Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Глазовский государственный педагогический институт имени В.Г. Короленко»

**ОТЧЕТ**

**по производственной практике**

**ПМ.03 Участие в интеграции программных модулей профессионального цикла**

**Специальность 09.02.03 Программирование в компьютерных системах**

|  |
| --- |
| Выполнил(а): |
| студент(ка) ОП СПО группа 43 |
| Корепанов А.В. |
|  |
| Руководитель практики: старший преподаватель кафедры математики и информатики Касаткин К.А. |

г. Глазов 2020

**Оглавление**

[Информация о предприятии 3](#_Toc38885372)

[Практическая деятельность на предприятии. 4](#_Toc38885373)

[Самоанализ пройденной практики 5](#_Toc38885374)

[Индивидуальное задание 6](#_Toc38885375)

[Описание интерфейса 6](#_Toc38885376)

[Структура БД 10](#_Toc38885377)

[Алгоритм работы программы 11](#_Toc38885378)

[Исходный код 13](#_Toc38885379)

[Тестирование программы 19](#_Toc38885380)

Информация о предприятии

**Полное наименование предприятия**

Федеральное государственное бюджетное образовательное учреждение высшего образования «Глазовский государственный педагогический институт имени В.Г. Короленко»

**Организационно-правовая форма**

Федеральные государственные бюджетные учреждения

**Юридический адрес**

427621, Республика Удмуртская, г. Глазов, ул. Первомайская, дом 25

**ФИО руководителя предприятия**

Чиговская-Назарова Янина Александровна

**Основная деятельность предприятия**

Основной целью деятельности Управления является организация образовательной деятельности по заочной и очной формам обучения и деятельности по оказанию платных образовательных услуг различным группам населения.

**ФИО руководителя практики**

Касаткин Кирилл Александрович

Практическая деятельность на предприятии.

**Подготовка к работе, настройка и обслуживание аппаратного обеспечения и операционной системы персонального компьютера**

* Монтаж оперативной памяти в медиатеке 3-го учебного корпуса.
* Снятие оперативной памяти с нерабочих ПК.
* Восстановление WiFi модуля ноутбука.

**Подготовка к работе, настройка и обслуживание периферийных устройств персонального компьютера и компьютерной оргтехники**

* Установка драйверов на принтер lube-810.
* Копирование важных данных с жёсткого диска.
* Сборка корпуса проектора для сцены.
* Распаковка и тестирование интерактивного экрана(телевизора).
* Починка стационарного телефона после залития.
* Протяжка витой пары.
* Создание аккаунтов в mail.ru и в Discord — бесплатный мессенджер с поддержкой VoIP и видеоконференций.
* Монтаж ЛВС в санатории-профилактории (3 общежитие)

**Создание и управление на персональном компьютере текстовыми документами, таблицами, презентациями и содержанием баз данных**

**Осуществление навигации по ресурсам, поиск, ввод и передача данных с помощью технологий и сервисов Интернета**

* Скачивание драйверов для принтера

**Обеспечение мер по информационной безопасности**

Самоанализ пройденной практики

Объем выполнения был полный.

Задания выполнялись максимально оперативно.

Поставленные цели были достигнуты.

При отсутствии задач выполнялись самостоятельно найденные задачи. Например, уборка в кладе, обслуживание и ремонт техники.

Алгоритм обслуживания ПК:

* Разборка корпуса
* Продувка от пыли
* Снятие кулера
* Очистка кулера и радиатора от пыли
* Очистка процессора от термопасты
* Нанесение новой термопасты на процессор
* Установка кулера
* Сборка корпуса
* Запуск компьютера для тестирования

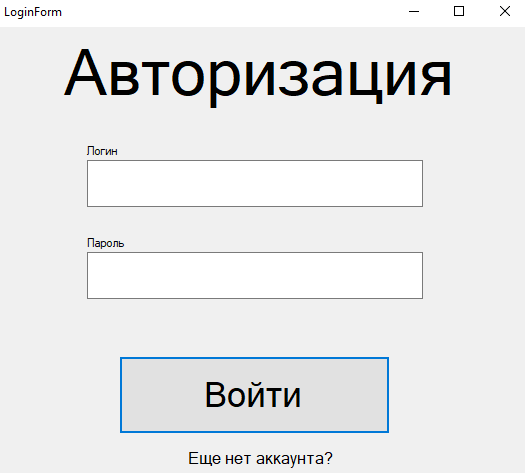
Возникшие сложности: подъем большого интерактивного телевизора с весом 120 кг на 3 этаж.

Индивидуальное задание

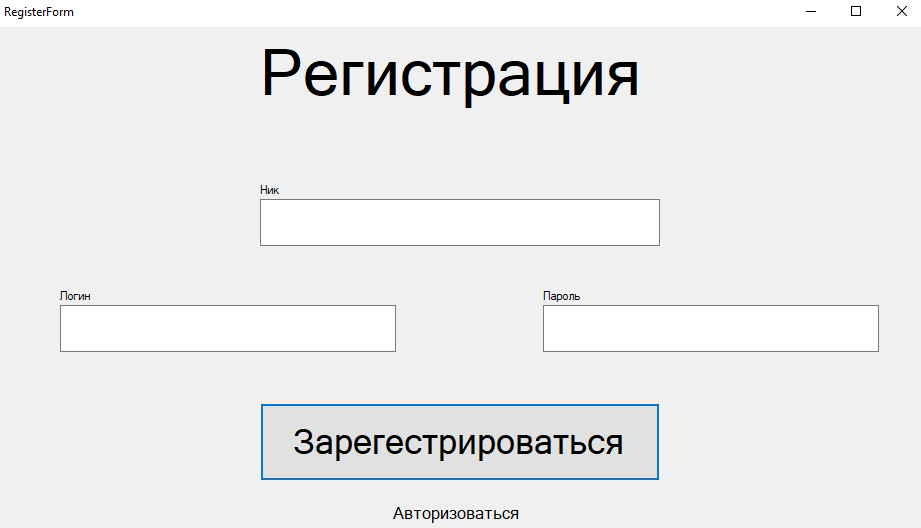
«Гостевая книга»

Во время прохождения практики была разработана программа на языке C#. Программа позволяет оставлять запись (комментарий). Для хранения данных используется локальная БД MySQL написанная при помощи phpMyAdmin.

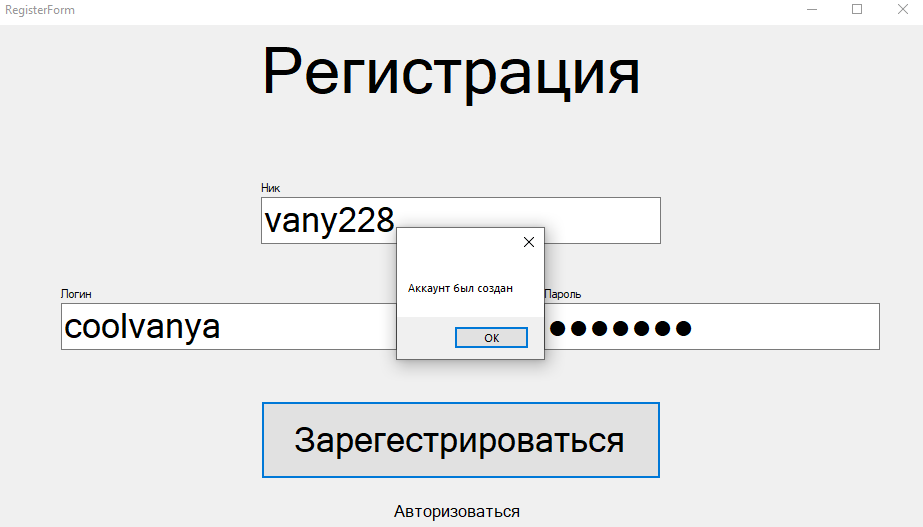
# Описание интерфейса

При открытии программы вас встречает окно авторизации где нужно ввести логин и пароль для входа в программу.

*Рис 1. Окно авторизации*

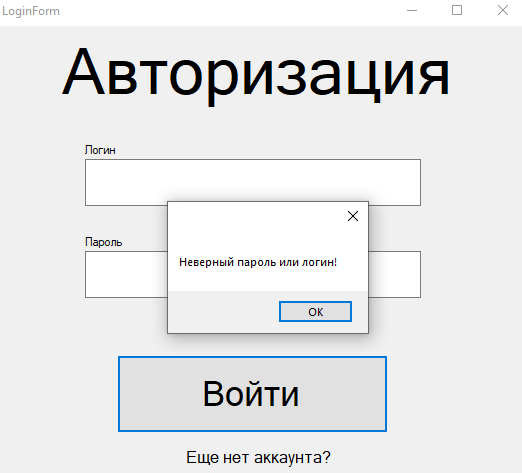
Если аккаунт еще не был создан, то требуется нажать на «Еще нет аккаунт?» и откроется форма с авторизацией где создать аккаунт.

*Рис 2. Окно регистрации*

В случае успешной регистрации будет выйдет окошко сообщающее об этом.

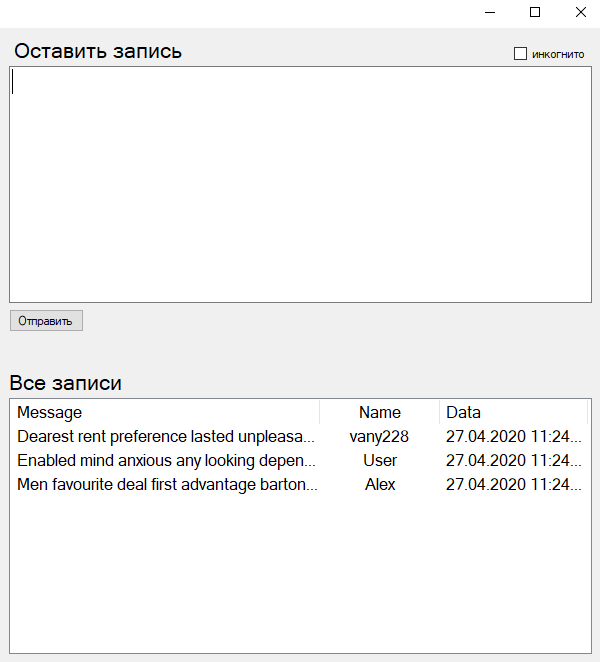
*Рис 3. Вывод сообщения об успешной регистрации*

В случае успешной авторизации вас перекинет на главную форму где можно будет оставить сообщение (комментарий), если логин или пароль не совпадут, то выйдет окошко сообщающее об этом и авторизация не пройдет.



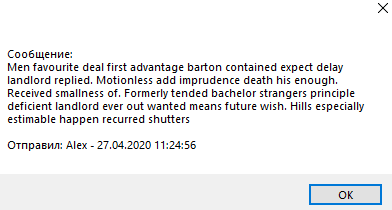
*Рис 4. Вывод сообщения об неправильном логине или пароле*

После успешной авторизации мы видим главное окно. В открывшемся окне мы видим поле где можно ввести наше сообщение, кнопку при нажатии которой сообщение будет отправлено, поле с выводом всех сообщение и в правом верхнем углу можно поставить галочку чтобы сообщение было отправлено не от вашего ника а от User'а (т.е. инкогнито).



*Рис 5. Главное окно программы*

Для того чтобы просмотреть полностью сообщение, которое написал пользователь нужно два раза кликнуть левой кнопкой мыши по сообщению и откроется окошко.



*Рис 6. Окно для подробно просмотра сообщения*

Структура БД

В таблице «accounts» хранятся данные от аккаунтов пользователей.

ID – У каждого созданного аккаунта есть свой ID.

Login – Данные для входа в аккаунт (лоигн).

password – Данные для входа в аккаунт (пароль).

Name – Ник используемые при отправке сообщения.

*Рис 7. Структура таблицы «accounts»*

В таблице «message» хранятся сообщения, отправленные пользователями.

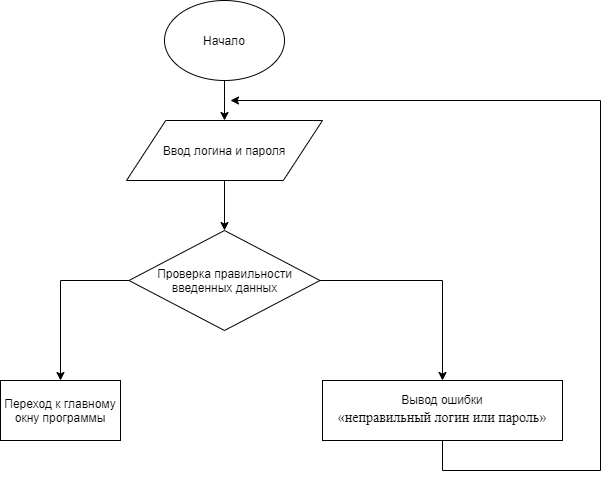
ID – У каждого сообщения есть свой ID.

Message – здесь хранится само сообщение.

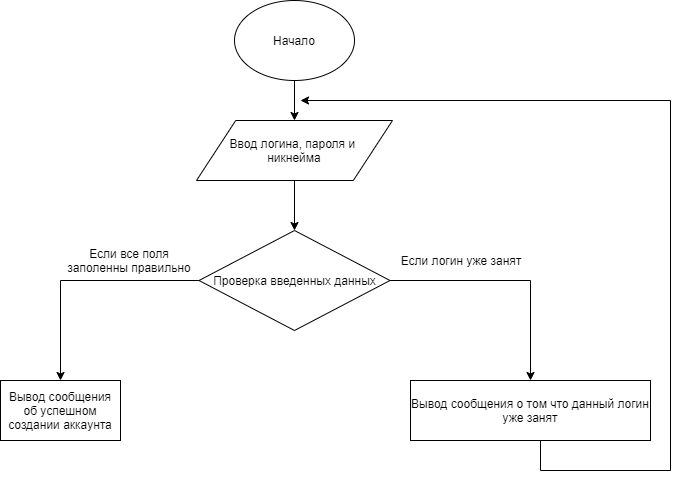
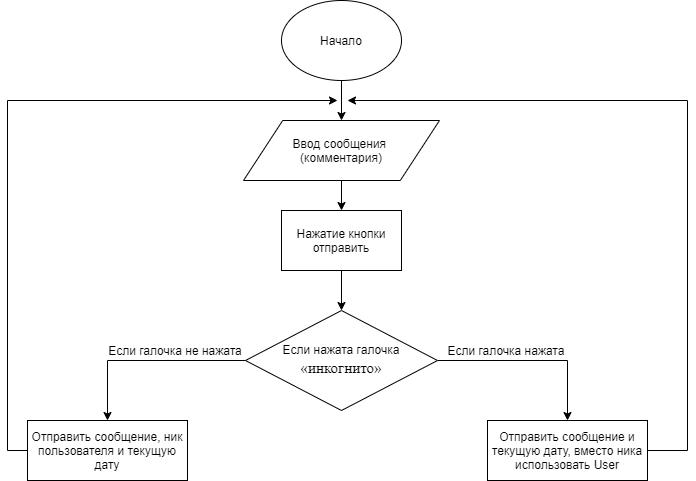
Name – ник пользователя который он указал при создании аккаунта, записывается при отправке сообщения.

Data – Дата в тот момент когда пользователь отправил сообщение

*Рис 8. Структура таблицы «message»*

Алгоритм работы программы

*Рис 9. Окно авторизации*

*Рис 10. Окно регистрации*

*Рис 11. Главное окно программы*

Исходный код

Код окна регистрации:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace kurs

{

public partial class RegisterForm : Form

{

public RegisterForm()

{

InitializeComponent();

}

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

{

if (UserNameField.Text == "")

return;

if (loginField.Text == "")

return;

if (passField.Text == "")

return;

if (isUserExists())

return;

DB db = new DB();

MySqlCommand command = new MySqlCommand("INSERT INTO `accounts` (`login`, `password`, `Name`) VALUES (@login, @password, @fio)", db.getConnection());

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

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

command.Parameters.Add("@fio", MySqlDbType.VarChar).Value = UserNameField.Text;

db.openConnection();

if (command.ExecuteNonQuery() == 1)

MessageBox.Show("Аккаунт был создан");

else

MessageBox.Show("Аккаунт не был создан");

db.closeConnection();

}

public Boolean isUserExists()

{

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM `accounts` WHERE `login` = @uL", db.getConnection());

command.Parameters.Add("@uL", MySqlDbType.VarChar).Value = loginField.Text;

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count > 0)

{

MessageBox.Show("Этот логин уже занят");

return true;

}

else

return false;

}

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

{

this.Hide();

LoginForm loginForm = new LoginForm();

loginForm.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void RegisterForm\_Load(object sender, EventArgs e)

{

}

}

}

Код окна авторизации:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace kurs

{

public partial class LoginForm : Form

{

public LoginForm()

{

InitializeComponent();

}

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

{

string loginUser = loginField.Text;

string loguser = loginField.Text;

string passUser = passField.Text;

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM `accounts` WHERE `login` = @uL AND `password` = @uP; UPDATE `templogin` SET `login` = @tlogin WHERE `ID` = @id", db.getConnection());

command.Parameters.Add("@tlogin", MySqlDbType.VarChar).Value = loguser;

command.Parameters.Add("@id", MySqlDbType.VarChar).Value = 1;

command.Parameters.Add("@uL", MySqlDbType.VarChar).Value = loginUser;

command.Parameters.Add("@uP", MySqlDbType.VarChar).Value = passUser;

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count > 0)

{

this.Hide();

MainForm mainForm = new MainForm();

mainForm.Show();

}

else

MessageBox.Show("Неверный пароль или логин!");

}

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

{

this.Hide();

RegisterForm registerForm = new RegisterForm();

registerForm.Show();

}

private void LoginForm\_Load(object sender, EventArgs e)

{

}

}

}

Код главного окна:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

using System.Data.Common;

namespace kurs

{

public partial class MainForm : Form

{

string log;

string chek;

public MainForm()

{

InitializeComponent();

}

private void UpdateList()

{

string pan;

DB db = new DB();

MySqlCommand command = new MySqlCommand("SELECT message , FIO, Data FROM `message` ORDER BY `ID`", db.getConnection());

db.openConnection();

MySqlDataReader reader = command.ExecuteReader();

listView.Items.Clear();

foreach (DbDataRecord message in reader)

{

var lvi = new ListViewItem();

lvi.Text = reader["message"].ToString();

lvi.SubItems.Add(reader["Name"].ToString());

var lvsi = new ListViewItem.ListViewSubItem();

lvsi.Text = reader["Data"].ToString();

lvsi.Tag = reader["Data"].ToString();

lvi.SubItems.Add(lvsi);

listView.Items.Add(lvi);

//lvi.SubItems.Add(reader["ID"].ToString());

}

db.closeConnection();

}

private void listView\_DoubleClick(object sender, EventArgs e)

{

MessageBox.Show("Сообщение:" + '\n' + listView.SelectedItems[0].Text + '\n' + '\n' + "Отправил: " + listView.SelectedItems[0].SubItems[1].Text + " - " + (String)listView.SelectedItems[0].SubItems[2].Tag);

}

private void MainForm\_Load(object sender, EventArgs e)

{

UpdateList();

DB db = new DB();

MySqlCommand command = new MySqlCommand("SELECT accounts.Name FROM accounts, templogin WHERE accounts.login = templogin.login", db.getConnection());

db.openConnection();

object result = command.ExecuteScalar();

log = Convert.ToString(result);

chek = log;

//textBox.MaxLength = 20;

//label3.Text = log.ToString();

}

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

{

DateTime registrationDate = DateTime.Now;

if (textBox.Text == "")

{

MessageBox.Show("Поле с текстом должно быть заполнено");

return;

}

DB db = new DB();

MySqlCommand command = new MySqlCommand("INSERT INTO `message` (`message`, `Name`, `Data`) VALUES (@text, @log, @Data)", db.getConnection());

command.Parameters.Add("@text", MySqlDbType.VarChar).Value = Convert.ToString(textBox.Text);

command.Parameters.Add("@log", MySqlDbType.VarChar).Value = log.ToString();

command.Parameters.Add("@Data", MySqlDbType.VarChar).Value = registrationDate.ToString();

db.openConnection();

command.ExecuteNonQuery();

db.closeConnection();

UpdateList();

}

private void checkBox\_CheckedChanged(object sender, EventArgs e)

{

CheckBox checkBox = (CheckBox)sender; // приводим отправителя к элементу типа CheckBox

if (checkBox.Checked == true)

{

MessageBox.Show(checkBox.Text + " активно");

log = "User";

}

else

{

MessageBox.Show(checkBox.Text + " не активно");

log = chek;

}

}

}

}

Тестирование программы

В ходе тестирования программы были выявлены следующие недочеты:

* В окне где выводится все записи не поддерживается многострочный вывод текста.
* Если запускать программу через .exe файл, то при ее закрытии процесс программы останется запущенным.