```
/*
 * 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 Chris Nevison
 * @author Barbara Cloud Wells
 * @author Cay Horstmann
 */

import info.gridworld.actor.Actor;
import info.gridworld.actor.Critter;
import info.gridworld.grid.Grid;
import info.gridworld.grid.Location;
import javafx.util.Pair;

import java.awt.Color;
import java.util.ArrayList;

public class KnightCritter extends Critter
{
        private static int m_r[] = {-1, 1, 2, -2, 2, -2, 1, -1};
        private static int m_c[] = {2, 2, 1, 1, -1, -1, -2, -2};
    public KnightCritter()
    {
        setColor(Color.orange);
    }

    public ArrayList<Actor> getActors()
    {
        ArrayList<Actor> actors = new ArrayList<Actor>();
        for (Location loc : getLocationsInDirections())
        {
            Actor a = getGrid().get(loc);
            if (a != null)
                actors.add(a);
        }

        return actors;
    }
    public ArrayList<Location> getMoveLocations()
    {
        ArrayList<Location> locs = new ArrayList<Location>();
        for (Location loc : getLocationsInDirections())
            if (getGrid().get(loc) == null)
                locs.add(loc);

        return locs;
    }

    public void makeMove(Location loc)
    {
        if (loc.equals(getLocation()))
        {
            double r = Math.random();
            int angle;
            if (r < 0.5)
                angle = Location.LEFT;
            else
                angle = Location.RIGHT;
            setDirection(getDirection() + angle);
        }
        else
            super.makeMove(loc);
    }

    public ArrayList<Location> getLocationsInDirections()
    {
        ArrayList<Location> locs = new ArrayList<Location>();
        Grid gr = getGrid();
        Location loc = getLocation();

        for (int i = 0; i < 8; ++i)
        {
        Location neighborLoc = new Location(loc.getRow() + m_r[i],
loc.getCol() + m_c[i]);
            if (gr.isValid(neighborLoc))
                locs.add(neighborLoc);
        }
        return locs;
    }
}
```