Лабораторная работа №3

по дисциплине

“Объектно-ориентированное программирование”

Выполнил студент

группы БВТ2005

Нурмагомедов Магомед

Москва 2020

**Цель работы:** завершить реализацию A\* алгоритма.

**Задание:**

1) Обеспечить реализацию метода equals ().

2) Обеспечить реализацию метода hashcode().

3) Реализоватьметодpublic int numOpenWaypoints().

4) Реализоватьметодpublic Waypoint getMinOpenWaypoint().

5) Реализоватьметодpublic booleanaddOpenWaypoint(Waypoint newWP).

6) Реализоватьметодpublic booleanisLocationClosed(Location loc).

7) Реализоватьметодpublic void closeWaypoint(Location loc).

**Результат**

@Override  
public booleanequals(Object o) {  
if (this == o)  
return true;  
 if (o == null || getClass() != o.getClass())  
return false;  
Location location = (Location) o;  
 return xCoord== location.xCoord&&yCoord== location.yCoord;  
}  
  
@Override  
public inthashCode() {  
return Objects.*hash*(xCoord, yCoord);  
}

*/\*\* Returns the current number of open waypoints. \*\*/*public intnumOpenWaypoints()  
{  
return openWaypoints.size();  
}

public Waypoint getMinOpenWaypoint() {  
if (openWaypoints.size() == 0) return null;  
  
ArrayList<Waypoint> waypoints = new ArrayList<>(openWaypoints.values());  
 double minCost = waypoints.get(0).getTotalCost();  
Waypoint minWaypoint = waypoints.get(0);  
intlength = waypoints.size();  
  
 for (inti = 0; i< length; i++) {  
if (waypoints.get(i).getTotalCost() <minCost) {  
minWaypoint = waypoints.get(i);  
minCost = minWaypoint.getTotalCost();  
}  
 }  
  
return minWaypoint;  
}

public booleanaddOpenWaypoint(Waypoint newWP)  
{  
if (openWaypoints.get(newWP.getLocation().hashCode()) == null) {  
openWaypoints.put(newWP.getLocation().hashCode(), newWP);  
 return false;  
} else {  
if (openWaypoints.get(newWP.getLocation().hashCode()).getPreviousCost() >newWP.getPreviousCost()) {  
openWaypoints.put(newWP.getLocation().hashCode(), newWP);  
 return true;  
}  
 }  
  
return false;  
}

public booleanisLocationClosed(intloc)  
{  
return closeWaypoints.containsKey(loc);  
}

public void closeWaypoint(intloc)  
{  
closeWaypoints.put(loc, openWaypoints.remove(loc));  
}

