МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Институт информационных технологий\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

наименование института (факультета)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_МПО ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

наименование кафедры \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Базы данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ наименование дисциплины в соответствии с учебным планом

ЛАБОРАТОРНАЯ РАБОТА № 7

ВЗАИМОДЕЙСТВИЕ С БД ACCESS ИЗ ПРИЛОЖЕНИЯ, СОЗДАННОГО НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ

Исполнитель

студент \_\_1ПИб-02-1оп-22\_\_

группа

\_\_\_Микуцких Г. А.\_\_\_

Фамилия, имя, отчество

Руководитель \_\_\_Селяничев О.Л.\_\_\_

Ф.И.О. преподавателя

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_2024\_ год

Часть I.

Создайте таблицу базы данных с полями, содержание которых - фамилия, год рождения, оценки по математике, информатике, иностранному языку. (Или работайте с имеющейся).

В Access`е была создана пустая БД (рис. 1-2) с полями: Код (Счётчик), Фамилия (Короткий текст), Дата рождения (Дата и время → Краткий формат даты), Математика (Байт), Информатика (Байт), Иностранный язык (Байт).

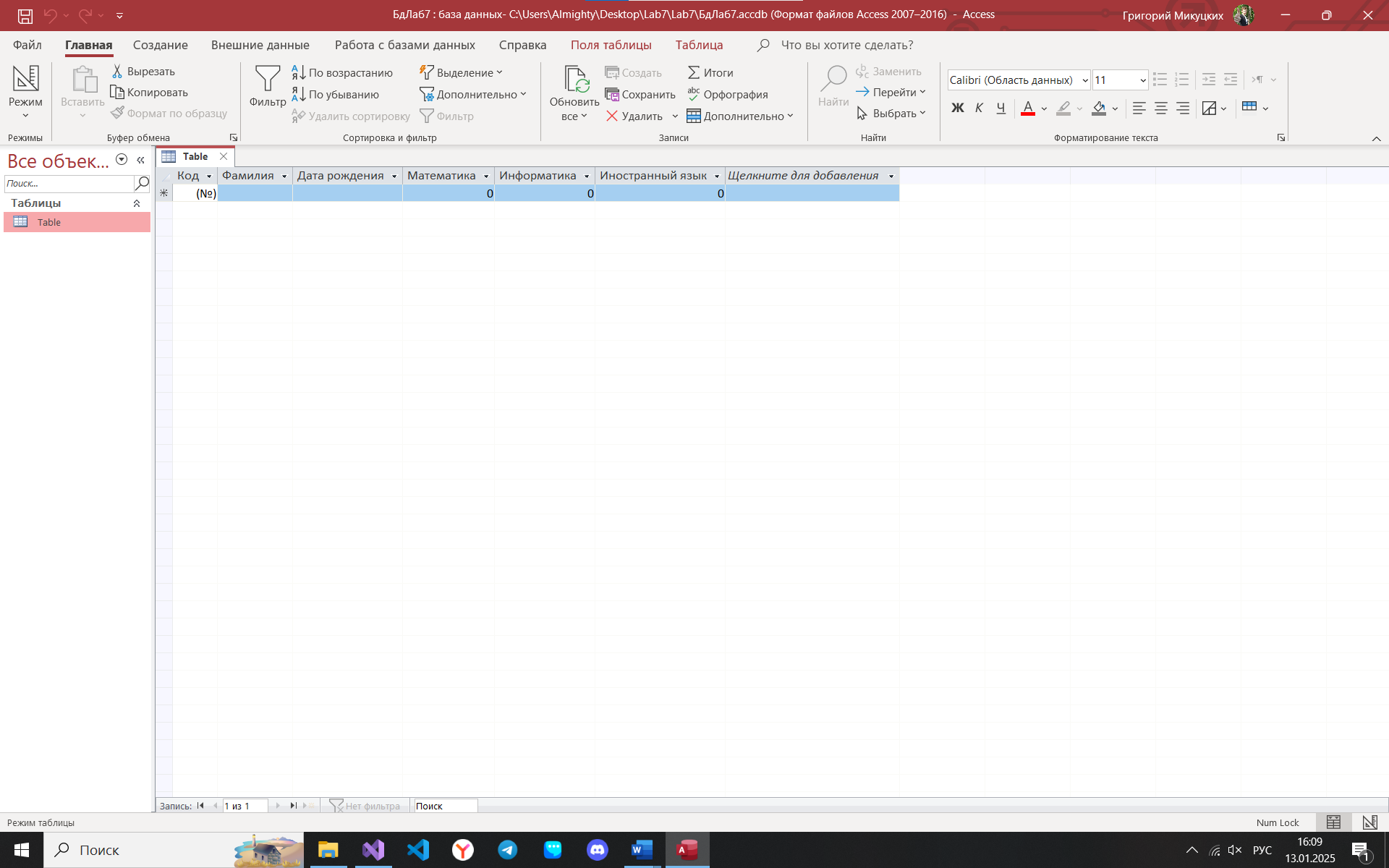


Рис. 1. Общий вид пустой БД

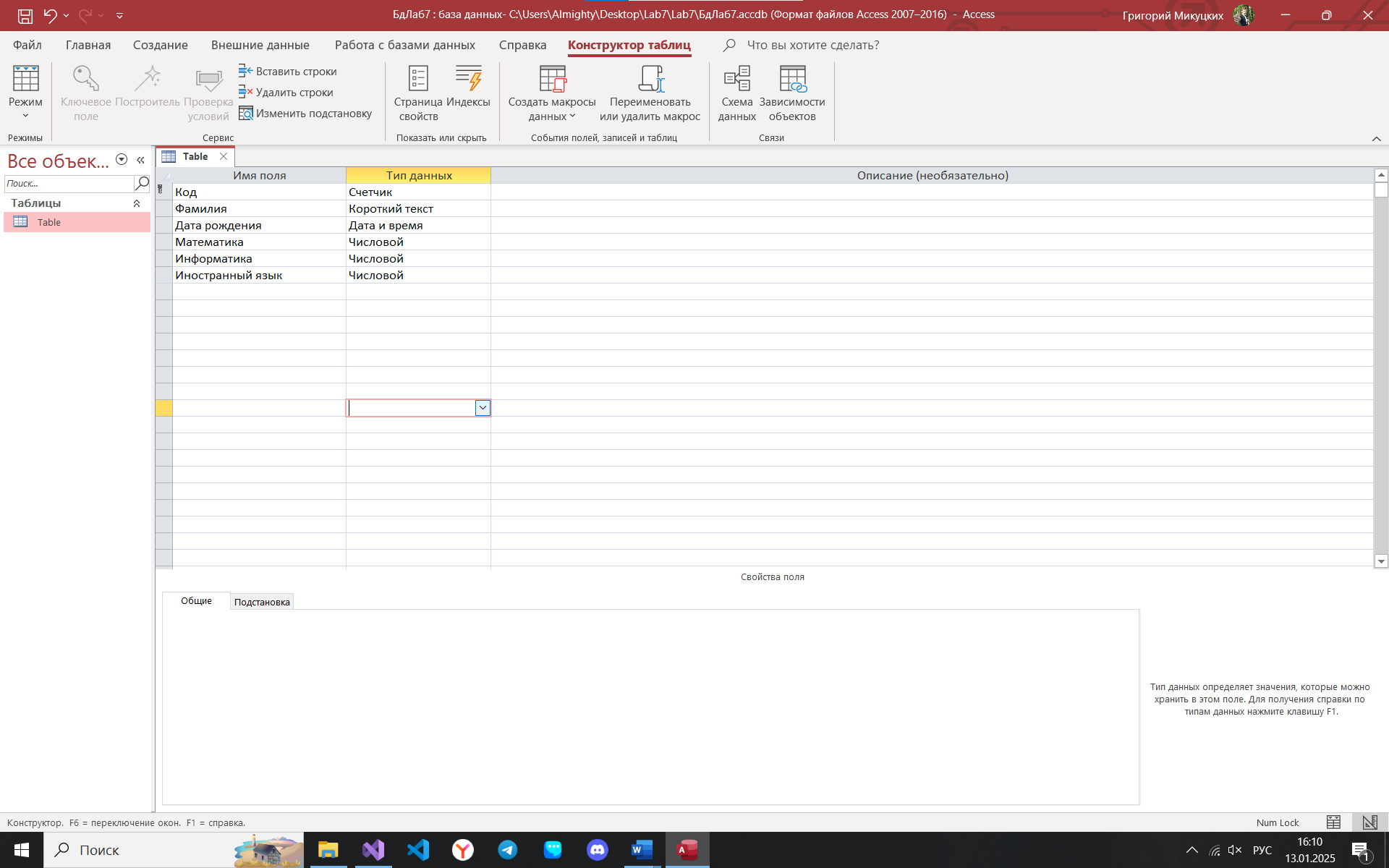


Рис. 2. Типы данных полей

Программа написана на ЯВУ C# и использует шаблон проекта «Windows Forms (.Net Framework 4.7.2)» (рис. 3).

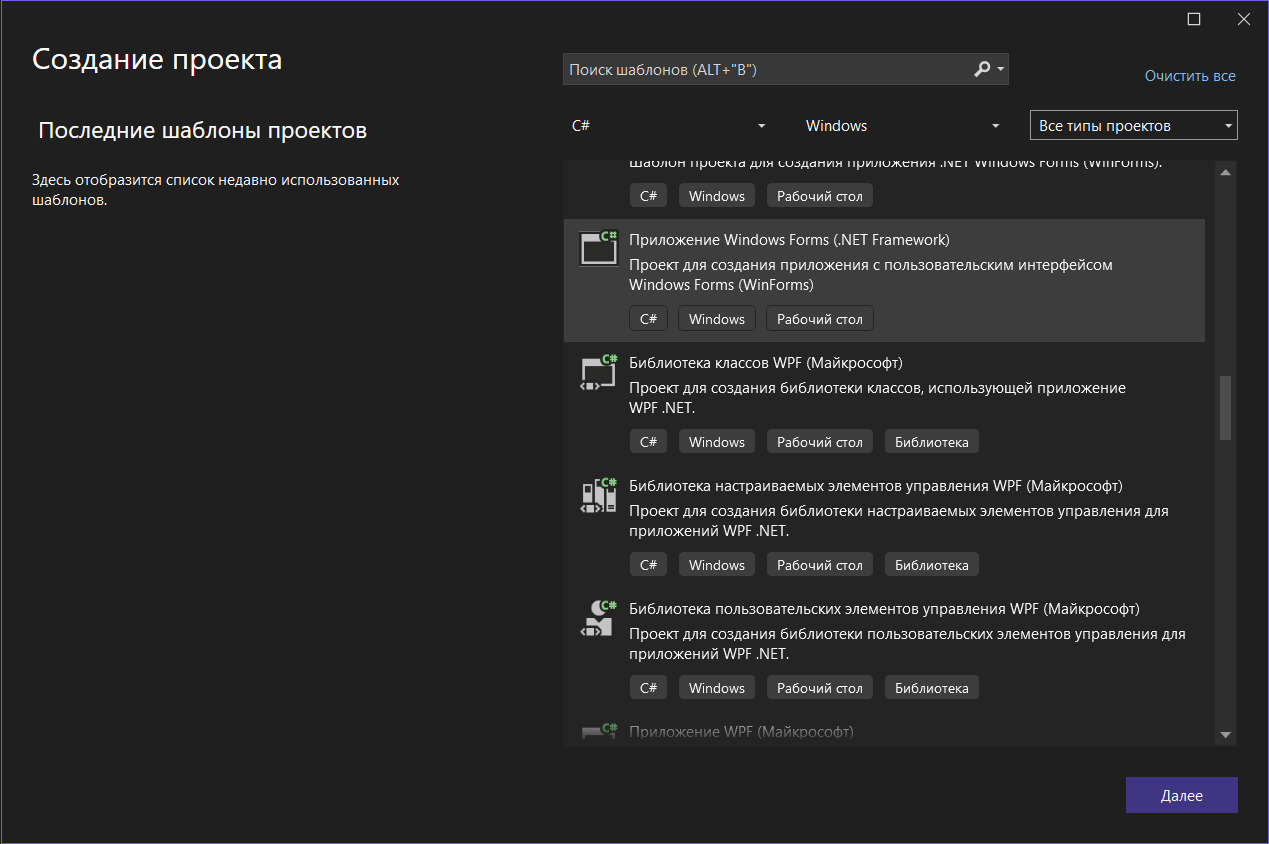


Рис. 3. Выбор шаблона «Windows Forms (.Net Framework)»

Прежде чем начать говнокодить, необходимо добавить в форму объект «DataGridView» и подключить к нему БД Access (рис. 4-12).

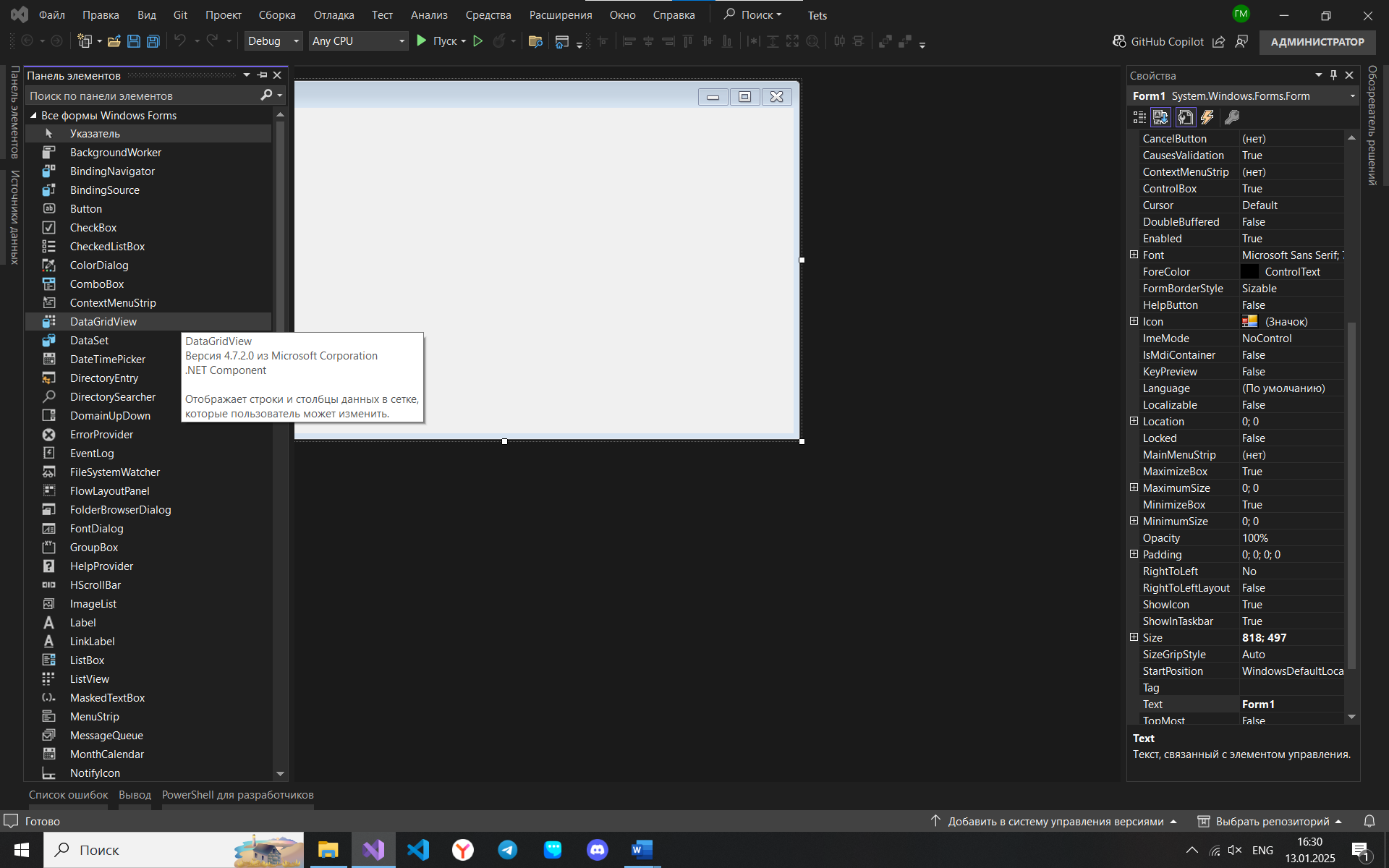


Рис. 4. «Вид» → «Панель элементов» → «DataGridView»

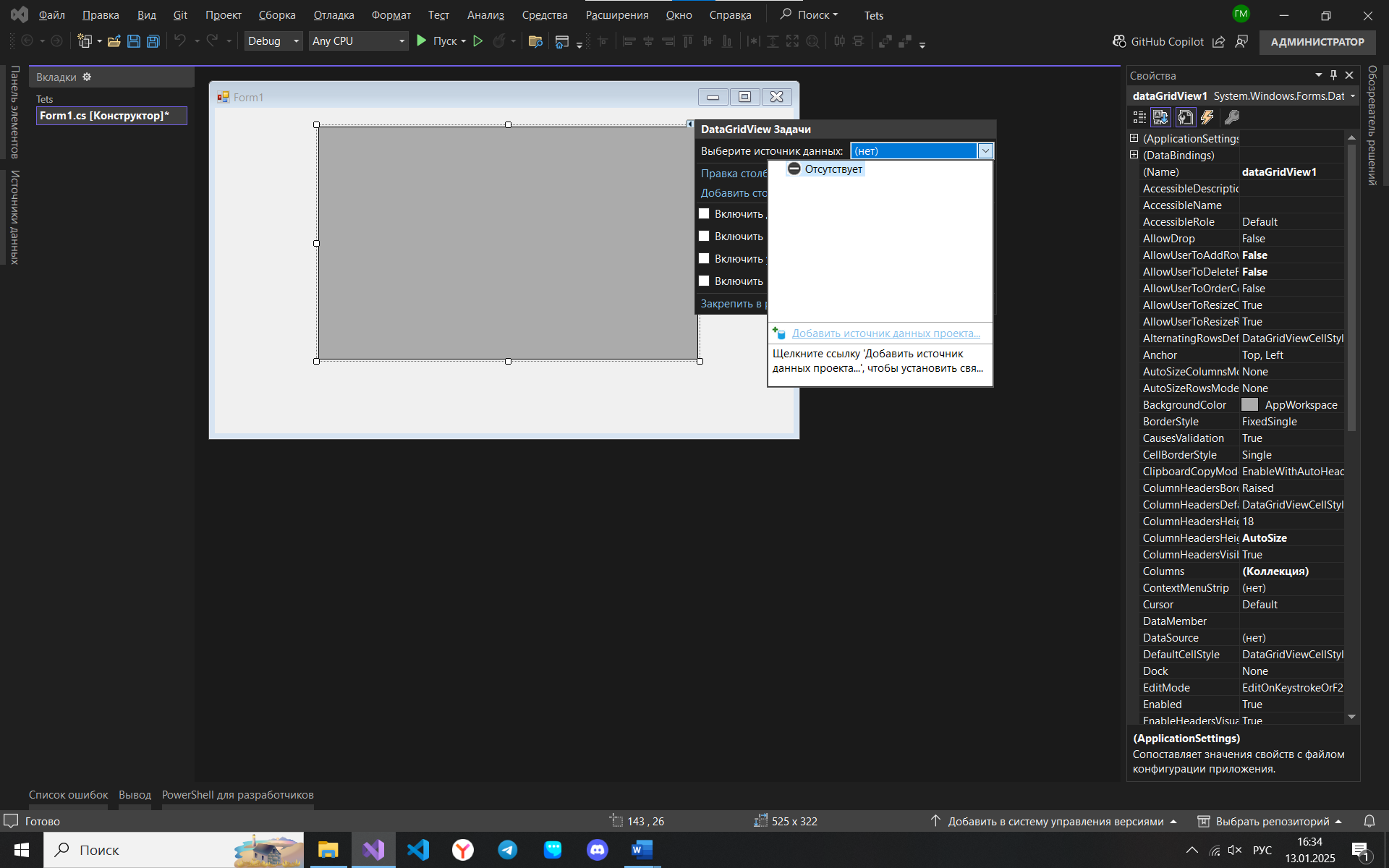


Рис. 5. «Выберите источник данных» → «Добавить источник данных…»

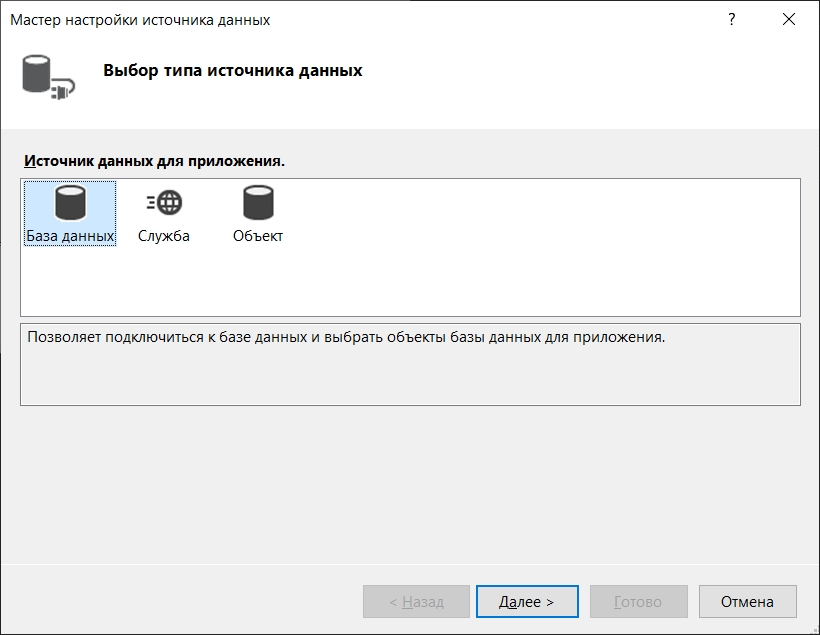


Рис. 6. «База данных»

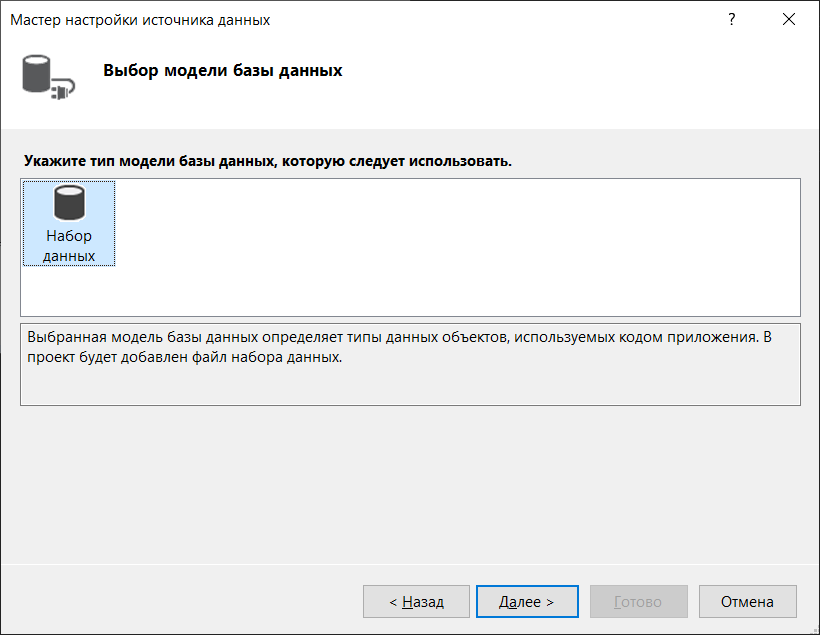


Рис. 7. «Набор данных»

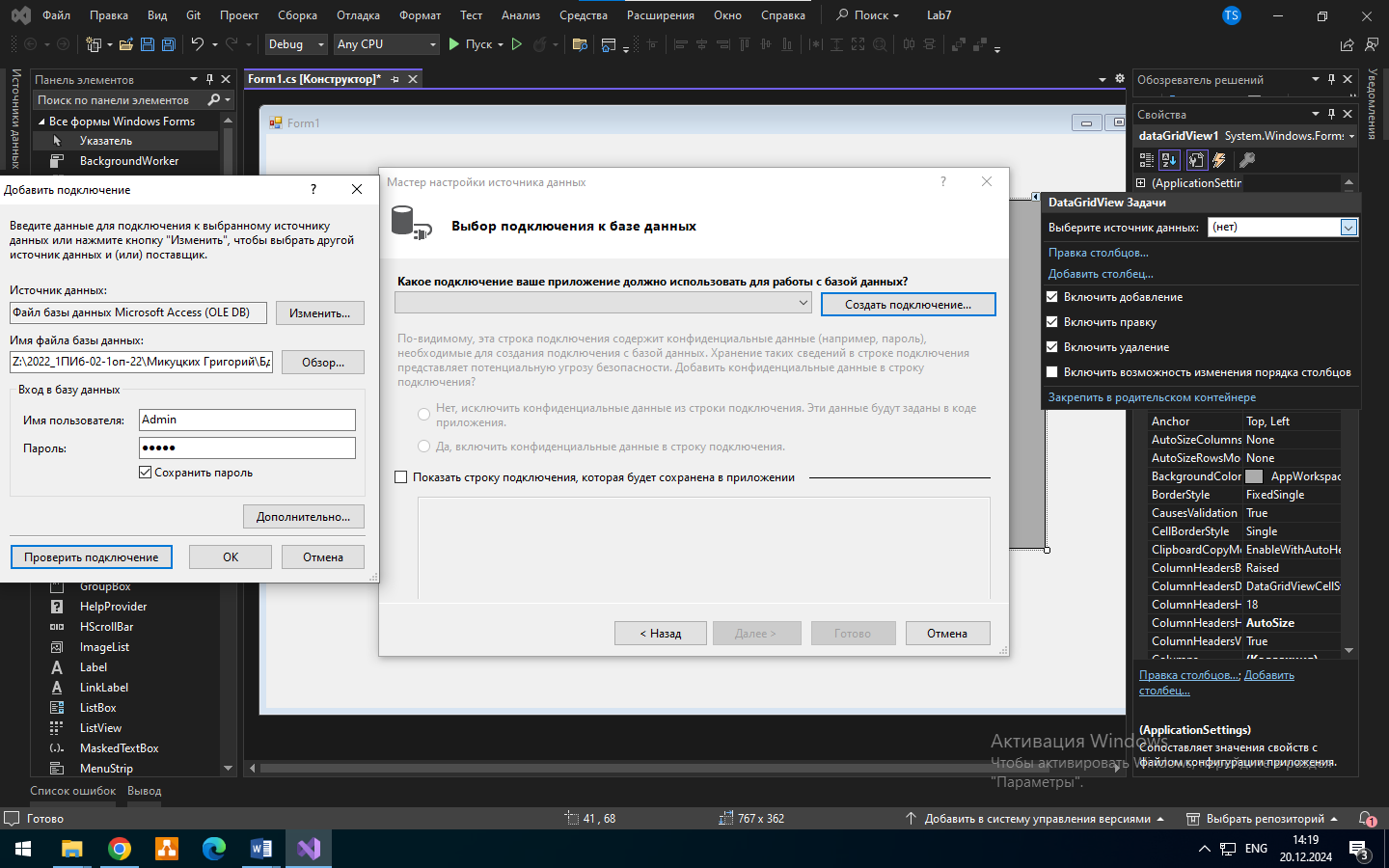


Рис. 8. «Создать подключение» → Файл БД MA (OLE DB) → Указание пути

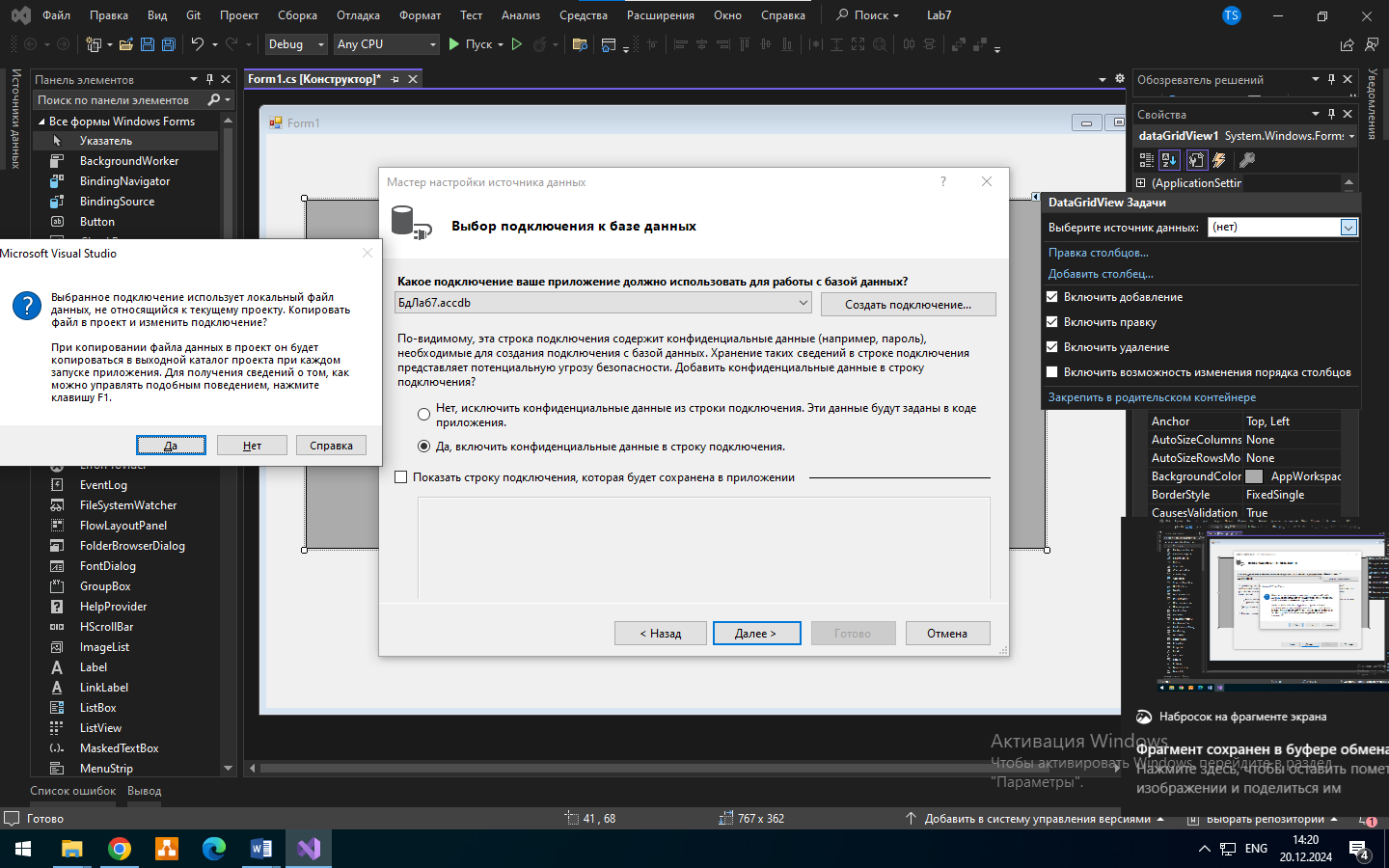


Рис. 9. Подтверждение копирования

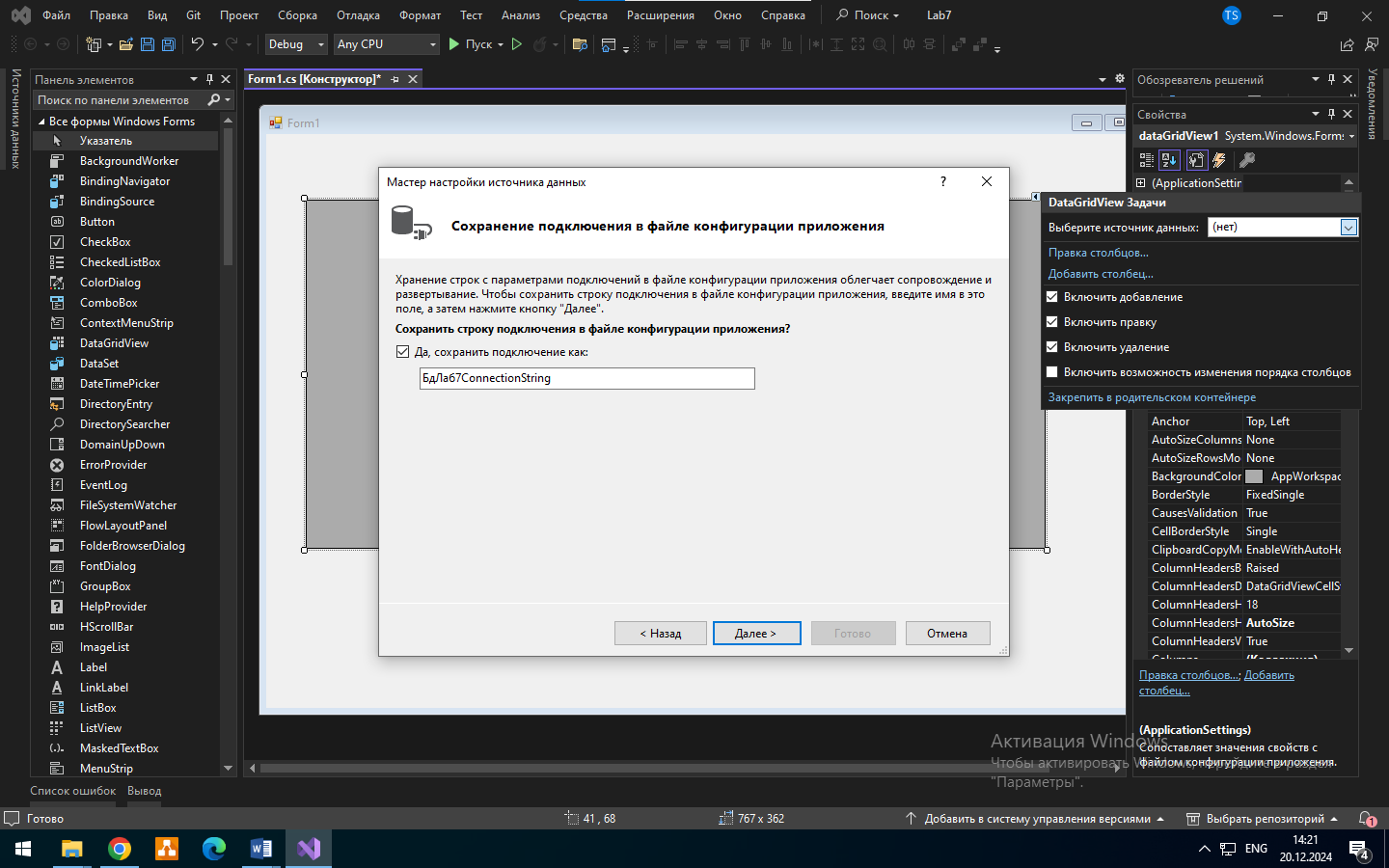


Рис. 10. Создание файла конфигурации

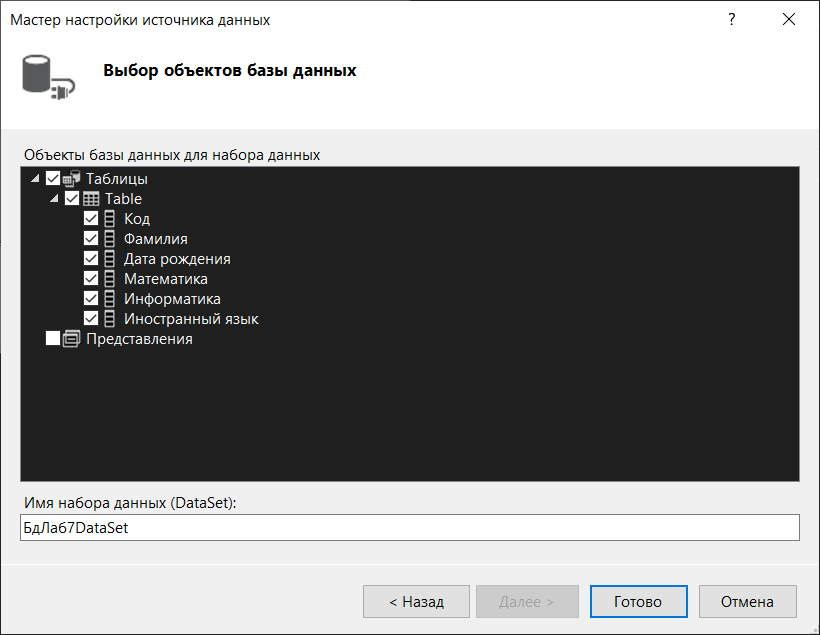


Рис. 11. Выбор всей таблицы БД для работы

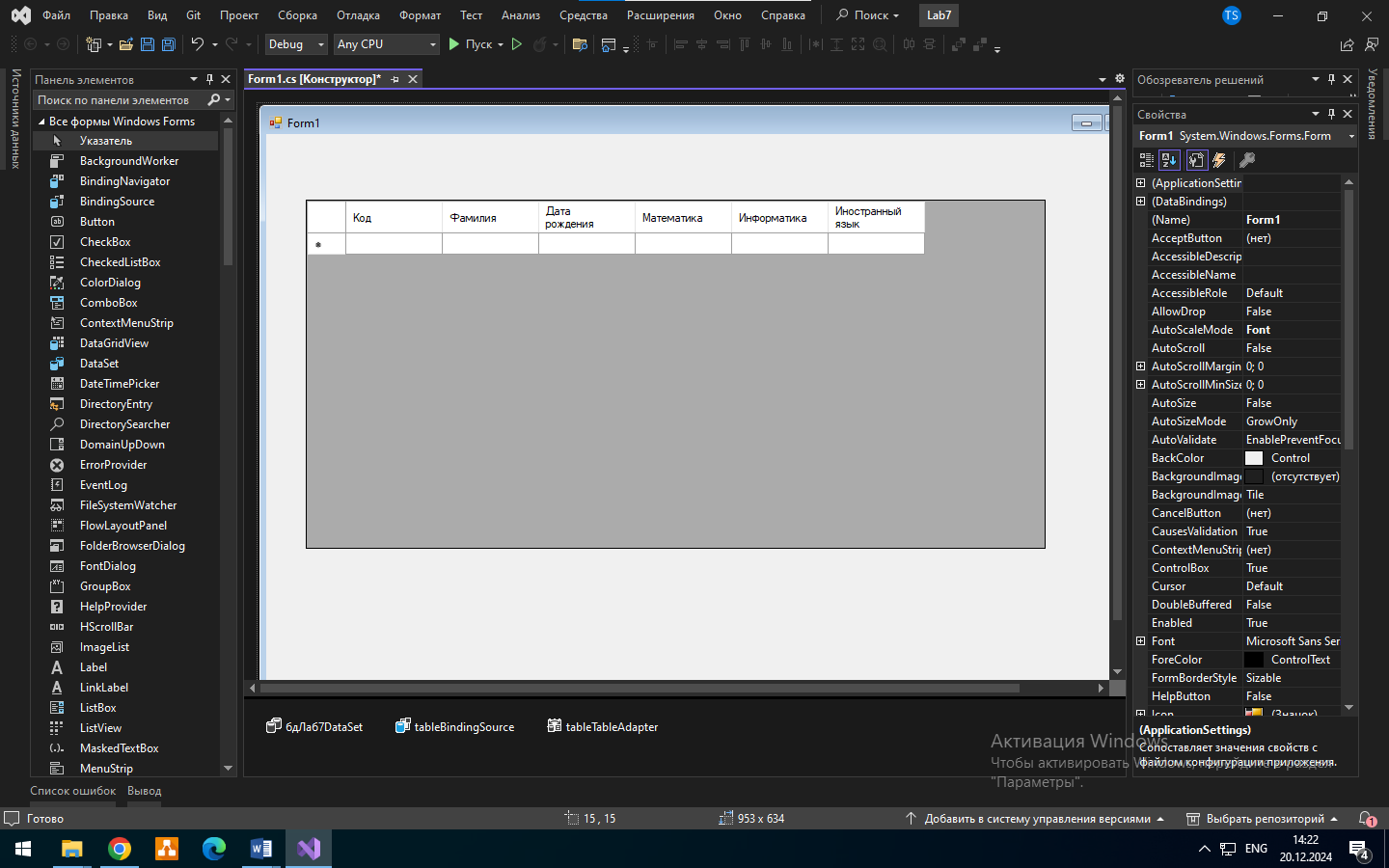


Рис. 12. БД Access подключена

Для взаимодействия с БД был создан интерфейс с помощью «Панели элементов» (рис. 13).

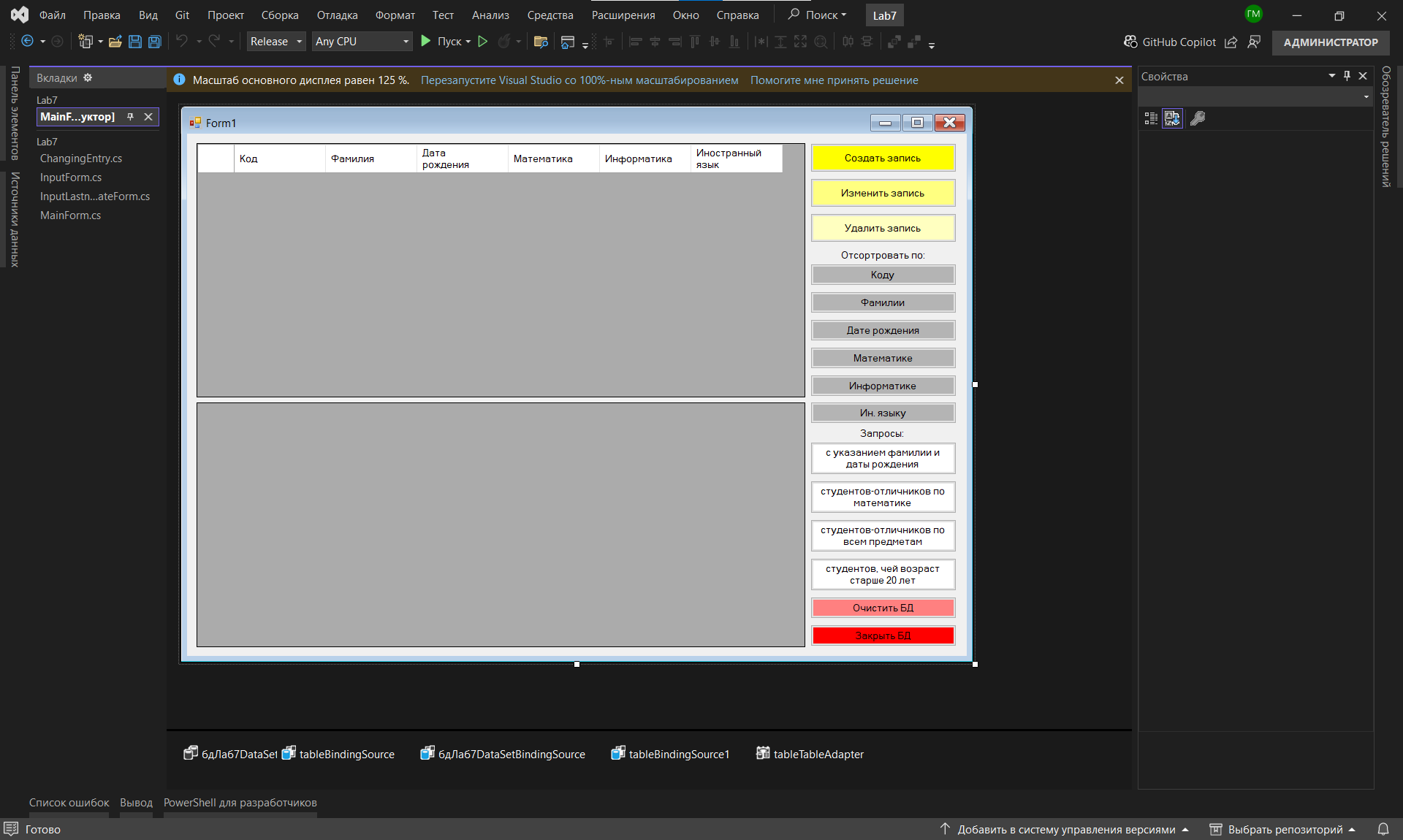


Рис. 13. Общий вид интерфейса приложения

Для корректной работы программы необходимо выбрать предпочтительной 32-разрядную версию ЦП (рис. 14).

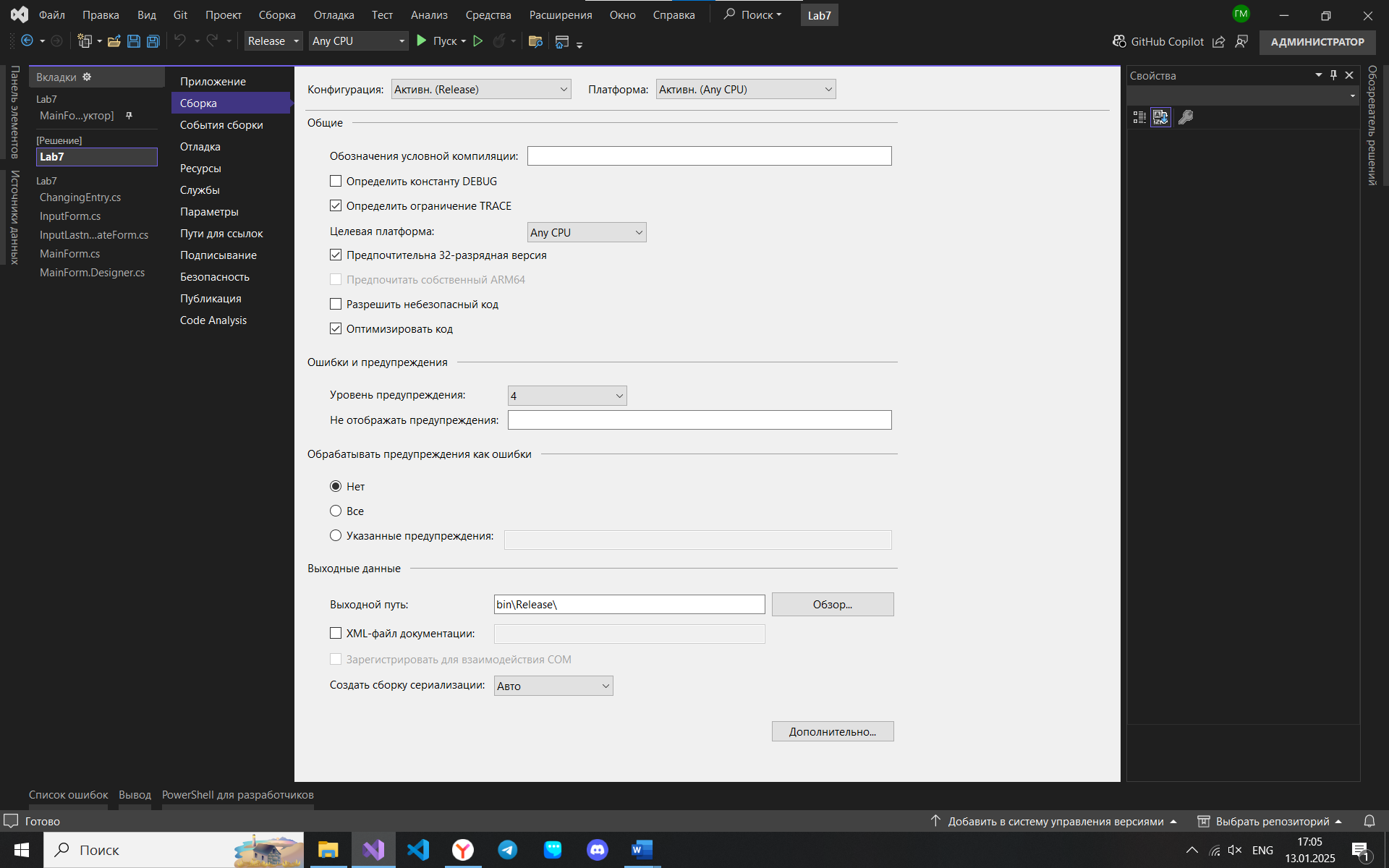


Рис. 14. «Проект» → «Свойства <Имя>»

В свойствах источника данных необходимо разрешить сохранение изменений после завершения работы с БД (рис. 15-16).

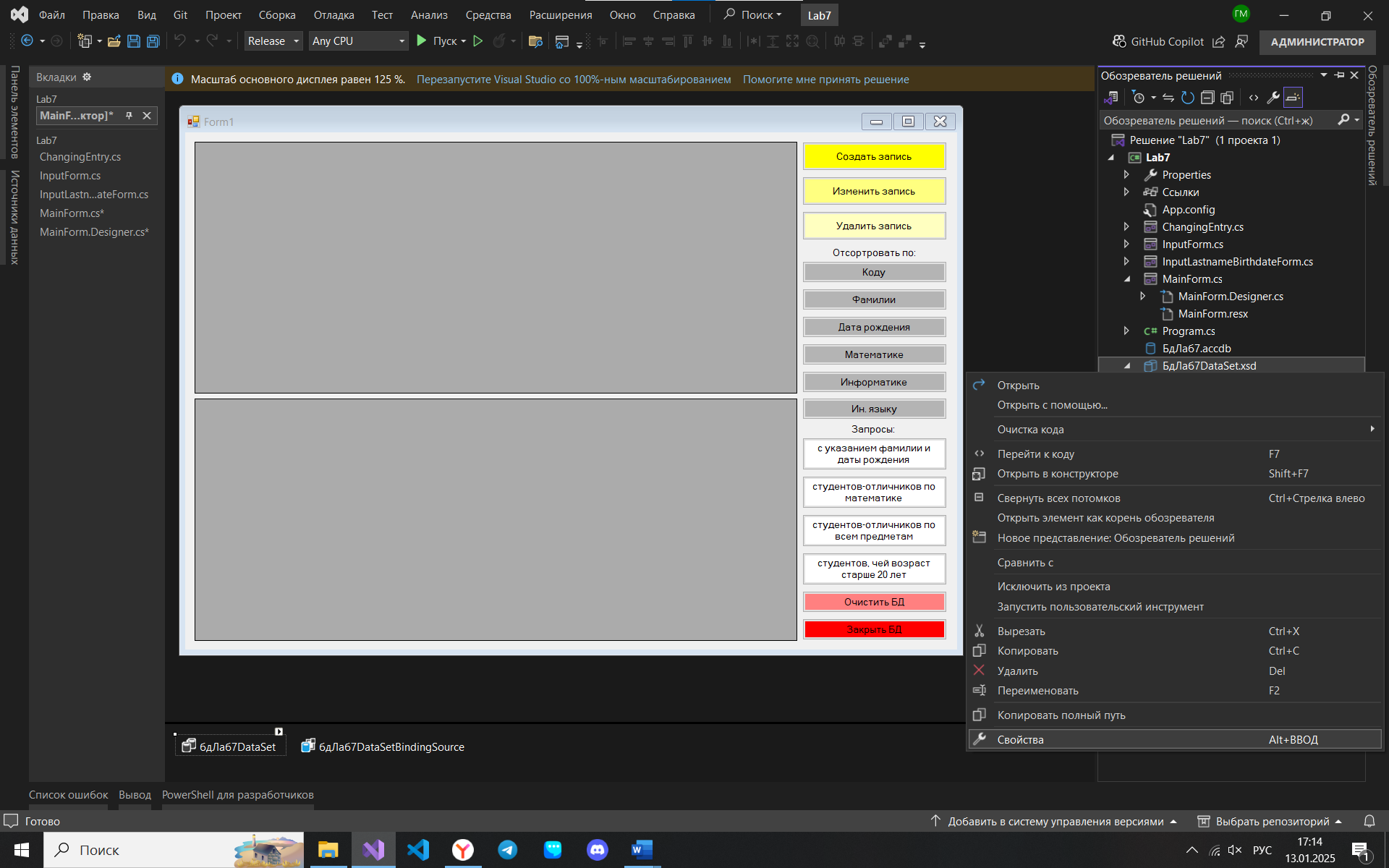


Рис. 15. Обозреватель решений → ПКМ по <имя>.xsd → «Свойства»

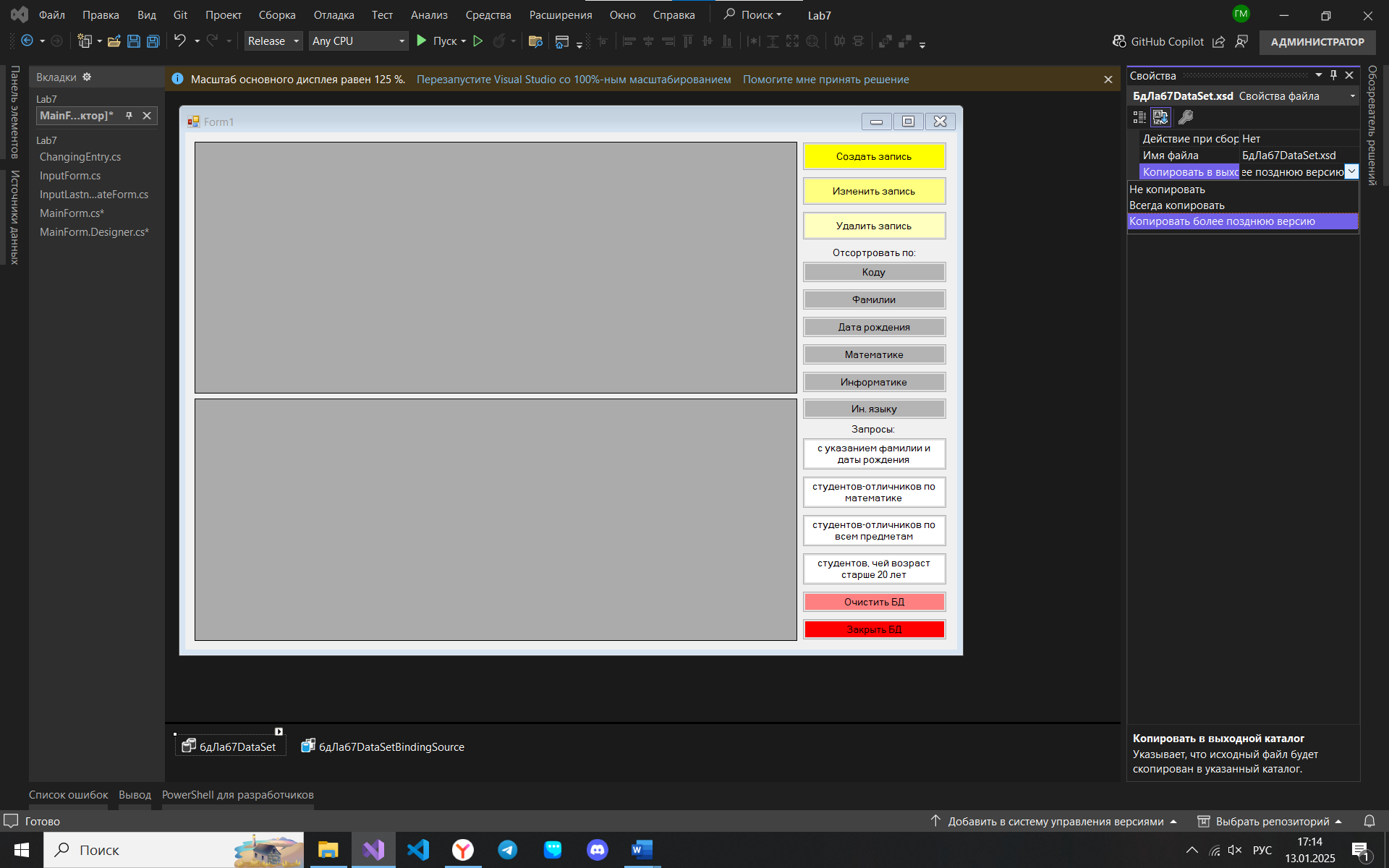


Рис. 16. «Копировать в выходной каталог» → «Копировать позднюю версию»

Часть II.

1. Внесение/изменение/удаление записей в таблицу/е.

Запрос для вывода всех записей БД: SELECT \* FROM [Table].

Запрос для создания записи: INSERT INTO [Table] ([Фамилия], [Дата рождения], [Математика], [Информатика], [Иностранный язык]) VALUES (@lastName, @birthdate, @math, @info, @forlan).

Процесс создания записей БД показан на рис. 17-18.

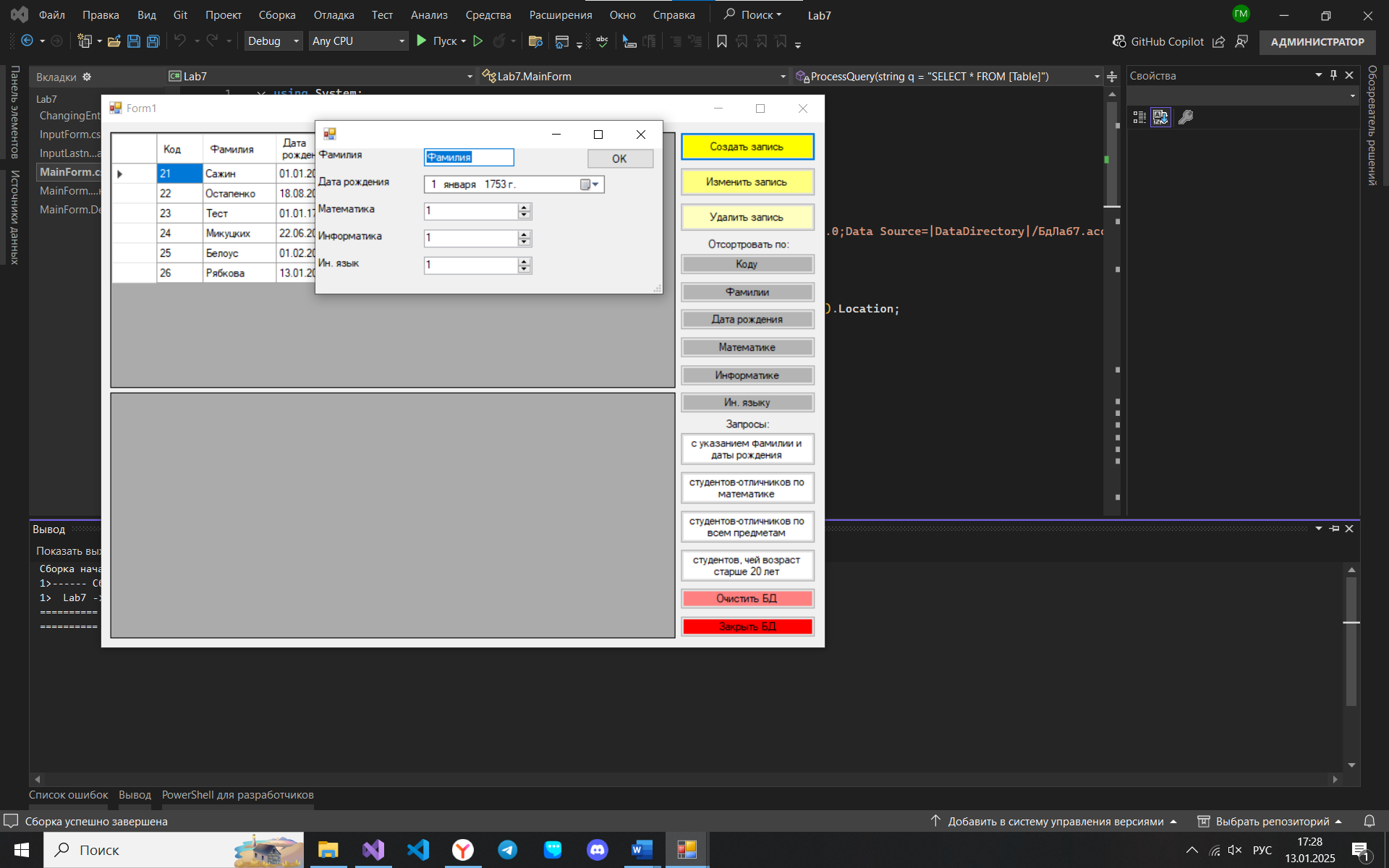


Рис. 17. Окно ввода записи в БД

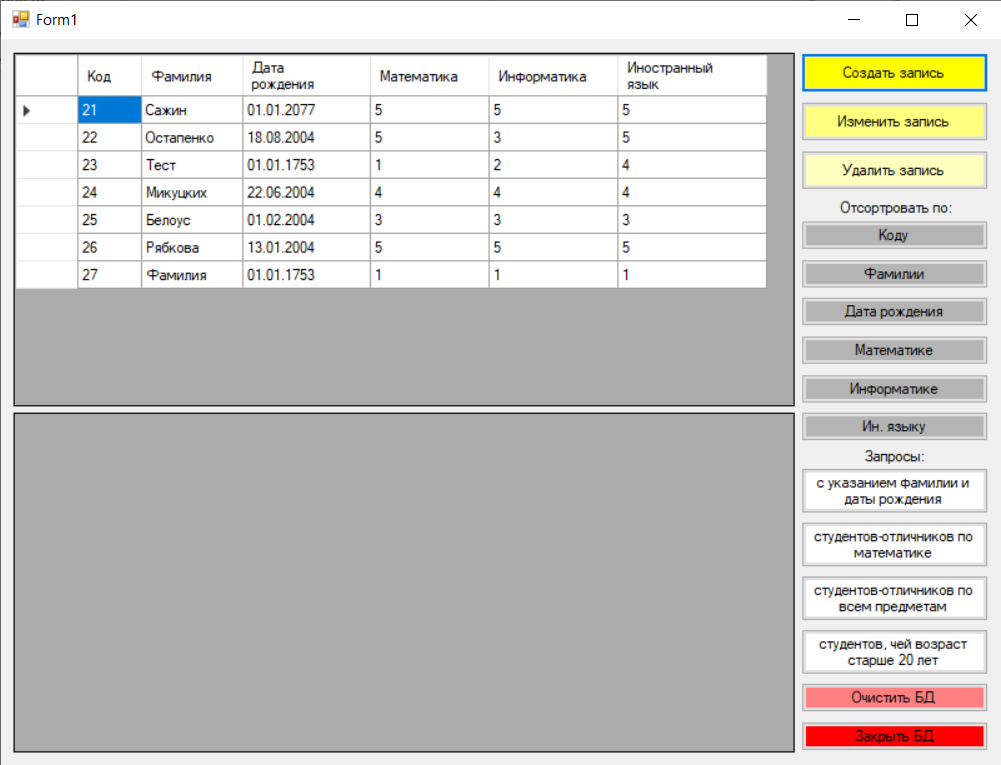


Рис. 18. Тестовая БД создана

Запрос изменения записи: SELECT \* FROM [Table] WHERE [Код] = @code.

Изменение записи требует ввода Кода студента и данных полей (рис. 19-20).

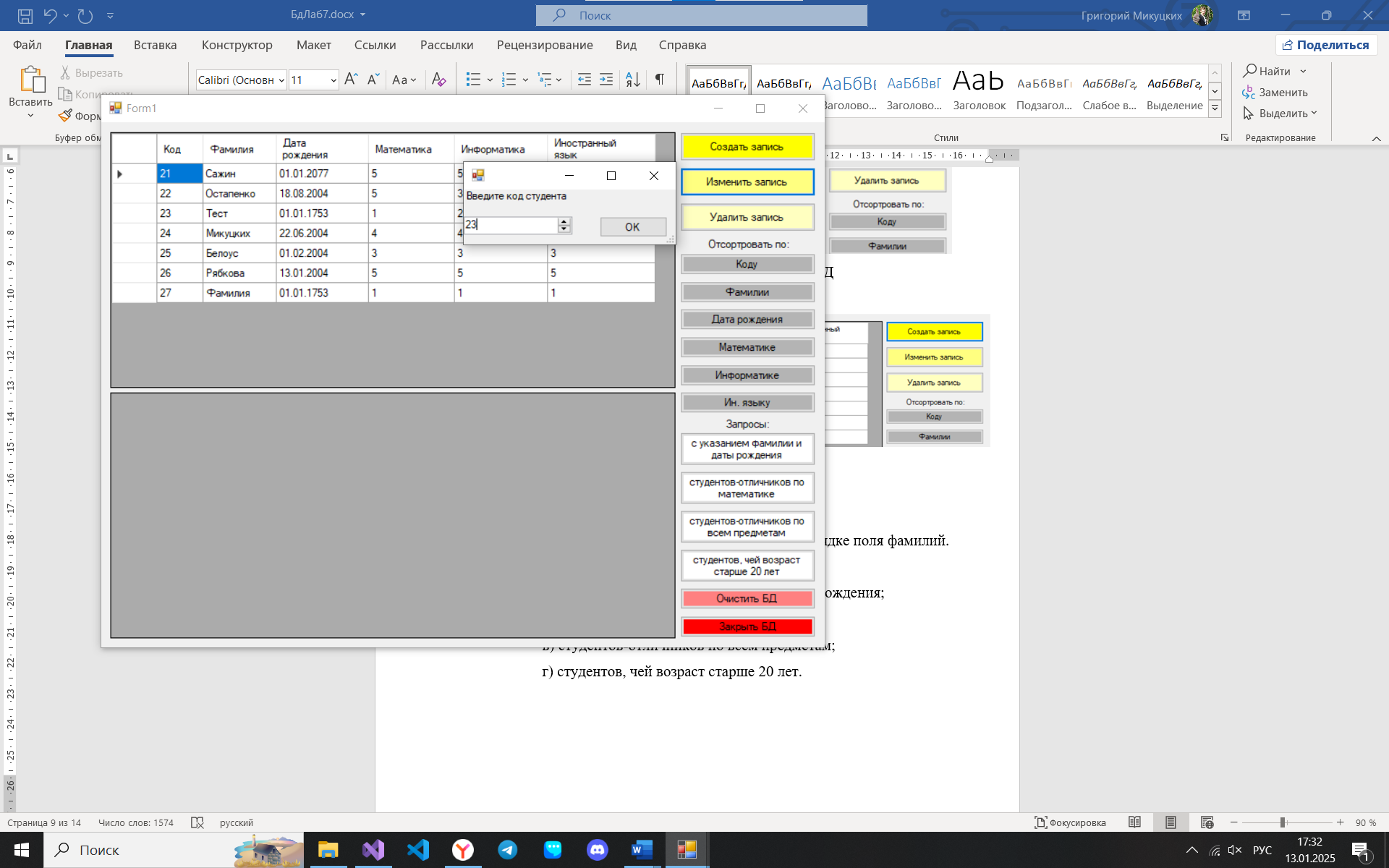


Рис. 19. Окно ввода Кода студента

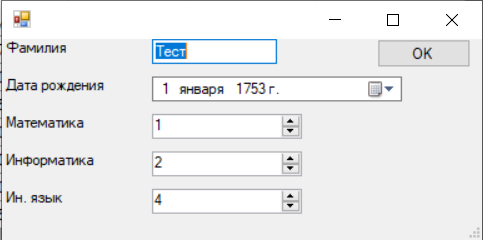


Рис. 20. Окно изменения данных записи

Запрос для удаления записи: DELETE FROM [Table] WHERE [Код] = @code.

Удаление записи требует ввода Кода (см. рис. 19), после запись удаляется.

2. Сортировка записей в алфавитном порядке поля фамилий (рис. 21).

Запрос: SELECT \* FROM [Table] ORDER BY [Фамилия].

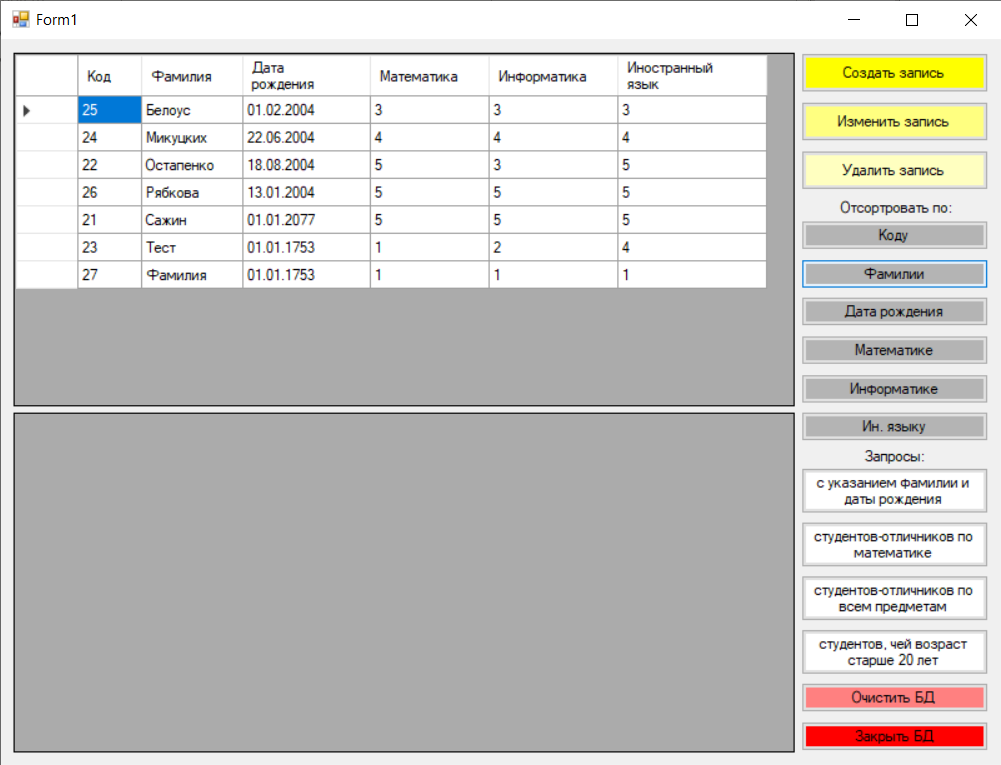


Рис. 21. БД, отсортированная по полю «Фамилия»

3. Сформируйте запросы на выборку:

а) студентов с указанием фамилии и даты рождения (рис. 22);

Запрос: SELECT \* FROM [Table] WHERE [Фамилия] = @lastname AND [Дата рождения] = @birthdate.

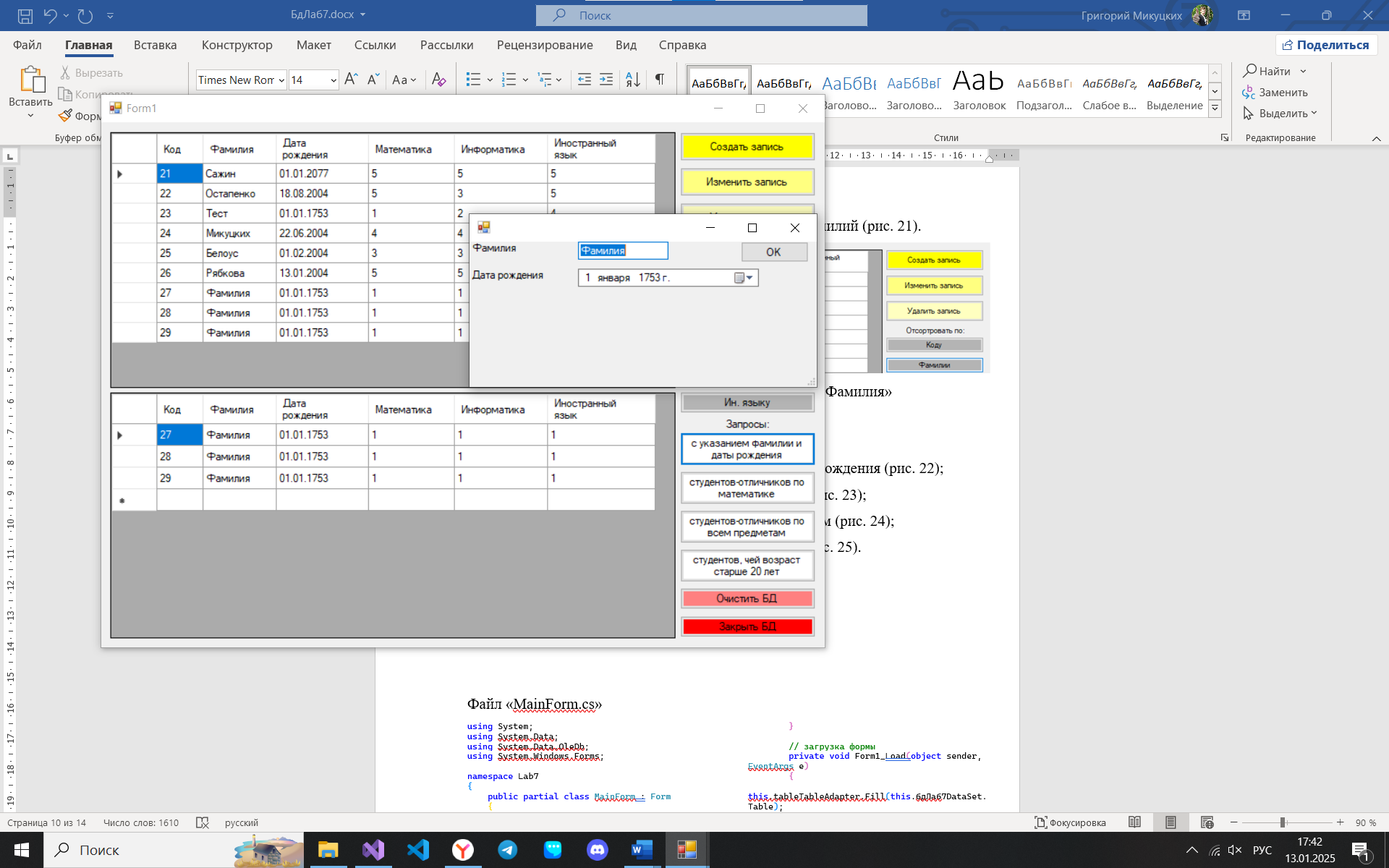


Рис. 22. Результат выборки студентов с указанием фамилии и даты рождения

б) студентов-отличников по математике (рис. 23);

Запрос: SELECT [Фамилия], [Математика] FROM [Table] WHERE [Математика] = 5.

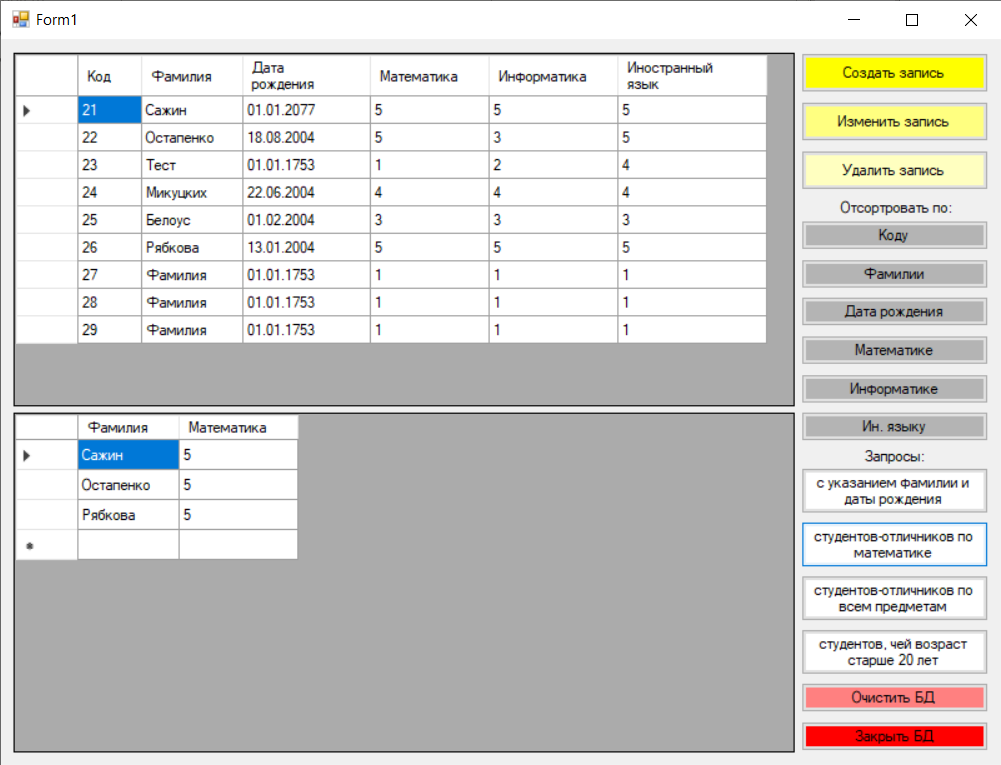


Рис. 23. Результат выборки студентов-отличников по математике

в) студентов-отличников по всем предметам (рис. 24);

Запрос: SELECT [Фамилия], [Математика], [Информатика], [Иностранный язык] FROM [Table] WHERE [Математика] = 5 AND [Информатика] = 5 AND [Иностранный язык] = 5.

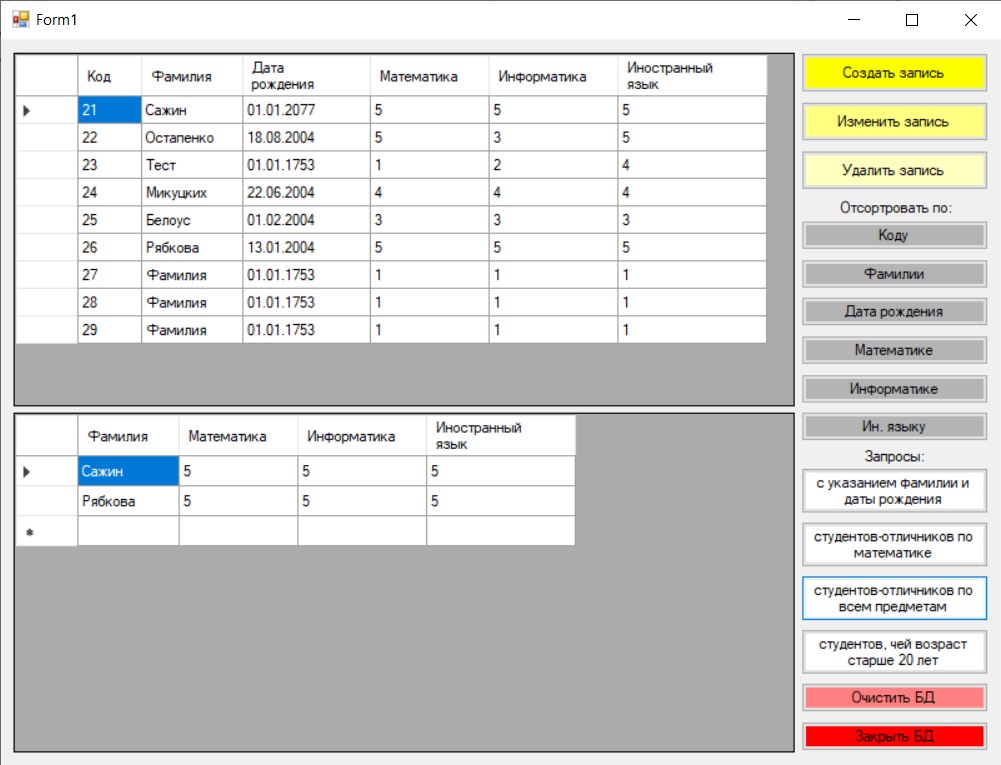
****

Рис. 24. Результат выборки студентов-отличников по всем предметам

г) студентов, чей возраст старше 20 лет (рис. 25).

Запрос: SELECT [Фамилия], [Дата рождения] FROM [Table] WHERE [Дата рождения] > 20.

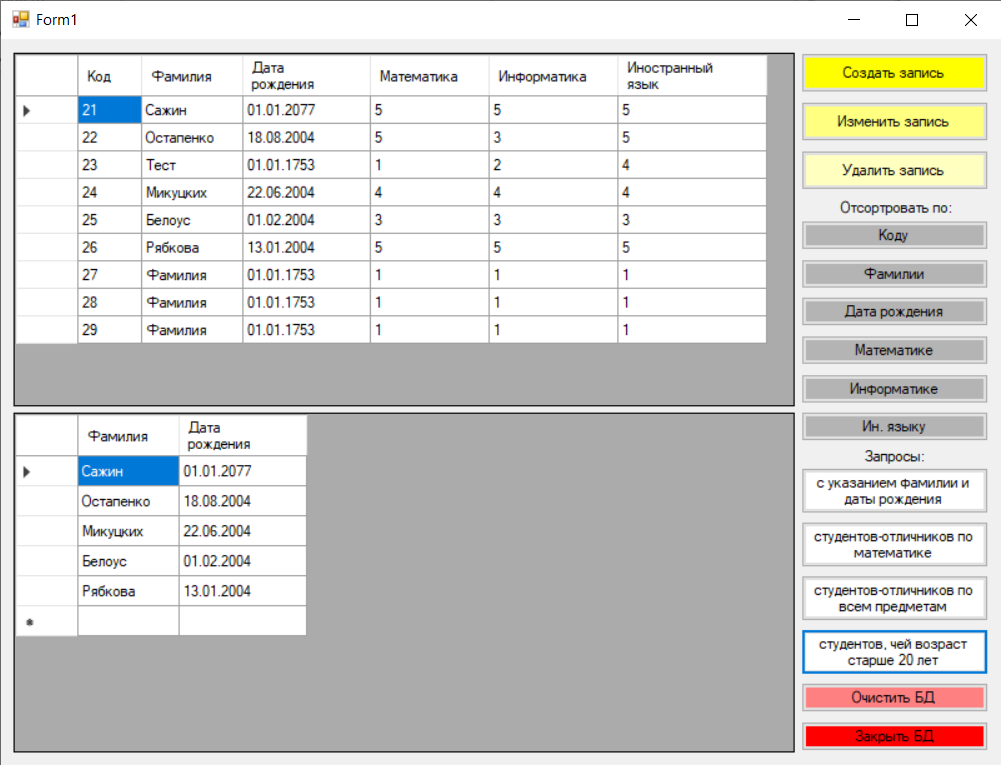


Рис. 25. Результат выборки студентов, чей возраст старше 20 лет

Результат работы можно увидеть, открыв БД в Microsoft Access (рис. 26).

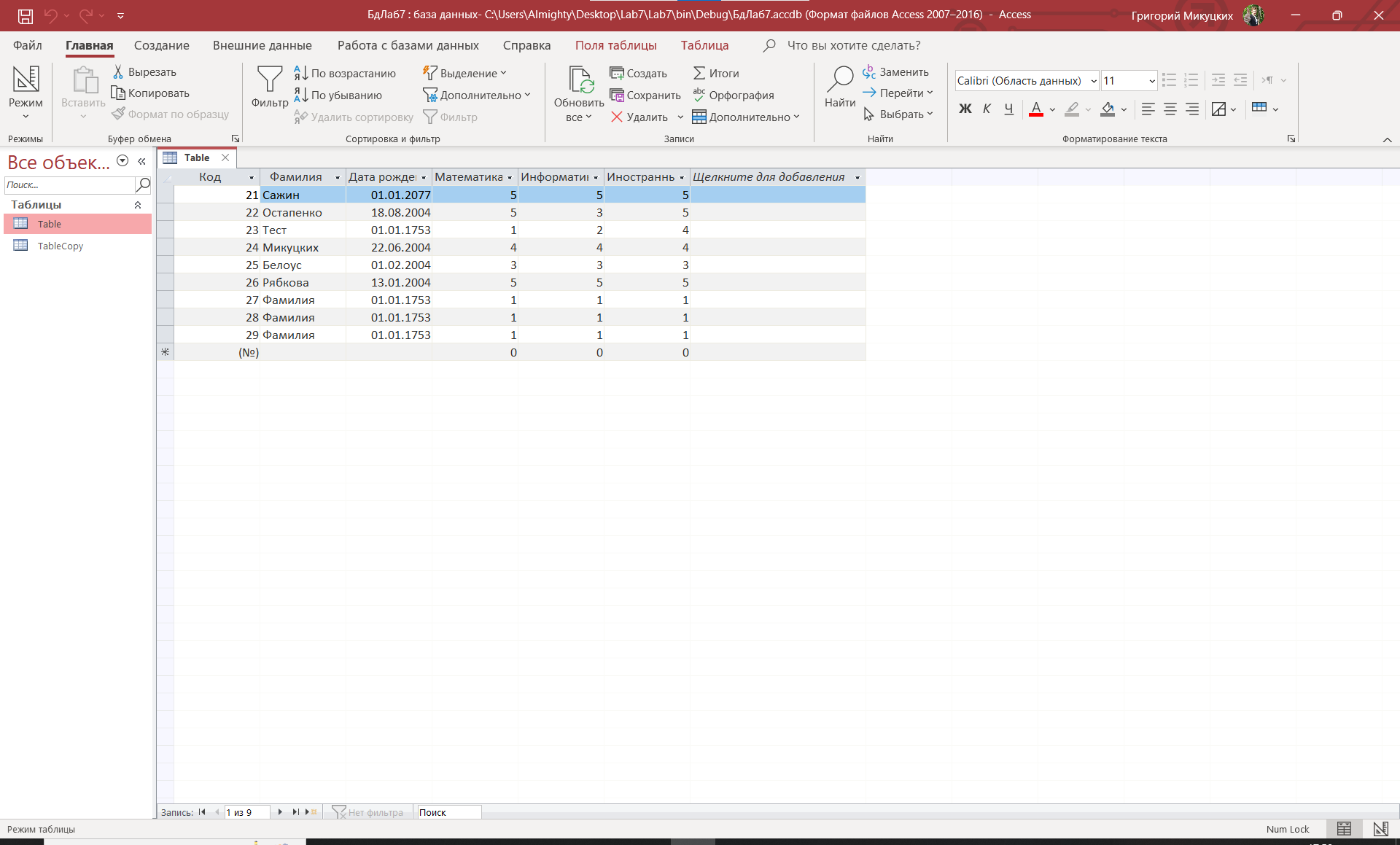


Рис. 26. Результат работы в приложении

Текст программы

Файл «MainForm.cs»

using System;

using System.Data;

using System.Data.OleDb;

using System.Windows.Forms;

namespace Lab7

{

public partial class MainForm : Form

{

public static string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=|DataDirectory|/БдЛаб7.accdb;";

private OleDbConnection \_myConnection;

public MainForm()

{

string executable = System.Reflection.Assembly.GetExecutingAssembly().Location;

string path = (System.IO.Path.GetDirectoryName(executable));

AppDomain.CurrentDomain.SetData("DataDirectory", path);

InitializeComponent();

\_myConnection = new OleDbConnection(connectionString);

}

// загрузка формы

private void Form1\_Load(object sender, EventArgs e)

{

this.tableTableAdapter.Fill(this.бдЛаб7DataSet.Table);

\_myConnection.Open();

ProcessQuery();

}

// закрытие формы

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

{

if (\_myConnection != null && \_myConnection.State == ConnectionState.Open)

{

this.tableTableAdapter.Update(this.бдЛаб7DataSet.Table);

\_myConnection.Close();

}

}

private void ProcessQuery(string q = "SELECT \* FROM [Table]")

{

OleDbDataAdapter adapter = new OleDbDataAdapter(q, \_myConnection);

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

dataGridView1.DataSource = dataTable;

dataGridView1.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.AllCells);

}

// создание записи

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

{

var inputForm = new InputForm();

if (inputForm.ShowDialog() != DialogResult.OK) return;

string query = "INSERT INTO [Table] ([Фамилия], [Дата рождения], [Математика], [Информатика]," +

"[Иностранный язык]) VALUES (@lastName, @birthdate, @math, @info, @forlan)";

OleDbCommand command = new OleDbCommand(query, \_myConnection);

command.Parameters.AddWithValue("@lastName", inputForm.GetLastname);

command.Parameters.AddWithValue("@birthdate", inputForm.GetBirthdate.ToOADate());

command.Parameters.AddWithValue("@math", inputForm.GetMath);

command.Parameters.AddWithValue("@info", inputForm.GetInfo);

command.Parameters.AddWithValue("@forlan", inputForm.GetForlan);

command.ExecuteNonQuery();

ProcessQuery();

}

// изменение записи

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

{

var changingEntry = new ChangingEntry();

if (changingEntry.ShowDialog() != DialogResult.OK) return;

string queryReader = "SELECT \* FROM [Table] WHERE [Код] = @code";

OleDbCommand commandReader = new OleDbCommand(queryReader, \_myConnection);

commandReader.Parameters.AddWithValue("@code", changingEntry.GetCode);

if (!reader.HasRows)

{

DialogResult error = MessageBox.Show("Такого кода не существует", "Ошибка ввода", MessageBoxButtons.OK);

if (error == DialogResult.OK) return;

}

reader.Read();

var lr2 = reader.GetString(1);

var br2 = reader.GetDateTime(2);

var m2 = reader.GetByte(3);

var i2 = reader.GetByte(4);

var fl2 = reader.GetByte(5);

reader.Close();

var inputForm2 = new InputForm(lr2, br2, m2, i2, fl2);

if (inputForm2.ShowDialog() != DialogResult.OK) return;

string query = $"UPDATE [Table] SET [Фамилия] = @lastName, [Дата рождения] = @birthdate," +

$"[Математика] = @math, [Информатика] = @info, [Иностранный язык] = @forlan WHERE [Код] = @code";

OleDbCommand command = new OleDbCommand(query, \_myConnection);

command.Parameters.AddWithValue("@lastName", inputForm2.GetLastname);

command.Parameters.AddWithValue("@birthdate", inputForm2.GetBirthdate.ToOADate());

command.Parameters.AddWithValue("@math", inputForm2.GetMath);

command.Parameters.AddWithValue("@info", inputForm2.GetInfo);

command.Parameters.AddWithValue("@forlan", inputForm2.GetForlan);

command.Parameters.AddWithValue("@code", changingEntry.GetCode);

command.ExecuteNonQuery();

ProcessQuery();

}

// удаление записи

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

{

var deleteEntry = new ChangingEntry();

if (deleteEntry.ShowDialog() != DialogResult.OK) return;

string query = "DELETE FROM [Table] WHERE [Код] = @code";

OleDbCommand command = new OleDbCommand(query, \_myConnection);

command.Parameters.AddWithValue("@code", deleteEntry.GetCode);

command.ExecuteNonQuery();

ProcessQuery();

}

// сортровка БД

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

{

ProcessQuery("SELECT \* FROM [Table] ORDER BY [Код]");

}

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

{

ProcessQuery("SELECT \* FROM [Table] ORDER BY [Фамилия]");

}

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

{

ProcessQuery("SELECT \* FROM [Table] ORDER BY [Дата рождения]");

}

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

{

ProcessQuery("SELECT \* FROM [Table] ORDER BY [Математика]");

}

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

{

ProcessQuery("SELECT \* FROM [Table] ORDER BY [Информатика]");

}

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

{

ProcessQuery("SELECT \* FROM [Table] ORDER BY [Иностранный язык]");

}

private void ProcessSelectionQuery(string q = "SELECT \* FROM [Table]")

{

OleDbDataAdapter adapter = new OleDbDataAdapter(q, \_myConnection);

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

dataGridView2.DataSource = dataTable;

dataGridView2.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.AllCells);

}

// запрос студентов с указанием фамилии и даты рождения

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

{

var inputLastnameBirthdateForm = new InputLastnameBirthdateForm();

if (inputLastnameBirthdateForm.ShowDialog() != DialogResult.OK) return;

string query = "SELECT \* FROM [Table] WHERE [Фамилия] = @lastname AND [Дата рождения] = @birthdate";

OleDbCommand command = new OleDbCommand(query, \_myConnection);

command.Parameters.AddWithValue("@lastName", inputLastnameBirthdateForm.GetLastname);

command.Parameters.AddWithValue("@birthdate", inputLastnameBirthdateForm.GetBirthdate.ToOADate());

OleDbDataAdapter adapter = new OleDbDataAdapter(query, \_myConnection);

DataTable dataTable = new DataTable();

adapter.SelectCommand = command;

adapter.Fill(dataTable);

dataGridView2.DataSource = dataTable;

dataGridView2.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.AllCells);

}

// запрос студентов-отличников по математике

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

{

ProcessSelectionQuery("SELECT [Фамилия], [Математика] FROM [Table] WHERE [Математика] = 5");

}

// запрос студентов-отличников по всем предметам

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

{

ProcessSelectionQuery("SELECT [Фамилия], [Математика], [Информатика], [Иностранный язык] FROM [Table]" +

"WHERE [Математика] = 5 AND [Информатика] = 5 AND [Иностранный язык] = 5");

}

// запрос студентов, чей возраст старше 20 лет

private void queryStudentsAgeMore20\_Click(object sender, EventArgs e)

{

ProcessSelectionQuery("SELECT [Фамилия], [Дата рождения] FROM [Table] WHERE [Дата рождения] > 20");

}

// очистка БД

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

{

ProcessQuery("DELETE \* FROM [Table]");

}

// закрыть БД и форму

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

{

this.tableTableAdapter.Update(this.бдЛаб7DataSet.Table);

\_myConnection.Close();

this.Close();

}

// выравнивание по ширине всей таблицы

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

dataGridView1.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.AllCells);

}

private void dataGridView2\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

dataGridView2.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.AllCells);

}

}

}

Файл « InputForm.cs»

using System;

using System.Windows.Forms;

namespace Lab7

{

internal class InputForm : Form

{

private TextBox \_lastname = new TextBox();

private DateTimePicker \_birthdate = new DateTimePicker();

private NumericUpDown \_math = new NumericUpDown();

private NumericUpDown \_info = new NumericUpDown();

private NumericUpDown \_forlan = new NumericUpDown();

private Label \_lnl = new Label();

private Label \_bdl = new Label();

private Label \_ml = new Label();

private Label \_il = new Label();

private Label \_fl = new Label();

public InputForm()

{

setSize();

formatFields();

createExitButton();

}

public InputForm(string lastname, DateTime birthdate, byte math, byte info, byte forlan)

{

setSize();

formatFields();

\_lastname.Text = lastname;

\_birthdate.Value = birthdate.Date;

\_math.Value = math;

\_info.Value = info;

\_forlan.Value = forlan;

createExitButton();

}

private void setSize()

{

this.Width = 400;

this.Height = 200;

}

private void formatFields()

{

\_lastname.Left = 120;

\_birthdate.Top = 30; \_birthdate.Left = 120;

\_math.Top = 60; \_math.Left = 120;

\_info.Top = 90; \_info.Left = 120;

\_forlan.Top = 120; \_forlan.Left = 120;

\_lastname.Text = "Фамилия";

\_birthdate.Value = \_birthdate.MinDate;

\_math.Maximum = 5; \_math.Minimum = 1;

\_info.Maximum = 5; \_info.Minimum = 1;

\_forlan.Maximum = 5; \_forlan.Minimum = 1;

\_lnl.Text = "Фамилия";

\_bdl.Text = "Дата рождения"; \_bdl.Top = 30;

\_ml.Text = "Математика"; \_ml.Top = 60;

\_il.Text = "Информатика"; \_il.Top = 90;

\_fl.Text = "Ин. язык"; \_fl.Top = 120;

this.Controls.Add(\_lastname);

this.Controls.Add(\_birthdate);

this.Controls.Add(\_math);

this.Controls.Add(\_info);

this.Controls.Add(\_forlan);

this.Controls.Add(\_lnl);

this.Controls.Add(\_bdl);

this.Controls.Add(\_ml);

this.Controls.Add(\_il);

this.Controls.Add(\_fl);

}

private void createExitButton()

{

var buttonOk = new Button() { Text = "OK", Left = 300 };

this.Controls.Add(buttonOk);

buttonOk.Click += (sender, args) =>

{

this.DialogResult = DialogResult.OK;

this.Close();

};

}

public string GetLastname => \_lastname.Text;

public DateTime GetBirthdate => \_birthdate.Value.Date;

public decimal GetMath => \_math.Value;

public decimal GetInfo => \_info.Value;

public decimal GetForlan => \_forlan.Value;

}

}

Файл «ChangingEntry.cs»

using System.Drawing;

using System.Windows.Forms;

namespace Lab7

{

internal class ChangingEntry : Form

{

NumericUpDown numeric;

public ChangingEntry()

{

this.Width = 250;

this.Height = 100;

var numl = new System.Windows.Forms.Label(); numl.Text = "Введите код студента";

numl.Size = new Size(numl.PreferredWidth, numl.PreferredHeight);

numeric = new NumericUpDown(); numeric.Top = 30;

this.Controls.Add(numl);

this.Controls.Add(numeric);

var buttonOk = new Button() { Text = "OK", Left = 150, Top = 30 };

this.Controls.Add(buttonOk);

buttonOk.Click += (sender, args) => { this.DialogResult = DialogResult.OK; this.Close(); };

}

public decimal GetCode => numeric.Value;

}

}

Файл «InputLastnameBirthdateForm.cs»

using System.Windows.Forms;

using System;

namespace Lab7

{

internal class InputLastnameBirthdateForm : Form

{

private TextBox \_lastname = new TextBox();

private DateTimePicker \_birthdate = new DateTimePicker();

private Label \_lnl = new Label();

private Label \_bdl = new Label();

public InputLastnameBirthdateForm()

{

setSize();

formatFields();

createExitButton();

}

private void setSize()

{

this.Width = 400;

this.Height = 200;

this.CenterToScreen();

}

private void formatFields()

{

\_lastname.Left = 120;

\_lastname.Text = "Фамилия";

\_birthdate.Top = 30; \_birthdate.Left = 120;

\_birthdate.Value = \_birthdate.MinDate;

\_lnl.Text = "Фамилия";

\_bdl.Text = "Дата рождения"; \_bdl.Top = 30;

this.Controls.Add(\_lastname);

this.Controls.Add(\_birthdate);

this.Controls.Add(\_lnl);

this.Controls.Add(\_bdl);

}

private void createExitButton()

{

var buttonOk = new Button() { Text = "OK", Left = 300 };

this.Controls.Add(buttonOk);

buttonOk.Click += (sender, args) =>

{

this.DialogResult = DialogResult.OK;

this.Close();

};

}

public string GetLastname => \_lastname.Text;

public DateTime GetBirthdate => \_birthdate.Value.Date;

}

}