FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ

A PROJECT REPORT ON

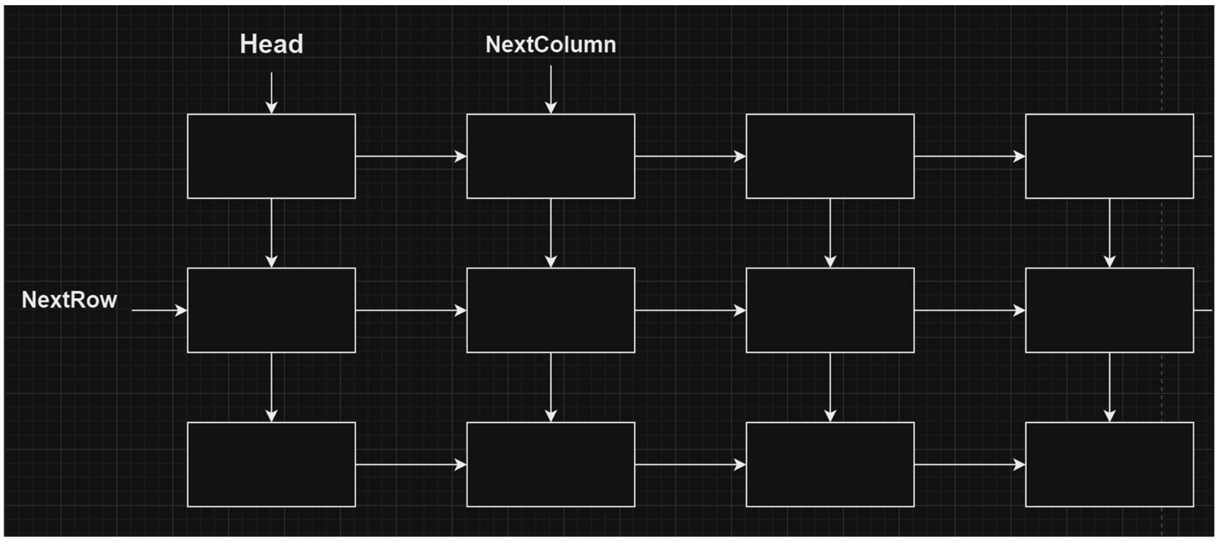
Multi-Player Tombola (Bingo) Game with Multi Linked List

SUBMITTED BY

NAME	:	Fatma Zehra Çılgın
ID	:	2121221014
COURSE	:	BLM10202E Data Structures

DESIGN

Multi-Linked List



Game

FatmaCilginGame and two FatmaCilginCard objects are created. Next, a manually entered card is identified using a 2D array of type Integer and assigned to the card object. A second card object is also randomly generated. Afterwards, the cards of both players are printed on the screen one by one and the game begins. The game loop has both players mark their cards according to the number drawn. For each number drawn, players' cards are printed on the screen and checked for zinc or bingo. If any player makes bingo, the relevant information is printed on the screen and the game ends.

IMPLEMENTATION DETAILS

1. FatmaCilginNode Sınıfı

```java
public class FatmaCilginNode<T> {

    T data;
    FatmaCilginNode<T> nextRow; // Bi
    FatmaCilginNode<T> nextColumn;//E


    public FatmaCilginNode(T data) {
        this.data = data;
        this.nextRow = null;
        this.nextColumn = null;
    }
```

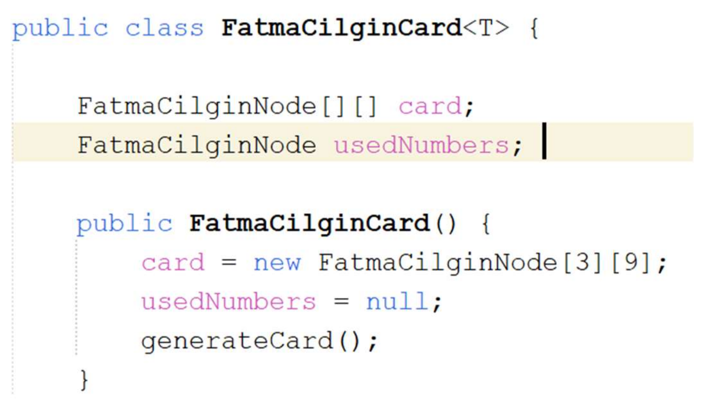
data: Represents the data to be stored in the node.

nextRow: A reference to the node in the next row in the same column. It is of generic type.

nextColumn: A reference that points to the node in the next column in the same row. It is of generic type.

FatmaCilginNode(T data): It is a constructor that creates a node by taking data.

2. FatmaCilginCard Sınıfı

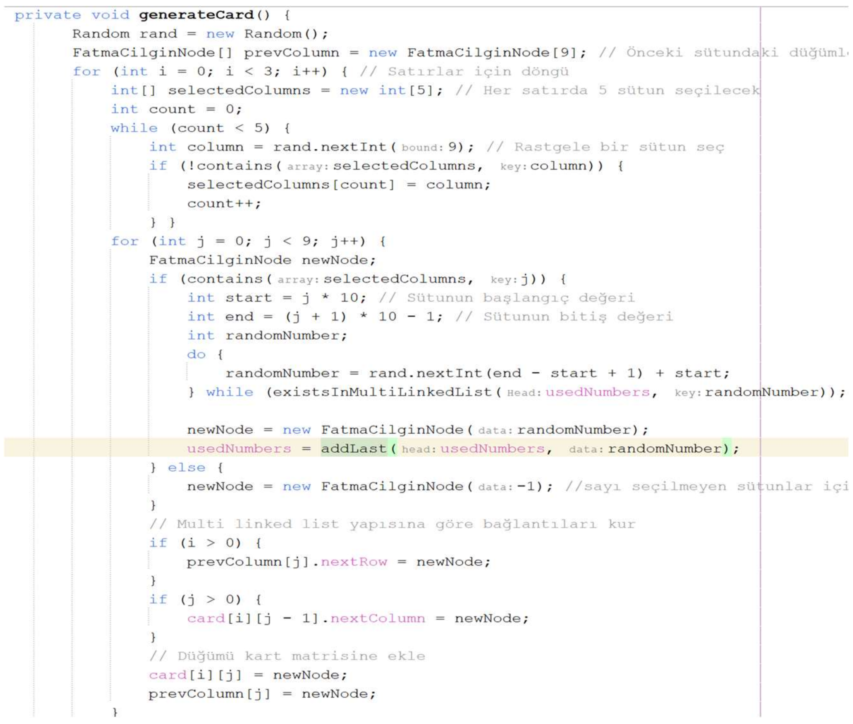


```java
public class FatmaCilginCard<T> {

    FatmaCilginNode[][] card;
    FatmaCilginNode usedNumbers;

    public FatmaCilginCard() {
        card = new FatmaCilginNode[3][9];
        usedNumbers = null;
        generateCard();
    }
```

card: It is a 3x9 matrix representing the bingo card. Each cell is represented by a FatmaCilginNode object.

usedNumbers: It is a linked list that holds used numbers.

FatmaCilginCard(): It is the constructor used to create the card. Generates random cards.



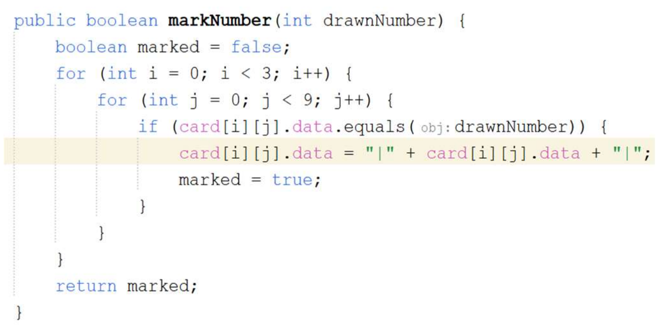
```java
private void generateCard() {
    Random rand = new Random();
    FatmaCilginNode[] prevColumn = new FatmaCilginNode[9]; // Önceki sütundaki düğüml
    for (int i = 0; i < 3; i++) { // Satırlar için döngü
        int[] selectedColumns = new int[5]; // Her satırda 5 sütun seçilecek
        int count = 0;
        while (count < 5) {
            int column = rand.nextInt( bound: 9); // Rastgele bir sütun seç
            if (!contains( array: selectedColumns,  key: column)) {
                selectedColumns[count] = column;
                count++;
            } }
        for (int j = 0; j < 9; j++) {
            FatmaCilginNode newNode;
            if (contains( array: selectedColumns,  key: j)) {
                int start = j * 10; // Sütunun başlangıç değeri
                int end = (j + 1) * 10 - 1; // Sütunun bitiş değeri
                int randomNumber;
                do {
                    randomNumber = rand.nextInt(end - start + 1) + start;
                } while (existsInMultiLinkedList( Head: usedNumbers,  key: randomNumber));

                newNode = new FatmaCilginNode( data: randomNumber);
                usedNumbers = addLast( head: usedNumbers,  data: randomNumber);
            } else {
                newNode = new FatmaCilginNode( data: -1); //sayı seçilmeyen sütunlar içi
            }
            // Multi linked list yapısına göre bağlantıları kur
            if (i > 0) {
                prevColumn[j].nextRow = newNode;
            }
            if (j > 0) {
                card[i][j - 1].nextColumn = newNode;
            }
            // Düğümü kart matrisine ekle
            card[i][j] = newNode;
            prevColumn[j] = newNode;
        }
```

generateCard(): An array of type FatmaCilginNode is created to hold the nodes (prevColumn) in the previous column. This will be used to make the connections when creating the next row. 5 columns are selected for each row. The indexes of the selected columns are stored in the selectedColumns array. A loop is created for each column and random numbers are assigned to each column. Each cell in the selected columns is assigned a random number between the column's starting and ending values. This random number must not have been used before for that column. When a number that has not been used before is found, this number is added to the card matrix by creating a FatmaCilginNode and is also added to the usedNumbers list. By looping over the prevColumn array, connections are made to the nodes in the previous column. This creates a link between the current row and the previous row. For each column, a connection is created between the node in the previous row and the current node. Nodes are added to the card matrix. These steps are completed for all rows and columns.

contains(int[] array, int key): Helper method that checks whether a certain value exists in an array. It returns true if it finds the searched value in the given array, otherwise it returns false.

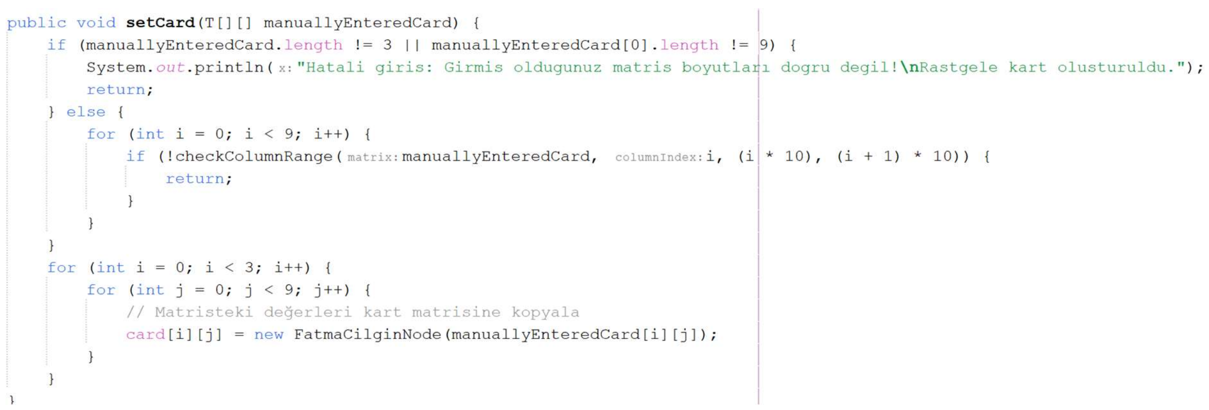
existsInMultiLinkedList(Node<T> startNode, int key): Helper method that checks whether a certain value exists in a multi-linked list. Returns true if it finds the searched value in the list starting from the given starting node, otherwise returns false.

add(Node<T> head, T data): Helper method that adds a new node to the multi-linked list. Creates a new node containing the given data and appends it to the end of the list.



```java
public boolean markNumber(int drawnNumber) {
    boolean marked = false;
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 9; j++) {
            if (card[i][j].data.equals( obj: drawnNumber)) {
                card[i][j].data = "|" + card[i][j].data + "|";
                marked = true;
            }
        }
    }
    return marked;
}
```

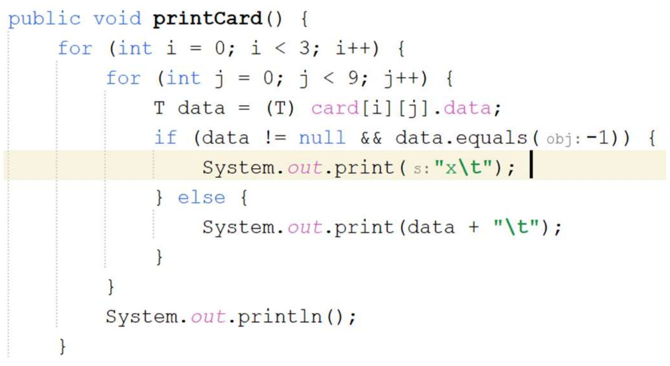
markNumber(int drawnNumber): It checks whether a certain number is marked on the card or not. If so, add "|" to the edges of the number if it is on the card. puts the mark.



```java
public void setCard(T[][] manuallyEnteredCard) {
    if (manuallyEnteredCard.length != 3 || manuallyEnteredCard[0].length != 9) {
        System.out.println( x: "Hatali giris: Girmis oldugunuz matris boyutları dogru degil!\nRastgele kart olusturuldu.");
        return;
    } else {
        for (int i = 0; i < 9; i++) {
            if (!checkColumnRange( matrix: manuallyEnteredCard,  columnIndex: i, (i * 10), (i + 1) * 10)) {
                return;
            }
        }
    }
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 9; j++) {
            // Matristeki değerleri kart matrisine kopyala
            card[i][j] = new FatmaCilginNode(manuallyEnteredCard[i][j]);
        }
    }
}
```

setCard(T[][] manuallyEnteredCard): It takes T[][] manuallyEnteredCard as a parameter. This is the card matrix that the user enters manually, the number of rows is 3 and the number of columns is 9. It checks whether the dimensions of the entered matrix are correct. If it is not a 3x9 matrix, it gives an error message and terminates the process. It then checks whether each column contains numbers within a certain range (between 10 times the column number and 10 times the column number). If there is a number outside this range, it returns an error message and terminates the process. If the matrix is suitable, it copies the values of the matrix to the card matrix.

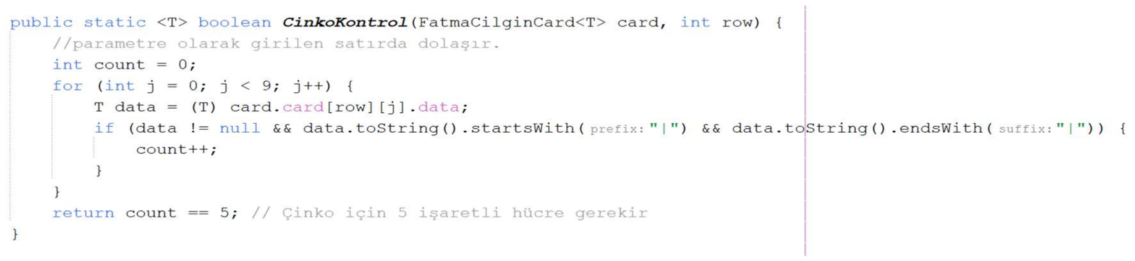
checkColumnRange(T[][] matrix, int columnIndex, int start, int end): Checks whether a given column contains numbers in the given range (for example, 10-19).



```java
public void printCard() {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 9; j++) {
            T data = (T) card[i][j].data;
            if (data != null && data.equals( obj: -1)) {
                System.out.print( s: "x\t");
            } else {
                System.out.print(data + "\t");
            }
        }
        System.out.println();
    }
```

printCard(): This is the method used to print the card. Prints 'x' for cells with '-1'. Prints data for other cells.

3. FatmaCilginGame Sınıfı

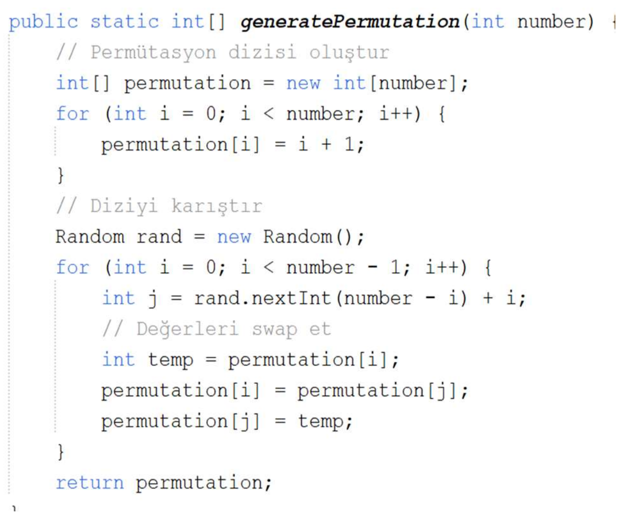


```java
public static <T> boolean CinkoKontrol(FatmaCilginCard<T> card, int row) {
    //parametre olarak girilen satırda dolaşır.
    int count = 0;
    for (int j = 0; j < 9; j++) {
        T data = (T) card.card[row][j].data;
        if (data != null && data.toString().startsWith( prefix: "|") && data.toString().endsWith( suffix: "|")) {
            count++;
        }
    }
    return count == 5; // Çinko için 5 işaretli hücre gerekir
}
```

CinkoKontrol(FatmaCilginCard<T> card, int row): "|" in the line entered as a parameter of the given card Checks the zinc status according to the number. “|” in the line If it starts and ends with , it increases the count by one and returns true when the count becomes 5.

Tombala(FatmaCilginCard<T> card): Checks the bingo status on the given card. It browses all the elements of the card and clicks “|” If it starts and ends with , it increases the count by one and returns true when the count becomes 15.

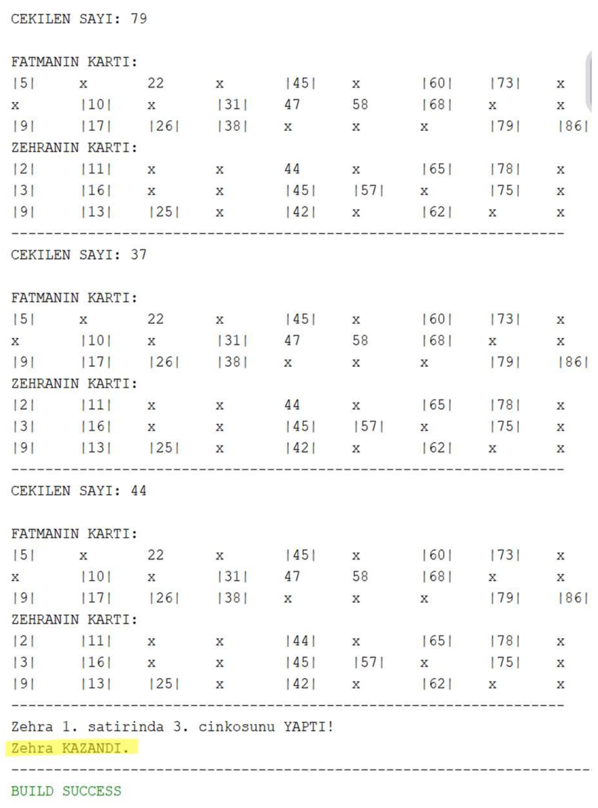
4. FatmaCilginMain Sınıfı



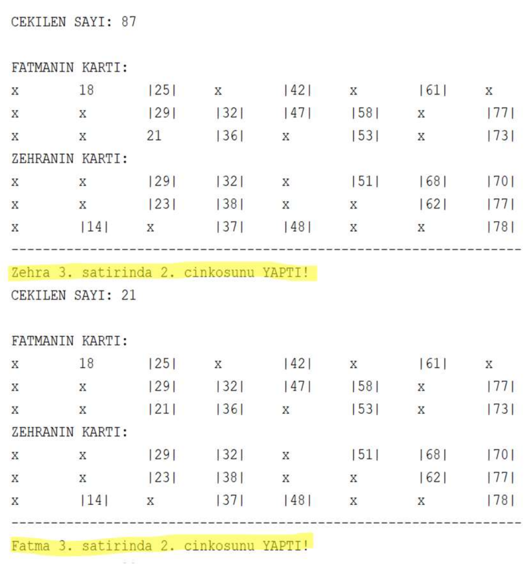
```java
public static int[] generatePermutation(int number) {
    // Permütasyon dizisi oluştur
    int[] permutation = new int[number];
    for (int i = 0; i < number; i++) {
        permutation[i] = i + 1;
    }
    // Diziyi karıştır
    Random rand = new Random();
    for (int i = 0; i < number - 1; i++) {
        int j = rand.nextInt(number - i) + i;
        // Değerleri swap et
        int temp = permutation[i];
        permutation[i] = permutation[j];
        permutation[j] = temp;
    }
    return permutation;
}
```

generatePermutation(int number): The method creates an int array with number of elements. Then, while creating the array, it fills each element sequentially with values from 1 to number. It then uses a loop to shuffle the array. In this loop, the position of each element in the array is changed randomly. At each loop step, a random number is generated and this number represents the indices of the elements in the array. This randomly generated number is replaced with the element from the current step of the loop. Finally, it returns the shuffled array.

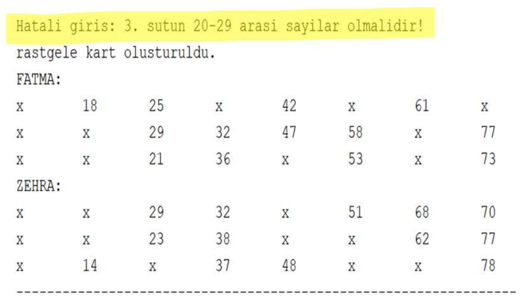
AN EXEMPLARY SCENARIO



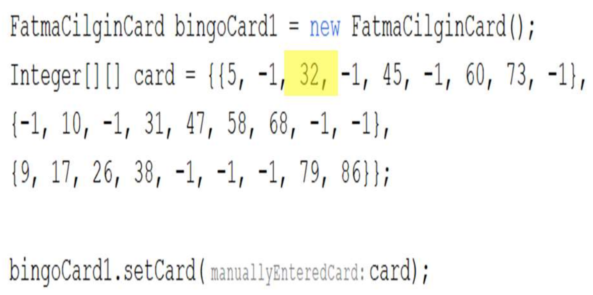
```
CEKILEN SAYI: 79

FATMANIN KARTI:
|5|     x       22      x       |45|    x       |60|    |73|    x
x       |10|    x       |31|    47      58      |68|    x       x
|9|     |17|    |26|    |38|    x       x       x       |79|    |86|
ZEHRANIN KARTI:
|2|     |11|    x       x       44      x       |65|    |78|    x
|3|     |16|    x       x       |45|    |57|    x       |75|    x
|9|     |13|    |25|    x       |42|    x       |62|    x       x
-----------------------------------------------------------------
CEKILEN SAYI: 37

FATMANIN KARTI:
|5|     x       22      x       |45|    x       |60|    |73|    x
x       |10|    x       |31|    47      58      |68|    x       x
|9|     |17|    |26|    |38|    x       x       x       |79|    |86|
ZEHRANIN KARTI:
|2|     |11|    x       x       44      x       |65|    |78|    x
|3|     |16|    x       x       |45|    |57|    x       |75|    x
|9|     |13|    |25|    x       |42|    x       |62|    x       x
-----------------------------------------------------------------
CEKILEN SAYI: 44

FATMANIN KARTI:
|5|     x       22      x       |45|    x       |60|    |73|    x
x       |10|    x       |31|    47      58      |68|    x       x
|9|     |17|    |26|    |38|    x       x       x       |79|    |86|
ZEHRANIN KARTI:
|2|     |11|    x       x       |44|    x       |65|    |78|    x
|3|     |16|    x       x       |45|    |57|    x       |75|    x
|9|     |13|    |25|    x       |42|    x       |62|    x       x
-----------------------------------------------------------------
Zehra 1. satirinda 3. cinkosunu YAPTI!
Zehra KAZANDI.
--------------------------------------------------------------------
BUILD SUCCESS
```



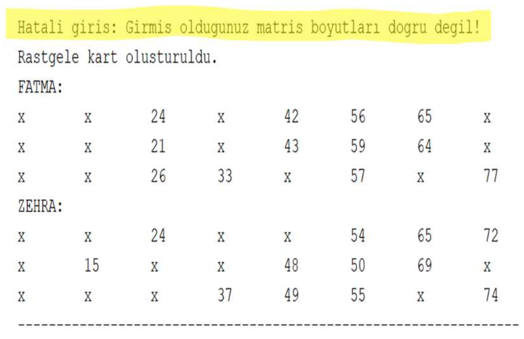
```
CEKILEN SAYI: 87

FATMANIN KARTI:
x       18      |25|    x       |42|    x       |61|    x
x       x       |29|    |32|    |47|    |58|    x       |77|
x       x       21      |36|    x       |53|    x       |73|
ZEHRANIN KARTI:
x       x       |29|    |32|    x       |51|    |68|    |70|
x       x       |23|    |38|    x       x       |62|    |77|
x       |14|    x       |37|    |48|    x       x       |78|
-----------------------------------------------------------------
Zehra 3. satirinda 2. cinkosunu YAPTI!
CEKILEN SAYI: 21

FATMANIN KARTI:
x       18      |25|    x       |42|    x       |61|    x
x       x       |29|    |32|    |47|    |58|    x       |77|
x       x       |21|    |36|    x       |53|    x       |73|
ZEHRANIN KARTI:
x       x       |29|    |32|    x       |51|    |68|    |70|
x       x       |23|    |38|    x       x       |62|    |77|
x       |14|    x       |37|    |48|    x       x       |78|
-----------------------------------------------------------------
Fatma 3. satirinda 2. cinkosunu YAPTI!
```



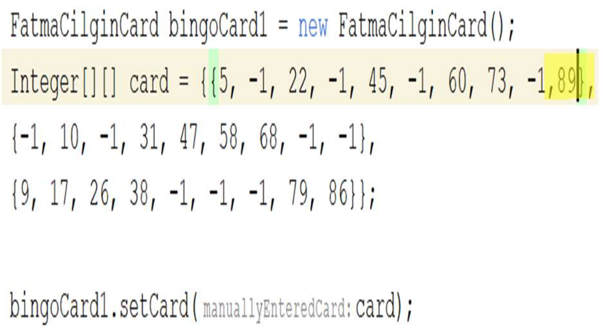
```
Hatali giris: 3. sutun 20-29 arasi sayilar olmalidir!
rastgele kart olusturuldu.
FATMA:
x       18      25      x       42      x       61      x
x       x       29      32      47      58      x       77
x       x       21      36      x       53      x       73
ZEHRA:
x       x       29      32      x       51      68      70
x       x       23      38      x       x       62      77
x       14      x       37      48      x       x       78
-----------------------------------------------------------------
```



```
FatmaCilginCard bingoCard1 = new FatmaCilginCard();
Integer[][] card = {{5, -1, 32, -1, 45, -1, 60, 73, -1},
{-1, 10, -1, 31, 47, 58, 68, -1, -1},
{9, 17, 26, 38, -1, -1, -1, 79, 86}};

bingoCard1.setCard( manuallyEnteredCard: card);
```



```
Hatali giris: Girmis oldugunuz matris boyutları dogru degil!
Rastgele kart olusturuldu.
FATMA:
x       x       24      x       42      56      65      x
x       x       21      x       43      59      64      x
x       x       26      33      x       57      x       77
ZEHRA:
x       x       24      x       x       54      65      72
x       15      x       x       48      50      69      x
x       x       x       37      49      55      x       74
-----------------------------------------------------------------
```



```
FatmaCilginCard bingoCard1 = new FatmaCilginCard();
Integer[][] card = {{5, -1, 22, -1, 45, -1, 60, 73, -1,89},
{-1, 10, -1, 31, 47, 58, 68, -1, -1},
{9, 17, 26, 38, -1, -1, -1, 79, 86}};

bingoCard1.setCard( manuallyEnteredCard: card);
```