

Documentation for the Speed It Up Junior 2021 Problem (https://github.com/IvanValBozhanin/Parking_Spaces)

Main class: This class's function is to execute the whole program. In its body it only contains the method for entering the data about the parking places — `inputDataForTheParkPlaces` (or the `generateRandomData` method, which was used while testing), as well as calling the constructor for the `ShowListPanel` class to show the data using GUI.

- The `inputDataForTheParkPlaces` method is using the terminal so it can read and save the information. It requires information for exactly 154 parking spaces. They should be inputted one by one following the process: write the number of the space; hit enter; write the area; hit enter; write the name of its user; hit enter; and write a description for it; hit enter. Should the user get lost while typing, s/he can read the prompt, which will display what information should be entered next. After each park space, before the final, there will be a message signaling whether or not should be typed information for the next park places. A message will appear when everything is typed correctly and the GUI is loading/has loaded.

`TypeOfSorting` enum: Easily makes distinct the different criteria for sorting when a query is invoked. Its members are: `NUMBER`, `AREA`, and `NAME`.

`ParkSpace` class: This class enables us to save the information for each of the parking spaces using a class, and to initialize some static members which will be needed for the program. The static members: `numberOfParkPlaces` (this can be used while testing, for example, with less or more than 154 parking spaces), `typeOfSorting`, and `parkingPlaces` are involved throughout the execution, regardless of the class's instance. Provides a constructor, mutators and accessors for the field, the methods `toString` (used for testing and debugging) and `compareTo` – involved when sorting the data later on. The method `getDataFromParkingPlaces` is used whenever we need to get the data from all parking places in the format of 2D array of strings.

`ShowListPanel` class: This is the class which actually delivers the information using Java Swing. The constructor is the main part of this class. It initializes the field variables, establishes the functionality of the `JTable` and the `JButton`'s. It sets all components into the `JFrame` and then configures some of its parameters to fit the needs of the solution to the problem. The table shows the four categories of information, presented in four different columns. Using the three buttons for sorting – by number, by area, and by name – we can get the table to display the result of this process (of course, with park place #74 at the top).

- An auxiliary class `MyTableCellRenderer` is used in order to define the properties of the final column. The final column presents description and since it can get quite long, we need to configure a text-wrapping (hyphenation) mechanism.
- The class has a method `updateTable` which is called initially in the constructor and every time we require a different way of sorting the information. Its main purpose is to “refresh” the information in the `JTable` and display the result from the queried sorting criteria.