edu.uci.ics.jung.graph

Class UndirectedSparseGraph<V,E>

```
java.lang.Object
  └edu.uci.ics.jung.graph.AbstractGraph<V,E>
      └edu.uci.ics.jung.graph.AbstractTypedGraph<V,E>
          └edu.uci.ics.jung.graph.UndirectedSparseGraph<V,E>
```

All Implemented Interfaces:

Graph<V,E>, Hypergraph<V,E>, UndirectedGraph<V,E>, Serializable

---

```
public class UndirectedSparseGraph<V,E>
extends AbstractTypedGraph<V,E>
implements UndirectedGraph<V,E>, Serializable
```

An implementation of `UndirectedGraph` that is suitable for sparse graphs.

See Also:

Serialized Form

Field Summary

`protected Map<E,Pair<V>>`	**`edges`**
`protected Map<V,Map<V,E>>`	**`vertices`**

Fields inherited from class edu.uci.ics.jung.graph.AbstractTypedGraph
`edge_type`

Constructor Summary

`UndirectedSparseGraph``()` Creates an instance.

Method Summary

`boolean`	**`addEdge`**`(E edge, Pair<? extends V> endpoints, EdgeType edgeType)` Adds `edge` to this graph with the specified `endpoints` and `EdgeType`.
`boolean`	**`addVertex`**`(V vertex)` Adds `vertex` to this graph.
`boolean`	**`containsEdge`**`(E edge)` Returns true if this graph's edge collection contains `edge`.

boolean	**containsVertex**(V vertex) Returns true if this graph's vertex collection contains vertex.
E	**findEdge**(V v1, V v2) Returns an edge that connects this vertex to v.
Collection<E>	**findEdgeSet**(V v1, V v2) Returns all edges that connects this vertex to v.
V	**getDest**(E directed_edge) If directed_edge is a directed edge in this graph, returns the destination; otherwise returns null.
int	**getEdgeCount**() Returns the number of edges in this graph.
Collection<E>	**getEdges**() Returns a view of all edges in this graph.
Pair<V>	**getEndpoints**(E edge) Returns the endpoints of edge as a Pair.
static <V,E> s15.Factory<UndirectedGraph<V,E>>	**getFactory**() Returns a Factory that creates an instance of this graph type.
Collection<E>	**getIncidentEdges**(V vertex) Returns the collection of edges in this graph which are connected to vertex.
Collection<E>	**getInEdges**(V vertex) Returns a Collection view of the incoming edges incident to vertex in this graph.
Collection<V>	**getNeighbors**(V vertex) Returns the collection of vertices which are connected to vertex via any edges in this graph.
Collection<E>	**getOutEdges**(V vertex) Returns a Collection view of the outgoing edges incident to vertex in this graph.
Collection<V>	**getPredecessors**(V vertex) Returns a Collection view of the predecessors of vertex in this graph.
V	**getSource**(E directed_edge) If directed_edge is a directed edge in this graph, returns the source; otherwise returns null.
Collection<V>	**getSuccessors**(V vertex) Returns a Collection view of the successors of vertex in this graph.

int	**getVertexCount**() Returns the number of vertices in this graph.
Collection<V>	**getVertices**() Returns a view of all vertices in this graph.
boolean	**isDest**(V vertex, E edge) Returns true if vertex is the destination of edge.
boolean	**isSource**(V vertex, E edge) Returns true if vertex is the source of edge.
boolean	**removeEdge**(E edge) Removes edge from this graph.
boolean	**removeVertex**(V vertex) Removes vertex from this graph.

Methods inherited from class edu.uci.ics.jung.graph.AbstractTypedGraph
getDefaultEdgeType, getEdgeCount, getEdges, getEdgeType, hasEqualEdgeType, validateEdgeType

Methods inherited from class edu.uci.ics.jung.graph.AbstractGraph
addEdge, addEdge, addEdge, addEdge, addEdge, degree, getIncidentCount, getIncidentVertices, getNeighborCount, getOpposite, getPredecessorCount, getSuccessorCount,getValidatedEndpoints, inDegree, isIncident, isNeighbor, isPredecessor, isSuccessor, outDegree, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Methods inherited from interface edu.uci.ics.jung.graph.Graph
addEdge, addEdge, getOpposite, getPredecessorCount, getSuccessorCount, inDegree, isPredecessor, isSuccessor, outDegree

Methods inherited from interface edu.uci.ics.jung.graph.Hypergraph
addEdge, addEdge, degree, getDefaultEdgeType, getEdgeCount, getEdges, getEdgeType, getIncidentCount, getIncidentVertices, getNeighborCount, isIncident, isNeighbor