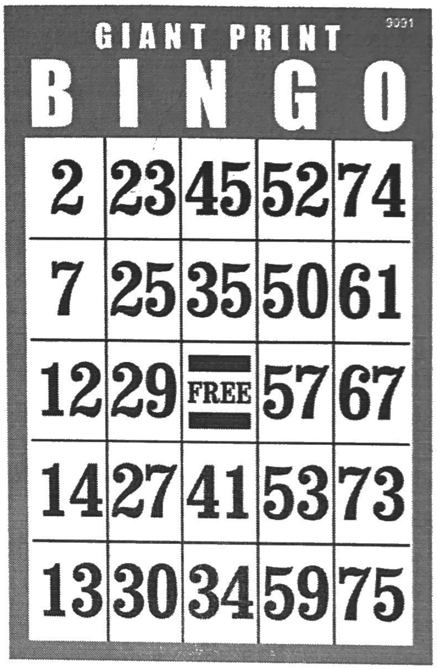
Challenge VI
BingoCard
COSC 2329 Component-Based Programming
Deadline: Friday, May 4, 2018 @ 11:59 PM
Late Deadline: No Lates Accepted



Assignment
Your assignment is to create four properly documented constructs in Java:

- a `BingoCardRowListBasedImpl_`*`LastName`* class (e.g., `BingoCardRowListBasedImpl_Kart`), which extends the class `BingoCardRowListBased_Abstract` class
- a `BingoCardRowSetBasedImpl_`*`LastName`* class (e.g., `BingoCardRowSetBasedImpl_Kart`), which extends the class `BingoCardRowSetBased_Abstract` class
- a `BingoCardColumnListBasedImpl_`*`LastName`* class (e.g., `BingoCardColumnListBasedImpl_Kart`), which extends the class `BingoCardColumnListBased_Abstract` class
- a `BingoCardDiagonalListBasedImpl_`*`LastName`* class (e.g., `BingoCardDiagonalListBasedImpl_Kart`), which extends the class `BingoCardDiagonalListBased_Abstract` class

`BingoCard` Concept

A `BingoCard` provides three main services, it can:

- report on what number is in each row and column
- mark entries
- report on whether it contains a certain number
- report on which numbers are marked
- report on whether it is in a winning configuration

Note that the `BingoCard`:

- is told which numbers are on the card in each row and column via a client-facing constructor call
- always reports that the free space is marked
- needs to have something added to the precondition for the mark() method

Deliverables

- A .zip file uploaded to Canvas that contains the following files (Look for "BingoCard" assignment or similar):
  - `BingoCardRowListBasedImpl_`*`LastName`*`.java` (e.g., `BingoCardRowListBasedImpl_Kart.java`)
  - `BingoCardRowSetBasedImpl_`*`LastName`*`.java` (e.g., `BingoCardRowSetBasedImpl_Kart.java`)
  - `BingoCardColumnListBasedImpl_`*`LastName`*`.java` (e.g., `BingoCardColumnListBasedImpl_Kart.java`)
  - `BingoCardDiagonalListBasedImpl_`*`LastName`*`.java` (e.g., `BingoCardDiagonalListBasedImpl_Kart.java`)
  - `BingoCardUtils_`*`LastName`*`.java` (e.g., `BingoCardUtils_Kart.java`)
  - Any supporting Utils/classes/interfaces that you created (note that the filename suffix on these files must be _*LastName*)
  - **Do not turn in the `BingoCard` interface or any of the abstract classes!** (What does this imply?)

Rules

- **ENSURE THAT YOU UNDERSTAND AND YOUR IMPLs ARE FAITHFUL TO THE MANDATORY INTERNAL REPRESENTATIONS! MY TESTS WILL CHECK FOR THIS!**
- My test cases do not change based on your submission.
- I will not violate the preconditions on my interface in my test cases.
- **USE THE PACKAGE 'bingo' for all of your files!**
- Use the Eclipse IDE
- Ensure that I, with only modest effort, can understand your code
- Ensure that the code is properly documented
- Ensure that the code is properly formatted
- **Test your code!** (What test cases can you think of?)
  - What are the "middle-of-the-road" (i.e., "vanilla") test cases?
  - What are the "corner" (i.e., "extreme") test cases?
- **Test your code some more!** (What other test cases can you think of?)
- Code that doesn't compile will not pass any tests and receive a score of 0

- Ensure that your files follow the naming convention under Deliverables
- WARNING: This specification may be misleading or incomplete! Part of the assignment is to read the assignment early, think about it, and ask any clarifying questions!

Java Interface

```
public interface BingoCard {
        public static final int ROW_COUNT = 5;
        public static final int COLUMN_COUNT = 5;
        public static final int FREE_SPACE_ROW = 3;
        public static final int FREE_SPACE_COLUMN = 3;
        public static final Integer FREE_SPACE = null;

        //part of pre: 1 <= row <= ROW_COUNT
        //part of pre: 1 <= column <= COLUMN_COUNT
        //part of post: column == 1 ("B") ==> 1 <= rv <= 15
        //part of post: column == 2 ("I") ==> 16 <= rv <= 30
        //part of post: column == 3 ("N") ==> ((31 <= rv <= 45) ||
        //                                    ((row = 3) && (rv == FREE_SPACE)));
        //part of post: column == 4 ("G") ==> 46 <= rv <= 60
        //part of post: column == 5 ("O") ==> 61 <= rv <= 75
        //part of post: ((column - 1)*15 + 1) <= rv <= ((column - 1) + 1)*15
        //part of post: rv == FREE_SPACE <==>
        //                      ((row == FREE_SPACE_ROW) && (column == FREE_SPACE_COLUMN))
        public Integer getEntry(int row, int column);

        //part of pre: 1 <= number <= 75
        //part of post: contains(number) <==>
        //                      (isMarked(row, column) for some
        //                              1 <= row <= ROW_COUNT, 1 <= column <= COLUMN_COUNT)
        public void mark(int number);

        //pre: true
        //part of post: rv == ((getEntry(1, 1) == number) || (getEntry(1, 2) == number) ||
        //                                  ... || (getEntry(1, COLUMN_COUNT) == number) ||
        //                      (getEntry(2, 1) == number) || (getEntry(2, 2) == number) ||
        //                                  ... || (getEntry(2, COLUMN_COUNT) == number) ||
        //...   (getEntry(ROW_COUNT, 1) == number) || (getEntry(ROW_COUNT, 2) == number) ||
        //                              ... || (getEntry(ROW_COUNT, COLUMN_COUNT) == number))
        public boolean contains(int number);

        //part of pre: 1 <= row <= ROW_COUNT
        //part of pre: 1 <= column <= COLUMN_COUNT
        public boolean isMarked(int row, int column);

        //pre: true
        //post: left to student
        public boolean isWinner();
}
```

Java Impl Representations

BingoCardRowListBasedImpl_*LastName*:

```
Internal Representative                View from Interface
          [                        --------------------------
 [ 2, 23, 45, 52, 74]              | (2)| 23 | 45 | 52 | 74 |
 [ 7, 25, 35, 50, 61]              --------------------------
   [12, 29, 57, 67]                |  7 |(25)| 35 | 50 | 61 |
 [14, 27, 41, 53, 73]              --------------------------
 [13, 30, 34, 59, 75]              | 12 | 29 |(FS)| 57 | 67 |
          ]                        --------------------------
                                   | 14 | 27 | 41 |(53)| 73 |
    {53, 25, 75, 2}                --------------------------
                                   | 13 | 30 | 34 | 59 |(75)|
                                   --------------------------
```

BingoCardRowSetBasedImpl_*LastName*:

```
Internal Representative                View from Interface
          [                        --------------------------
 { 2, 52, 23, 74, 45}              | (2)| 23 | 45 | 52 | 74 |
 {50, 35,  7, 25, 61}              --------------------------
 { *, 67, 57, 12, 29}              |  7 |(25)| 35 | 50 | 61 |
 {53, 41, 73, 27, 14}              --------------------------
 {34, 59, 75, 13, 30}              | 12 | 29 |(FS)| 57 | 67 |
          ]                        --------------------------
                                   | 14 | 27 | 41 |(53)| 73 |
    {75, 53, 2, *, 25}             --------------------------
                                   | 13 | 30 | 34 | 59 |(75)|
                                   --------------------------
```

*(Note that * indicates FREE_SPACE)*

BingoCardColumnListBasedImpl_*LastName*:

Internal Representative

```
  1,1    2,1  3,1  4,1  5,1
               [
    [ 2,  7, 12, 14, 13]
1,2 [23, 25, 29, 27, 30] 5,2
1,3 [45, 35,  *, 41, 34] 5,3
1,4 [52, 50, 57, 53, 59] 5,4
1,5 [74, 61, 67, 73, 75] 5,5
               ]
```

{*, 25, 75, 2, 53}
*(Note that * indicates FREE_SPACE)*

View from Interface

```
---------------------------------
| (2)| 23 | 45 | 52 | 74 |
---------------------------------
|  7 |(25)| 35 | 50 | 61 |
---------------------------------
| 12 | 29 |(FS)| 57 | 67 |
---------------------------------
| 14 | 27 | 41 |(53)| 73 |
---------------------------------
| 13 | 30 | 34 | 59 |(75)|
---------------------------------
```

BingoCardDiagonalListBasedImpl_*LastName*:

Internal Representative

```
       [
      [13]
    [30, 14]
  [34, 27, 12]
[59, 41, 29,  7]
[75, 53, 25,  2]
[73, 57, 35, 23]
  [67, 50, 45]
    [61, 52]
      [74]
       ]
```

{25, 2, 75, 53}

View from Interface

```
---------------------------------
| (2)| 23 | 45 | 52 | 74 |
---------------------------------
|  7 |(25)| 35 | 50 | 61 |
---------------------------------
| 12 | 29 |(FS)| 57 | 67 |
---------------------------------
| 14 | 27 | 41 |(53)| 73 |
---------------------------------
| 13 | 30 | 34 | 59 |(75)|
---------------------------------
```

```
      [
    [(5,1)]                          5-1 = 4
   [(5,2),(4,1)]                     5-2 = 3
  [(5,3),(4,2),(3,1)]                4-2 = 2
 [(5,4),(4,3)(3,2)(2,1)]             3-2 = 1
 [(5,5),(4,4),(2,2),(1,1)]           1-1 = 0
 [(4,5),(3,4),(2,3),(1,2)]           4-5 = -1
  [(3,5),(2,4),(1,3)]                2-4 = -2
   [(2,5),(1,4)]                     1-4 = -3
    [(1,5)]                          1-5 = -4
      ]
```