Exercise 7- Client-Server (Socket)

Create two separate Eclipse projects, KmeansClient and KmeansServer. Distribute the classes defined so far between the two projects.

CLIENT

he client system must connect to the server via the address and port on which the server is listening. Once the connection is established, the user can choose whether to start a new clustering process or retrieve clusters previously serialised in some file.

Include the MainTest class that establishes the connection to the Server and, once the connection has been made, sends and receives messages, depending on the choice made by the user. Through a menu, the client user selects the activity to be performed, discovery/reading clusters. If the choice is a discovery activity, the number of clusters to be discovered, the name of the database table, and the name of the file in which to serialise the discovered clusters are sent to the server. If the choice is a read activity, the name of the file in which the clusters to be recovered are serialised is sent to the Server. In both activities, the client either acquires the result transmitted by the server or displays it on the screen. Making use of the Keyboard class for keyboard input (PROVIDED BY THE TEACHER)

Define the exception class ServerException which is raised by the server system and passed on to the client from the connection stream. The exception is handled by the MainTest class

SERVER

The server collects classes for the execution of kmeans (cluster discovery, (de)serialisation).

- Defining the MultiServer class

Attributes

int PORT = 8080;

Methods

public static void main(String[] args): Instantiates an object of type MultiServer.

public MultiServer(int port): Constructor class. Initialises the port and invokes run()

void run() Instantiates an object instance of the ServerSocket class which waits for connection requests from the client. Each time a new connection request is made, ServerOneClient is instantiated.

Implement the run method so that it is able to acquire client requests (see server-side implementation of MainTest)

- Define the ServerOneClient class by extending the Thread class.

  Attributes

  ```
  Socket socket;
  ObjectInputStream in;
  ObjectOutputStream out;
  KmeansMiner kmeans;
  ```

  Methods

  public ServeOneClient**(Socket s) throws IOException**: Constructor class. Initialises the socket, in and out attributes. Starts the thread.

  public void run**()** Rewrites the run method of the Thread superclass in order to handle client requests.