Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Ордена Трудового Красного Знамени

федеральное государственное бюджетное образовательное учреждение высшего образования

МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ

Кафедра «Математической кибернетики и информационных технологий»

Информационные технологии и программирование

Лабораторная работа № 3

Выполнил:

студент группы БВТ2102

Гаджиявов Джамал

Москва 2022

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

**Задание:**

1) Обеспечить реализацию метода equals ().

2) Обеспечить реализацию метода hashcode().

3) Реализовать метод public int numOpenWaypoints().

4) Реализовать метод public Waypoint getMinOpenWaypoint().

5) Реализовать метод public boolean addOpenWaypoint(Waypoint newWP).

6) Реализовать метод public boolean isLocationClosed(Location loc).

7) Реализовать метод public void closeWaypoint(Location loc).

**Результат**

@Override  
public boolean equals(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 int hashCode() {  
 return Objects.*hash*(xCoord, yCoord);  
}

*/\*\* Returns the current number of open waypoints. \*\*/*public int numOpenWaypoints()  
{  
 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);  
 int length = waypoints.size();  
  
 for (int i = 0; i < length; i++) {  
 if (waypoints.get(i).getTotalCost() < minCost) {  
 minWaypoint = waypoints.get(i);  
 minCost = minWaypoint.getTotalCost();  
 }  
 }  
  
 return minWaypoint;  
}

public boolean addOpenWaypoint(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 boolean isLocationClosed(int loc)  
{  
 return closeWaypoints.containsKey(loc);  
}

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

