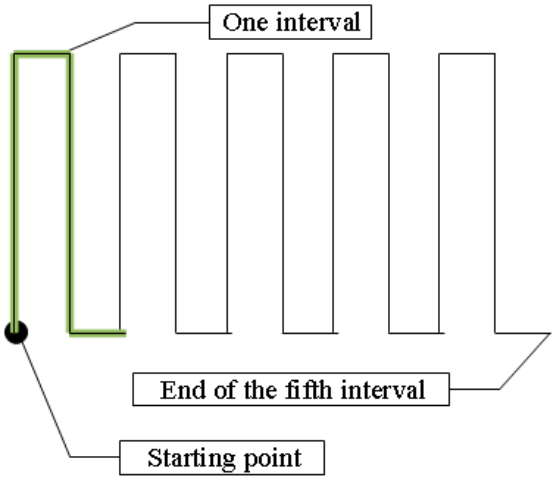
Assignment

Instructions

A search and rescue agency needs a tool that will standardize their grid searching patterns in the field. The agency wants to preload GPS waypoints into their GPS receivers so that their search parties follow a consistent pattern and the area has full coverage. The problem we have to solve is as follows: given a starting position, an x increment, a y increment and the total number of intervals, generate the waypoints i.e. X and Y coordinates separated by a colon.



STEP 01

Open IntelliJ

Create a new Project

Use the template

Call the project mod_05_GridSearch

Ensure the project location is:
H:\var\gist\7010\wksp_dt\mod_05_GridSearch

Set the base package to the standard:
bcit.gist7010

STEP 02	**Build the 'code skeleton' to retrieve user input and start the processing output** Here are some suggestions and requirements: **Use redirection (required)** **Use the Scanner class to retrieve data from the user (required)** The data for a grid will be on one line and in the following format (required): xcoord ycoord xdistance ydistance intervals Here is an example of a line of data: 10 10 10 100 5 **Have a string variable to test if the data is a number or the word "END"** xcoordAsString **Have several double variables** **(do not use the Big-D-Double wrapper class)** xstart ystart xinterval yinterval iterations
STEP 03	**Use a for-loop within a while-loop (required)** The while-loop from the tutorial is a great place to start. Your for-loop would be inside the else-block.
STEP 04	**Get xCoordAsString using the scanner-variable keyboard** Where? The while loop The while loop will process the file testing for the keyword "END"
STEP 05	**Inside the else but before the for-loop print "BEGIN" (required)**

STEP 06	**Inside the else but before the for-loop initialize the following variables:** xstart ➡xCoordAsString ➡ Double.parseDouble() ystart ➡ keyboard xinterval ➡ keyboard yinterval ➡ keyboard iterations ➡ keyboard
STEP 07	**Just before the for-loop, print the starting x and y values separated by a colon**
STEP 08	**Within the for loop have 4 print statements (UP, OVER, DOWN, OVER):** ● The y heading north but the same x ● The y heading north and the x heading east ● The y at the original value but the x heading east ● The y at the original value but the 2 times x heading east
STEP 09	**Perform the move** xStart = xStart + 2 * x interval Where? Just before the end of the for-loop reset the current x variable
STEP 10	**End the search grid** The end of the file will be marked with the keyword 'END' (required)
STEP 11	**Compare your output file to the one below:** **Sample input:** 10 10 10 100 5 5 5 5 25 10 END

Sample Output:

```
BEGIN
10.000000:10.000000
10.000000:110.000000
20.000000:110.000000
20.000000:10.000000
30.000000:10.000000
30.000000:110.000000
40.000000:110.000000
40.000000:10.000000
50.000000:10.000000
50.000000:110.000000
60.000000:110.000000
60.000000:10.000000
70.000000:10.000000
70.000000:110.000000
80.000000:110.000000
80.000000:10.000000
90.000000:10.000000
90.000000:110.000000
100.000000:110.000000
100.000000:10.000000
110.000000:10.000000
END
BEGIN
5.000000:5.000000
5.000000:30.000000
10.000000:30.000000
10.000000:5.000000
15.000000:5.000000
15.000000:30.000000
20.000000:30.000000
20.000000:5.000000
25.000000:5.000000
25.000000:30.000000
30.000000:30.000000
30.000000:5.000000
35.000000:5.000000
35.000000:30.000000
40.000000:30.000000
40.000000:5.000000
45.000000:5.000000
45.000000:30.000000
50.000000:30.000000
50.000000:5.000000
55.000000:5.000000
55.000000:30.000000
```

	60.000000:30.000000 60.000000:5.000000 65.000000:5.000000 65.000000:30.000000 70.000000:30.000000 70.000000:5.000000 75.000000:5.000000 75.000000:30.000000 80.000000:30.000000 80.000000:5.000000 85.000000:5.000000 85.000000:30.000000 90.000000:30.000000 90.000000:5.000000 95.000000:5.000000 95.000000:30.000000 100.000000:30.000000 100.000000:5.000000 105.000000:5.000000 END

Marking Guide

Item	Value
The tutorial works	2
The program works	5
Indentation is correct and consistent	1
Comments	2
Total	**10**