

You've completed Minesweeper. Discover more exercises



Open in editor

```
Iteration 1
Submitted via Editor, 24 minutes ago

9  CONSTANTS:
10  c_left      TYPE 1 VALUE 1,
11  c_right     TYPE 1 VALUE 2,
12  c_up_left   TYPE 1 VALUE 3,
13  c_up        TYPE 1 VALUE 4,
14  c_up_right  TYPE 1 VALUE 5,
15  c_down_right TYPE 1 VALUE 6,
16  c_down      TYPE 1 VALUE 7,
17  c_down_left TYPE 1 VALUE 8,
18  c_all_directions TYPE 1 VALUE 8
19  .
20  * exercism.org doesn't know these SAP Types
21  TYPES: BEGIN OF ty_submatch_result,
22         offset TYPE 1,
23         length TYPE 1,
24         END OF ty_submatch_result.
25  TYPES: ty_submatch_result_tab TYPE STANDARD TABLE OF ty_submatch_result WITH DEFAULT KEY.
26
27  TYPES: BEGIN OF ty_match_result,
28         line TYPE 1,
29         offset TYPE 1,
30         length TYPE 1,
31         submatches TYPE ty_submatch_result_tab,
32         END OF ty_match_result.
33
34  TYPES: ty_match_result_tab TYPE STANDARD TABLE OF ty_match_result WITH DEFAULT KEY.
35
36  PRIVATE SECTION.
37  METHODS add_to_mine_counter
38  IMPORTING
39    i_direction TYPE i
40    i_mine_pos TYPE ty_match_result
41    i_minefield TYPE string_table.
42  RETURNING
43    VALUE(result_field) TYPE string_table.
44
45  ENDMETHOD.
46
47  CLASS zcl_minesweeper IMPLEMENTATION.
48
49  METHOD annotate.
50  DATA lt_mine_pos TYPE ty_match_result_tab.
51  FIELD-SYMBOLS: <ls_mine_pos> TYPE ty_match_result.
52
53  result = input.
54  FIND ALL OCCURRENCES OF '+' IN TABLE input RESULTS lt_mine_pos.
55  DATA(lv_direction) = 1.
56  LOOP AT lt_mine_pos ASSIGNING <ls_mine_pos>.
57    lv_direction = c_left.
58    WHILE lv_direction <= c_all_directions.
59      result = add_to_mine_counter( i_direction = lv_direction i_minefield = result i_
60      lv_direction = lv_direction + 1.
61    ENDMETHOD.
62  ENDOLOOP.
63  ENDMETHOD.
64
65
66  METHOD add_to_mine_counter.
67  *exercism.org doesn't know SAP Type text1000
68  TYPES: ty_text1000 TYPE c LENGTH 1000.
69  DATA: lv_counter_position TYPE i,
70        lv_out_of_bounds TYPE abap_bool,
71        lv_counter_line TYPE i.
72
73  CONSTANTS: lc_mine TYPE 1 VALUE '-1'.
74
75  DATA(lv_mineposition) = i_mine_pos->offset.
76  data(lv_minefield_len) = strlen( i_minefield[ 1 ] ).
77
78  result_field = i_minefield.
79
80  CASE i_direction.
81    WHEN c_left.
82      lv_counter_position = lv_mineposition - 1.
83      lv_counter_line = i_mine_pos->line.
84      lv_out_of_bounds = COND abap_bool( WHEN lv_mineposition = 0 THEN abap_true ELSE
85      WHEN c_up_left.
86      lv_counter_position = lv_mineposition - 1.
87      lv_counter_line = i_mine_pos->line - 1.
88      lv_out_of_bounds = COND abap_bool( WHEN lv_mineposition = 0 THEN abap_true
89      WHEN lv_counter_line = 0 THEN abap_true
90      ELSE abap_false ).
91    WHEN c_up.
92      lv_counter_position = lv_mineposition.
93      lv_counter_line = i_mine_pos->line + 1.
94      lv_out_of_bounds = COND abap_bool( WHEN lv_counter_line = 0 THEN abap_true
95      ELSE abap_false ).
96    WHEN c_up_right.
97      lv_counter_position = lv_mineposition + 1.
98      lv_counter_line = i_mine_pos->line + 1.
99      lv_out_of_bounds = COND abap_bool( WHEN lv_counter_line = 0 THEN abap_true
100      WHEN lv_mineposition = lv_minefield_len - 1 T
101      ELSE abap_false ).
102    WHEN c_right.
103      lv_counter_position = lv_mineposition + 1.
104      lv_out_of_bounds = COND abap_bool( WHEN lv_mineposition = lv_minefield_len - 1 T
105      lv_counter_line = i_mine_pos->line.
106    WHEN c_down_right.
107      lv_counter_position = lv_mineposition + 1.
108      lv_counter_line = i_mine_pos->line + 1.
109      lv_out_of_bounds = COND abap_bool( WHEN lv_counter_line > lines( i_minefield ) T
110      WHEN lv_mineposition = lv_minefield_len - 1 T
111      ELSE abap_false ).
112    WHEN c_down.
113      lv_counter_position = lv_mineposition.
114      lv_counter_line = i_mine_pos->line + 1.
115      lv_out_of_bounds = COND abap_bool( WHEN lv_counter_line > lines( i_minefield )
116      ELSE abap_false ).
117    WHEN c_down_left.
118      lv_counter_position = lv_mineposition - 1.
119      lv_counter_line = i_mine_pos->line + 1.
120      lv_out_of_bounds = COND abap_bool( WHEN lv_mineposition = 0 THEN abap_true
121      WHEN lv_counter_line > lines( i_minefield )
122      ELSE abap_false ).
123
124  ENDCASE.
125
126  IF lv_out_of_bounds = abap_false.
127    DATA(lv_field_value) = COND 1( LET lv_count = substring( val = i_minefield[ lv_cou
128      WHEN lv_count = '' THEN 0
129      WHEN lv_count = '+' THEN lc_mine
130      ELSE lv_count
131      ).
132    IF lv_field_value <> lc_mine.
133      DATA(lv_line_with_counter) = CONV ty_text1000( result_field[ lv_counter_line ] )
134
135      lv_line_with_counter+lv_counter_position() = [( lv_field_value + 1 )].
136      result_field[ lv_counter_line ] = [( lv_line_with_counter WIDTH = lv_minefield_l
137    ENDF.
138  ENDMETHOD.
139
140  ENDMETHOD.
```

ALL TESTS PASSED

12 tests passed

- Test 1 NO\_ROWS
- Test 2 NO\_COLUMNS
- Test 3 NO\_MINES
- Test 4 ONLY\_MINES
- Test 5 MINE\_SURROUNDED\_BY\_SPACES
- Test 6 SPACE\_SURROUNDED\_BY\_MINES
- Test 7 HORIZONTAL\_LINE
- Test 8 HORIZONTAL\_LINE\_MINES\_AT\_ENDS
- Test 9 VERTICAL\_LINE
- Test 10 VERTICAL\_LINE\_MINES\_AT\_ENDS
- Test 11 CROSS
- Test 12 LARGE\_MINFIELD