

Local Search :

The screenshot shows an IDE with a project named "Bot-AI_Side [route-optimizer]". The project structure on the left includes a package hierarchy: `com.university.routing.algorithms`, which contains `AStarAlgorithm`, `localSearch` (highlighted with a red circle), `TSPGeneticSolver`, and `UseAlgorithms`. There is also a `Map` package with various services and a `models` package with `Graph`, `Node`, and `Main`.

The main editor displays the `localSearch.java` file. The code implements a local search algorithm to optimize a path. It starts with an import statement, followed by the class declaration `public class localSearch` (circled in red). The class contains two static methods: `optimizePath` and `applyLocalSearch`.

```
3 > import ...
6
7 public class localSearch { // Локальный поиск
8
9     2 usages new *
    public static List<String> optimizePath(Graph graph, List<String> path) {
10         boolean improvement = true;
11         while (improvement) {
12             improvement = false;
13             for (int i = 1; i < path.size() - 2; i++) {
14                 for (int j = i + 1; j < path.size() - 1; j++) {
15                     if (swapImprovesPath(graph, path, i, j)) {
16                         Collections.swap(path, i, j);
17                         improvement = true;
18                     }
19                 }
20             }
21         }
22         return path;
23     }
24
25     1 usage new *
    public static List<String> applyLocalSearch(Graph graph, List<String> path) {
26         // Добавляем первую точку в конец пути, чтобы сделать его замкнутым
27         path.add(path.get(0));
28     }
29 }
```