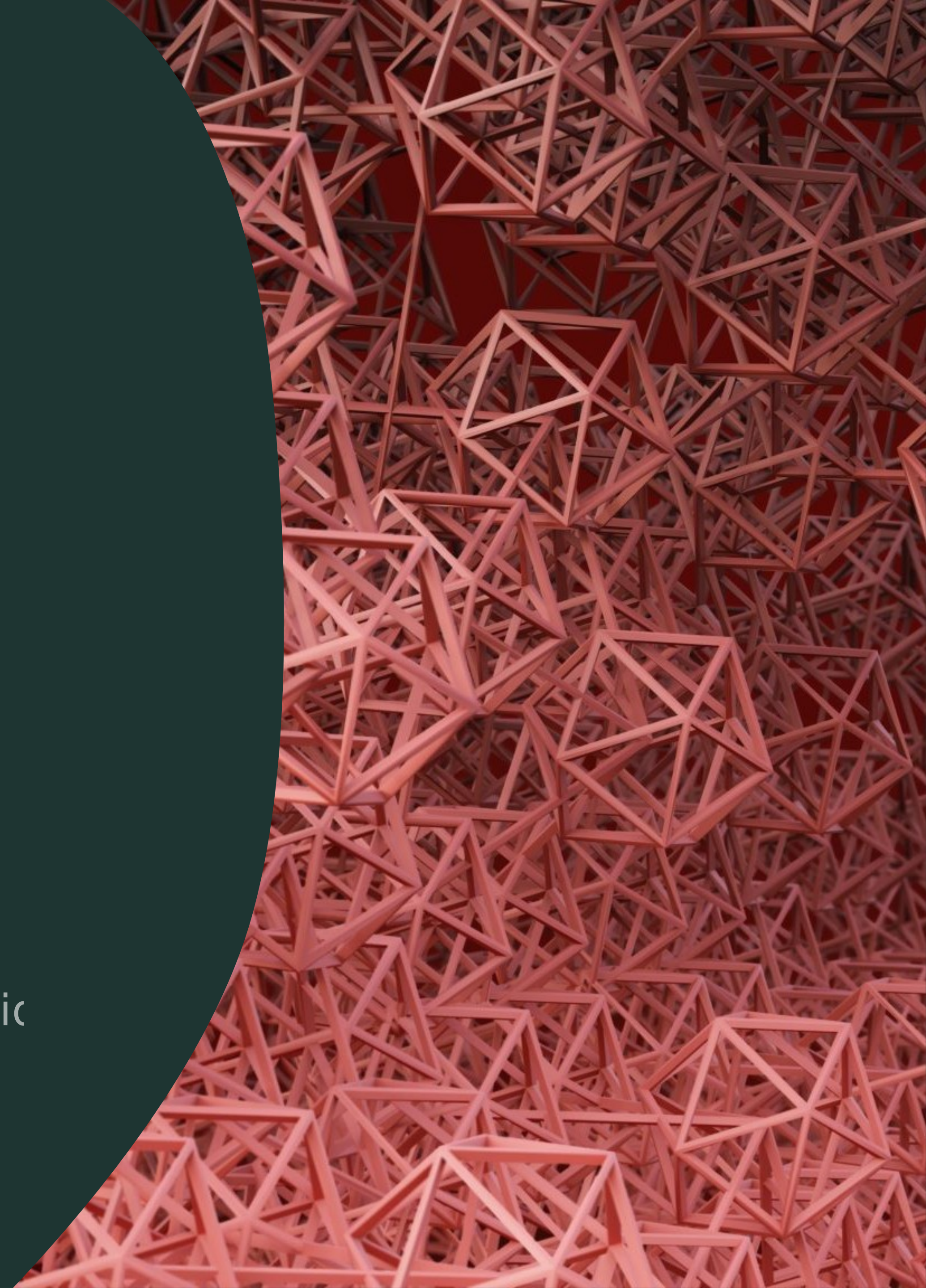
JAVA GAME PROJECT

~Made by Pratham Pate
and Nilay Patel

```
public int[][] grid

  SudokuGame()
  {
     grid = new int[9][9]; //empty
  }

  SudokuGame(String start){
     grid = new int[9][9]
     for(int x = 0; x <9; x++
     {
        for(int y =0; y<9; y++
        {
           grid[x][y] = 0
        }
     }
```

- This 9 * 9 Sudoku grid was created using 2-D Arrays
- We used two for loops to print the 81 values of the grid



```
public void printMySudokuGame(
  {
      for (int i = 0; i < 9; ++i
      {
          if (i % 3 == 0)
              System.out.println(" -----------------------")
          for (int j = 0; j < 9; ++j
          {
              if (j % 3 == 0) System.out.print("| ")
              System.out.print(grid[i][j] == 0 ? " ": Integer.toString(grid[i][j]))
              System.out.print(' ')
          }
          System.out.println("|")
      }
      System.out.println(" -----------------------"
  }
```

- This method is used to create the grid structure an print the verticle and horizontal dashed lines in th grid, forming the structure of Sudoku

- public boolean insertVal(int row, int col, int myVa
- {
- System.out.println("Entered insertVal " + "row " + row + "column " + col + " myVal "
myVal);
- if(checkRow(row, col, myVal) == false
- return false
- if(checkCol(row, col, myVal) == false
- return false
- if(checkBox(row, col, myVal) == false
- return false
- grid[row][col] = myVal
- return true
- }

- This method is for inserting the value in the gri
- After inserting the row column and value, the code will check the insert value is existing in the row, column and box. If the value is present already, it returns false and if the value is not existing it will insert tl given value in the grid

```
public boolean removeVal(int row, int co
   {
       grid[row][col] = 0

       return true
}
```

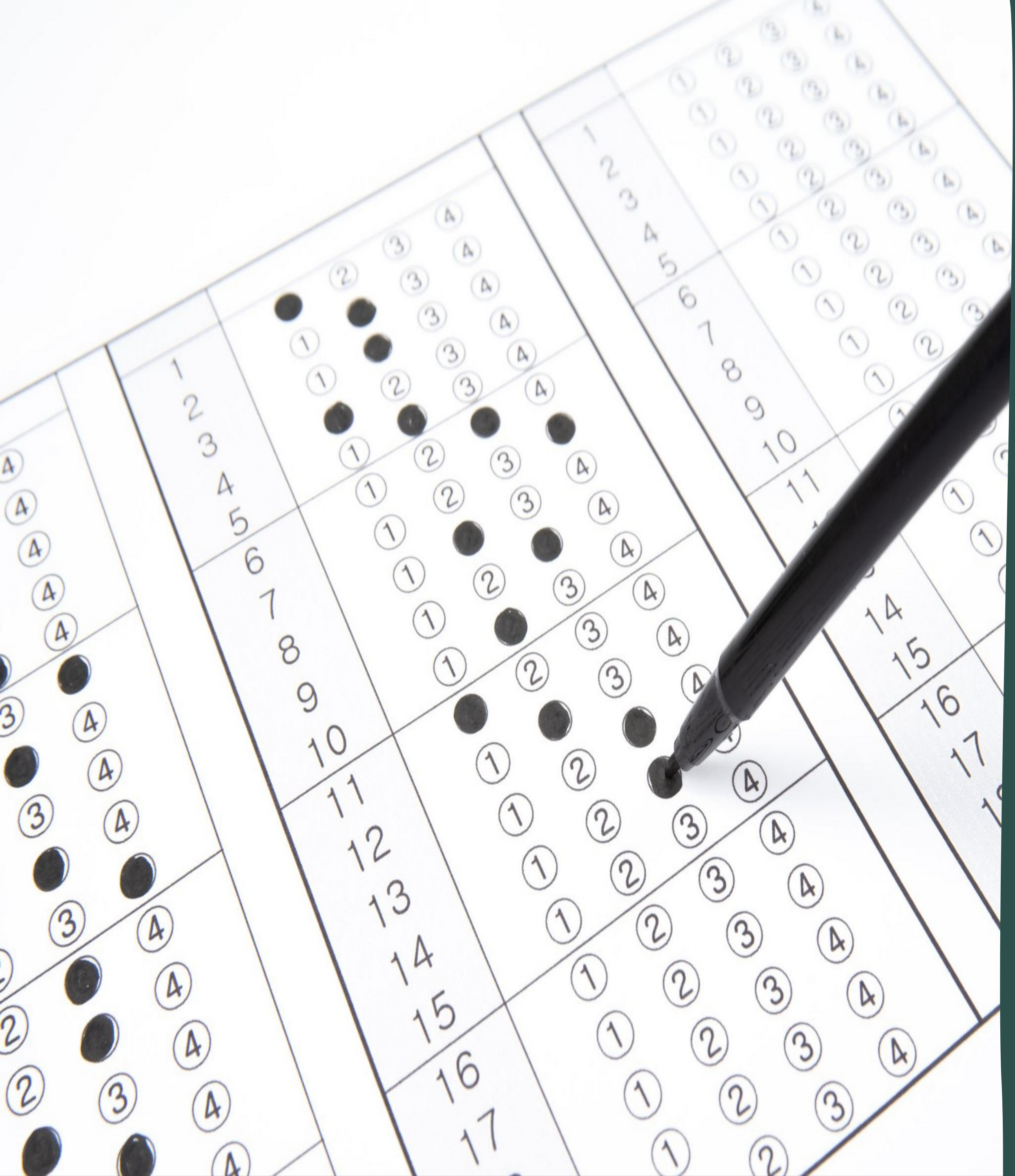
- This method is used to remove the existing value from the gri

```
private boolean checkRow(int row, int col, int myVa
   {
       for (int a = 0; a < 9; ++a)  // ro
           if (myVal == grid[row][a]
           {
               System.out.println(myVal + " Already in Row: " + row
               return false
           }
       return true
}
```

- The method check row is used too validate the input value in th row, if the the value is existing it returns a message that value already present in row

```
• private boolean checkCol(int row, int col, int myVal
•    {
•        for (int b = 0; b < 9; ++b)  // colum
•            if (myVal == grid[b][col]
•            {
•                System.out.println(myVal + " Already i
Column: " + col);
•                return false
•            }
•        return true

•    }
```

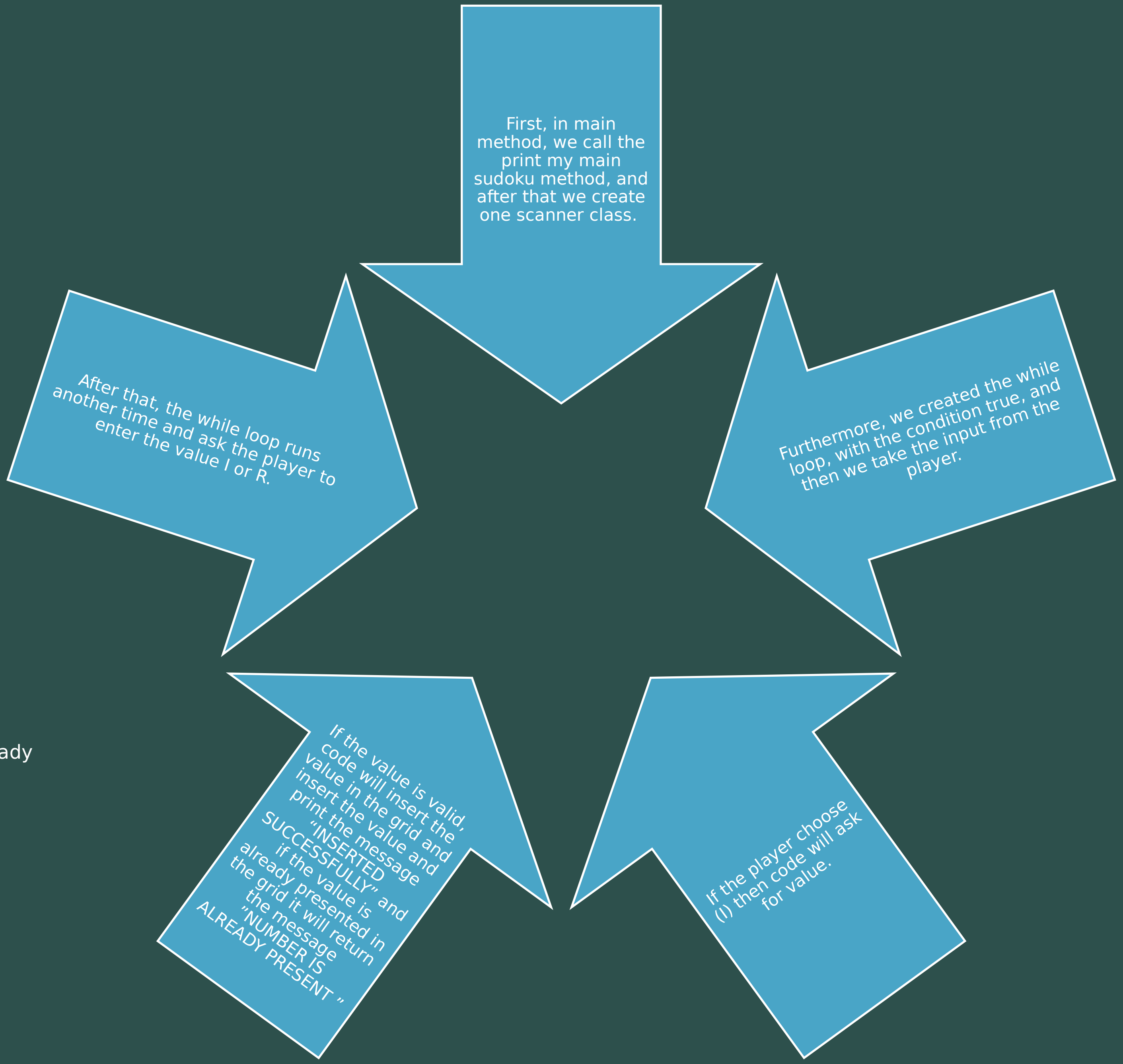
- The method check column is used to validate the inpu value in the column, if the value is existing it returns message that the value is already present in the colum



```
private boolean checkBox(int row, int col, int myVa
  {
      int boxRowOffset = (row / 3)*3
      int boxColOffset = (col / 3)*3
      for (int c = 0; c < 3; ++c) // bo
          for (int d = 0; d < 3; ++d
              if (myVal == grid[boxRowOffset+c][boxColOffset+d
              {
                  System.out.println(myVal + " Already in Box "
                   return false
              }
      return true


  }
}
```

- Firstly, this method will identify in which 3*3 box value is inserted. After that, it checks that the value is existing in the 3*3 box , i value is existing it will return the message that it is already in box

```
import java.util.Scanner;
   public class GameTester
   {
   public static void main(String args[])
   {
       SudokuGame i = new SudokuGame("start");
       i.printMySudokuGame();
       Scanner guess = new Scanner(System.in);
       {
        int row, col, val;
        String action;
           while(true)
           {
               System.out.println("Enter I for insert or R for remove: ");
               action = guess.next();

               System.out.println("Row: ");
               row = guess.nextInt();
               System.out.println("Column: ");
               col = guess.nextInt();

               if(action.equals("I"))
               {
                    System.out.println("Value: ");
                    val = guess.nextInt();
                   if(i.insertVal(row, col, val))
                   {
                       i.printMySudokuGame();
                       System.out.println("The Number is Inserted Sucessfully");
                   }
                   else
                   {
                       i.printMySudokuGame();
                       System.out.println("Try Again, Number you have entered is already
present");
                   }
               }
               else
               {
                i.removeVal(row, col);
                i.printMySudokuGame();
                System.out.println("The Number is removed Sucessfully");
               }
                   //guess.close();
           }
       }
   }
}
```



First, in main method, we call the print my main sudoku method, and after that we create one scanner class.

Furthermore, we created the while loop, with the condition true, and then we take the input from the player.

If the player choose (I) then code will ask for value.

If the value is valid, code will insert the value in the grid and insert the value and print the message "INSERTED SUCCESSFULLY" and if the value is already presented in the grid it will return the message "NUMBER IS ALREADY PRESENT "

After that, the while loop runs another time and ask the player to enter the value I or R.

- Thankyou