Test 1

```
private static void test1(String fileName) {
        int[][] board = read(fileName);
        for (int i = 1; i < board.length - 1; i++) {
            for (int j = 1; j < board[i].length - 1; j++) {
                System.out.printf ("%2d", board[i][j]);
            }
        System.out.println();
        }
    }
```

Test 2

```
private static void test2(String fileName) {
        int[][] board = read(fileName);
        int[][] temp = new int[board.length][board[0].length];
        for (int i = 1; i < board.length - 1; i++) {
            for (int j = 1; j < board.length - 1; j++) {
                temp[i][j] = cellValue(board, i, j);
            }
        }
        board = temp;
        for (int i = 1; i < board.length - 1; i++) {
            for (int j = 1; j < board[i].length - 1; j++) {
                System.out.printf ("%2d", board[i][j]);
            }
        System.out.println();
        }
    }
```

Test 3

```
private static void test3(String fileName, int Ngen) {
        int[][] board = read(fileName);
        for (int gen = 0; gen < Ngen; gen++) {
            System.out.println("Generation " + gen + ":");
            print(board);
            board = evolve(board);
        }
    }
```

Read

```
public static int[][] read(String fileName) {
        In in = new In(fileName);
        int rows = Integer.parseInt(in.readLine());
        int cols = Integer.parseInt(in.readLine());
        int[][] board = new int[rows + 2][cols + 2];
        String line = "";
        for (int i = 1; i < board.length-1; i++) {
            line = in.readLine();
            for (int j = 1; j < board[i].length-1; j++) {
                    if (line.length() == 0 || line.length() < j)
                        break;
                    if (line.charAt(j-1) == 'x') board[i][j] = 1;
            }
        }
        return board;
    }
```

Evolve

```
public static int[][] evolve(int[][] board) {
        int[][] b = new int[board.length][board[0].length];
        for (int i = 1; i < board.length - 1; i++) {
            for (int j = 1; j < board.length - 1; j++) {
                b[i][j] = cellValue(board, i, j);
            }
        }
        return b;
    }
```

CellValue

```
    public static int cellValue(int[][] board, int i, int j) {
        int cnt = count(board, i, j);
        if (board[i][j] == 0 && cnt == 3) return 1;
        if (board[i][j] == 1) {
            if (cnt < 2) {
                return 0;
            } else if (cnt == 2 || cnt == 3) {
                return 1;
            } else {
                return 0;
            }
        }
        return 0;
    }
```

Count

```java
public static int count(int[][] board, int i, int j) {
        int cnt = 0;
        for (int l = i - 1; l <= i + 1; l++) {
            for (int k = j - 1; k <= j + 1; k++) {
                    if (board[l][k] == 1) {
                        cnt++;
                    }
            }
        }
        if (board[i][j] == 1) cnt--;
        return cnt;
    }
```

Print

```
    public static void print(int[][] arr) {
        for (int i = 1; i < arr.length - 1; i++) {
            for (int j = 1; j < arr[i].length - 1; j++) {
                System.out.printf ("%2d", arr[i][j]);
            }
        System.out.println();
        }
    }
```