Assignment 1

Mesk alhamaideh

Section 1

Karel learns java

Introduction

Blankkarel solves the assignment by splitting the map into four equal parts and paints the center of each part

Code

The code consists of three functions (moveStraight, corner, fillCenter) in addition to the run function.

MoveStraight function receives a number of steps to walk and a flag to know whether to place beepers or not.

FillCenter function receives a flag (square) to decide whether to paint a square center or a single point.

Corner function receives three flags (doubleLine, squareCenter,lastCorner),

It fills the rest of the line with beepers if it's a doubleLine, or if it's a single line except for the last corner.

After filling the rest of the line it moves to the start of the new quarter.

The start of the new quarter is the same point we are at in these cases:

- We are in the last quarter (no quarters left).
- The map is divided with a single line not double lines.

Otherwise the start of the next quarter is the point above the current point on the y-axis.

Run

The run function starts by measuring the width of the map, deciding whether to divide the map with single or double lines and whether the center of each part is square or a single point center, calculating width/2 then moving to the start point for the while loop to do the work.

Filling the map

The while loop fills the map solving one quarter and repeating the same process four times for the whole map to be done.

Solving a single quarter as follows:

- Place a beeper
- Fill a straight line of length width/2
- Turn left
- Fill half a line (width/4)
- Move to center
- Fill center
- Return back to line
- Fill the remaining of the line and stand at the starting point of the next quarter.

*note: first two bullet point keep applying if the map is divided with double lines, and only applied once if its divided using a single line.

Optimizations

The code splits the map into four sections using the minimum amount of steps needed, it is also modular and I tried to make it cohesive (each function solves one specific sub-problem within the code, and is reusable) and loosely coupled (functions don't interact with each other except through the run function) and the code does not exceed 100 lines.