Scenery configuration for graph methods:

|  |  |  |
| --- | --- | --- |
| Name | Class | Scenarios (Examples) |
| settingUpVertexes | TestsMatrixGraph | *Initializes a graph using the adjacency matrix implementation* |
| settingUpVertexes | TestsMatrixGraph | *Adds a new vertex with its id and data associated to it.* |
| settingUpVertexes | TestsMatrixGraph | *Tries to add a new vertex that shares the id with an already existing vertex.* |
| settingUpVertexes | TestsMatrixGraph | *Tries to add a new vertex that contains null data associated to it.* |
| settingUpVertexes | TestsMatrixGraph | *Tries to add a new vertex that shares the same data with an already existing vertex* |
| settingUpEdges | TestMatrixGraph | *Initializes several vertexes in the graph and creates and edge between them.* |
| settingUpEdges | TestMatrixGraph | *Tries to create an edge between and existing and a nonexistent vertex.* |
| settingUpEdges | TestMatrixGraph | *Tries to create an edge between two existing edges in the graph, but with negative weight.* |
| settingUpEdges | TestMatrixGraph | *Tries to create an edge between one vertex and itself using a valid weight.* |
| setUpForDijkstra | TestMatrixGraph | *Initializes a graph with several vertexes that do not have any existing connection between them* |
| setUpForDijkstra1 | TestMatrixGraph | *Initializes a graph with several vertexes that do have an existing connection but it’s the same between them* |
| setUpForDijkstra1 | TestMatrixGraph | *Initializes a graph with several vertexes that do have an existing connection but it’s the same between them.* |
| setUpForDijkstra2 | TestMatrixGraph | *Initializes graph with several connections between them and multiple weights.* |
| settingUpForMatrix | TestMatrixGraph | *Initializes a graph that does not contain any graph, so the representative matrix is empty.* |
| settingUpForMatrix | TestMatrixGraph | *Initializes a graph that does contain vertexes and connections between them, so the matrix contains the information.* |
| settingUpForMatrix1 | TestMatrixGraph | *Initializes a graph that only contains a vertex pointing to nothing* |
| settingUpTrasversal | TestMatrixGraph | *Initializes a graph that contains multiple vertexes that contain multiple with different weights between them.* |
| settingUpTrasversal1 | TestMatrixGraph | *Initializes an empty graph with any path.* |
| settingUpTrasversal2 | TestMatrixGraph | *Initializes a graph with multiple vertexes but none of them are accessible between them.* |
| Setting List Graph | TestListGaph | *Initializes a graph implemented using adjacency lists and adds a vertex with (Key = A, Data = 1), then checks if the addition to the Hash table is done* |
| Setting Double Vertex | TestListGraph | *Initializes a graph implemented using adjacency lists and tries to add two vertexes that contain the same associated key. (Key =A, Data = 1) and Key =A, Data = 2), then checks that the second vertex was not added.* |
| Setting Multiple Vertexes | TestListGraph | *Initializes a graph implemented using adjacency lists and adds multiple vertexes containing non repeated data, ("A", 1); ("B", 2); ("C", 3); then verifies that they were added successfully. ("A", 1);("B", 2);*  *("C", 3l); then checks that the Hash contains them* |
| Setting Null Vertex | TestListGraph | *Initializes a graph implemented using adjacency lists and adds multiple vertexes containing non repeated data, ("A", 1); ("B", 2); ("C", 3); then verifies that they were added successfully. ("A", 1);("B", 2);*  *("C", null); then checks that the third vertex was not added.* |
| Setting Valid Edges | TestListGraph | *Initializes a graph implemented using adjacency lists, creates multiple vertexes with multiple data and creates edges containing a valid* |
| Setting Valid Edges | TestListGraph |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Designing test cases for the graph implemented with adjacency matrix: