Model - stores all the low level data and does all the calculations necessary and returns data
View - The entity that interacts with the model to get whatever data needed, and then returns an output that the client needs
Controller - Interacts with the view to get the high level data needed

Path.java:
- Path(Double d, String a) - Model
- getDistance() - View
- getDirection() - View
- compareTo(Path B2) - Model

Coordinate.java:
- Coordinate(int xc, int yc) - Model
- getX() - View
- getY() - View

Building.java:
- Building(String i, String n) - Model
- getId() - View
- getName() - View
- compareTo(Building B2) - Model
- equals(Object obj) - Model
- hashCode() - Model

CampusPaths.java:
- CampusPaths() - Model
- getDirection(Coordinate point1, Coordinate point2) - Model
- getDistance(Coordinate point1, Coordinate point2) - Model
- createNewGraph(String filename, String filename2) - Model
- findPath(String B1, String B2) - View
- listBuildings() - View
- main() - Controller

CampusParser.java:
- readData(...) - Model

Dijkstra.java:
- run_Dijkstra(...) - View
- compare(...) - Model