Class Vertex

java.lang.Object
    Vertex

```
public class Vertex
extends java.lang.Object
```

Field Summary

Fields

Modifier and Type	Field and Description
`static double`	`INF`

Constructor Summary

Constructors

Constructor and Description
`Vertex(int id, java.lang.String label)` Instantiates a new Vertex

Method Summary

All Methods **Instance Methods** **Concrete Methods**

Modifier and Type	Method and Description
`void`	`addAdj(Edge e)` Adds the given Edge to this Vertex's neighbor list.
`void`	`addAdj(Vertex vdst, double w)` Creates and adds a new Edge with vdst and w to this Vertex's neighbor list.

void	**addCost**(double c) Adds c to the current cost of this Vertex ('s path)
java.util.TreeMap<java.lang.String,**Edge**>	**getAdj**() Returns the adjacency list (neighbors) as a TreeMap of outgoing edges sorted by dst vertex label.
Edge	**getAdj**(java.lang.String dst) Returns the edge connecting this vertex to the Vertex labeled dst, or returns null if no such edge exists.
double	**getCost**() Returns the cost of this vertex
int	**getID**() Returns the id of this vertex
java.lang.String	**getLabel**() Returns the label of this vertex
Vertex	**getPred**() Returns the predecessor vertex of this vertex
boolean	**isMarked**() Returns whether this Vertex is processed (marked, or visited) or not.
void	**mark**() Marks this Vertex as processed (or visited; no need to check)
int	**nAdj**() Returns the number of neighbors this vertex has (size of adjacency list)
void	**reset**() Resets this Vertex for graph algorithms (marked to false, cost to +infinity, pred to null)
void	**reset**(boolean mark, double newCost, **Vertex** newPred) Resets this Vertex for graph algorithms using the parameters.

void	**setCost**(double c) Sets the cost of this Vertex ('s path) to c
void	**setPred**(**Vertex** p) Sets the predecessor vertex of this vertex to p
java.lang.String	**toString**() Returns the information of this vertex in String format
void	**unmark**() Marks this Vertex as _not_ processed; no need to check

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Detail

INF

```
public static final double INF
```

See Also:

Constant Field Values

Constructor Detail

Vertex

```
public Vertex(int id,
              java.lang.String label)
```

Instantiates a new Vertex

Parameters:

`id` - the ID of the Vertex

`label` - the label of the Vertex

Method Detail

getID

```
public int getID()
```

Returns the id of this vertex

Returns:

`Returns the id of this vertex`

getLabel

```
public java.lang.String getLabel()
```

Returns the label of this vertex

Returns:

`Returns the label of this vertex`

getPred

```
public Vertex getPred()
```

Returns the predecessor vertex of this vertex

Returns:

`Returns the predecessor vertex of this vertex`

getCost

```
public double getCost()
```

Returns the cost of this vertex

Returns:

`Returns the cost of this vertex`

isMarked

```
public boolean isMarked()
```

Returns whether this Vertex is processed (marked, or visited) or not.

Returns:

`Returns whether this Vertex is processed`

getAdj

```
public java.util.TreeMap<java.lang.String,Edge> getAdj()
```

Returns the adjacency list (neighbors) as a TreeMap of outgoing edges sorted by dst vertex label.

Returns:

`Returns the adjacency list`

reset

```
public void reset(boolean mark,
                  double newCost,
                  Vertex newPred)
```

Resets this Vertex for graph algorithms using the parameters.

Parameters:

`mark` - the mark value

`newCost` - the new cost

`newPred` - the new predecessor

reset

```
public void reset()
```

Resets this Vertex for graph algorithms (marked to false, cost to +infinity, pred to null)

mark

```
public void mark()
```

Marks this Vertex as processed (or visited; no need to check)

unmark

```
public void unmark()
```

Marks this Vertex as _not_ processed; no need to check

addCost

```
public void addCost(double c)
```

Adds c to the current cost of this Vertex ('s path)

Parameters:

`c - added cost`

setCost

```
public void setCost(double c)
```

Sets the cost of this Vertex ('s path) to c

Parameters:

`c - new cost`

setPred

```
public void setPred(Vertex p)
```

Sets the predecessor vertex of this vertex to p

Parameters:

`p - the new predecessor`

nAdj

```
public int nAdj()
```

Returns the number of neighbors this vertex has (size of adjacency list)

Returns:

`number of neighnors`

getAdj

```
public Edge getAdj(java.lang.String dst)
```

Returns the edge connecting this vertex to the Vertex labeled dst, or returns null if no such edge exists.

Parameters:

`dst - the label of the dst vertex`

Returns:

`the target edge, null if not found`

toString

```
public java.lang.String toString()
```

Returns the information of this vertex in String format

Overrides:

`toString in class java.lang.Object`

addAdj

```
public void addAdj(Vertex vdst,
                   double w)
```

Creates and adds a new Edge with vdst and w to this Vertex's neighbor list.

Parameters:

`vdst - the destination vertex`

`w - the cost from source to destination`

addAdj

```
public void addAdj(Edge e)
```

Adds the given Edge to this Vertex's neighbor list.

Parameters:

`e - the edge being added`

PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD