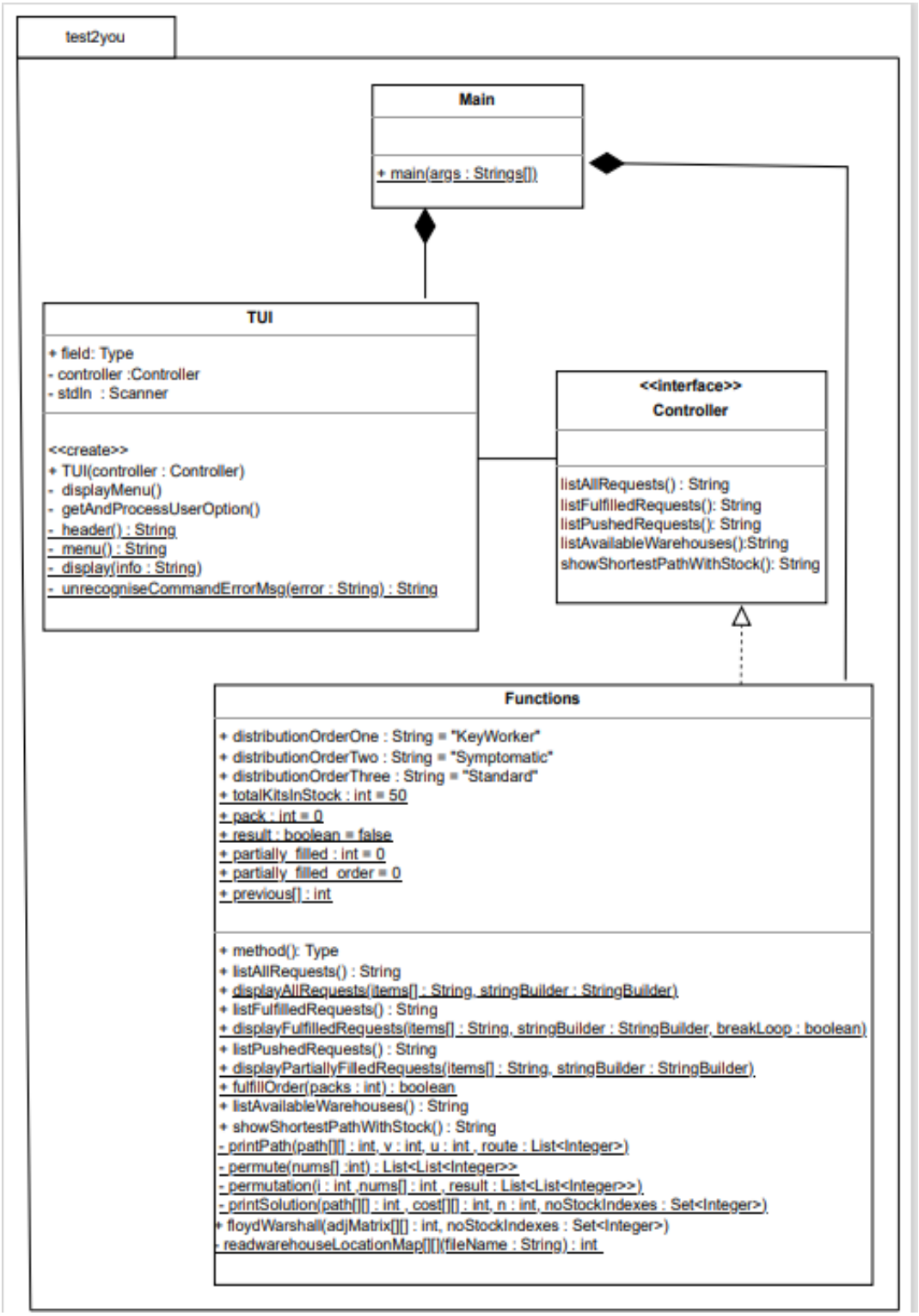
UML Class Diagram:



Functional Requirements:

1.

- Read the customer orders file.
- If data is present in the file, compare the line-by-line data with the string keyworker, if the keyworker is present in the comparing line then the displayAllRequests() method is used to sort the data in order.
- Appen the data from the displayAllRequest() method, and returns the data.
- If data is not present in the file "No Record found" in print.
- After comparing all lines with the keyworker, next compare Symptomatic with all the data in the file then sort, append, and returns the data, and then Standard is compared with all the data in the file then sort, append, and returns the data.

2.

- Read the customer order file.
- If the file consists of the data, compare the line-by-line data with the string keyworker, if the keyworker is present in the comparing line then the displayFulfilledRequests() method is used to display the fulfilled data with the help of fulfilledorder() method. fulfilledOrder() method is used to find out how many kits are fulfilled and partially filled data. If the kits are available displayFulfilledRequests() sorts the data and then appends and return the data.
- The above step is continued with another string Symptomatic and standard.
- If the file doesn't consist of the data, then print the "No Record Found".

3

- The kits are not available for distribution, the data is pushed to the next day's orders.
- Functional requirement 2 finds the fulfilled data, and the same process is followed to find the unfulfilled requests. The method fulfilledorder(), is used to find out the unfulfilled data, sort the data, and append and return the data.

4 .

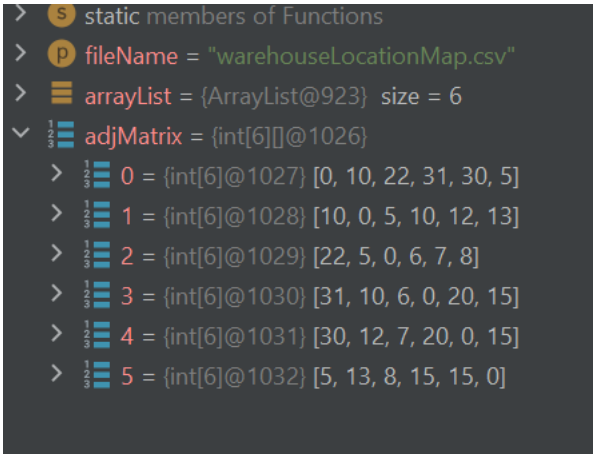
- Read the warehouse Stock file. If the file consists of the data if stock is not equal to zero add the data to the object and return it.

5.

- Floyd Warshall Algorithm:
- This algorithm is used to find out the shortest path between all the pairs of vertices in a weighted graph.
- Works for both direct and undirected graphs.
- It doesn't work for negative cycles.
- Follows the dynamic programming approach.
- It consists of 3 loops, so time complexity is $O(n^3)$. Space complexity is $O(n^2)$.

```
for (int k = 0; k < n; k++)
{
    for (int v = 0; v < n; v++)
    {
        for (int u = 0; u < n; u++)
        {
            if (cost[v][k] != Integer.MAX_VALUE
                    && cost[k][u] != Integer.MAX_VALUE
                    && (cost[v][k] + cost[k][u] < cost[v][u]))
            {
                cost[v][u] = cost[v][k] + cost[k][u];
                path[v][u] = path[k][u];
            }
        }
    }
}
```

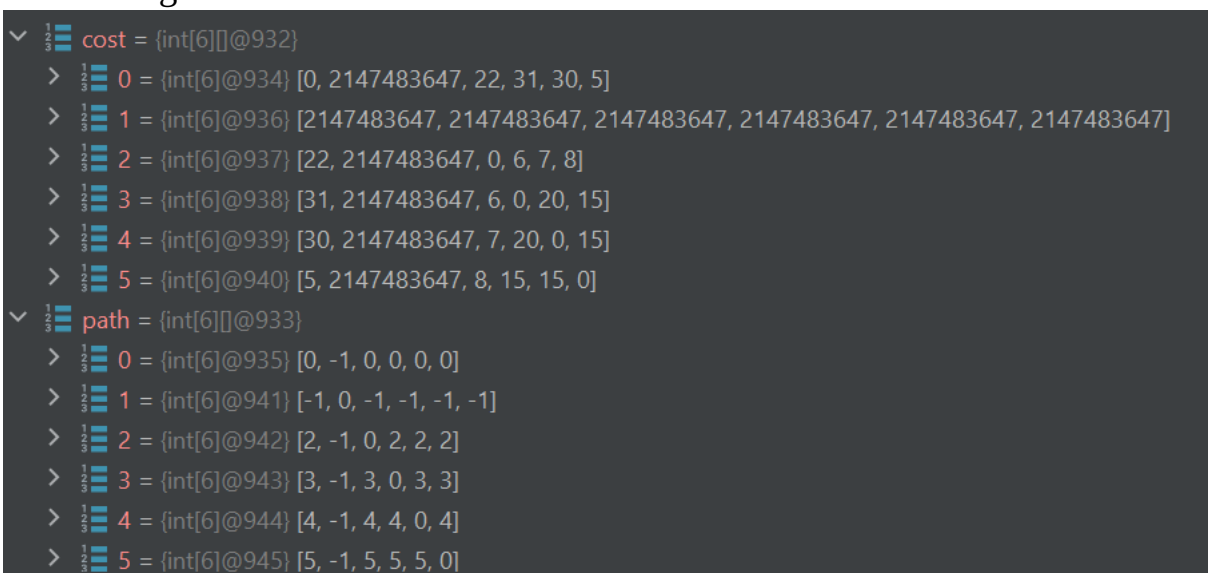
- Shortest path:
- Read the WarehouseLocationMap file. In warehouselocationMap() method is used to reads the distances of warehouses and returns adjMatrix .



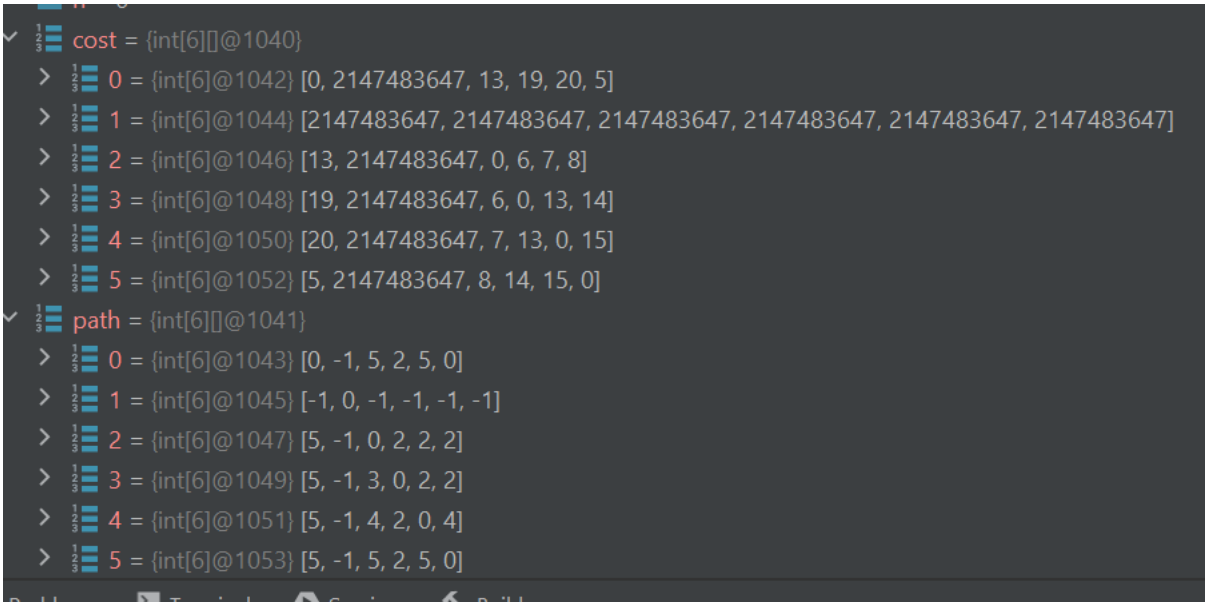
- Set the I to the MaxInteger.
- If Stock is not available in the warehouse, suppose not available in B(i.e.. index 1), the index 1 in the matrix is set to a max integer([0][1],[1][0],[1][1],[1][2],[2][1],[3][1] and so on…). So this path is not travelled.

Floydwarshell():

- In place of index 1(i.e., B) path is set to -1 and the cost(distance) is set to a max integer.

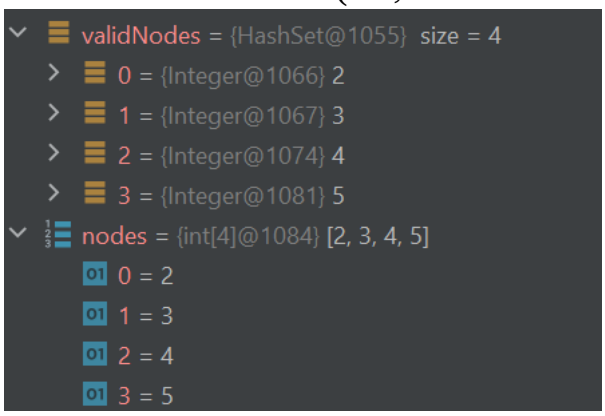


- Find the shortests distances of all pairs and set the path.



printSoultion():

- finding the valid nodes of ware houses (i.e., stock available paths)



- Permute() method Finds the best paths among all the possible paths. Permutations() method finds the different number available of paths . printpath() add the path and print the shortest path of the warehouses

References:

[1] Floyd-Warshall Algorithm (programiz.com)

[2] StringBuilder (Java SE 18 & JDK 18) (oracle.com)

[3] Different ways of Reading a text file in Java - GeeksforGeeks