GitHub Stuff\project-5-minewalker-WildestPantaloons\RandomPath.java

```java
import java.awt.Point;
import java.util.ArrayList;
import java.util.Random;

/**
 * Static utility class that generates a random path through a square grid.
 *
 * @author mvail
 */
public class RandomPath {

    /**
     * Private constructor prevents creation of an object. Must use static method
     * RandomPath.getPath() to get a path.
     */
    private RandomPath() { }

    /**
     * Returns a list of Points creating a random path connecting Point(0,0) to
     * Point(gridDimension-1,gridDimension-1) in a square grid with the given
     * dimension. Note that Point.x corresponds to grid row and Point.y
     * corresponds to grid column to match 2D array indexing convention.
     *
     * @param gridDimension a positive integer greater than 1
     * @return a list of Points connecting Point(0,0) to Point(gridDimension-1,gridDimension-1)
     */
    public static ArrayList<Point> getPath(int gridDimension) {
        if (gridDimension < 2) {
            throw new RuntimeException("RandomPath dimension cannot be less than 2.");
        }

        Random rand = new Random();

        Point startingPoint = new Point(0,0); //upper left
        Point endingPoint = new Point(gridDimension-1, gridDimension-1); //lower right

        ArrayList<Point> path = new ArrayList<Point>();
        path.add(startingPoint);

        Point lastPoint = startingPoint; //last Point in the current list

        while (!lastPoint.equals(endingPoint)) {
            if (rand.nextBoolean()) { //move right if possible
                if (lastPoint.y < endingPoint.y) {
                    path.add(new Point(lastPoint.x, lastPoint.y+1));
                } else {
                    path.add(new Point(lastPoint.x+1, lastPoint.y));
                }
            } else { //move down if possible
                if (lastPoint.x < endingPoint.x) {
                    path.add(new Point(lastPoint.x+1, lastPoint.y));
                } else {
                    path.add(new Point(lastPoint.x, lastPoint.y+1));
                }
            }
            lastPoint = path.get(path.size() - 1);
        }

        return path;
    }

}

```