# Class Graph

java.lang.Object
    Graph

---

```
public class Graph
extends Object
```

## *Field Summary*

### Fields

| Modifier and Type | Field and Description |
|---|---|
| `private Map<Integer,Set<Integer>>` | `adj` |
| `private int` | `E` |
| `private static String` | `NEWLINE` |
| `private int` | `V` |

## *Constructor Summary*

### Constructors

| Constructor and Description |
|---|
| `Graph()`<br>Initializes an empty graph |

## *Method Summary*

**All Methods** Instance Methods Concrete Methods

| Modifier and Type | Method and Description |
|---|---|
| `void` | `addEdge(int v, int w)`<br>Adds the undirected edge v-w to this graph. |
| `boolean` | `addVertex(int v)`<br>Adds the vertex v to this graph |
| `int` | `degree(int v)`<br>Returns the degree of vertex v. |

| | |
|---|---|
| int | **edges**()<br>Returns the number of edges in this graph. |
| Iterator<Integer> | **getAdjacent**(int v)<br>Returns the vertices adjacent to vertex v. |
| String | **toString**()<br>Returns a string representation of this graph. |
| private void | **validateVertex**(int v)<br>Ensures the argument is a valid vertex in the graph |
| int | **vertices**()<br>Returns the number of vertices in this graph. |

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## *Field Detail*

### NEWLINE

```
private static final String NEWLINE
```

### V

```
private int V
```

### E

```
private int E
```

### adj

```
private Map<Integer,Set<Integer>> adj
```

## *Constructor Detail*

### Graph

```
public Graph()
```

Initializes an empty graph

## *Method Detail*

### vertices

```
public int vertices()
```

Returns the number of vertices in this graph.

**Returns:**

`the number of vertices in this graph`

### edges

```
public int edges()
```

Returns the number of edges in this graph.

**Returns:**

`the number of edges in this graph`

### validateVertex

```
private void validateVertex(int v)
```

Ensures the argument is a valid vertex in the graph

**Throws:**

`IllegalArgumentException` - if v is not a valid vertex

### addVertex

```
public boolean addVertex(int v)
```

Adds the vertex v to this graph

**Parameters:**

`v - one vertex in the graph`

**Returns:**

`true if v was added, false otherwise`

### addEdge

```
public void addEdge(int v,
                    int w)
```

Adds the undirected edge v-w to this graph. The arguments must be valid vertices in the graph.

**Parameters:**

`v - one vertex in the edge`

`w - the other vertex in the edge`

**Throws:**

`IllegalArgumentException - if either vertex does not exist`

### getAdjacent

```
public Iterator<Integer> getAdjacent(int v)
```

Returns the vertices adjacent to vertex v.

**Parameters:**

`v - the vertex`

**Returns:**

`an Iterator containing the vertices adjacent to vertex v`

**Throws:**

`IllegalArgumentException - if v is not a valid vertex`

### degree

```
public int degree(int v)
```

Returns the degree of vertex v.

**Parameters:**

`v - the vertex`

**Returns:**

`the degree of vertex v`

**Throws:**

`IllegalArgumentException` - if v is not a valid vertex

### toString

```
public String toString()
```

Returns a string representation of this graph.

**Overrides:**

`toString` in class `Object`

**Returns:**

the number of vertices *V*, followed by the number of edges *E*, followed by the *V* adjacency lists