**Министерство цифрового развития, связи и массовых**

**коммуникаций Российской Федерации**

**Ордена Трудового Красного Знамени**

**федеральное государственное бюджетное**

**учреждение высшего образования**

**«Московский Технический Университет Связи и Информатики»**

Кафедра «Математическая кибернетика и информационные

технологии»

**ЛАБОРАТОРНАЯ РАБОТА №3**

по дисциплине «Кроссплатформенные технологии программирования»

по теме «Алгоритм A\*»

Выполнил: студент группы БСТ2001

Савкин Д. И.

Проверил: ст. преп. каф. «Информатика»

Мосева М. С.

Москва 2022

# Цели и задачи

Загрузить исходные коды программы по поиску пути между двумя точками пространства с помощью алгоритма «A\*» и реализовать несколько методов с целью доведения программы для работоспособного состояния.

# Ход выполнения работы

Для выполнения задания, откроем исходный код класса Location и реализуем методы equals(…) и hashCode() следующим образом:



Это необходимо для того, чтобы экземпляры данного класса правильно работали внутри хэш-таблицы в качестве ключа. Сравнение экземпляров зависит от значений координат, что гарантирует истину в том случае, если два объекта указывают на одну и ту же точку в пространстве.

В классе AStarState реализуем метод getMinOpenWaypoint(), отвечающий за возврат из таблицы точки маршрутка с наименьшей «ценой»:



Далее реализуем метод addOpenWaypoint(…), отвечающий за добавление или обновление существующей точки маршрутка, если она более выгодна по «цене»:



Далее реализуем метод numOpenWaypoints(), который возвращает количество открытых точек маршрута:



Далее реализуем метод closeWaypoint(…), который помечает точку маршрута по указанному положению как закрытую.



Далее реализуем метод isLocationClosed(…), который возвращает истину, если точка маршрута по указанному положению находится в закрытом состоянии:



# ЗАКЛЮЧЕНИЕ

В ходе работы были реализованы методы с целью доведения программы для работоспособного состояния. Результат работы программы представлен на рисунке 1.

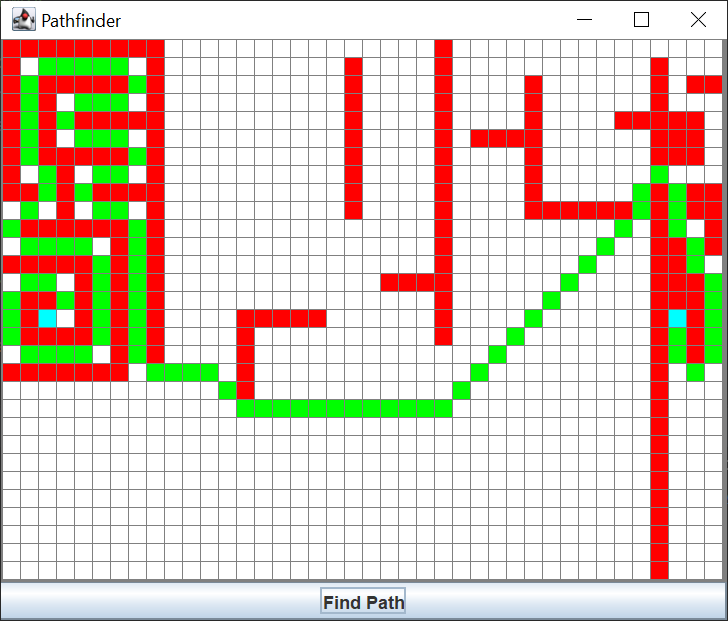


Рисунок 1 — Графическая визуализация алгоритма поиска пути «A\*»