

Assignment 1

Due: 24.02.06 – 17:59:59

Part 1) Reflection and serialization

Your task is to write an application that will allow a user to create/destroy/manipulate instances of classes that are unknown at compile time. Your program should offer an interface (GUI) that will enable a user to perform the following operations:

Create new instance

The user should be able to create an instance of a class by specifying the name of the class and the location where the application can find the class file. In case of an unsuccessful creation the users should be informed by displaying an error message. In case of successful creation the user should be informed by displaying a “successful creation” message AND a logical name (unique string) that has been assigned to the new instance. Assume that every class has one no argument constructor.

Destroy existing instance

The user should be able to destroy a created instance by using a logical name of an instance. In case of successful destruction the user should be informed with a “successful removal” message – in case of failure an error message should be displayed.

Display instance

The user should be able to review a created instance. The system should display the fields (name, type and content) and methods of the instance.

Set Fields

The user should be able to manipulate the fields of the instance that are of a primitive data type (built-in types e.g. int, boolean etc.)

Invoke method

The user should be able to invoke a method of an existing instance. To simplify this task assume that only methods with return type string and one string argument are invoked.

Save Instance

Allow the user to save an instance to file. The user should be allowed to select the filename.

Load Instance

Allow the user to load an instance if she/he names the file and provides the system with the name of the class and the location of the class-file.

Part 2) Sockets & Server

Transform the application of part 1 into a client server application that will allow up to 30 users to remotely perform the operations mentioned in part 1. To do so you will have to develop a communication protocol and its implementation. The clients will be thin clients that will connect to the server. The users will be able to activate to perform the above-mentioned 7 operations (create, destroy, display, set, invoke, load, save) by using the GUI of the client. Rather than executing the commands the client will send the commands via a (TCP/IP) socket-connection to the server, which will perform them. This means that all 7 operations will be executed only in the server. The server will send the results of the operations to the client. The client will be used only as a remote GUI for accessing the server.

To keep the program flexible you can expect that the user will inform the client of the IP and Port of the server.

Part 3) Using multiple Servers

It is often advisable to enable the distribution of client load by using more than one server. This means that you have to modify your server and client code from part 2 to handle more than one server.

When starting a server it should be informed if there are already other servers running. In case that there are other servers it should contact them and inform them about its presence. The new server should try to offload the existing servers. New client connections to any server should be redirected to the server with the lowest number of connections.

Please note that new clients could try to access an object created on an “older” server.

Part 4) Handling Server shutdown/reboots

To add more flexibility to the multiple server scenario ensure that a server can be shut-down. If a shutdown is initiated, the data of the server should be moved to another machine. In addition to not allowing new connections, current clients should be redirected to another machine.

Handin (via E-Handin):

Documented source code

All Binaries

Batch files to compile and run the different parts of the assignment

A detailed description of how you tested your code

Marking: Functionality = 80 %, Test Report = 10 %, Documented Source Code = 10 %