Semestrálne zadanie

authors: Sebastian Nikolas Ondirko,

Stanislav Karabinoš

**package klient;**

// import kniznic potrebnych k spravnej funkionalite programu na komunikaciu medzi klientom a serverom

import java.io.BufferedReader;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.InputStreamReader;

import java.net.Socket;

/\*\*

\* This class implements java socket client

\* @author Ondirko, Karabinos

\* @version 1.3

\*

\*/

// trieda Klient, pomocou ktorej komunikujeme so serverom

public class Klient {

// Hlavna funkcia main Klienta

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

{

// Zadefinovanie socketu s nazvom -s- a prepojenie na lokalneho hosta s portom 3333

Socket s = new Socket("localhost",3333);

// Zadefinovanie dis = datovy vstupny tok

DataInputStream dis = new DataInputStream(s.getInputStream());

// Zadefinovanie dos = datovy vystupny tok

DataOutputStream dos = new DataOutputStream(s.getOutputStream());

// Zadefinovanie isr = citaka vstupneho toku

InputStreamReader isr = new InputStreamReader(System.in);

// Zadefinovanie stdin = bufer

BufferedReader stdin = new BufferedReader(isr);

// Zadefinovanie prazdnych retazcov s1 a s2

String s1="",s2="";

// Pozdrav a oboznamenie s programom

System.out.println("Dobrý deň, ukončenie aplikacie je zadanim retazca -stop- \n");

// Po zapnuti servera komunikacia začina požiadavkou klienta, a preto vyzyvame klienta k zadani Id

System.out.println("Zadaj Id: ");

// Nekonečny cyklus komunikacie klienta a servera ktory konči zadanim retazca -stop-

while(!s1.equals("stop"))

{

// Nacitanie retazca z konzoli do retazca s2

s2=stdin.readLine();

// Poslanie serveru hodnotu retazca s2

dos.writeUTF(s2);

// Poslanie

dos.flush();

// Nacitanie spravy od servera a nacitanie do premennej s1

s1=dis.readUTF();

// vypisanie spravi od servera na konzolu

System.out.println("Server: "+s1);

}

// Zatvorenie datoveho vystupneho toku

dos.close();

// Zatvorenie socketu s

s.close();

}

}

**package server;**

// import kniznic potrebnych k spravnej funkionalite programu na komunikaciu medzi serverom a klientom

import java.io.BufferedReader;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.InputStreamReader;

import java.net.ServerSocket;

import java.net.Socket;

// import kniznic potrebnych k spravnej funkionalite programu na komunikaciu servera a databazy

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.sql.Statement;

/\*\*

\*

\* @author Ondirko, Karabinos

\* @version 1.20

\*

\*/

// trieda Server, pomocou ktorej komunikujeme so s klientom a databazou

public class Server

{

// Hlavna funkcia main Servera

public static void main(String args[]) throws Exception, SQLException

{

Server myWrapper = new Server() ;

// Zadefinovanie premennej x

int x=0;

// Zadefinovanie socketu s nazvom -s- a prepojenie s portom 3333

ServerSocket serverSocket = new ServerSocket(3333);

Socket socket = serverSocket.accept();

// Zadefinovanie dis = datovy vstupny tok

DataInputStream dataInputStream = new DataInputStream(socket.getInputStream());

// Zadefinovanie dos = datovy vystupny tok

DataOutputStream dataOutputStream = new DataOutputStream(socket.getOutputStream());

// Zadefinovanie isr = citaka vstupneho toku

InputStreamReader inputStreamReader = new InputStreamReader(System.in);

// Zadefinovanie stdin = bufer

BufferedReader buffer = new BufferedReader(inputStreamReader);

// Zadefinovanie retazcov

String socket1="",socket2="",socket3="",rola="",student="student",referent="referent",maria="maria",robert="robert",vypis="vypis",helpID="",helpName="";

// Nekonečny cyklus komunikacie medzi serverom, klientom a databazou ktory konči zadanim retazca -stop- od klienta

while(!socket1.equals("stop"))

{

// Klient zacina komunikaciu, nacitanie vstupnych dat od klienta do retazca socket1

socket1=dataInputStream.readUTF();

// Prva podmienka ktora preveri ci je zadane Id v databaze, a zistenie ci je to referent alebo student

if(x == 0)

{

// Odskusanie pripojenia na databazu, zadanie prihlasovacieho mena a hesla a nasledna praca s databazou ak v tabulke CLIENTS sa ID = socketu1

try (Connection connect = DriverManager.getConnection("jdbc:derby://localhost:1527/ServerDatabase", "test", "test");

PreparedStatement preparedStatement = connect.prepareStatement("SELECT TYPE\_OF\_CLIENT FROM CLIENTS WHERE ID=?");)

{

// Dosadenie hodnoty socket1 za ?

preparedStatement.setString(1, socket1);

// Ak plati prvy try tak tu uz pracujeme v danej tabulke a vyberame data

try(ResultSet result = preparedStatement.executeQuery())

{

// Dalsia hodnota

result.next();

// Vytiahnutie TYPE\_OF\_CLIENT do rola

rola = result.getString("TYPE\_OF\_CLIENT");

// Naplnenie retazca helpID zo socketu1

helpID = socket1;

// Nastavenie premennej x na 1

x=1;

}

// Zachytenie vynimky

}catch (SQLException ex)

{

// Vypisanie do konzoly chyby komunikacie s databazou

System.out.println("Cannot communicate with database");

}

}

// Vypisanie poziadavky klienta zo socketu1 na konzolu

System.out.println("Client: "+socket1);

// Ak bolo Id zadane spravne tak uz vieme ze pracujeme so studentom

if((x == 1)&&(rola.equals(student)))

{

// Odskusanie pripojenia na databazu, zadanie prihlasovacieho mena a hesla a nasledna praca s databazou ak v tabulke CLIENTS sa ID = socketu1

try (Connection connect = DriverManager.getConnection("jdbc:derby://localhost:1527/ServerDatabase", "test", "test");

PreparedStatement preparedStatement = connect.prepareStatement("SELECT NAME FROM CLIENTS WHERE ID=?");)

{

// Dosadenie hodnoty socket1 za ?

preparedStatement.setString(1, socket1);

// Ak plati prvy try tak tu uz pracujeme v danej tabulke a vyberame data

try(ResultSet result = preparedStatement.executeQuery())

{

// Dalsia hodnota

result.next();

// Vytiahnutie NAME do name

String name = result.getString("NAME");

// Nastavenie premennej x na 2

x=2;

}

// Zachytenie vynimky

}catch (SQLException ex)

{

// Vypisanie do konzoly chyby komunikacie s databazou

System.out.println("Cannot communicate with database");

}

}

if((x == 2)&&(rola.equals(student)))

{

// Odskusanie pripojenia na databazu, zadanie prihlasovacieho mena a hesla a nasledna praca s databazou ak v tabulke CLIENTS sa ID = socketu1

try (Connection connect = DriverManager.getConnection("jdbc:derby://localhost:1527/ServerDatabase", "test", "test");

PreparedStatement preparedStatement = connect.prepareStatement("SELECT NAME,ID FROM CLIENTS WHERE PASSWORD=?");)

{

// Dosadenie hodnoty socket1 za ?

preparedStatement.setString(1, socket1);

// Ak plati prvy try tak tu uz pracujeme v danej tabulke a vyberame data

try(ResultSet result = preparedStatement.executeQuery())

{

// Dalsia hodnota

result.next();

// Vytiahnutie NAME do name

String name = result.getString("NAME");

// Nastavenie premennej x na 60

x=60;

// Zadanie retazca do socketu2

socket2="Ak chcete sa zapisat zadajte 1, Ak chcete vypisat kde vase schodzky zadajte 2";

// Poslanie klientovi hodnotu retazca socket2

dataOutputStream.writeUTF(socket2);

// Poslanie

dataOutputStream.flush();

// Nacitanie vstupnych dat od klienta do retazca socket1

socket1=dataInputStream.readUTF();

// Ak socket1 = 1

if(socket1.equals("1"))

{

// Zadanie retazca do socketu2

socket2="Ak chcete sa zapisat pri marii zadajte 7 , Ak chcete sa zapisat pri robertovi zadajte 8";

// Poslanie klientovi hodnotu retazca socket2

dataOutputStream.writeUTF(socket2);

// Poslanie

dataOutputStream.flush();

// Nacitanie vstupnych dat od klienta do retazca socket1

socket1=dataInputStream.readUTF();

// Ak socket1 = 7

if(socket1.equals("7"))

{ // Nastavenie premennej x na 7

x=7;

}

// Ak socket1 = 8

if(socket1.equals("8"))

{ // Nastavenie premennej x na 8

x=8;

}

}

if(socket1.equals("2"))

{ // Nastavenie premennej x na 4

x=4;

}

// Zachytenie vynimky

}catch (SQLException ex)

{

// Vypisanie do konzoly chyby komunikacie s databazou

System.out.println("Cannot communicate with database");

}

}

}

if(x == 7)

{

// Odskusanie pripojenia na databazu, zadanie prihlasovacieho mena a hesla a nasledna praca s databazou ak v tabulke CLIENTS sa ID = socketu1

try (Connection connect = DriverManager.getConnection("jdbc:derby://localhost:1527/ServerDatabase", "test", "test");

PreparedStatement preparedStatement = connect.prepareStatement("SELECT NAME,SURNAME,ID FROM CLIENTS WHERE ID=?");)

{

// Dosadenie hodnoty helpID za ?

preparedStatement.setString(1, helpID);

try(ResultSet result = preparedStatement.executeQuery())

{

// Dalsia hodnota

result.next();

// Vytiahnutie NAME do name

String name = result.getString("NAME");

// Vytiahnutie SURNAME do surname

String surname = result.getString("SURNAME");

String id = result.getString("ID");

Statement statement = connect.createStatement();

statement.executeUpdate("insert into TEST.MARIA(name,surname,id) values('"+name+"','"+surname+"','"+id+"')");

statement.executeUpdate("UPDATE CLIENTS SET EXAM\_Maria='Zapisany u Marii' WHERE ID='"+id+"' ");

System.out.println("Data was inserted ");

x=0;

}

// Zachytenie vynimky

}catch (SQLException ex)

{

// Vypisanie do konzoly chyby komunikacie s databazou

System.out.println("Cannot communicate with database");

}

}

if(x == 8)

{

// Odskusanie pripojenia na databazu, zadanie prihlasovacieho mena a hesla a nasledna praca s databazou ak v tabulke CLIENTS sa ID = socketu1

try (Connection connect = DriverManager.getConnection("jdbc:derby://localhost:1527/ServerDatabase", "test", "test");

PreparedStatement preparedStatement = connect.prepareStatement("SELECT NAME,SURNAME,ID FROM CLIENTS WHERE ID=?");)

{

preparedStatement.setString(1, helpID);

try(ResultSet result = preparedStatement.executeQuery())

{

result.next();

String name = result.getString("NAME");

String surname = result.getString("SURNAME");

String id = result.getString("ID");

Statement statement = connect.createStatement();

statement.executeUpdate("insert into TEST.ROBERT(name,surname,id) values('"+name+"','"+surname+"','"+id+"')");

statement.executeUpdate("UPDATE CLIENTS SET EXAM\_ROBERT='Zapisany u Roba' WHERE ID='"+id+"' ");

System.out.println("Data was inserted ");

x=0;

}

// Zachytenie vynimky

}catch (SQLException ex)

{

// Vypisanie do konzoly chyby komunikacie s databazou

System.out.println("Cannot communicate with database");

}

}

if((x == 1)&&(rola.equals(referent)))

{

// Odskusanie pripojenia na databazu, zadanie prihlasovacieho mena a hesla a nasledna praca s databazou ak v tabulke CLIENTS sa ID = socketu1

try (Connection connect = DriverManager.getConnection("jdbc:derby://localhost:1527/ServerDatabase", "test", "test");

PreparedStatement preparedStatement = connect.prepareStatement("SELECT NAME FROM CLIENTS WHERE ID=?");)

{

preparedStatement.setString(1, socket1);

try(ResultSet result = preparedStatement.executeQuery())

{

result.next();

String name = result.getString("NAME");

helpName=name;

x=2;

}

// Zachytenie vynimky

}catch (SQLException ex)

{

// Vypisanie do konzoly chyby komunikacie s databazou

System.out.println("Cannot communicate with database");

}

}

if((x == 2)&&(rola.equals(referent)))

{

// Odskusanie pripojenia na databazu, zadanie prihlasovacieho mena a hesla a nasledna praca s databazou ak v tabulke CLIENTS sa ID = socketu1

try (Connection connect = DriverManager.getConnection("jdbc:derby://localhost:1527/ServerDatabase", "test", "test");

PreparedStatement preparedStatement = connect.prepareStatement("SELECT NAME,ID FROM CLIENTS WHERE PASSWORD=?");)

{

preparedStatement.setString(1, socket1);

try(ResultSet result = preparedStatement.executeQuery())

{

result.next();

String name = result.getString("NAME");

x=60;

socket2="Ak chcete sa vypisat prihlasenych studentov zadajte 1, Ak chcete posunut 2";

dataOutputStream.writeUTF(socket2);

dataOutputStream.flush();

socket1=dataInputStream.readUTF();

if(socket1.equals("1"))

{ x=10;

}

if(socket1.equals("2"))

{ x=20;

}

}

// Zachytenie vynimky

}catch (SQLException ex)

{

// Vypisanie do konzoly chyby komunikacie s databazou

System.out.println("Cannot communicate with database");

}

}

if((x == 10)&&(rola.equals(referent)))

{

// Odskusanie pripojenia na databazu, zadanie prihlasovacieho mena a hesla a nasledna praca s databazou ak v tabulke CLIENTS sa ID = socketu1

try (Connection connect = DriverManager.getConnection("jdbc:derby://localhost:1527/ServerDatabase", "test", "test");

PreparedStatement preparedStatement = connect.prepareStatement("SELECT NAME,SURNAME,ID FROM FROM '"+helpName+"' ");)

{

preparedStatement.setString(1, helpName);

try(ResultSet result = preparedStatement.executeQuery())

{

result.next();

String name = result.getString("NAME");

socket2=name;

dataOutputStream.writeUTF(socket2);

String surname = result.getString("SURNAME");

socket2=surname;

dataOutputStream.writeUTF(socket2);

String id = result.getString("ID");

socket2=id;

dataOutputStream.writeUTF(socket2);

x=5;

}

// Zachytenie vynimky

}catch (SQLException ex)

{

// Vypisanie do konzoly chyby komunikacie s databazou

System.out.println("Cannot communicate with database");

}

}

if((x == 4)&&(rola.equals(student)))

{

// Odskusanie pripojenia na databazu, zadanie prihlasovacieho mena a hesla a nasledna praca s databazou ak v tabulke CLIENTS sa ID = socketu1

try (Connection connect = DriverManager.getConnection("jdbc:derby://localhost:1527/ServerDatabase", "test", "test");

PreparedStatement preparedStatement = connect.prepareStatement("SELECT EXAM\_MARIA, EXAM\_ROBERT FROM CLIENTS WHERE ID=?");)

{

preparedStatement.setString(1, helpID);

try(ResultSet result = preparedStatement.executeQuery())

{

result.next();

String examMaria = result.getString("EXAM\_MARIA");

String examRobert = result.getString("EXAM\_ROBERT");

socket2=examMaria;

dataOutputStream.writeUTF(socket2);

socket2=examRobert;

dataOutputStream.writeUTF(socket2);

x=5;

}

// Zachytenie vynimky

}catch (SQLException ex)

{

// Vypisanie do konzoly chyby komunikacie s databazou

System.out.println("Cannot communicate with database");

}

}

if((x == 20)&&(rola.equals(referent)))

{

// Odskusanie pripojenia na databazu, zadanie prihlasovacieho mena a hesla a nasledna praca s databazou ak v tabulke CLIENTS sa ID = socketu1

try (Connection connect = DriverManager.getConnection("jdbc:derby://localhost:1527/ServerDatabase", "test", "test");

PreparedStatement preparedStatement = connect.prepareStatement("ERASE NAME,SURNAME,ID FROM '"+helpName+"' ");)

{

preparedStatement.setString(1, helpName);

try(ResultSet result = preparedStatement.executeQuery())

{

Statement statement = connect.createStatement();

statement.executeUpdate("delete from TEST.ROBERT(name,surname,id) values()");

result.next();

String name = result.getString("NAME");

socket2=name;

dataOutputStream.writeUTF(socket2);

String surname = result.getString("SURNAME");

socket2=surname;

dataOutputStream.writeUTF(socket2);

String id = result.getString("ID");

socket2=id;

dataOutputStream.writeUTF(socket2);

x=5;

}

// Zachytenie vynimky

}catch (SQLException ex)

{

// Vypisanie do konzoly chyby komunikacie s databazou

System.out.println("Cannot communicate with database");

}

}

// Hodnota x je nastavena na 0, server vyzve klienta na zadanie Id

if(x == 0)

{

// Nastavenie retazca s2 na "Enter Id"

socket2="Enter ID";

}

if(x == 2)

{

// Nastavenie retazca s2 na "Enter password"

socket2="Enter password";

}

if(x == 60)

{

// Nastavenie retazca s2 na "Nechcel vypisat"

socket2="Nechcel vypisat";

}

if(x == 5)

{

// Nastavenie retazca s2 na "Vypisane"

socket2="Vypisane";

// Nastavenie premennej x na 0

x=0;

}

// Poslanie klientovi hodnotu retazca socket2

dataOutputStream.writeUTF(socket2);

// Poslanie

dataOutputStream.flush();

}

// Zatvorenie datoveho vstupneho toku

dataInputStream.close();

// Zatvorenie socketu

socket.close();

// Zatvorenie serveroveho socketu

serverSocket.close();

}

}

**Dodatok**

Funkcia „if((x == 20)&&(rola.equals(referent)))“ ,ktorá umožňuje referentovi si pozrieť počet prihlasených použivateľov k nemu na „skúšku“ nefunguje úplne správne. Sýstem nám nevyhodzuje žiadnú chybu v konzole a aj napriek tomu sa daný príkaz opakovane zacyklí. Na to nadväzujúca funkcia odoslania správ čakajúcim študentom tým pádom nefunguje resp. nemali sme možnosť overiť jej funkčnosť.