Question: `Market Navigator`

Description

In this project you are asked to find the smallest path from the market (Migros) to houses. Imagine we have a car that gets items from the market, visits each house once and returns back to the market. **You are required to:**

- Complete the `House` class, which has its name and coordinates as attributes. You are required to create getters and setters for all attributes. Optionally, you can create helper methods to help you solve the problem. Your getters and setters should be named like `getX` and `setX` as we conventionally do.
- Complete the `pathFinder` method in `MarketNavigator` class. **This method should:**
- Take filename as the parameter. You can use example files given in the project to test your code: `coordinates.txt` and `coordinates2.txt`
- Read the contents of the file line by line, and create houses. For example the coordinates2.txt contains: `Migros 24 32` , `A 24 35` , `B 28 35` . With these, you should create 3 houses named 'Migros', 'A' and 'B' with the given coordinates as their x and y attributes. Note that as an example Migros house should have 24 as x and 32 as y. While you calculate the distance between two houses, use their x and y coordinates.
- **Do your calculation by:** Start from Migros house and visit each of the other houses once. At the end, return to Migros. Your returned value should be the sum of distances that you have traveled. Keep in mind that you can find many different such paths, but we want the shortest one as the answer. You can see an example visualization of a path below, it starts from Migros house, visits each house once and returns back to Migros.
- After you do the calculation return the length of the path as an integer. To do that, **round** the decimal value that you find with `Math.round()` , as already given in the code. Using different methods to do rounding may give different results.

"Check for .png file in repo."

Notes:

- Do not modify anything else in the code, except the specified area.
- Hint: Try random house permutations and select the shortest.