|  |  |  |
| --- | --- | --- |
| **Name** | **Class** | **Scenarios** |
| SetupScenary1() | WeTrustTest | Empty graph G |
| setupScenary2() | WeTrustTest | A graph with connected vertices |
| setupScenary3() | WeTrustTest | The initial matrix with the weights      The matrix with the minimum weights between the vertices |
| setupScenary4() | WeTrustTest | Graph G, vertices connected with weighted edges. |

**Tests Design**

**Design test case**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| WeTrust | Floyd-Warshall | setupScenary3 | double[][] | It should return a new matrix with the minimum weights possible between every vertex |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| WeTrust | Dijkstra | setupScenary3 | double[][] | It should return the minimum weight between the vertices. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| WeTrust | BFS | setupScenary2 | Graph G  Vertex V (The initial vertex) | The method should have visited all the vertices in the right order (previously stated). Verified with an ArrayList. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| WeTrust | DFS | setupScenary2 | Graph G  Vertex V (The initial vertex) | The method should have visited all the vertices in the right order (previously stated). Verified with an ArrayList. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| WeTrust | Kruskal | setupScenary4 | Graph G  Vertices connected with corresponding weight | The total weight of the resulting subgraph must be the right value (the lowest possible having all the vertex connected). Also, the number or edges must be the number of vertex - 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| WeTrust | Prim | setupScenary4 | Graph G  Vertices connected with corresponding weight  Vertex V (The initial vertex) | The total weight of the resulting subgraph must be the right value (the lowest possible having all the vertex connected). Also, the number or edges must be the number of vertex - 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| AdjacencyMatrix | createGraph | Empty |  | An empty graph has been created. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| AdjacencyList | createGraph | Empty |  | An empty graph has been created. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| AdjacencyMatrix | addVertex | setupScenary1 | Graph G  Vertex V | Vertex V has been added to Graph G. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| AdjacencyList | addVertex | setupScenary1 | Graph G  Vertex V | Vertex V has been added to Graph G. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| AdjacencyMatrix | addEdge | setupScenary2 | Graph G  Vertex V  Vertex V1 | Vertex V and Vertex V1 now share an edge. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| AdjacencyList | addEdge | setupScenary2 | Graph G  Vertex V  Vertex V1 | Vertex V and Vertex V1 now share an edge. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| AdjacencyMatrix | removeVertex | setupScenary2 | Graph G  Vertex V | Vertex V is not in the graph anymore. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| AdjacencyList | removeVertex | setupScenary2 | Graph G  Vertex V | Vertex V is not in the graph anymore. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| AdjacencyMatrix | removeEdge | setupScenary2 | Graph G  Vertex V  Vertex V1 | Vertex V and Vertex V1 do not share an edge anymore. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| AdjacencyList | removeEdge | setupScenary2 | Graph G  Vertex V  Vertex V1 | Vertex V and Vertex V1 do not share an edge anymore. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| AdjacencyMatrix | vertexAdjacent | setupScenary2 | Graph G  Vertex V | Returns a list with all the vertices adjacent to V. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| AdjacencyList | vertexAdjacent | setupScenary2 | Graph G  Vertex V | Returns a list with all the vertices adjacent to V. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| AdjacencyMatrix | areConnected | setupScenary2 | Graph G  Vertex V  Vertex V1 | TRUE if V and V1 share an edge, FALSE if not. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective : To verify if the corresponding method is working properly** | | | | |
| **Class** | **Method** | **Scenarios** | **Entry values** | **Result** |
| AdjacencyList | areConnected | setupScenary2 | Graph G  Vertex V  Vertex V1 | TRUE if V and V1 share an edge, FALSE if not. |