

Московский приборостроительный техникум

Практическая работа №1

Дисциплина: ОП.04 «Основы алгоритмизации и программирования»

Тема: Разработка проекта авторизации с помощью Windows Forms на языке C#

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Разработчик веб и мультимедийных приложений

Разработал:
преподаватель Колмыкова В.А..

Рассмотрено и одобрено на заседании ЦМК
Профессиональных модулей 09.02.07-ВД

Протокол от _____ г. № _____

Председатель ЦМК _____ / М.С. Прищеп /

Квалификация: Разработчик веб и мультимедийных приложений

Цель работы

Изучение и практическое применение элементов управления и контейнеров для разработки приложений типа Windows Forms на языке C#. Изучение способов подключения базы данных и её взаимодействия с приложением.

Задание

Руководствуясь перечисленными ниже требованиями, выполнить действия по созданию проекта оконного приложения Windows Forms в интегрированной среде разработки MS Visual Studio. Проект будет содержать в себе множество модулей, выполняемые постепенно, в соответствии с заданием.

Данная практическая работа подразумевает создание формы авторизации пользователя в системе и разработка её полного функционала.

Квалификация: Разработчик веб и мультимедийных приложений

Этапы выполнения:

Этап 1. Установка локального веб-сервера

Первым делом необходимо перейти на сайт <https://www.apachefriends.org/> для установки веб-сервера.



Рисунок 1 – Скачивание установщика программы

После установки необходимо перейти в каталог, куда был скачен сервер и открыть приложение «xampp-control».

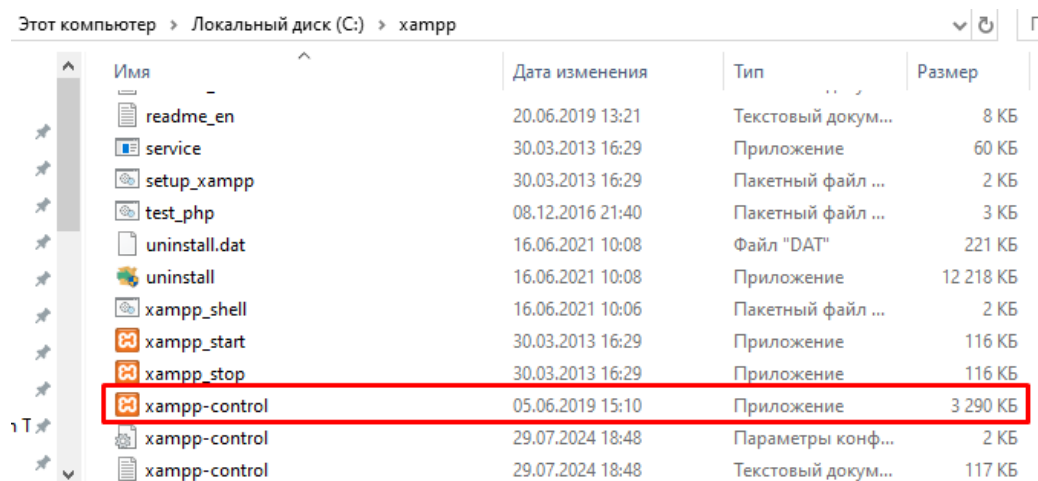


Рисунок 2 – Открытие программы

Квалификация: Разработчик веб и мультимедийных приложений

Приложение будет запущено, необходимо включить модули: apache и mysql, после чего нажать на кнопку admin напротив модуля mysql.

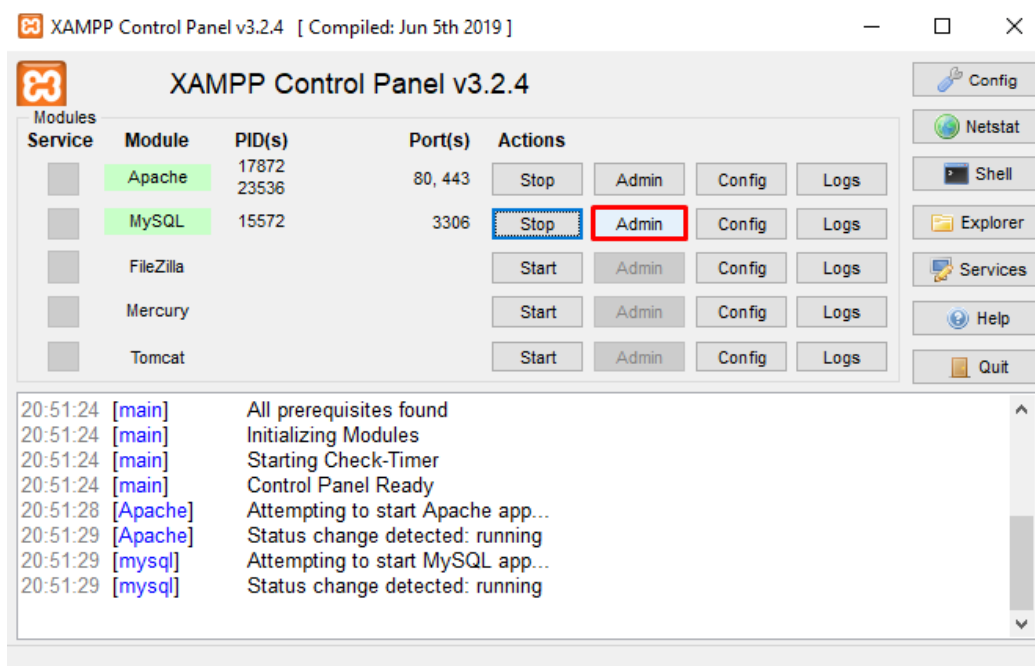


Рисунок 3 – Запуск модулей и открытие веб-приложения

Квалификация: Разработчик веб и мультимедийных приложений

Этап 2. Первый запуск phpMyAdmin

phpMyAdmin — веб - приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. RHPMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд, предоставляя дружественный интерфейс. (<https://ru.wikipedia.org/wiki/PhpMyAdmin>)

После нажатия на кнопку «Admin», откроется новая вкладка в браузере с веб-приложением phpMyAdmin. В правом меню необходимо выбрать пункт «Создать БД» (рисунок 4). При создании базы данных нужно дать ей имя и правильную кодировку (utf8_general_ci) (рисунок 5).

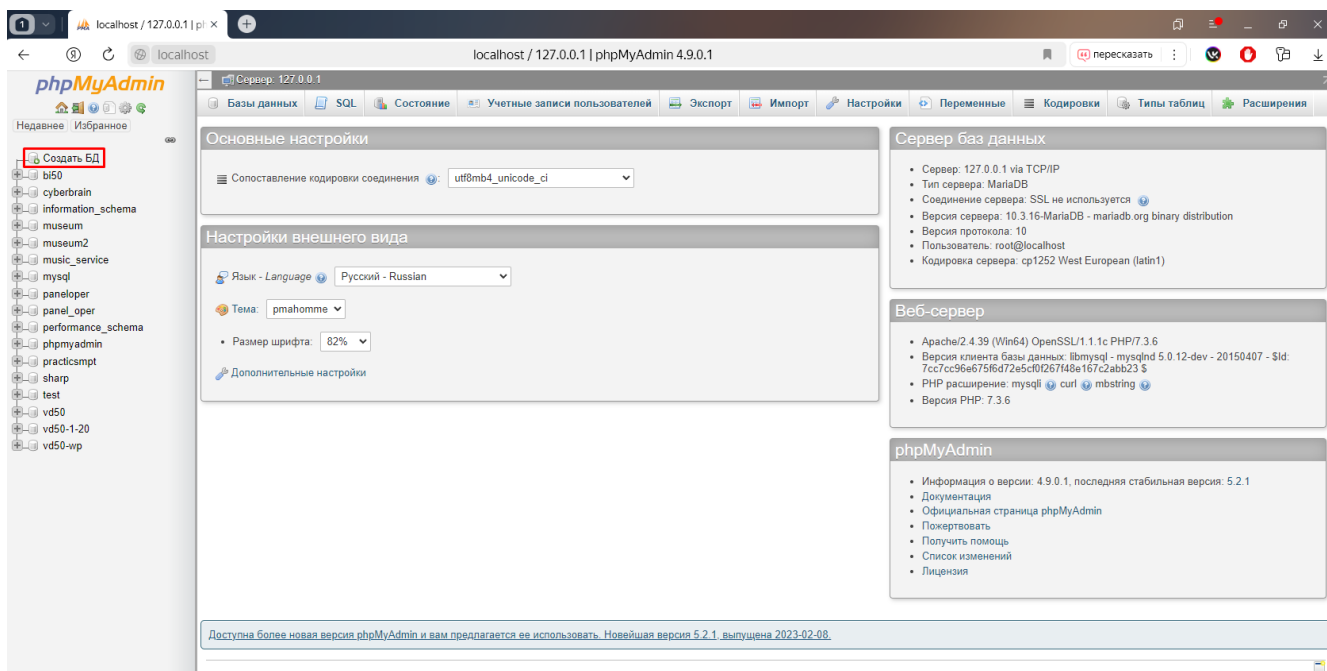


Рисунок 4 – Главная страница phpMyAdmin

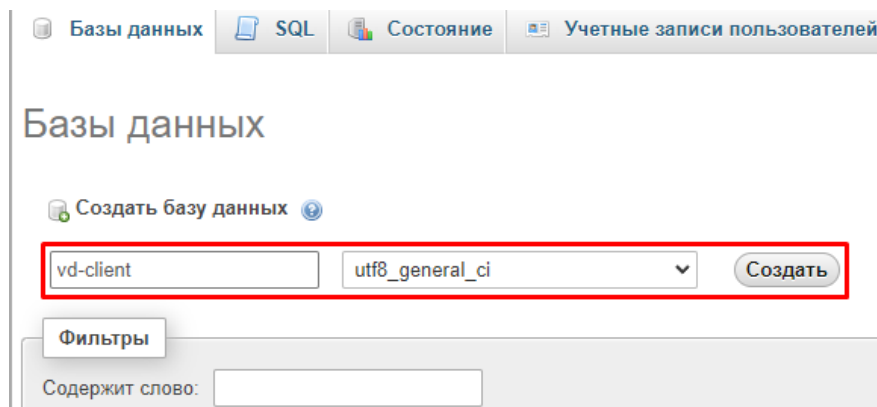


Рисунок 5 – Создание базы данных

Квалификация: Разработчик веб и мультимедийных приложений

Этап 3. Импорт базы данных

После создания базы данных в правой части приложения в верхней панели необходимо выбрать вкладку «Импорт». К заданию был приложен файл с разрешением .sql – его необходимо загрузить и нажать кнопку «Вперед».

Сервер: 127.0.0.1 » База данных: vd-client

Структура SQL Поиск Запрос по шаблону Экспорт **Импорт** Операции

Импорт в базу данных "vd-client"

Импортируемый файл:

Файл может быть сжат в архив (gzip, bzip2, zip) или находиться без сжатия.
Имя сжатого файла должно заканчиваться в виде `.[формат].[сжатие]`. Пример: `.sql.zip`

Обзор вашего компьютера: **Выберите файл** clientka.sql (Максимальный размер: 2,048КиБ)

Вы также можете просто перетащить файл на любой странице.

Кодировка файла: utf-8

Частичный импорт:

☒ Разрешить скрипту разбивать процесс импорта при приближении временного лимита. (Может быть использовано п...

Пропустить указанное число запросов (для SQL), начиная со следующего: 0

Прочие параметры:

☒ Включить проверку внешних ключей

Формат:

SQL

Параметры формата:

Режим совместимости SQL: NONE

☒ Не использовать атрибут AUTO_INCREMENT для нулевых значений

Вперед

Рисунок 6 – Импортирование базы данных

Квалификация: Разработчик веб и мультимедийных приложений

Далее слева появиться созданная база данных с импортированными таблицами. Необходимо нажать на таблицу «users», после чего откроется таблица с тестовым значением, где прописан логин и пароль пользователя – его нужно запомнить.

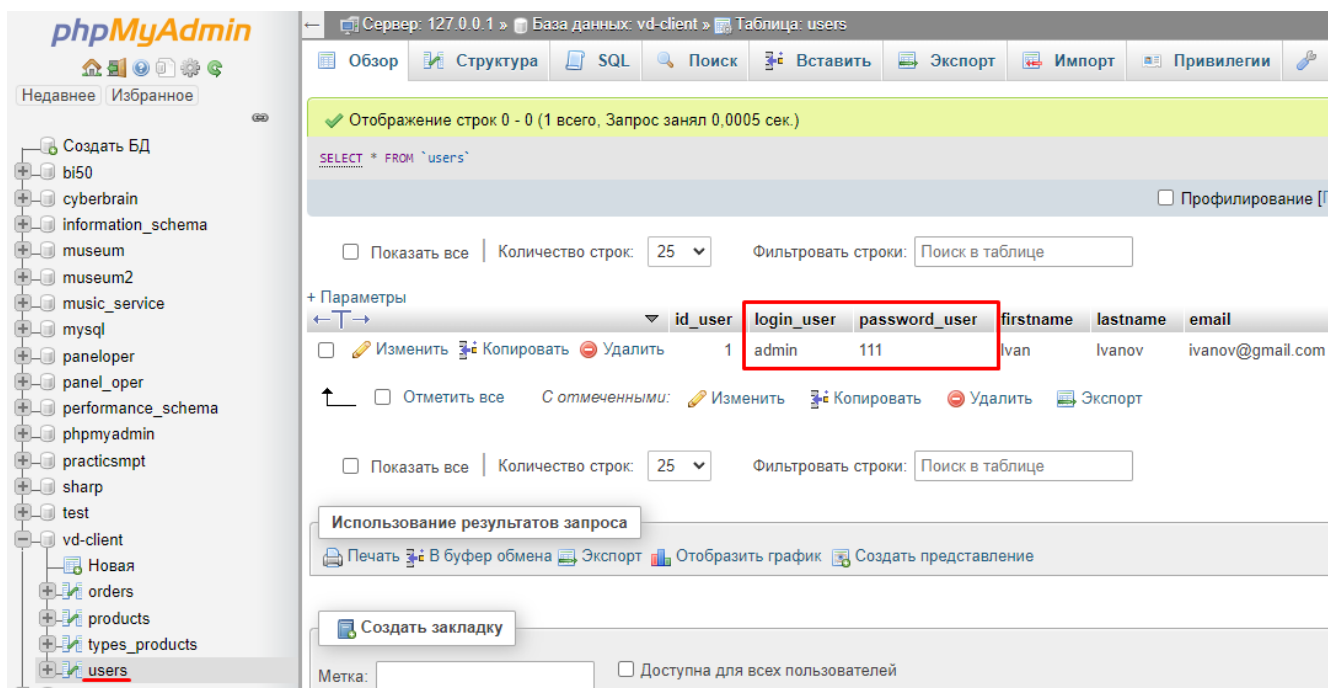


Рисунок 7 – Данные в таблице пользователей

Квалификация: Разработчик веб и мультимедийных приложений

Этап 4. Создание формы авторизации

1. При создании проекта необходимо внимательно выбрать шаблон приложения, в данном случае, для дальнейшей работы, нужен шаблон: «Приложение Windows Forms (.NET Framework)»;
2. Изменить заголовок главной формы на «Авторизация»;
3. Добавить элемент управления «Panel», которые необходимо будет закрепить в родительском контейнере, после чего добавить ему любой цвет;
4. Добавить еще раз элемент управления «Panel», дать ему цвет и расположить наверху формы, чтобы это выглядело как шапка;
5. Создать в проекте папку и назвать её «Images», после добавить в нее две подходящие картинки. Далее необходимо расположить элемент «PictureBox» и выбрать картинку из папки с проектом. Первая картинка с аватаром должна располагаться в левой части приложения по горизонтали и по середине по вертикали;
6. Добавить элемент «Label» на форму и настроить его стилизацию (Шрифт: Microsoft Sans Serif, начертание: обычный, размер: 28, набор символов: кириллица). После изменить название на «Login in system», изменить расположение элемента на форме (TextAlign: MiddleCenter), изменить прикрепление элемента к контейнеру (Dock: Fill) и отключить автоматический подбор размера (AutoSize: False);
7. Добавить элемент «Label» на форму и настроить его стилизацию (Шрифт: Microsoft Sans Serif, начертание: полужирный, размер: 20, набор символов: кириллица). После дать обозначение данному элементу «X» - он будет отвечать за закрытие формы. Расположение данного закрытия приложения должно быть в верхнем правом углу;
8. Добавить на форму два элемента управления «TextBox», которые необходимо назвать в соответствии с их значением. Первый элемент (TextBox1) – «loginUser»,

Квалификация: Разработчик веб и мультимедийных приложений

второй (TextBox2) – «passwordUser». Для поля с вводом пароля необходимо указать запрет на видимость вводимых значений (UseSystemPasswordChar: True);

9. Добавить на форму два элемента «Label», чтобы отобразить подсказки для полей ввода (TextBox). Label1 – «Логин:», label2 – «Пароль:»;

10. Добавить на форму элемент управления «Button» и расширить его до размеров поле ввода. После дать видимое наименование «Login», и невидимое наименование «loginButton». Также необходимо стилизовать текст внутри кнопки (Шрифт: Microsoft Sans Serif, начертание: полужирный, размер: 20, набор символов: кириллица);

11. Осуществить настройку формы: позиционирование формы при открытии должно происходить по центру (StartPosition: CenterScreen), рамка приложения должна быть скрыта

«Стилизация формы, описанная выше, не является обязательной частью. Можно подобрать цвета и картинки, которые захочется».

На рисунке 8 представлена форма, которая должна получиться в ходе выполнения вышеперечисленных действий.



Рисунок 8 – Форма авторизации

Квалификация: Разработчик веб и мультимедийных приложений

Этап 5. Подключение базы данных к приложению

1. Открыть вкладку «Проект» и выбрать «Управление пакетами NuGet».

NuGet – это бесплатный пакетный менеджер с открытым исходным кодом, служащий для .NET и .NET Core механизмом совместного использования кода, поддерживаемым Microsoft.

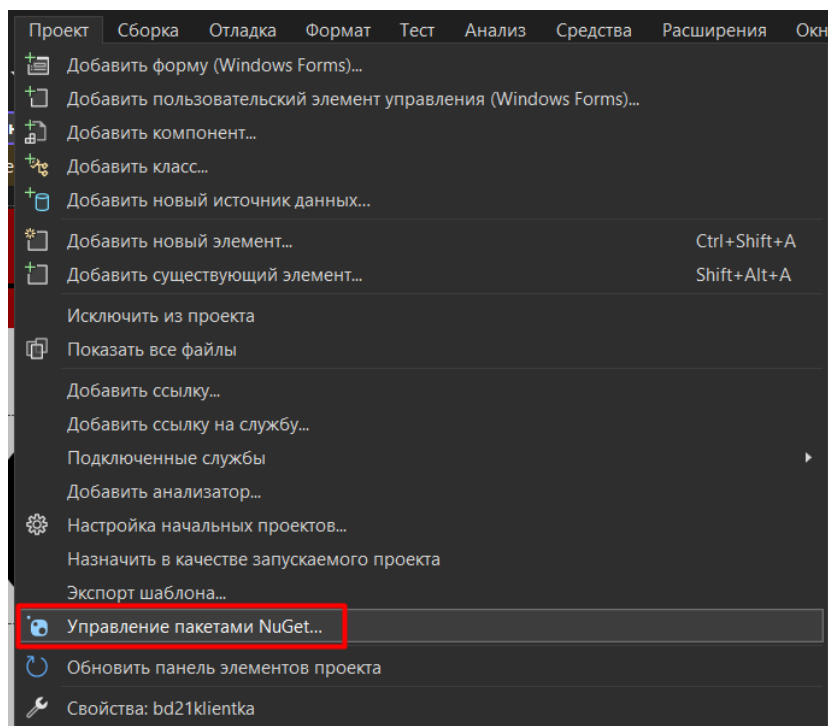


Рисунок 9 – Выбор вкладки «Управление пакетами NuGet»

2. Во вкладке обзор, в поисковой строке ввести MySQL.Data, в высвеченных вариантах выбрать первую строчку и нажать на нее для установки. Данный пакет будет необходим в дальнейшем, чтобы связать приложение с базой данных.

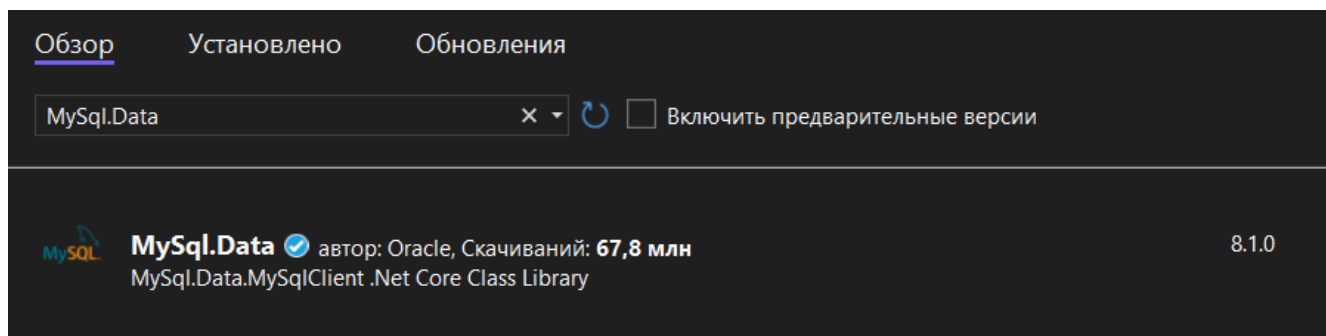


Рисунок 10 – Установка необходимого пакета

Квалификация: Разработчик веб и мультимедийных приложений

3. Теперь необходимо добавить новый элемент «Класс», в котором будет описываться подключение приложения к базе данных, его необходимо назвать DB.

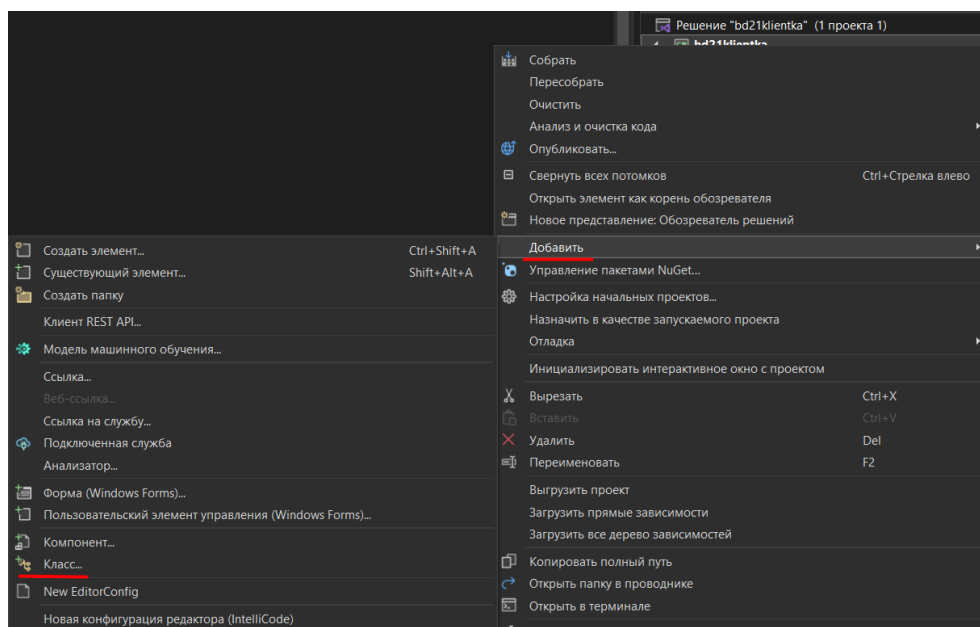


Рисунок 11 – Добавление класса

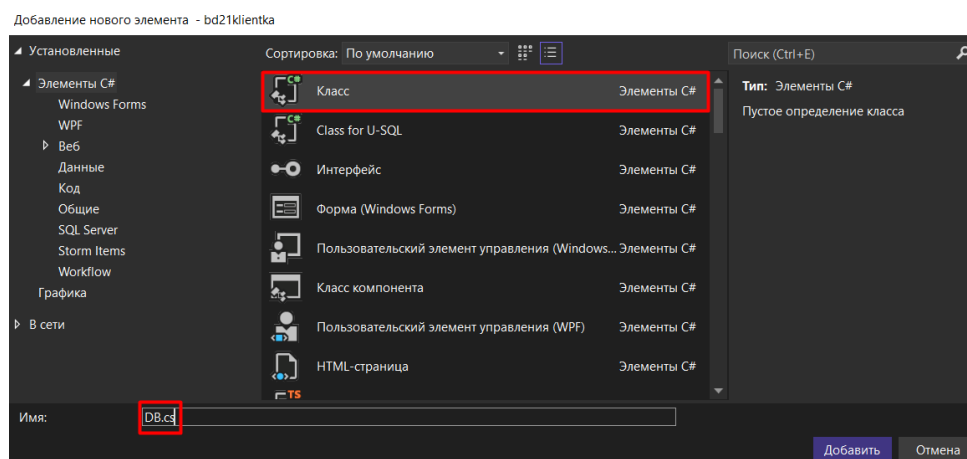


Рисунок 12 – Название класса

4. Перед началом написания кода необходимо добавить библиотеку, скаченную ранее (using MySql.Data.MySqlClient). Далее создается переменная с модификацией private, в которой содержатся данные для создания подключения к БД. В параметрах MySqlConnection указывается используемый сервер (localhost), логин пользователя в PhpMyAdmin (root), пароль пользователя в PhpMyAdmin (если используется XAMPP – пароль пустой, если OpenServer – пароль: root) и имя базы данных.

Квалификация: Разработчик веб и мультимедийных приложений

MySQLConnection – это класс представляет открытое соединение с базой данных MySQL в C#. Можно передавать строку подключения конструктору MySqlConnection класса для инициализации нового экземпляра MySqlConnection класса, который может подключаться к базе данных.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using MySQL.Data.MySqlClient;
7
8 namespace bd21klientka
9 {
10     Ссылка: 0
11     internal class DB
12     {
13         private MySqlConnection conn = new MySqlConnection("server=localhost;user=root;password=;database=bd50_21;");
```

Рисунок 13 – Подключение библиотеки и БД

5. Создать переменную openConnection(), которая будет доступна в любой части кода. Внутри прописывается условие: если conn (переменная подключения к бд) с параметром state (позволяет объекту варьировать свое поведение в зависимости от внутреннего состояния) равна текущему состоянию подключения к источнику данных (к БД) с параметром closed (соединение закрыто), то переменная conn должна открывать соединение.

Если простыми словами, то если нет соединения к подключению к БД, то нужно его открыть. Далее создается публичная переменная, которая делает все то же самое, что и прошлая, но наоборот. Если соединение есть, значит нужно его закрыть. Последняя переменная отвечает за передачу подключения к БД.

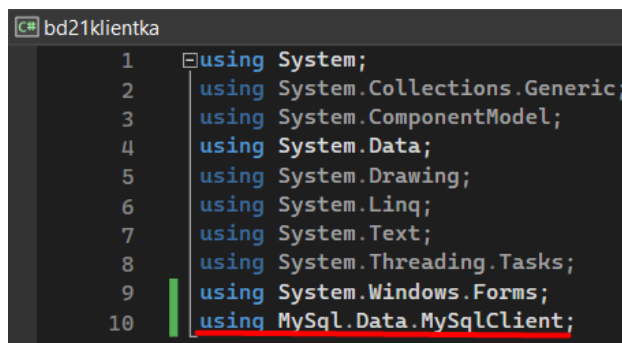
```
private MySqlConnection conn = new MySqlConnection("server=localhost;user=root;password=;database=bd50_21;");
Ссылка: 0
public void openConnection()
{
    if (conn.State == System.Data.ConnectionState.Closed)
    {
        conn.Open();
    }
}
Ссылка: 0
public void closeConnection()
{
    if (conn.State == System.Data.ConnectionState.Open)
    {
        conn.Close();
    }
}
Ссылка: 0
public MySqlConnection GetConnection()
{
    return conn;
}
```

Рисунок 14 – Подключение к базе

Квалификация: Разработчик веб и мультимедийных приложений

Этап 6. Реализация авторизации пользователя

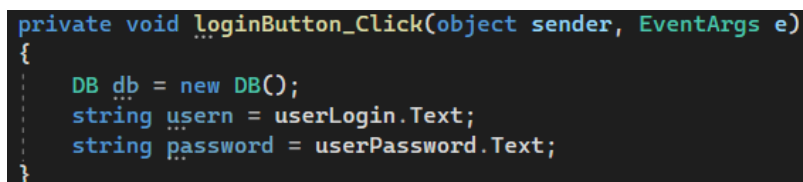
1. После создания класса, можно перейти на главную форму проекта. В каждой форме будет необходимо добавлять библиотеку с MySQL.



```
1 using System;  
2 using System.Collections.Generic;  
3 using System.ComponentModel;  
4 using System.Data;  
5 using System.Drawing;  
6 using System.Linq;  
7 using System.Text;  
8 using System.Threading.Tasks;  
9 using System.Windows.Forms;  
10 using MySQL.Data.MySqlClient;
```

Рисунок 15 – Добавление библиотеки

2. Кликнув дважды на кнопку (Login) на главной форме, откроется приватная процедура с событием «клик» на кнопку. Внутри нее необходимо прописать создание переменной с классом DB (подключение к БД). Далее необходимо создать строковые переменные, которые будут в себе содержать вписанное в поля ввода текст.



```
private void loginButton_Click(object sender, EventArgs e)  
{  
    DB db = new DB();  
    string usern = userLogin.Text;  
    string password = userPassword.Text;  
}
```

Рисунок 16 – Создание необходимые переменных

3. Создаются несколько переменных: переменная dt с классом DataTable, которая будет хранить данные в виде таблицы, переменная adapter с классом MySqlDataAdapter, который используется для получения данных из источника данных и заполнения таблиц, и переменная cmd с классом MySqlCommand, которая предоставляет инструкцию SQL для выполнения запроса в базе данных. В скобках указывается необходимый запрос, в данном случае это вывод всех атрибутов из таблицы пользователей, где атрибут логина равен переменной, в которой содержится вводимое в поле ввода значение и тоже самое с атрибутом пароля. В конце указывается переменная db (подключение к бд) с параметром GetConnection – который возвращает подключение.

Квалификация: Разработчик веб и мультимедийных приложений

```
private void loginButton_Click(object sender, EventArgs e)
{
    DB db = new DB();
    string usern = userLogin.Text;
    string password = userPassword.Text;

    DataTable dt = new DataTable();
    MySqlDataAdapter adapter = new MySqlDataAdapter();
    MySqlCommand cmd = new MySqlCommand($"SELECT * FROM user WHERE login_user = '{usern}' AND password_user = '{password}';", db.GetConnection());
```

Рисунок 17 – Создание переменных для взаимодействия с БД и написание запроса

4. Далее переменной adapter с параметром SelectCommand (он содержит SQL-команды, отправляемые в хранилище данных при вызовах методов Fill() и Update()) присваивается переменная cmd. Это значит, что переменной adapter присваивается запрос, который содержался в переменной cmd. После выполняется команда с переменной adapter и параметром Fill, который в свою очередь выполняет команду SQL SELECT (указанную в свойстве SelectCommand) для запроса к базе данных и загрузки этих данных в объект DataTable. В скобках команды указывается созданная ранее переменная таблицы – dt.

Теперь необходимо условие: если в таблице количество строк больше нуля, то вывод сообщения об успешной авторизации, иначе вывод сообщения об ошибке.

```
DataTable dt = new DataTable();
MySqlDataAdapter adapter = new MySqlDataAdapter();
MySqlCommand cmd = new MySqlCommand($"SELECT * FROM users

adapter.SelectCommand = cmd;
adapter.Fill(dt);

if (dt.Rows.Count > 0)
{
    MessageBox.Show("Авторизация прошла успешно");
}
else
{
    MessageBox.Show("Неверный логин или пароль");
}
```

Рисунок 18 – Выполнение запроса и проверка на функционирование

Квалификация: Разработчик веб и мультимедийных приложений

Составление отчета

Оформление отчёта

Для оформления отчёта допускается использование любых текстовых редакторов, в которых могут быть реализованы предъявляемые к оформлению отчёта требования, например, Microsoft Office, Open Office или подобные. Отчёт о выполнении работы оформляется в соответствии с методическими указаниями.

Содержание отчёта

Отчёт о выполнении работы должен включать:

1. Титульный лист.
2. Цель работы.
3. Задание.
4. Скриншоты, иллюстрирующие отображение всех форм проекта в режиме проектирования и в режиме выполнения программы.
5. Вывод.
6. Исходных код программы (файлы проекта с расширением *.cs).

Пример:

Содержание	
Цель работы	1
Задание	1
Выполнение работы	1
Вывод	1
Код программы.....	1