| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-all.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | | |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/info/gridworld/grid/Location.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?info/gridworld/grid/UnboundedGrid.html)    [**NO FRAMES**](http://docs.google.com/UnboundedGrid.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |

## **info.gridworld.grid**

Class UnboundedGrid<E>

[java.lang.Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true)  
 [info.gridworld.grid.AbstractGrid](http://docs.google.com/info/gridworld/grid/AbstractGrid.html)<E>  
 **info.gridworld.grid.UnboundedGrid<E>**

**All Implemented Interfaces:** [Grid](http://docs.google.com/info/gridworld/grid/Grid.html)<E>

public class **UnboundedGrid<E>**extends [AbstractGrid](http://docs.google.com/info/gridworld/grid/AbstractGrid.html)<E>

An UnboundedGrid is a rectangular grid with an unbounded number of rows and columns.

The implementation of this class is testable on the AP CS AB exam.

| **Constructor Summary** | |
| --- | --- |
| [**UnboundedGrid**](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html#UnboundedGrid())()            Constructs an empty unbounded grid. |

| **Method Summary** | |
| --- | --- |
| [E](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html) | [**get**](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html#get(info.gridworld.grid.Location))([Location](http://docs.google.com/info/gridworld/grid/Location.html) loc)            Returns the object at a given location in this grid. |
| int | [**getNumCols**](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html#getNumCols())()            Returns the number of columns in this grid. |
| int | [**getNumRows**](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html#getNumRows())()            Returns the number of rows in this grid. |
| [ArrayList](http://java.sun.com/j2se/1.5.0/docs/api/java/util/ArrayList.html?is-external=true)<[Location](http://docs.google.com/info/gridworld/grid/Location.html)> | [**getOccupiedLocations**](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html#getOccupiedLocations())()            Gets the locations in this grid that contain objects. |
| boolean | [**isValid**](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html#isValid(info.gridworld.grid.Location))([Location](http://docs.google.com/info/gridworld/grid/Location.html) loc)            Checks whether a location is valid in this grid. |
| [E](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html) | [**put**](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html#put(info.gridworld.grid.Location,%20E))([Location](http://docs.google.com/info/gridworld/grid/Location.html) loc, [E](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html) obj)            Puts an object at a given location in this grid. |
| [E](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html) | [**remove**](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html#remove(info.gridworld.grid.Location))([Location](http://docs.google.com/info/gridworld/grid/Location.html) loc)            Removes the object at a given location from this grid. |

| **Methods inherited from class info.gridworld.grid.**[**AbstractGrid**](http://docs.google.com/info/gridworld/grid/AbstractGrid.html) |
| --- |
| [getEmptyAdjacentLocations](http://docs.google.com/info/gridworld/grid/AbstractGrid.html#getEmptyAdjacentLocations(info.gridworld.grid.Location)), [getNeighbors](http://docs.google.com/info/gridworld/grid/AbstractGrid.html#getNeighbors(info.gridworld.grid.Location)), [getOccupiedAdjacentLocations](http://docs.google.com/info/gridworld/grid/AbstractGrid.html#getOccupiedAdjacentLocations(info.gridworld.grid.Location)), [getValidAdjacentLocations](http://docs.google.com/info/gridworld/grid/AbstractGrid.html#getValidAdjacentLocations(info.gridworld.grid.Location)), [toString](http://docs.google.com/info/gridworld/grid/AbstractGrid.html#toString()) |

| **Methods inherited from class java.lang.**[**Object**](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) |
| --- |
| [clone](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#clone()), [equals](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#equals(java.lang.Object)), [finalize](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#finalize()), [getClass](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#getClass()), [hashCode](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#hashCode()), [notify](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#notify()), [notifyAll](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#notifyAll()), [wait](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#wait()), [wait](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#wait(long)), [wait](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### UnboundedGrid

public **UnboundedGrid**()

Constructs an empty unbounded grid.

| **Method Detail** |
| --- |

### getNumRows

public int **getNumRows**()

**Description copied from interface:** [**Grid**](http://docs.google.com/info/gridworld/grid/Grid.html#getNumRows()) Returns the number of rows in this grid.

**Returns:**the number of rows, or -1 if this grid is unbounded

### getNumCols

public int **getNumCols**()

**Description copied from interface:** [**Grid**](http://docs.google.com/info/gridworld/grid/Grid.html#getNumCols()) Returns the number of columns in this grid.

**Returns:**the number of columns, or -1 if this grid is unbounded

### isValid

public boolean **isValid**([Location](http://docs.google.com/info/gridworld/grid/Location.html) loc)

**Description copied from interface:** [**Grid**](http://docs.google.com/info/gridworld/grid/Grid.html#isValid(info.gridworld.grid.Location)) Checks whether a location is valid in this grid.

Precondition: loc is not null

**Parameters:**loc - the location to check **Returns:**true if loc is valid in this grid, false otherwise

### getOccupiedLocations

public [ArrayList](http://java.sun.com/j2se/1.5.0/docs/api/java/util/ArrayList.html?is-external=true)<[Location](http://docs.google.com/info/gridworld/grid/Location.html)> **getOccupiedLocations**()

**Description copied from interface:** [**Grid**](http://docs.google.com/info/gridworld/grid/Grid.html#getOccupiedLocations()) Gets the locations in this grid that contain objects.

**Returns:**an array list of all occupied locations in this grid

### get

public [E](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html) **get**([Location](http://docs.google.com/info/gridworld/grid/Location.html) loc)

**Description copied from interface:** [**Grid**](http://docs.google.com/info/gridworld/grid/Grid.html#get(info.gridworld.grid.Location)) Returns the object at a given location in this grid.

Precondition: loc is valid in this grid

**Parameters:**loc - a location in this grid **Returns:**the object at location loc (or null if the location is unoccupied)

### put

public [E](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html) **put**([Location](http://docs.google.com/info/gridworld/grid/Location.html) loc,  
 [E](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html) obj)

**Description copied from interface:** [**Grid**](http://docs.google.com/info/gridworld/grid/Grid.html#put(info.gridworld.grid.Location,%20E)) Puts an object at a given location in this grid.

Precondition: (1) loc is valid in this grid (2) obj is not null

**Parameters:**loc - the location at which to put the objectobj - the new object to be added **Returns:**the object previously at loc (or null if the location was previously unoccupied)

### remove

public [E](http://docs.google.com/info/gridworld/grid/UnboundedGrid.html) **remove**([Location](http://docs.google.com/info/gridworld/grid/Location.html) loc)

**Description copied from interface:** [**Grid**](http://docs.google.com/info/gridworld/grid/Grid.html#remove(info.gridworld.grid.Location)) Removes the object at a given location from this grid.

Precondition: loc is valid in this grid

**Parameters:**loc - the location of the object that is to be removed **Returns:**the object that was removed (or null if the location is unoccupied)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-all.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | | |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/info/gridworld/grid/Location.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?info/gridworld/grid/UnboundedGrid.html)    [**NO FRAMES**](http://docs.google.com/UnboundedGrid.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |