1. Map2D.java - представляет карту, по которой перемещается алгоритм А\*, включая в себя информацию о проходимости ячеек
2. Location.java - этот класс представляет координаты конкретной ячейки на карте. В данном классе нам необходимо реализовать методы equals() – сравнивает данное местоположение с другими. И hashcode()- предоставляет хэш-код для каждого местоположения.
3. Waypoint.java - представляет отдельные вершины в сгенерированном пути
4. AStarPathfinder.java - этот класс реализует алгоритм поиска пути А\* в виде статического метода
5. AStarState.java - этот класс хранит набор открытых и закрытых вершин, и предоставляет основные операции, необходимые для функционирования алгоритма поиска А\*. В данный класс мы добавляем два нестатических поля типа HashMap<Location, Waypoint>, одно для «открытых вершин», другое для «закрытых вершин». После мы реализуем методы:
   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.
   3. public boolean addOpenWaypoint(Waypoint newWP)- этот метод должен делать:
      * 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.
   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.
6. AStarApp.java - простое Swing-приложение, которое обеспечивает редактируемый вид 2D-карты, и запускает поиск пути по запросу
7. JMapCell.java - это Swing -компонент, который используется для отображения состояния ячеек на карте

Кроме классов Location и AStarState в данной лабораторной работе мы больше ничего не меняем и не дополняем.