**Федеральное агентство связи**

**Ордена трудового Красного Знамени федеральное государственное бюджетное**

**образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математическая кибернетика и информационные технологии

Отчет по лабораторной работе №3

по дисциплине «Объектно-ориентированное программирование»

Выполнил: студент группы БВТ2002

ФИО: Мосева Алеся Сергеевна

Проверил:Полянцева Ксения Андреевна

Москва, 2020

**Цель работы**: необходимо реализовать правильную работу алгоритма A-star.

**Задания:**

**Локации**

Для начала необходимо подготовить класс Location для совместного

использования с классами коллекции Java. Поскольку вы будете использовать

контейнеры для хеширования для выполнения данного задания, то для этого

необходимо:

* Обеспечить реализацию метода equals ().
* Обеспечить реализацию метода hashcode().

Добавьте реализацию каждого из этих методов в класс Location, следуя шаблонам в классе. После этого вы можете использовать класс Location в качестве ключевого типа в контейнерах хеширования, таких как HashSet и HashMap.

**Состояния А\***

После того, как класс Location готов к использованию, вы можете

завершить реализацию класса AStarState. Это класс, который поддерживает

наборы открытых и закрытых вершин, поэтому он действительно обеспечивает

основную функциональность для реализации алгоритма А\*.

Как упоминалось ранее, состояние А\* состоит из двух наборов вершин,

один из открытых вершин и другой из закрытых. Чтобы упростить алгоритм,

вершины будут храниться в хэш-карте, где местоположение вершин является

ключом, а сами вершины являются значениями. Таким образом, у вас будет

такой тип:

HashMap<Location, Waypoint>

(Очевидный вывод из всего этого заключается в том, что с каждым

местоположением на карте может быть связана только одна вершина.)

Добавьте два (нестатических) поля в класс AStarState с таким типом, одно

для "открытых вершин" и другой для "закрытых вершин". Кроме того, не

забудьте инициализировать каждое из этих полей для ссылки на новую пустую

коллекцию.

После создания и инициализации полей, вы должны реализовать

следующие методы в классе AStarState:

1) public int numOpenWaypoints()

Этот метод возвращает количество точек в наборе открытых вершин.

2) 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.

Не удаляйте вершину из набора после того, как вы вернули ее; просто

верните ссылку на точку с наименьшей общей стоимостью.

3) public boolean addOpenWaypoint(Waypoint newWP)

Это самый сложный метод в классе состояний А\*. Данный метод

усложняет то, что он должен добавлять указанную вершину только в том

случае, если существующая вершина хуже новой. Вот что должен делать этот

метод:

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. (Убедитесь, что

используете не общую стоимость.) Другими словами, если путь через новую

вершину короче, чем путь через текущую вершину, замените текущую вершину

на новую

Как вы могли заметить, что в таком случае вам потребуется извлечь

существующую вершину из «открытого набора», если таковая имеется. Данный

шаг довольно прост - замените предыдущую точку на новую, используя метод

HashMap.put(), который заменит старое значение на новое. Пусть данный метод

вернет значение true, если новая вершина была успешно добавлена в набор, и

false в противном случае.

4) public boolean isLocationClosed(Location loc)

Эта функция должна возвращать значение true, если указанное

местоположение встречается в наборе закрытых вершин, и false в противном

случае. Так как закрытые вершины хранятся в хэш-карте с расположениями в

качестве ключевых значений, данный метод достаточно просто в реализации.

5) 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.

Процесс должен быть простым:

Удалите вершину, соответствующую указанному местоположению

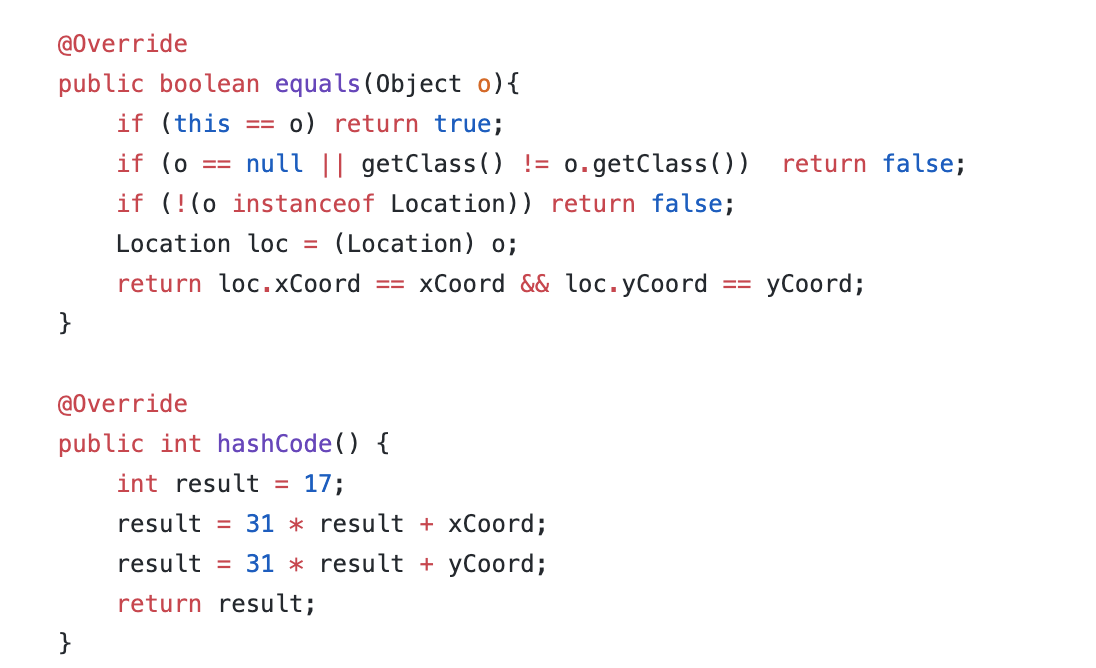
из набора «открытых вершин».

Добавьте вершину, которую вы удалили, в набор закрытых вершин.

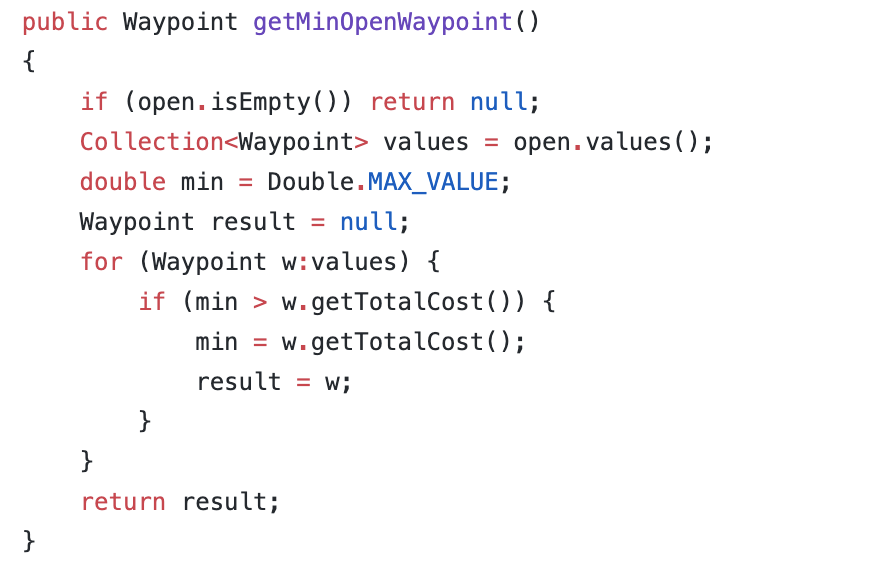
Ключом должно являться местоположение точки.

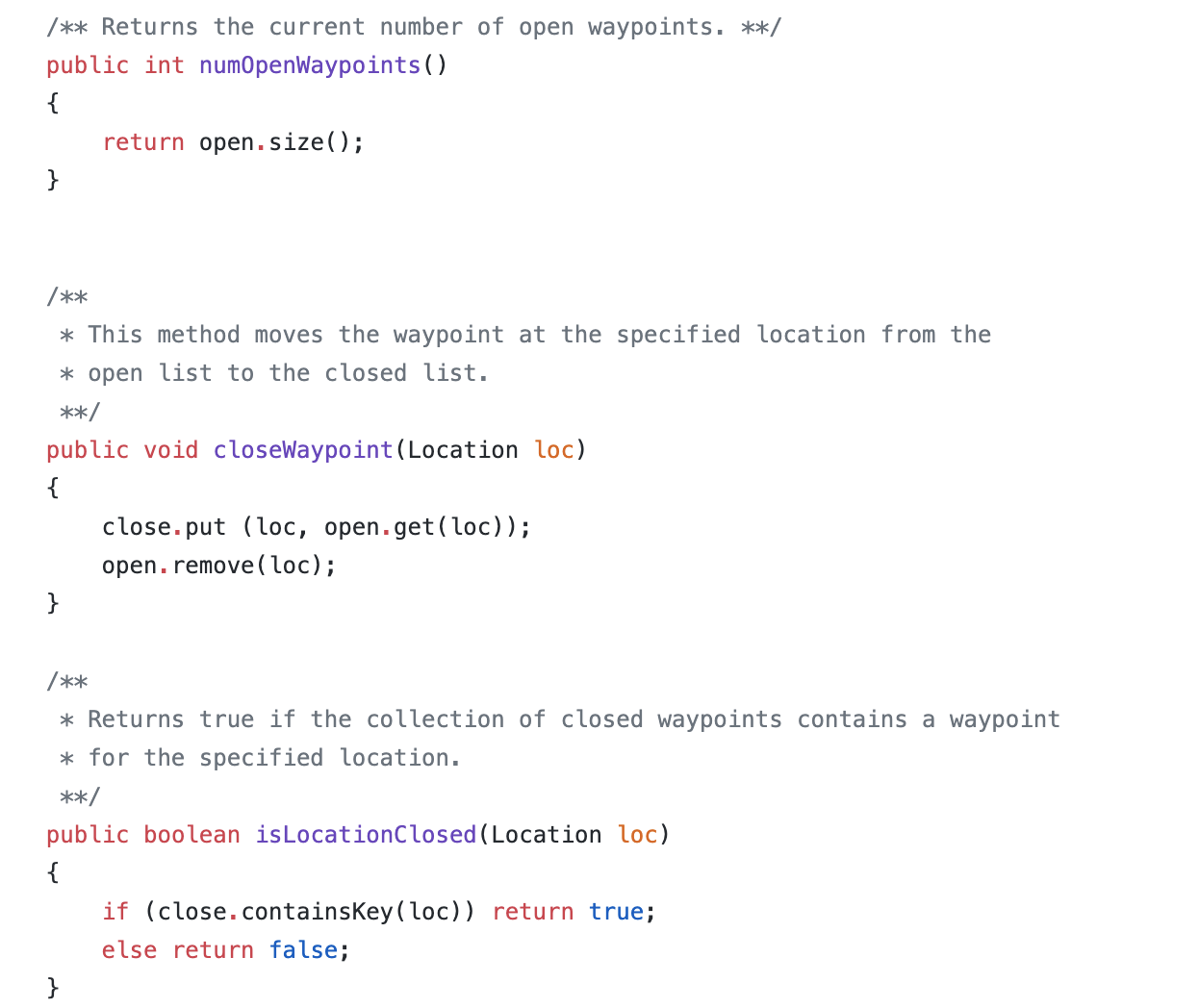
**Код программы:**

1. Реализация методов equals() и hashсode() в классе Location

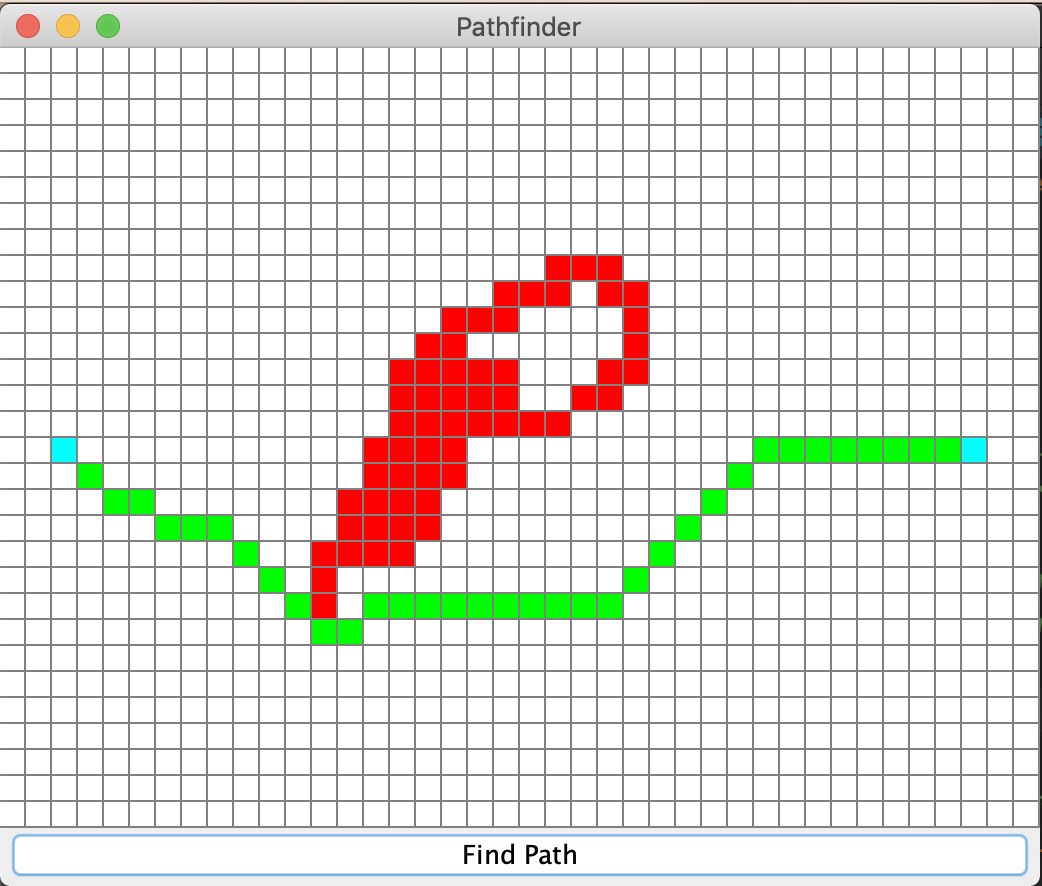


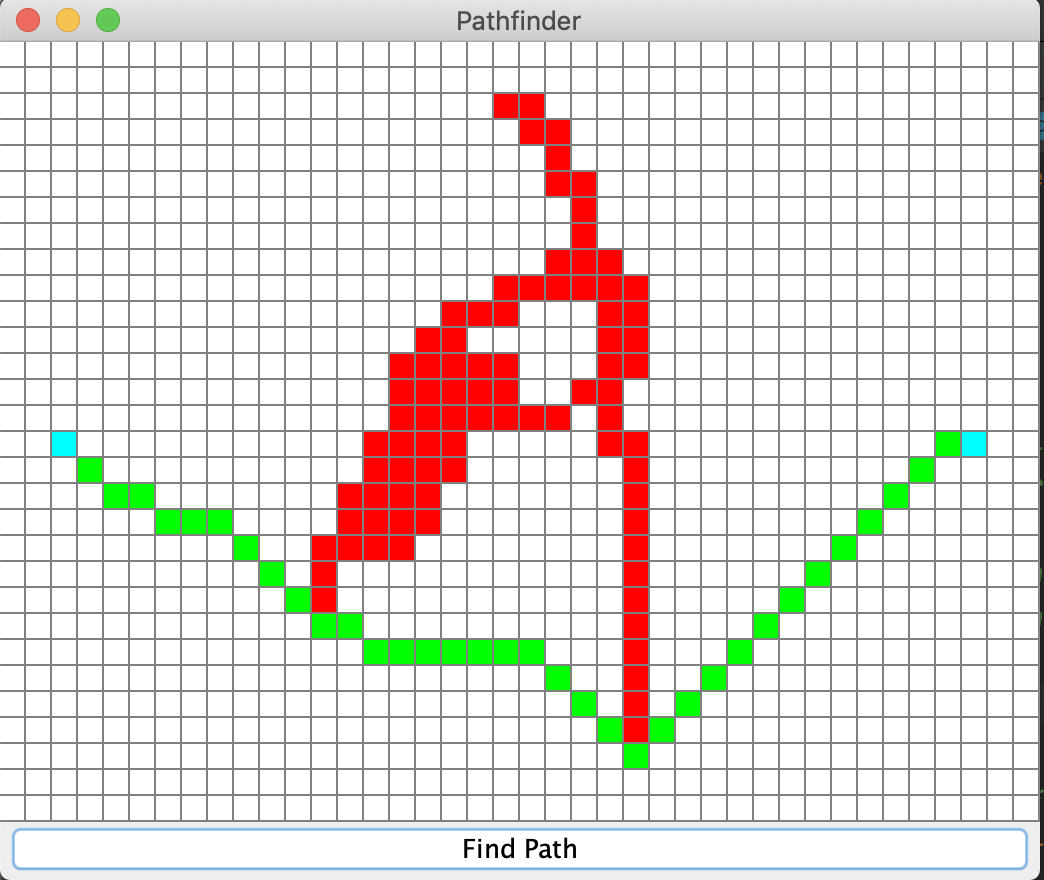
1. Файл Point3d.java





**Скриншоты выполнения:**





**Заключение:**

Таким образом, мы разобрали реализацию алгоритма нахождения кратчайшего пути A star.