|  | /\* |
| --- | --- |
|  | \* AP(r) Computer Science GridWorld Case Study: |
|  | \* Copyright(c) 2005-2006 Cay S. Horstmann (http://horstmann.com) |
|  | \* |
|  | \* This code is free software; you can redistribute it and/or modify |
|  | \* it under the terms of the GNU General Public License as published by |
|  | \* the Free Software Foundation. |
|  | \* |
|  | \* This code is distributed in the hope that it will be useful, |
|  | \* but WITHOUT ANY WARRANTY; without even the implied warranty of |
|  | \* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the |
|  | \* GNU General Public License for more details. |
|  | \* |
|  | \* @author Cay Horstmann |
|  | \*/ |
|  |  |
|  | package info.gridworld.grid; |
|  |  |
|  | import java.util.ArrayList; |
|  |  |
|  | /\*\* |
|  | \* <code>AbstractGrid</code> contains the methods that are common to grid |
|  | \* implementations. <br /> |
|  | \* The implementation of this class is testable on the AP CS AB exam. |
|  | \*/ |
|  | public abstract class AbstractGrid<E> implements Grid<E> |
|  | { |
|  | public ArrayList<E> getNeighbors(Location loc) |
|  | { |
|  | ArrayList<E> neighbors = new ArrayList<E>(); |
|  | for (Location neighborLoc : getOccupiedAdjacentLocations(loc)) |
|  | neighbors.add(get(neighborLoc)); |
|  | return neighbors; |
|  | } |
|  |  |
|  | public ArrayList<Location> getValidAdjacentLocations(Location loc) |
|  | { |
|  | ArrayList<Location> locs = new ArrayList<Location>(); |
|  |  |
|  | int d = Location.NORTH; |
|  | for (int i = 0; i < Location.FULL\_CIRCLE / Location.HALF\_RIGHT; i++) |
|  | { |
|  | Location neighborLoc = loc.getAdjacentLocation(d); |
|  | if (isValid(neighborLoc)) |
|  | locs.add(neighborLoc); |
|  | d = d + Location.HALF\_RIGHT; |
|  | } |
|  | return locs; |
|  | } |
|  |  |
|  | public ArrayList<Location> getEmptyAdjacentLocations(Location loc) |
|  | { |
|  | ArrayList<Location> locs = new ArrayList<Location>(); |
|  | for (Location neighborLoc : getValidAdjacentLocations(loc)) |
|  | { |
|  | if (get(neighborLoc) == null) |
|  | locs.add(neighborLoc); |
|  | } |
|  | return locs; |
|  | } |
|  |  |
|  | public ArrayList<Location> getOccupiedAdjacentLocations(Location loc) |
|  | { |
|  | ArrayList<Location> locs = new ArrayList<Location>(); |
|  | for (Location neighborLoc : getValidAdjacentLocations(loc)) |
|  | { |
|  | if (get(neighborLoc) != null) |
|  | locs.add(neighborLoc); |
|  | } |
|  | return locs; |
|  | } |
|  |  |
|  | /\*\* |
|  | \* Creates a string that describes this grid. |
|  | \* @return a string with descriptions of all objects in this grid (not |
|  | \* necessarily in any particular order), in the format {loc=obj, loc=obj, |
|  | \* ...} |
|  | \*/ |
|  | public String toString() |
|  | { |
|  | String s = "{"; |
|  | for (Location loc : getOccupiedLocations()) |
|  | { |
|  | if (s.length() > 1) |
|  | s += ", "; |
|  | s += loc + "=" + get(loc); |
|  | } |
|  | return s + "}"; |
|  | } |
|  | } |