```
import java.util.ArrayList;

public class SlidingPuzzle
{
    private int[][] values;

    public SlidingPuzzle(int sideLength)
    {
        values = new int[sideLength][sideLength];
        initialize(sideLength);
    }

    public int getValueAt(int row, int col)
    {
        return values[row][col];
    }

    /* The given code creates and initializes an ArrayList
     containing the numbers 0 to (side*side-1).  This
     method fills in the private instance variable values with
     random values from this ArrayList temp, which will initialize
     the puzzle to a random starting state.
     */
    private void initialize(int side)
    {
        ArrayList<Integer> numbers = new ArrayList<Integer>();
        for (int num = 0; num < side * side; num++)
            numbers.add(num);
        // to be completed in part (A)
        for(int r=0; r<values.length; r++){
            for(int c=0; c<values[0].length; c++){
                int index = (int) (Math.random() * numbers.size());
                values[r][c] = numbers.remove(index);
            }
        }
    }

    /* returns true if the numbers are in "sliding puzzle"
     order, false otherwise; note that the blank (represented
     by a 0) can be anywhere
    */
    public boolean isDone()
    {
        int curr = 1;
        for(int r=0; r<values.length; r++){
            for(int c=0; c<values[0].length; c++){
                if(values[r][c] == curr) curr++;
                else if(values[r][c] != 0) return false;


            }
        }

        return true;
    }
```

```
    /* The player has clicked on a number at (row, col).  This method
     should "slide" that number to the neighboring blank (represented
     by the number 0), if such a neighboring blank exists.  If there
     is no neighboring blank, this method should do nothing.  If you
     slide a tile, you should change the private instance variable
     values to reflect this slide.  Note: you'll need to do lots of
     "bounds checking" to make sure you avoid an
     ArrayIndexOutOfBoundsException!
     */
    public void slide(int row, int col)
    {
        // to be completed in part (c)
        if(row != 0 && values[row-1][col] == 0){
            values[row-1][col] = values[row][col];
            values[row][col] = 0;
        }
        if(col !=0 && values[row][col-1] == 0){
            values[row][col-1] = values[row][col];
            values[row][col] = 0;
        }
        if(row != values.length-1 && values[row+1][col] == 0){
            values[row+1][col] = values[row][col];
            values[row][col] = 0;
        }
        if(col != values[0].length-1 && values[row][col+1] == 0){
            values[row][col+1] = values[row][col];
            values[row][col] = 0;
        }
    }


}
```