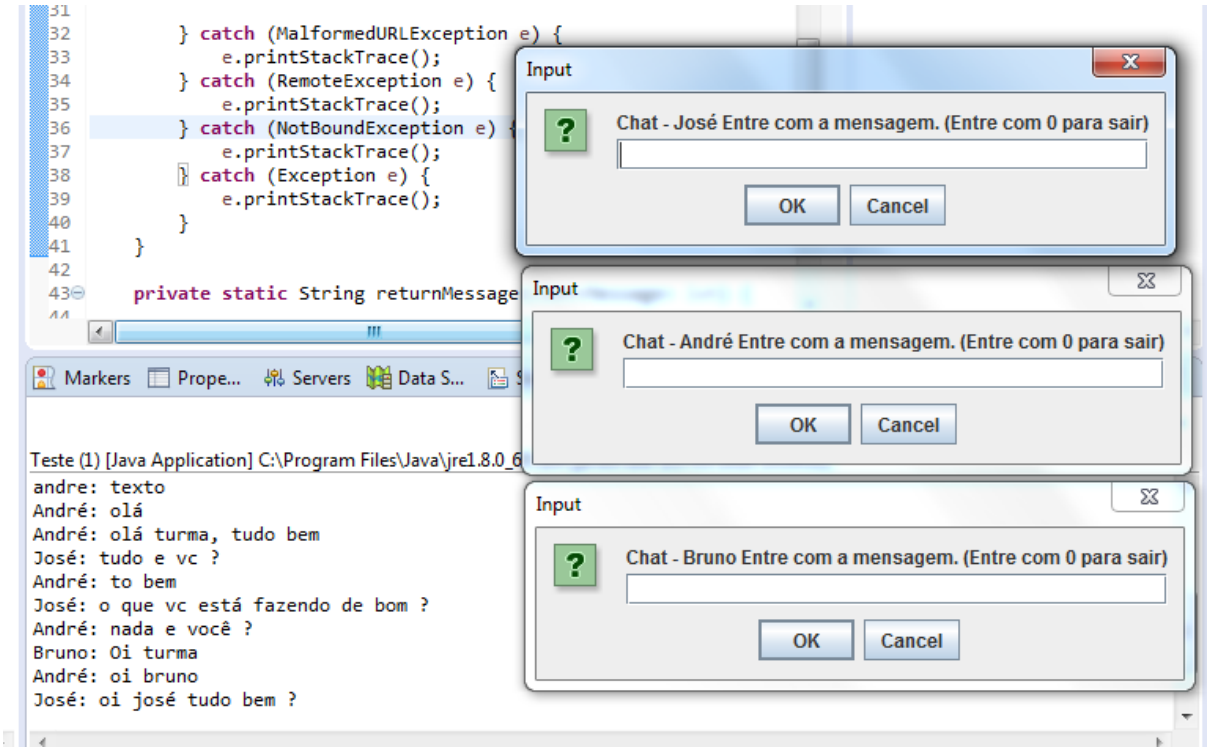
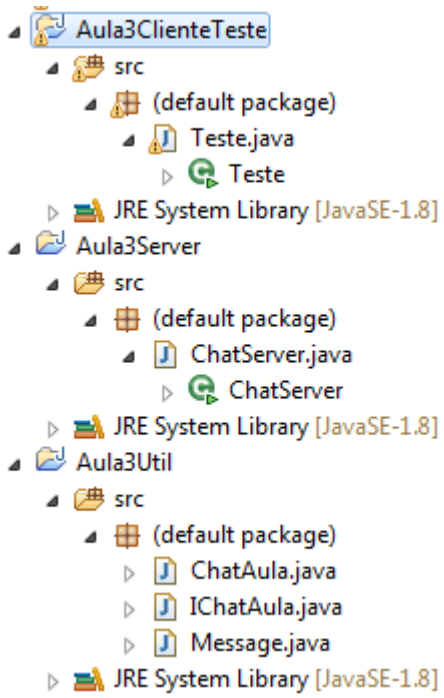
Chat – RMI



Estrutura do projeto:



Client:

```
 8 import javax.swing.JOptionPane;
 9
10 public class Teste {
11
12     public static void main(String[] args) {
13
14         String nome = "";
15         String msgp = "";
16
17         nome = JOptionPane
18                 .showInputDialog("Bem vindo, ao Chat, Qual é o seu nome? ");
19
20         try {
21
22             while (msgp != "0") {
23                 msgp = JOptionPane.showInputDialog("Chat - " + nome
24                         + " Entre com a mensagem. (Entre com 0 para sair)");
25                 IChatAula objChat = (IChatAula) Naming
26                         .lookup("rmi://localhost:8282/chat");
27                 Message msg = new Message(nome, msgp);
28                 objChat.sendMessage(msg);
29                 System.out.println(returnMessage(objChat.retrieveMessage()));
30             }
31
32         } catch (MalformedURLException e) {
33             e.printStackTrace();
34         } catch (RemoteException e) {
35             e.printStackTrace();
36         } catch (NotBoundException e) {
37             e.printStackTrace();
38         } catch (Exception e) {
39             e.printStackTrace();
40         }
41     }
42
43     private static String returnMessage(List<Message> lst) {
44
45         String valor = "";
46         for (Message message : lst) {
47             valor += message.getUsuario() + ": " + message.getMensagem() + "\n";
48         }
49         return valor;
50     }
51
52 }
```

Server:

```
1  import java.rmi.Naming;
4
5
6  public class ChatServer {
7
8      public ChatServer() throws RemoteException {
9          try {
10             LocateRegistry.createRegistry(8282);
11             Naming.rebind("rmi://localhost:8282/chat", new ChatAula());
12             System.out.println("Rodando...");
13         } catch (Exception e) {
14              e.printStackTrace();
15         }
16     }
17     public static void main(String[] args) throws RemoteException {
18        new ChatServer();
19     }
20
21 }
22
```

IChatAula:

```
1  import java.rmi.Remote;
2  import java.rmi.RemoteException;
3  import java.util.List;
4
5
6  public interface IChatAula extends Remote{
7
8          void sendMessage(Message msg) throws RemoteException;
9          List<Message> retrieveMessage() throws RemoteException;
10 }
11
```

ChatAula:

```java
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
import java.util.List;

public class ChatAula extends UnicastRemoteObject implements IChatAula {

    public ChatAula() throws RemoteException {
        super();
    }

    private static final long serialVersionUID = 4998385221633210l4L;

    @Override
    public void sendMessage(Message msg) throws RemoteException {
        Message.setLstMessage(msg);
    }

    @Override
    public List<Message> retrieveMessage() throws RemoteException {
        return Message.getLstMessage();
    }
}

```

Message:

```java
import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;

public class Message implements Serializable{

    private static final long serialVersionUID = -2723630512719669 64L;

    private String user;
    private String message;

    private static List<Message> lstMessage = new ArrayList<Message>();

    public Message(String user, String message){
        this.user = user;
        this.message = message;
    }

    public String getUsuario() {
        return user;
    }
    public void setUsuario(String user) {
        this.user = user;
    }
    public String getMensagem() {
        return message;
    }
    public void setMensagem(String message) {
        this.message = message;
    }

    public static List<Message> getLstMessage() {
        return lstMessage;
    }

    public static void setLstMessage(Message msg) {
        lstMessage.add(msg);
    }
}
```

Atividade:

1) Implementar o código acima.
2) Criar uma interface para o chat utilizando Swing/AWT (é permitido a utilização do NetBeans).
3) Armazenar as conversas em uma base de dados (MySQL) e também em um arquivo *.txt, salvar em um diretório qualquer.
4) Implementar uma validação para quando o usuário sair do Chat.
5) Implementar uma validação para quando o usuário entrar no Chat.