Alexander Keith

Project 1

1/26/17

Class: TestPackages

1. import java.io.\*;
2. /\*\* \* Testing Class containing the main method in order to test the Packages Class \*  \* @author (Alexander Keith)  \* @version (1/26) \*/
3. public class TestPackages {
4. public static void main(String[] args) throws IOException {
5. Packages test = new Packages();
6. System.out.println("ALL PACKETS \n" + test.toString());
7. System.out.println("ALL HEAVY PACKETS");
8. test.displayHeavyPackages();
9. System.out.println();
10. System.out.println("ALL IN-STATE PACKETS");
11. test.displayInStatePackages();
12. System.out.println();
13. System.out.println("The packet that is the heaviest is: \n" + test.MaxWeightPacket());
14. System.out.print("The average weight of all packets is: ");
15. test.displayAverageWeight();
16. System.out.println();
17. System.out.println("LOCAL PACKETS");
18. test.displayLocalPackets();
19. System.out.println();
20. System.out.println("NONLOCAL PACKETS");
21. test.displayNonLocalPackets();
22. }
23. }

Class: Packages

1. import java.util.\*;
2. import java.io.\*;
3. import java.text.DecimalFormat;
4. /\*\* \* Write a description of class Packages here. \*  \* @author (Alexander Keith)  \* @version (1/26) \*/
5. public class Packages {
6. private ArrayList < Packets > shipmentList = new ArrayList < Packets > ();
7. private double totalWeight = 0.0;
8. DecimalFormat fmt = new DecimalFormat("0.##");
9. public Packages() throws IOException {
10. **private** int id;
11. **private** double weight;
12. **private** String dest = "", orig = "";
13. **private** Scanner fileScan;
14. fileScan = new Scanner(new File("packetData.txt"));
15. while (fileScan.hasNext()) {
16. id = fileScan.nextInt();
17. weight = fileScan.nextDouble();
18. dest = fileScan.next().toUpperCase().trim();
19. orig = fileScan.next().toUpperCase().trim();
20. shipmentList.add(new Packets(id, weight, dest, orig));
21. totalWeight = totalWeight + weight;
22. }
23. }
24. public String toString() {
25. **private** String p1 = "";
26. for (Packets p: shipmentList) {
27. p1 = p1 + p.toString();
28. }
29. return p1;
30. }
31. public void displayHeavyPackages() {
32. for (Packets h: shipmentList) {
33. if (h.isHeavy() == true) {
34. System.out.print(h.toString());
35. }
36. }
37. }
38. public void displayInStatePackages() {
39. for (Packets s: shipmentList) {
40. if (s.isInState() == true) {
41. System.out.print(s.toString());
42. }
43. }
44. }
45. public Packets MaxWeightPacket() {
46. **private** int heavy = 0;
47. for (Packets w: shipmentList) {
48. if (w.getWeight() > shipmentList.get(heavy).getWeight()) {
49. heavy = shipmentList.indexOf(w);
50. }
51. }
52. return shipmentList.get(heavy);
53. }
54. public void displayAverageWeight() {
55. System.out.println(fmt.format(totalWeight / shipmentList.size()));
56. }
57. public void displayLocalPackets() {
58. for (Packets l: shipmentList) {
59. if (l.destinationLocation() == "Local") {
60. System.out.print(l.toString());
61. }
62. }
63. }
64. public void displayNonLocalPackets() {
65. for (Packets l: shipmentList) {
66. if (l.destinationLocation() == "NonLocal") {
67. System.out.print(l.toString());
68. }
69. }
70. }
71. }

Class: Packets

1. /\*\* \* Describes Packets. \*  \* @author (Alexander Keith)  \* @version 1/25 \*/
2. public class Packets {
3. private int idNumber;
4. private double weight;
5. private String stateDest = "", stateOrig = "";
6. public Packets(int id, double weight, String dest, String orig) {
7. idNumber = id;
8. this.weight = weight;
9. stateDest = dest;
10. stateOrig = orig;
11. }
12. public String toString() {
13. return idNumber + "\t" + weight + "\t" + stateDest + "\t" + stateOrig + "\n";
14. }
15. public boolean isHeavy() {
16. return weight > 10;
17. }
18. public boolean isInState() {
19. return stateDest.equals(stateOrig);
20. }
21. public double getWeight() {
22. return weight;
23. }
24. public String destinationLocation() {
25. if (stateDest.equals("CT") || stateDest.equals("MA") || stateDest.equals("RI") || stateDest.equals("NY")) {
26. return "Local";
27. } else {
28. return "NonLocal";
29. }
30. }
31. }

Test Run:

UML:

|  |
| --- |
| Packages |
| - shipmentList: ArrayList<Packets>  - totalWeight: Double  - fmt: DecimalFormat |
| + Packages(): void  + toString() : String  + displayHeavyPackages() : void  + displayInStatePackages() : void  + MaxWeightPacakges() : Packets  + displayAverageWeight() : void  + displayLocalPackets() : void  + displayNonLocalPackets() : void |

|  |
| --- |
| TestPackages |
|  |
| + main (String [] agrs) : void |

|  |
| --- |
| Packets |
| - idNumber : int  - weight : double  - stateDest : String  - stateOrig : String |
| + Packets( id:int, weight:double, dest:String, orig:String) : void  + to String() : String  + isHeavy() : boolean  + isInState() : Boolean  + getWeight() : double  + destinationLocation() : String |