Divide the program from laboratory work # 5 into the client and server modules. The server module must implement all collection management functions in an interactive mode, except for displaying text in accordance with the subject area. The client module should query the server for the current state of the collection, generate a plot, output it to the console, and exit.

Objects that are stored in a collection must have the following characteristics:

* name, name or similar text identifier;
* size or a similar numeric parameter;
* the characteristic determining the location of the object on the plane / in space;
* time / date of birth / creation of the object.
* If you already have similar characteristics, you do not need to add them.

The following requirements must be fulfilled:

* Collection from LR No. 5 is replaced by its thread-safe analog.
* The operations of processing collection objects must be implemented using the Stream API using lambda expressions.
* Objects between the client and the server must be transmitted in serialized form.
* Objects in the collection sent to the client must be sorted by default.
* Upon receiving the request, the server must create a separate thread, which must generate and send a response to the client.
* The client must correctly process the temporary unavailability of the server.
* The exchange of data between the client and the server must be done via the UDP protocol.
* On the server side, datagrams should be used and on the client side, the network channel.

The report on the work must contain:

* The text of the task.
* The diagram of classes of the developed program (both client, and server application).
* Source code of the program.
* Conclusions on the work.

Questions for the protection of laboratory work:

* Networking - client-server architecture, basic protocols, their similarities and differences.
* The TCP protocol. Classes Socket and ServerSocket.
* The UDP protocol. Classes DatagramSocket and DatagramPacket.
* Data transmission over the network. Serialization of objects.
* Serializable interface. Object graph, serialization and deserialization of fields and methods.
* Multithreaded programs. Concepts. The Thread class and the Runnable interface.
* State of the flow. Synchronization of the flow.
* The java.util.concurrent package. Interface Lock and its implementation.
* Atomic operations.
* Java Stream APi. Creating pipelines. Intermediate and terminal operations.