

Hints.java

```
package gotocph;

public class Hints {
    public String result = "";

    @Override
    public String toString() { return result; }

    public void calculateFor(String textRepresentation) {
        final String[] lines = textRepresentation.split("\n");

        for (int r = 0; r < lines.length; r++) {
            for (int c = 0; c < lines[r].length(); c++) {
                result += cellFor(lines, c, r);
            }
            result += "\n";
        }
    }

    private static String cellFor(final String[] lines, final int
column, int row) {
        String line = lines[row];
        if (line.charAt(column) == '*') return "*";
        int nearbyMines = 0;

        if (0 < column) {
            nearbyMines += line.charAt(column - 1) == '*' ? 1 : 0;
        }

        if (column < line.length() - 1) {
            nearbyMines += line.charAt(column + 1) == '*' ? 1 : 0;
        }

        return String.valueOf(nearbyMines);
    }
}
```

MinesweeperTest.java

```
package test.gotocph;

import gotocph.Hints;
import org.junit.Ignore;
import org.junit.Test;

import static org.hamcrest.MatcherAssert.assertThat;
import static org.hamcrest.Matchers.equalTo;

public class MinesweeperTest {

    @Test public void
    represents_a_mine_as_a_star() {
        assertThat(hintsFrom("*\n"), equalTo("*\n"));
    }

    @Test public void
    represents_no_mine_nearby_as_0() {
        assertThat(hintsFrom(".\n"), equalTo("0\n"));
    }

    @Test public void
    adds_nearby_mines_to_total_for_cell() {
        assertThat(hintsFrom("..*\n"), equalTo("01*\n"));
        assertThat(hintsFrom("*. *\n"), equalTo("*2*\n"));
    }

    @Ignore @Test public void
    adds_mine_from_line_below() {
        assertThat(hintsFrom(".\n*\n"), equalTo("1\n*\n"));
    }

    private String hintsFrom(String textRepresentation) {
        final Hints hints = new Hints();

        hints.calculateFor(textRepresentation);
        return hints.toString();
    }
}
```