**SUPER SECRET JAVA SERVER**

REFERENCE SHEET

**Class Client**

Client(InetAddress address, int port, String client, String local\_folder\_path) throws Exception

Parameters:

address - address

port - port number

client - client name, which with he is associated on server

local\_folder\_path - path to client local folder with files that is synchronized with server

public void run()

Starts new thread that control synchronization with server.

public static void main(String[] args) throws IOException

Asks for client name and local folder path, then runs new client thread.

**Class ControlClient**

public ControlClient(Socket socket, String local\_folder\_path, String client) throws IOException

ClientControl constructor that takes socket, his local folder path and his name.

Parameters:

socket - socket took from Client class

local\_folder\_path - path to client local folder with files that is synchronized with server

client - client name, which with he is associated on server

public static MyMap get\_file\_list(String path, String client)

Gets files list from client local folder.

Parameters:

path - path to folder from which files are taken.

client - client name

Returns:

MyMap with key being client name and value being list of files

public static void send\_files\_map(MyMap map, OutputStream stream) Exception

Sends client files to server.

Parameters:

map - map of client files, with key being client name and value being list of client files

stream - stream in which map needs to be send

public List<String> get\_server\_files\_list() throws Exception

Returns list of files, that needs to be saved in client folder, from server.

public void get\_files(List< String> dontHave)

For every file that client don't have it starts a new thread to download it.

Parameters:

dontHave - files that client don't have, and needs to download them

public void run()

Synchronizes files between client and server.

**Class Discs**

public Discs(String disc, String client, String file\_path, PrintWriter pw)

Discs constructor that takes disc, client, filepath, and PrintWriter as parameters.

Parameters:

disc - disc on which given operations will be made

client - client name for who given operations will be made

file\_path - path to file which is needed to be saved on server

pw - stream which is used to communicate with client, to send him messages

public static void delay(int sleep) throws Exception

Delays a thread on given seconds.

Parameters:

sleep - number on seconds of which thread will be sleeping

public void run()

Saves files on given disc and updates .csv file

public String get\_file\_name(String file\_path)

Gets file name from full path

Parameters:

file\_path - file path from which file name is needed to be taken

Returns:

just a file name

public void copy\_file(String to, String from)

Copies file given file in given path.

Parameters:

to - path to folder in which the file is needed to be saved

from - file that needs to be saved

public void update\_list(String file, String client, String disc) throws IOException

Updates .csv list which contains lists of client and it's files saved on given disc

Parameters:

file - file that needs to be added

client - client to whom file is needed to be added

disc - disc on which .csv file is needed to be updated

public static MyMap get\_csv(String disc) throws IOException

Reeds clients and their files from .csv file.

Parameters:

disc - disc from which files and clients are needed to be taken

Returns:

MyMap whit all clients and their files that have them saved on given disc

**Class Multi\_Client**

public Multi\_Client(ServerSocket server, Socket socket) throws IOException

Multi\_Client constructor that takes ServerSocket and Socket as arguments.

Parameters:

server - ServerSocket took form Server class

socket - Socket took form Server class

public void init() throws IOException

Initializes PrintWriter, Scanner and String

public String choose\_disc() throws IOException

Chooses disc with the least amount of space taken

public MyMap get\_client\_files\_map() throws Exception

Gets MyMap from client with his files list.

public List< String> uncommon\_files(List< String> want, List<String> dont)

Returns a list of uncommon files between two lists.

Parameters:

want - list from which you delets files to leave files hat you want

dont - list of files that you dont want

public static void send\_files\_list(List<String> list, OutputStream stream) throws Exception

Sends list of files that client dont have to him

Parameters:

list - list that will be send to client

stream - stream used to send list

public static List<String> get\_clients\_list(String client) throws Exception

Gets lists of clients from all discs

Parameters:

client - client that will be removed from list

public void synchro\_client(MyMap[] filelist, MyMap client\_files) throws Exception

Synchronizes saving files on client side.

Parameters:

filelist - list of files from all discs

client\_files - list of client files

public void synchro\_server(MyMap[] filelist, MyMap client\_files, String client\_path) throws Exception

Synchronizes saving files on server side.

Parameters:

filelist - list of files from all discs

client\_files - list of client files

client\_path - path to client folder

public void synchro(InputStream stream, String client\_path) throws Exception

Synchronizes all sending and receiving files between client and server

Parameters:

stream - stream from which client files will be taken

client\_path - path to client folder

public void run()

If client and server are ready it starts synchronization, and control status of it.

**Class MyMap**

It takes string as key, and list of strings as values.

**Class SaveFiles**

public SaveFiles(String what, String where, PrintWriter pw)

SaveFiles constructor that takes files, folder path, and PrintWriter as parameters.

Parameters:

what - file that needs o be saved

where - path to folder in where the file is needed to be saved

pw - PrintWriter to communicate with server

public void run()

Saves given file in given path.

**Class Server**

public Server(int port) throws IOException

Server construcor that tahes port as an argument.

Parameters:

port - number of port

public void run()

For every new client it starts a thread to handle synchronization.

**Class Share**

public Share(String file, String from\_client, String to\_client)

Share constructor that takes file, and two clients ad parameters.

Parameters:

file - file that is shared with other client

from\_client - client who shares a file

to\_client - client to who file is shared

public void run()

Starts sharing process.

public void find\_file(String file, String from\_client, String to\_client) throws Exception

Finds given file in discs and saves it to given client.

Parameters:

file - file that is shared with other client

from\_client - client who shares a file

to\_client - client to who file is shared

**Class ClientFx**

public List< String> get\_files()

Gets client file list as MyMap and returns lists of client files.

public void refresh()

Refreshes client files list and list of other files every 10 seconds.

public void status()

Refreshes status of synchronization every 1 second.

public void Client\_Panel(Stage stage) throws Exception

View for actual client panel.

public void start(Stage stage)

View for Log in panel, that changed to client panel if needed.

**Class ServerFx**

public List<String> get\_files(int disc)

Gets files list form given disc.

Parameters:

disc - disc from which files list needs to be read

public void setLists(int i)

Sets list for given disc

public void refresh()

Refreshes files lists for all discs every 5 seconds.

public void status()

Refreshes status of synchronization every 1 second.

public void start(Stage stage)

View for server panel.