**3.1 Структуры данных**

Таблица 1 – Основные структуры данных, необходимые для поиска кратчайшего пути

|  |  |
| --- | --- |
| **Класс** | **Назначение** |
| Ore | Класс, который описывает тип руды и путь к картинке, изображающей данный тип |
| OreTypes | Перечисление, описывающие все типы руд, которые могут быть на поле |
| Cell | Класс, описывающий позицию клетки и тип руды, который клетка может принимать |
| Field | Класс поля состоит из двумерного массива клеток. Имеет определенную высоту и ширину |
| Node | Класс, описывающий элемент графа.  Содержит номер руды, к которой принадлежит, клетку из которой пришли в данный элемент, клетку самого элемента, предыдущий элемент, для восстановления кратчайшего пути, вес ребра к данному элементу, минимальный путь к данному элементу от точки старта |
| Graph | Основной класс, описывающий граф.  Содержит поле, массив руд с номером руды и клетками принадлежащими каждой, клетку и номер старта, клетку и номер ада, минимальный путь от старта до ада, клетки и элементы, входящие в кратчайший путь |

**3.2 Основные методы**

Таблица 2 – Основные методы класса Graph

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| findVein(int x, int y, OreTypes type, ArrayList<Cell> vein) | Параметры:   * int x, y – позиция клетки, с которой начинается новая жила * OreTypes type – тип руды новой жилы * ArrayList<Cell> vein – массив, в который будут записаны все клетки принадлежащие одной жиле   Метод ищет все клетки принадлежащие одной жиле |
| findVeins() | Находит все жилы на поле |
| findMinWayBetweenTwoVeins(ArrayList<Cell> vein1, ArrayList<Cell> vein2, Integer numberVein1, Integer numberVein2) | Параметры:   * ArrayList<Cell> vein1 – массив клеток жилы, из которой ищем кратчайший путь * ArrayList<Cell> vein2 – массив клеток жилы, в которую ищем кратчайший путь * Integer numberVein1 – номер жилы, из которой ищем кратчайший путь * Integer numberVein2 – номер жилы, в которую ищем кратчайший путь   Метод ищет кратчайший путь между двумя жилами, сравнивая поочередно расстояния между всеми клетками жил |
| findMinWayBetweenVeins() | Находит кратчайшие пути между всеми жилами |
| setNodesToGraph() | Добавляет элементы в граф |
| setCellsOfGround(Cell from, Cell to) | Параметры:   * Cell from – клетка выхода из жилы * Cell to - клетка входа в жилу   Добавляет в массив клеток кратчайшего пути клетки земли, которые встречаются при переходе из одной жилы в другую |
| dijkstra() | Реализует работу алгоритма Дейкстры |
| shortestWay() | Восстанавливает кратчайший путь полученный в результате работы алгоритма Дейкстры |
| getCellsOfShortestWay() | Возращает все клетки, которые входят в кратчайший путь.  Кратчайший путь – путь от точки старта до точки ада  В состав клеток входят:   * клетки входа/выхода в/из жил * клетки земли, для перехода из одной жилы в следующую * клетка старта * клетка ада |
| getVeins() | Возращает все жилы расположенные на поле |
| getGraph() | Возращает переменную HashMap<Integer, ArrayList<Node>> graph, которая хранит в себе номер жилы и все клетки принадлежащие данному номеру жилы |
| getLengthMinWay() | Возвращает длину минимального пути от точки старта до точки ада |