Федеральное агенство связи

Ордена Трудового Красного Знамени федеральное государственное

бюджетное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра Математической кибернетики и

информационных технологий

Лабораторная работа №3

по дисциплине: «Технология разработки программного обеспечения»

на тему: «Алгоритм A\* («A star»)»

Выполнил студент

группы БФИ1902

Рахимов Е.К.

Проверила:

Мосева М.С.

Москва, 2020 г.

Оглавление

[1. Цель лабораторной работы 2](#_Toc58524139)

[2. Задание на лабораторную работу 3](#_Toc58524140)

[3. Ход лабораторной работы 4](#_Toc58524141)

[3.1 Листинг программы 4](#_Toc58524142)

[3.2 Результат выполнения программы 7](#_Toc58524143)

[Список использованных источников 8](#_Toc58524144)

# 1. Цель лабораторной работы

Цель данной лабораторной работы — изучить основы синтаксиса Java с помощью нескольких простых задач программирования, научиться использовать компилятор Java и виртуальную машину Java для запуска программы.

# 2. Задание на лабораторную работу

1) **Location**:

Переопределить 2 метода – equals и hashcode

2) **AStarState**:

Создать HashMap<Location, Waypoint> для открытых и закрытых вершин. Реализовать метод public int numOpenWaypoints() - этот метод возвращает количество точек в наборе открытых вершин. Реализовать метод public Waypoint getMinOpenWaypoint() – эта функция должна проверить все вершины в наборе открытых вершин, и после этого она должна вернуть ссылку на вершину с наименьшей общей стоимостью. Если в "открытом" наборе нет вершин, функция возвращает NULL.

Реализовать метода public boolean addOpenWaypoint(Waypoint newWP). Если в наборе «открытых вершин» в настоящее время нет вершины для данного местоположения, то необходимо просто добавить новую вершину. • Если в наборе «открытых вершин» уже есть вершина для этой локации, добавьте новую вершину только в том случае, если стоимость пути до новой вершины меньше стоимости пути до текущей.

Реализовать метод public boolean isLocationClosed(Location loc) - эта функция должна возвращать значение true, если указанное местоположение встречается в наборе закрытых вершин, и false в противном случае.

Реализовать метод public void closeWaypoint(Location loc) - эта функция перемещает вершину из набора «открытых вершин» в набор «закрытых вершин». Так как вершины обозначены местоположением, метод принимает местоположение вершины.

# 3. Ход лабораторной работы

## 3.1 Листинг программы

public class Location  
{  
 public int xCoord;  
 public int yCoord;  
  
  
 public Location(int x, int y)  
 {  
 xCoord = x;  
 yCoord = y;  
 }  
  
 public Location()  
 {  
 this(0, 0);  
 }  
   
 public boolean equals(Object obj)   
 {   
  
 if (obj instanceof Location)  
 {  
 Location other = (Location) obj;   
   
 if (xCoord == other.xCoord &&   
 yCoord == other.yCoord)   
 {   
 return true;   
 }   
 }   
   
 return false;   
 }  
   
 public int hashCode()   
 {   
 int result = 20;  
  
 result = 16 \* result + (xCoord \* 13);  
 result = 22 \* result + (yCoord \* 11);  
   
 return result;   
 }  
}

public class AStarState  
{  
   
 private Map2D map;  
  
   
 private HashMap<Location, Waypoint> openWaypoints;  
   
   
 private HashMap<Location, Waypoint> closedWaypoints;   
  
   
 public AStarState(Map2D map)  
 {  
 if (map == null)  
 throw new NullPointerException("map cannot be null");  
  
 this.map = map;  
   
 openWaypoints = new HashMap<Location, Waypoint>();  
 closedWaypoints = new HashMap<Location, Waypoint>();  
 }  
   
 public Map2D getMap()  
 {  
 return map;  
 }  
  
   
 public Waypoint getMinOpenWaypoint()  
 {  
 Waypoint sol = null;  
   
 float min = Float.*POSITIVE\_INFINITY*;  
 float totalCost = 0;  
   
 for(Waypoint p : openWaypoints.values())  
 {  
 totalCost = p.getTotalCost();  
   
 if(min > totalCost)  
 {  
 min = totalCost;  
 sol = p;  
 }  
 }  
   
 return sol;  
 }  
  
   
 public boolean addOpenWaypoint(Waypoint newWP)  
 {  
 Waypoint other = openWaypoints.get(newWP.loc);  
   
 if(other == null || newWP.getPreviousCost() < other.getPreviousCost())  
 {  
 openWaypoints.put(newWP.loc, newWP);  
 return true;  
 }  
  
 return false;  
 }  
  
  
   
 public int numOpenWaypoints()  
 {  
 return openWaypoints.size();  
 }  
  
   
 public void closeWaypoint(Location loc)  
 {  
 Waypoint point = openWaypoints.remove(loc);  
   
 if(point != null)  
 {  
 closedWaypoints.put(loc, point);  
 }  
 }  
  
   
 public boolean isLocationClosed(Location loc)  
 {  
 return closedWaypoints.containsKey(loc);  
 }

## 3.2 Результат выполнения программы

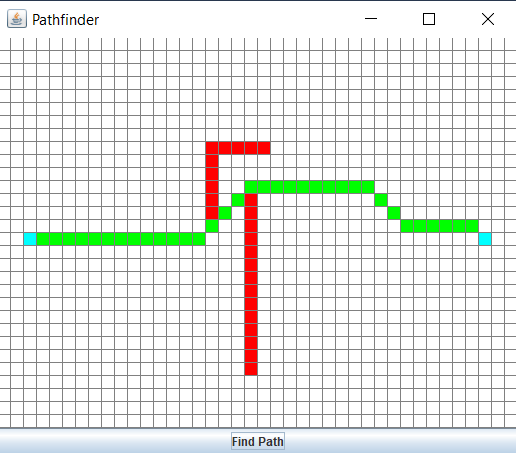


Рисунок 1 – результат выполнения

# Список использованных источников

1) ГОСТ 7.32-2017 Система стандартов по информации, библиотечному и издательскому делу. Отчёт о научно-исследовательской работе. Структура и правила оформления

2) ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание. Общие требования и правила составления