRQs.



The materials covered in this unit prepared me for this FRQ.



I'm looking forward to answering more FRQs throughout the course.

