package Base;

import java.util.List;

public class Plotter { //Contains methods to print & plot objects in a grid

private int width;

private int height;

private String[][] World;

private final int centerX;

private final int centerY;

public Plotter(int xRadius, int yRadius) {

width = xRadius \* 2 + 3;

height = yRadius \* 2 + 3;

initializeWorldGrid(width, height);

centerX = width / 2;

centerY = height / 2;

}

//To initialize grid for plotting multiple objects

private void initializeWorldGrid(int width, int height) {

World = new String[width][height];

//Initializing grid/frame buffer

for (int i = 0; i < width; i++) {

for (int j = 0; j < height; j++) {

World[i][j] = ".";

}

}

//Drawing ordinate & abscissa markings

int centerX = width / 2;

int centerY = height / 2;

for (int j = 0; j < height; j++) {

int value = centerY - j;

World[centerX][j] = String.valueOf(value); //Ordinate

}

for (int i = 0; i < width; i++) {

int value = i - centerX;

World[i][centerY] = String.valueOf(value); //Abscissa

}

World[centerX][centerY] = "0"; //Center

}

//To initialize grid for an individual object

public static String[][] initializeObjectGrid(List<Point> Object, char character) {

//Setting initial min/max value

double minX = Double.MAX\_VALUE;

double maxX = Double.MIN\_VALUE;

double minY = Double.MAX\_VALUE;

double maxY = Double.MIN\_VALUE;

//Finding min & max for x & y for all points in the list

for (Point point : Object) {

if (point.x < minX) minX = point.x;

if (point.x > maxX) maxX = point.x;

if (point.y < minY) minY = point.y;

if (point.y > maxY) maxY = point.y;

}

// Calculating grid dimensions

int width = (int) Math.round(maxX - minX) + 1;

int height = (int) Math.round(maxY - minY) + 1;

//Initializing grid/frame buffer

String[][] grid = new String[width][height];

for (int i = 0; i < width; i++) {

for (int j = 0; j < height; j++) {

grid[i][j] = ".";

}

}

// Plotting coordinates

for (Point point : Object) {

int X = (int) (Math.round(point.x - minX));

int Y = (int) (Math.round(point.y - minY));

grid[X][height - Y - 1] = String.valueOf(character);

}

return grid;

}

//To print object coordinates

public static void printCoordinates(List<Point> Object) {

for (Point point : Object) {

System.out.print(point.getCoordinates() + ' ');

}

}

//To plot object in its ObjectGrid

public static void plotObject(List<Point> Object, char character) {

//Initializing grid/frame buffer

String[][] grid = Plotter.initializeObjectGrid(Object, character);

// Displaying coordinate grid

for (int j = 0; j < grid[0].length; j++) {

for (String[] strings : grid) {

String cell = strings[j];

System.out.printf("%3s", cell);

}

System.out.println();

}

}

//To plot objects in the World Grid

public void WorldPlotObject(List<Point> Object, char character) {

for (Point point : Object) {

// if (Point.x <= centerX && Point.y <= centerY) {

int X = (int) (Math.round(point.x));

int Y = (int) (Math.round(point.y));

World[centerX + X][centerY - Y] = String.valueOf(character);

// }

}

}

//To display the World Grid

public void WorldDisplay() {

for (int j = 0; j < height; j++) {

for (int i = 0; i < width; i++) {

String cell = World[i][j];

System.out.printf("%3s", cell);

}

System.out.println();

}

}

//To clear objects in World Grid

public void ClearObject(List<Point> Object) {

WorldPlotObject(Object, '.');

}

//To completely clear the WorldGrid

public void ClearWorld() {

initializeWorldGrid(width, height);

}

//To resize World Grid

public void ResizeWorld(int xRadius, int yRadius) {

width = xRadius \* 2 + 1;

height = yRadius \* 2 + 1;

initializeWorldGrid(width, height);

}

}