Alexander Keith

CS 2 MW 3:00

Project 3

Packet:

1. public class Packet {
2. private int idNumber;
3. private double weight;
4. private String destination;
5. public Packet(int id, double wei, String dest) {
6. idNumber = id;
7. weight = wei;
8. destination = dest;
9. }
10. public boolean isHeavy() {
11. if (weight > 10.0) {
12. return true;
13. } else {
14. return false;
15. }
16. }
17. public String toString() {
18. return idNumber + " " + " " + weight + " " + destination;
19. }
20. public double getWeight() {
21. return weight;
22. }
23. public String getDestination() {
24. return destination;
25. }
26. }

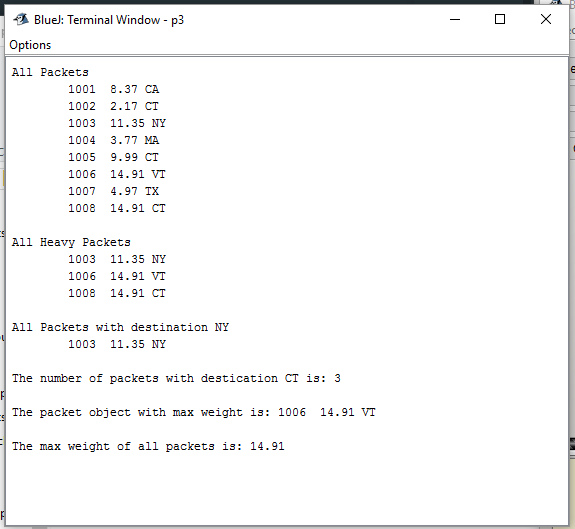
Recursion:

1. public class Recursion {
2. public String toString(Packet[] list, int n) {
3. n = n - 1;
4. if (n == 0) {
5. return "\t" + list[n].toString();
6. } else {
7. return toString(list, (n)) + "\n\t" + list[n].toString();
8. }
9. }
10. public void displayHeavyPackets(Packet[] list, int n) {
11. n = n - 1;
12. if (n == 0) {
13. if (list[n].isHeavy()) {
14. System.out.print("\t" + list[n].toString());
15. }
16. } else {
17. displayHeavyPackets(list, n);
18. if (list[n].isHeavy()) {
19. System.out.print("\t" + list[n].toString() + "\n");
20. }
21. }
22. }
23. public void displayPacketsToDest(Packet[] list, int n, String dest) {
24. n = n - 1;
25. if (n == 0) {
26. if (list[n].getDestination().equals(dest)) {
27. System.out.print(list[n].toString());
28. }
29. } else {
30. displayPacketsToDest(list, n, dest);
31. if (list[n].getDestination().equals(dest)) {
32. System.out.print("\t" + list[n].toString() + "\n");
33. }
34. }
35. }
36. public int countPacketsToDest(Packet[] list, int n, String dest) {
37. n = n - 1;
38. if (n == 0) {
39. if (list[n].getDestination().equals(dest)) {
40. return 1;
41. } else {
42. return 0;
43. }
44. } else {
45. if (list[n].getDestination().equals(dest)) {
46. return 1 + countPacketsToDest(list, n, dest);
47. } else {
48. return 0 + countPacketsToDest(list, n, dest);
49. }
50. }
51. }
52. public Packet maxWeightPacket(Packet[] list, int n) {
53. n = n - 1;
54. if (n == 0) {
55. return list[n];
56. } else if (list[n].getWeight() > maxWeightPacket(list, n).getWeight()) {
57. return list[n];
58. } else {
59. return maxWeightPacket(list, n);
60. }
61. }
62. public double maxWeight(Packet[] list, int n) {
63. n = n - 1;
64. if (n == 0) {
65. return list[n].getWeight();
66. } else if (list[n].getWeight() > maxWeight(list, n)) {
67. return list[n].getWeight();
68. } else {
69. return maxWeight(list, n);
70. }
71. }
72. }

TestPackages:

1. import java.util.\*;
2. import java.io.\*;
3. public class TestPackages {
4. public static void main(String[] args) throws IOException {
5. Packet[] packetList = new Packet[100];
6. Recursion rec = new Recursion();
7. Scanner fScan = new Scanner(new File("packetData.txt"));
8. int id, counter = 0;
9. double weight;
10. String dest;
11. Packet pack; //NumberFormat fmt = new DecimalFormat("#.##");
12. while (fScan.hasNext()) {
13. id = fScan.nextInt();
14. weight = fScan.nextDouble();
15. dest = fScan.next();
16. pack = new Packet(id, weight, dest);
17. packetList[counter] = pack;
18. counter++;
19. }
20. System.out.println("All Packets");
21. System.out.println(rec.toString(packetList, counter));
22. System.out.println("\nAll Heavy Packets");
23. rec.displayHeavyPackets(packetList, counter);
24. System.out.println("\nAll Packets with destination NY");
25. rec.displayPacketsToDest(packetList, counter, "NY");
26. System.out.println("\nThe number of packets with destication CT is: " + rec.countPacketsToDest(packetList, counter, "CT"));
27. System.out.println("\nThe packet object with max weight is: " + rec.maxWeightPacket(packetList, counter));
28. System.out.println("\nThe max weight of all packets is: " + rec.maxWeight(packetList, counter));
29. }
30. }

Test run:



UML:

|  |
| --- |
| Packet |
| - idNumber : int  - weight : double  - destination : String |
| + Packet(int id, double wei, String dest) : void  + isHeavy() : Boolean  + toString() : String  + getWeight() : double  + getDestination() : String |

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

|  |
| --- |
| Recursion |
|  |
| + toString(Packet[] list, int n) : String  + displayHeavyPackets(Packet[], int n) : void  + displayPacketsToDest(Packet[], int n, String dest) : void  + countPacketsToDest(Packet[], int n, String dest) : int  + maxWeightPacket(Packet[], int n) : Packet  + maxWeight(Packet[], int n) : double |