**Занятие № 22**

**Номер учебной группы:** П-16

**Фамилия, инициалы учащегося:** Хомич В.И.

**Дата выполнения работы:** 3.12.2022

**Тема работы:** «Кодирование дополнительных модулей программной системы»

**Цель работы:** научиться выполнять кодирование дополнительных модулей программной системы.

**Ход работы**

**Задание 1**

Применяя язык программирования или визуальную среду провести кодирование дополнительных модулей программной системы.

Применяя язык программирования, провел кодирование дополнительного модуля, отвечающего за соединение базы данных с программным продуктом.

**1)Код модуля:**

namespace ProgramMed

{

class DB

{

MySqlConnection connection = new MySqlConnection("server = localhost; port = 3306; username = root; password = root; database = basec# ");

public void openConnection()

{

if (connection.State == System.Data.ConnectionState.Closed)

connection.Open();

}

public void closeConnection()

{

if (connection.State == System.Data.ConnectionState.Open)

connection.Close();

}

public MySqlConnection getConnection()

{

return connection;

}

}

}

Также было произведено кодирование форм регистрации и авторизации в программном продукте.

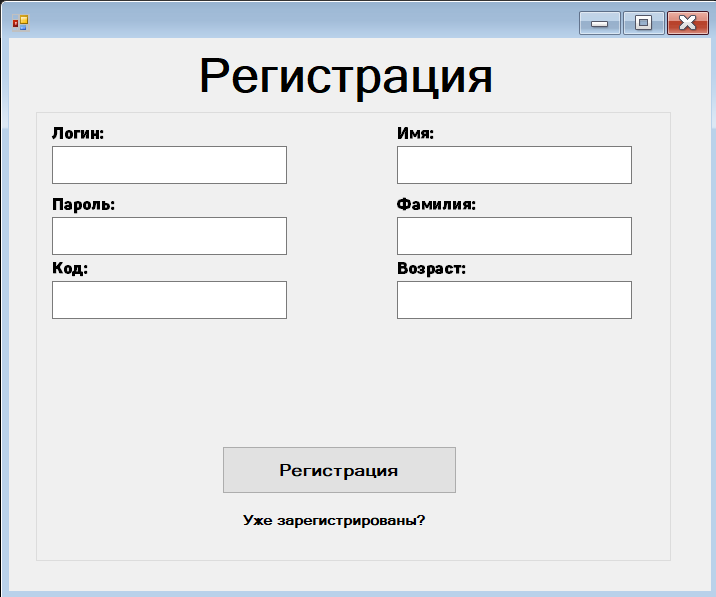


Рисунок 1 «Форма регистрации»

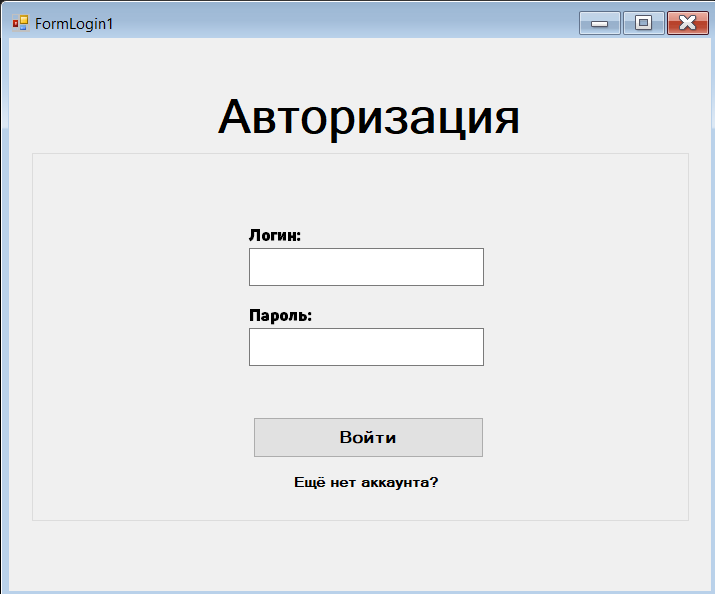


Рисунок 2 «Форма авторизации»

**2) Код формы регистрации:**

namespace ProgramMed

{

public partial class FormReg1 : Form

{

public FormReg1()

{

InitializeComponent();

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

System.Windows.Forms.Application.Exit();

}

private void label1\_Click(object sender, EventArgs e)

{

Form ifrm = new FormLogin2();

ifrm.Left = this.Left; // задаём открываемой форме позицию слева равную позиции текущей формы

ifrm.Top = this.Top; // задаём открываемой форме позицию сверху равную позиции текущей формы

ifrm.Show(); // отображаем Form2

this.Hide(); // скрываем Form1 (this - текущая форма)

}

private void buttonLog\_Click(object sender, EventArgs e)

{

//Проверяю заполнили ли поля при регистрации

if (String.IsNullOrEmpty(LogtextBox1.Text))

{

MessageBox.Show("Заполните все поля!");

return;

}

if (String.IsNullOrEmpty(PasstextBox2.Text))

{

MessageBox.Show("Заполните все поля!");

return;

}

if (String.IsNullOrEmpty(NametextBox3.Text))

{

MessageBox.Show("Заполните все поля!");

return;

}

if (String.IsNullOrEmpty(SurntextBox4.Text))

{

MessageBox.Show("Заполните все поля!");

return;

}

if (String.IsNullOrEmpty(AgetextBox1.Text))

{

MessageBox.Show("Заполните все поля!");

return;

}

//Проверяем есть ли такой пользователь, если есть то выходим и добавление не происходит

//То есть если поле возращает True то выходим и пользователь не регистрируется

if (isUserExists())

return;

//Добавление пользователя в базу.

DB db = new DB();

MySqlCommand command = new MySqlCommand("INSERT INTO `clients`( `Логин`,`Пароль`,`Имя`,`Фамилия`,`Возраст`,`Роль`) VALUES ( @login, @password, @name, @Surname, @Age, @Role)", db.getConnection());

command.Parameters.Add("@login", MySqlDbType.VarChar).Value = LogtextBox1.Text;

command.Parameters.Add("@password", MySqlDbType.VarChar).Value = PasstextBox2.Text;

command.Parameters.Add("@name", MySqlDbType.VarChar).Value = NametextBox3.Text;

command.Parameters.Add("@Surname", MySqlDbType.VarChar).Value = SurntextBox4.Text;

command.Parameters.Add("@Age", MySqlDbType.VarChar).Value = AgetextBox1.Text;

if (CodetextBox1.Text == "")

{

command.Parameters.Add("@Role", MySqlDbType.VarChar).Value = null;

}

else

command.Parameters.Add("@Role", MySqlDbType.VarChar).Value = CodetextBox1.Text;

//Открываем базу данных и если введённые данные в поля верны, проходит регистрация

db.openConnection();

if (command.ExecuteNonQuery() == 1)

MessageBox.Show("Регистрация прошла успешно");

else

MessageBox.Show("Что-то пошло не так, попробуй позже");

//Закрываем базу данных

db.closeConnection();

}

//Поле для проверки идентичных логинов

public Boolean isUserExists()

{

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("Select \* FROM `clients` WHERE `Логин`=@LU " , db.getConnection());

command.Parameters.Add("@LU", MySqlDbType.VarChar).Value = LogtextBox1.Text;

//получаем данные и заносим в перемунную table, далее проверяем table, если там будет хоть 1 строка(запись) то выводим сообщение

//что такой логин уже есть и выходим, либо "Иначе, выходим"

adapter.SelectCommand = command;

adapter.Fill(table);

db.openConnection();

if (table.Rows.Count > 0)

{

MessageBox.Show("Такой логин уже есть!");

db.closeConnection();

return true;

}

else

return false;

}

private void label1\_MouseEnter(object sender, EventArgs e)

{

label1.ForeColor = Color.DeepSkyBlue; // смена цвета текста

}

private void label1\_MouseLeave(object sender, EventArgs e)

{

label1.ForeColor = Color.Black; // смена цвета текста

}

private void LogtextBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (int)Keys.Space)

e.KeyChar = '\0';

LogtextBox1.MaxLength = 15;

}

private void PasstextBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (int)Keys.Space)

e.KeyChar = '\0';

PasstextBox2.MaxLength = 15;

}

private void CodetextBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (int)Keys.Space)

e.KeyChar = '\0';

CodetextBox1.MaxLength = 4;

}

private void AgetextBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

char number = e.KeyChar;

if (!Char.IsDigit(number) && number != 8) // цифры и клавиша BackSpace

{

e.Handled = true;

}

AgetextBox1.MaxLength = 2;

}

private void NametextBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!Char.IsDigit(e.KeyChar)) return;

else

e.Handled = true;

NametextBox3.MaxLength = 15;

}

private void SurntextBox4\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!Char.IsDigit(e.KeyChar)) return;

else

e.Handled = true;

SurntextBox4.MaxLength = 15;

}

}

}

**3)Код формы авторизации:**

namespace ProgramMed

{

public partial class FormLogin2 : Form

{

public FormLogin2()

{

InitializeComponent();

}

public void buttonLogin\_Click(object sender, EventArgs e)

{

//Объявляем переменные у которых тип строка

String LoginUser = textBoxlogin.Text;

String PassUser = textBoxpassword.Text;

if (String.IsNullOrEmpty(textBoxlogin.Text))

{

MessageBox.Show("Заполните все поля!");

return;

}

if (String.IsNullOrEmpty(textBoxpassword.Text))

{

MessageBox.Show("Заполните все поля!");

return;

}

//на основе класса DB создаём переменную db чтобы использовать поля прописанные в классе

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("Select \* FROM `clients` WHERE `Логин`=@LU AND `Пароль` =@PU", db.getConnection());

//Даём значения ссылкам и используем ранее созданные переменные

command.Parameters.Add("@LU", MySqlDbType.VarChar).Value = LoginUser;

command.Parameters.Add("@PU", MySqlDbType.VarChar).Value = PassUser;

//Заносим полученные данные в таблицу, если есть хоть 1 строка(запись) то тогда появляется нужное окно

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count > 0)

{

var rol = table.Rows[0][6].ToString();

if (rol == "")

{

this.Hide();

MenuclientsForm3 Clients = new MenuclientsForm3();

Clients.Show();

}

if (rol == "2")

{

this.Hide();

StaffForm4 staff = new StaffForm4();

staff.Show();

}

if (rol == "3")

{

this.Hide();

AdminForm5 adm = new AdminForm5();

adm.Show();

}

}

else

MessageBox.Show("Введены некорректные данные!");

}

//Метод который скрывает первую форму авторизации и отображает форму регистрации

private void label3\_Click\_1(object sender, EventArgs e)

{

Form ifrm = new FormReg1();

ifrm.Left = this.Left; // задаём открываемой форме позицию слева равную позиции текущей формы

ifrm.Top = this.Top; // задаём открываемой форме позицию сверху равную позиции текущей формы

ifrm.Show(); // отображаем Form1

this.Hide(); // скрываем Form2 (this - текущая форма)

}

private void FormLogin2\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.Exit(); //закрывает программу, возможно вариант выше более верный

}

private void label3\_MouseLeave(object sender, EventArgs e)

{

label3.ForeColor = Color.Black;

}

private void label3\_MouseEnter(object sender, EventArgs e)

{

label3.ForeColor = Color.DeepSkyBlue;

}

private void textBoxlogin\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (int)Keys.Space)

e.KeyChar = '\0';

textBoxlogin.MaxLength = 15;

}

private void textBoxpassword\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (int)Keys.Space)

e.KeyChar = '\0';

textBoxpassword.MaxLength = 15;

}

}

}

**Задание 2**

**Ответил на контрольные вопросы:**

**1.После чего начинается кодирование модулей?**

После завершения проектирования каждого модуля и проведения их обзора и утверждения.

**2. Перечислите правила кодирования модулей**

‒ Каждый модуль должен иметь заголовок, оформленный по стандарту, который включается в "шапку" модуля.

‒ Код должен быть непротиворечивым, что уменьшает его сложность.

‒ Код должен быть структурным, так как это уменьшает ошибки и улучшает сопровождаемость.

**3. Что происходит после кодирования отдельных модулей?**

Интеграция модулей в систему.

**4. Что означает нисходящий подход?**

Одновременно выполняемые внутреннее проектирование, кодирование и тестирование осуществляются для отдельных модулей в том порядке, чтобы тестирование каждого модуля не зависело от еще не написанных модулей или от еще неподготовленных данных.

**5. Для чего нужна интеграция модулей?**

Для обеспечения взаимодействия между модулями и обмена данными между ними.