

Object Oriented Programming 2 Final Project Report

Alihan BOZKIR 151220182032

Uml Diagram:

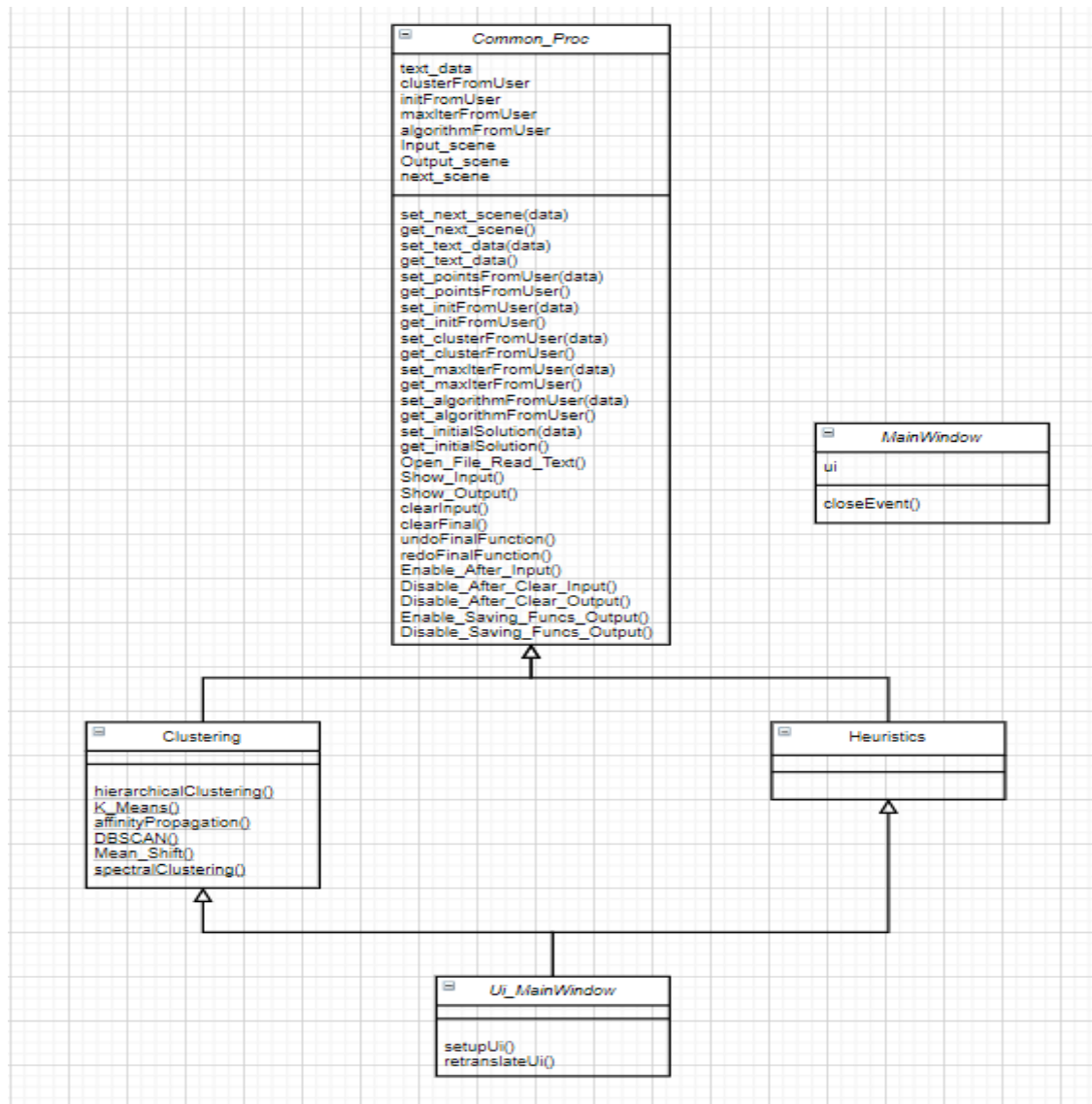


Figure 1 UML Diagram of the Project

MainWindow Class:

Our MainWindow class acts as a runner. QMainWindow is the parent class of our MainWindow class. Our program activates the GUI of our software from the moment this class is run and brings it to the screen of our computer.

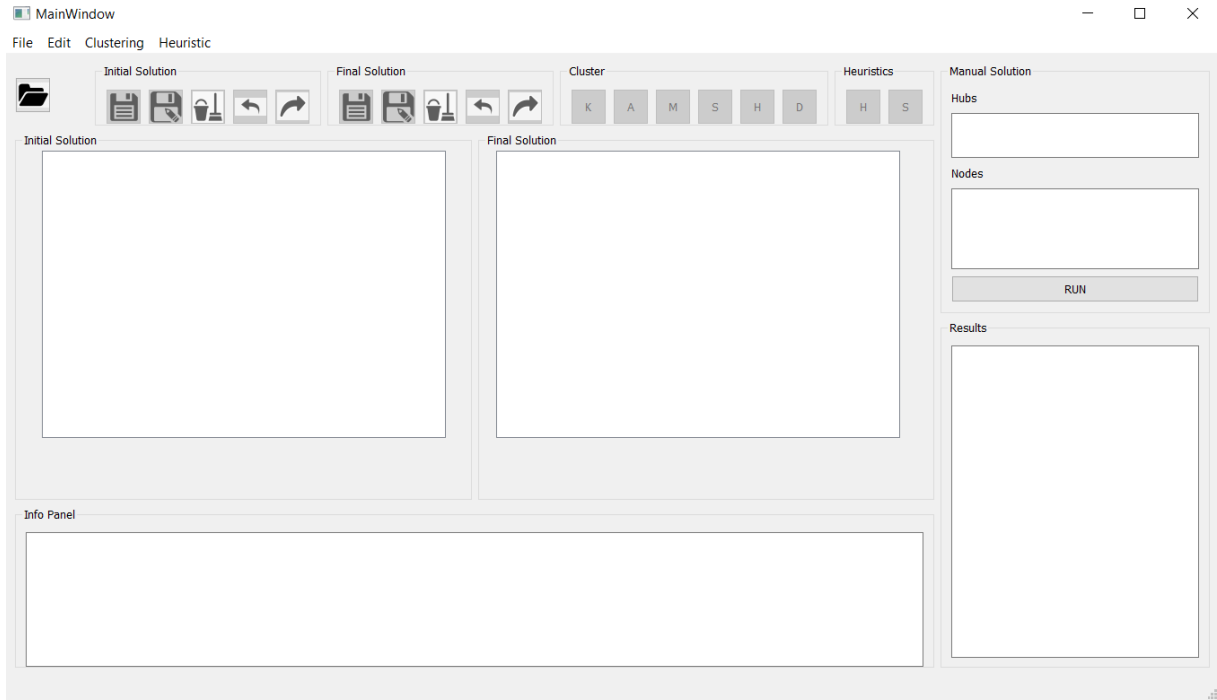


Figure 2 GUI of our software

Common_Proc Class:

Common_Proc class contains all the attributes required for our program and their set and get methods. Although our CClass is a direct parent class of classes with different functions (Clustering and Heuristics) in the program, its main purpose is to contain every method that is done jointly, regardless of the work done in the program. Since our Common_Proc Class has an indirect parent class to the Ui_MainWindow class, we can manipulate the signal and slot functions from the Common_Proc Class to ensure changes in the interface when any operation is made.

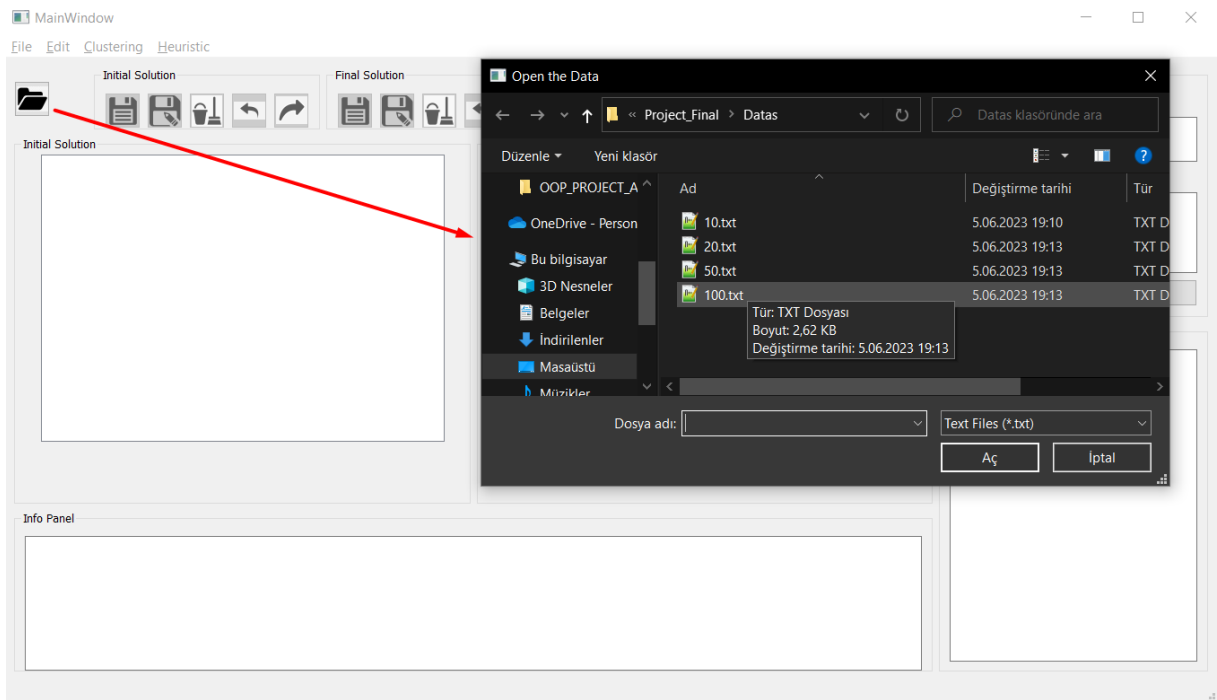


Figure 3 Opening data selector window

Clustering Class:

Our Clustering Class currently includes 6 clustering methods. In case of interacting with the relevant actions or buttons associated with these methods, it obtains all the data necessary for it from its parent class, Common_Proc class, with the get functions and makes the necessary manipulations on our data. After the necessary operations are completed, our data is updated with the set functions.

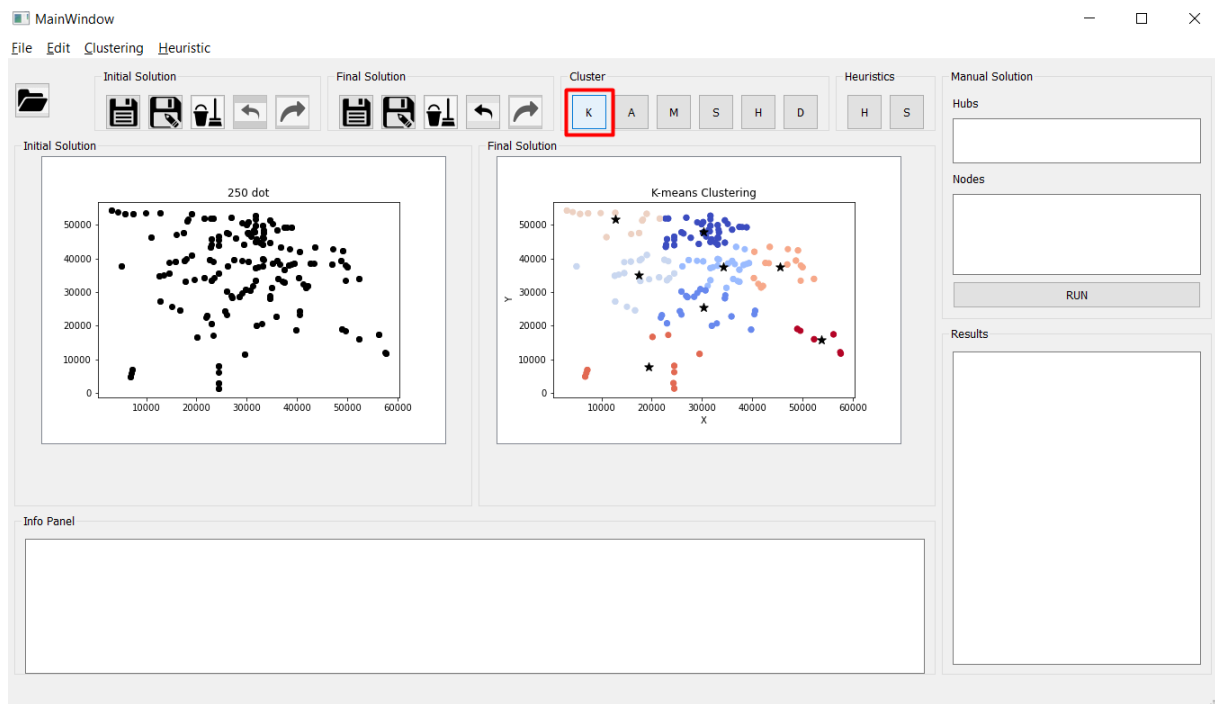


Figure 4 An example of K-means Clustering method

Ui_MainWindow Class:

Our Ui_MainWindow class contains the software of our interface. Clustering and Heuristics classes are the parent classes of our Ui_MainWindow class. Our Class is obtained by converting our interface from the .ui extension to .py extension after our interface is made with the Qt Designer program. By using the lambda: structure for the necessary signal and slot operations for our Class obtained in this way, with the structures on the screen (Buttons, Actions, etc.). When interacting, it is ensured what to do in the next step, and changes to our interface by interacting with our necessary Classes.