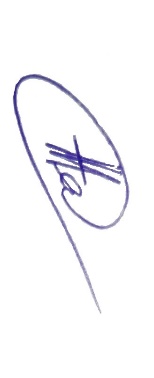
**МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ**

**Лабораторная работа №4**

«A\*»

Вариант 16

 Выполнил: Некрасов А. А.   
БСТ1601

Проверил: Городничев М.Г

Москва 2018

Оглавление

[Цель работы: 2](#_Toc507492825)

[Используемые инструменты: 2](#_Toc507492826)

[Блок-схемы алгоритмов 2](#_Toc507492827)

[Код программы: 2](#_Toc507492828)

[Компиляция и выполнение программы: 3](#_Toc507492829)

[Вывод: 3](#_Toc507492830)

Цель работы: Научиться работать с классами.

Используемые инструменты: для написания кода программы будем использовать редактор кода Visual Studio Code. Для создания, компилирования и дебага программ будем использовать jdk v9.0.4.

## Функции которые были добавлены в файл Location

//функция для сравнения точек

public boolean equals(Object obj)

{

    if (obj == null) {

     return false;

    }

    if (obj instanceof Location)

{

Location other = (Location) obj;

if (xCoord == other.xCoord &&

yCoord == other.yCoord)

{

return true;

}

}

return false;

}

//подсчет хешкода

public int hashCode()

{

int result = 17;

result = 9 \* result + (xCoord \* 8);

result = 14 \* result + (yCoord \* 10);

return result;

}

## Функции которые были добавлены в файл AStarState

/\*\*

\* Этот метод добавляет маршрутную точку в коллекцию

\* "открытые маршрутные точки"

\* (или потенциально обновляет уже существующую).

\* Если уже не открыть точку на местоположение новой

\* путевой точки, то новая точка будет добавлена в

\* коллекцию. Однако, если там уже есть точка на

\* новой путевой точки местоположения новой путевой

\* точки заменяет старый новый путевой "старой цене" значение меньше текущей

\* путевой точки "старой цене" значение.

\*\*/

public boolean addOpenWaypoint(Waypoint newWP)

{

Waypoint other = openWp.get(newWP.getLocation());

if(other == null || newWP.getPreviousCost() < other.getPreviousCost())

{

openWp.put(newWP.getLocation(), newWP);

return true;

}

return false;

}

/\*\* Возвращает текущее количество открытых точек.

\*\*/

public int numOpenWaypoints()

{

return openWp.size();

}

//Этот метод перемещает точку в указанном месте из открытого списка в закрытый список.

public void closeWaypoint(Location loc)

{

Waypoint p = openWp.remove(loc);

if(p != null)

{

closeWp.put(loc, p);

}

}

//Возвращает True, если коллекция закрытых точек содержит точки в указанном месте.

public boolean isLocationClosed(Location loc)

{

return closeWp.containsKey(loc);

}

}

Вывод: В рамках этой работы я научился писать простейшие программы на java, компилировать программный код и выполнять скомпилированные программы.