**Главная форма:**

using System;

using System.Drawing;

using System.IO;

using System.Windows.Forms;

using Word = Microsoft.Office.Interop.Word;

namespace Uchet

{

public partial class Main : Form

{

private string User;

public Main(string u)

{

InitializeComponent();

User = u;

}

private string connectionstring;

// = "Data Source=WIN-AIP7CN6UDV7\\SQLEXPRESS;Initial Catalog=Kazakov;Integrated Security=True"

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

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Install". При необходимости она может быть перемещена или удалена.

installTableAdapter.Fill(kazakovDataSet.Install);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.ProgPred". При необходимости она может быть перемещена или удалена.

progPredTableAdapter.Fill(kazakovDataSet.ProgPred);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Sotr". При необходимости она может быть перемещена или удалена.

sotrTableAdapter.Fill(kazakovDataSet.Sotr);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.PC1". При необходимости она может быть перемещена или удалена.

pC1TableAdapter.Fill(kazakovDataSet.PC1);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.PC". При необходимости она может быть перемещена или удалена.

pCTableAdapter.Fill(kazakovDataSet.PC);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.License". При необходимости она может быть перемещена или удалена.

licenseTableAdapter.Fill(kazakovDataSet.License);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Programs". При необходимости она может быть перемещена или удалена.

programsTableAdapter.Fill(kazakovDataSet.Programs);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Prepodav". При необходимости она может быть перемещена или удалена.

prepodavTableAdapter.Fill(kazakovDataSet.Prepodav);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Predmet". При необходимости она может быть перемещена или удалена.

predmetTableAdapter.Fill(kazakovDataSet.Predmet);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Kabs". При необходимости она может быть перемещена или удалена.

kabsTableAdapter.Fill(kazakovDataSet.Kabs);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Teachers". При необходимости она может быть перемещена или удалена.

teachersTableAdapter.Fill(kazakovDataSet.Teachers);

if (User == "Администратор")

{

button1.Visible = true;

button2.Visible = true;

button3.Visible = true;

button4.Visible = true;

button7.Visible = true;

button8.Visible = true;

button9.Visible = true;

button10.Visible = true;

button11.Visible = true;

button12.Visible = true;

tabPage3.Parent = tabControl1;

}

else

{

tabPage3.Parent = null;

}

StreamReader sr = new StreamReader("Uchet.exe.config");

while (!sr.ReadLine().Contains("<connectionStrings>"))

{

sr.ReadLine();

}

sr.ReadLine();

string s = sr.ReadLine();

s = s.Remove(0, 30);

s = s.Remove(s.Length - 1);

connectionstring = s;

}

private int pr = 0;

private void Main\_Activated(object sender, EventArgs e)

{

if (pr > 0)

{

//Обновление всех адаптеров таблиц при каждой фокусировке на форму

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Install". При необходимости она может быть перемещена или удалена.

installTableAdapter.Fill(kazakovDataSet.Install);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.ProgPred". При необходимости она может быть перемещена или удалена.

progPredTableAdapter.Fill(kazakovDataSet.ProgPred);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Sotr". При необходимости она может быть перемещена или удалена.

sotrTableAdapter.Fill(kazakovDataSet.Sotr);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.PC1". При необходимости она может быть перемещена или удалена.

pC1TableAdapter.Fill(kazakovDataSet.PC1);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.PC". При необходимости она может быть перемещена или удалена.

pCTableAdapter.Fill(kazakovDataSet.PC);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.License". При необходимости она может быть перемещена или удалена.

licenseTableAdapter.Fill(kazakovDataSet.License);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Programs". При необходимости она может быть перемещена или удалена.

programsTableAdapter.Fill(kazakovDataSet.Programs);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Prepodav". При необходимости она может быть перемещена или удалена.

prepodavTableAdapter.Fill(kazakovDataSet.Prepodav);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Predmet". При необходимости она может быть перемещена или удалена.

predmetTableAdapter.Fill(kazakovDataSet.Predmet);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Kabs". При необходимости она может быть перемещена или удалена.

kabsTableAdapter.Fill(kazakovDataSet.Kabs);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Teachers". При необходимости она может быть перемещена или удалена.

teachersTableAdapter.Fill(kazakovDataSet.Teachers);

for (int i = 0; i < dataGridView5.Rows.Count; i++)

{

if (Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Year == DateTime.Today.Year)

{

if (Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Month - DateTime.Today.Month < 2 && Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Month - DateTime.Today.Month >= 0)

{

dataGridView5.Rows[i].DefaultCellStyle.BackColor = Color.Red;

}

else

if (Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Month - DateTime.Today.Month < 0)

{

dataGridView5.Rows[i].DefaultCellStyle.BackColor = Color.Gray;

}

}

else

{

if (Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Year < DateTime.Today.Year)

{

dataGridView5.Rows[i].DefaultCellStyle.BackColor = Color.Gray;

}

}

}

}

}

#region Подсистема "Преподаватели"

#region Добавление

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

{

//Открытие формы добавления записей, относящихся к подсистеме "Преподаватели"

Add\_Prep\_Form f = new Add\_Prep\_Form();

f.Show();

pr++;

}

#endregion

#region Поиск методом выбора

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

{

pr = 0;

//Фильтр таблицы кабинетов

if ((sender as DataGridView).SelectedCells.Count > 0)

{

int teach = Convert.ToInt32(dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[0].Value);

kabsBindingSource.Filter = "teach = " + teach;

if (kabsBindingSource.Count == 0)

{

MessageBox.Show("У выбранного преподавателя нет кабинетов, за которые он несет ответственность");

}

//фильтр ассоциативной таблицы

prepodavBindingSource.Filter = "tnum = " + teach;

int count = prepodavBindingSource.Count;

//Создание строки фильтра для таблицы предметов и фильтрация

string filt;

if (count != 0)//если есть предметы у выбранного преподавателя

{

filt = "(";

for (int i = 0; i < kazakovDataSet.Prepodav.Rows.Count; i++)

{

if (kazakovDataSet.Prepodav[i][0].ToString() == teach.ToString())

{

filt += Convert.ToInt32(kazakovDataSet.Prepodav[i][1]) + ", ";

}

}

filt = filt.Remove(filt.Length - 1 - 1);

predmetBindingSource.Filter = "pnum in " + filt + ")";

}

else

{

predmetBindingSource.Filter = "pnum = 0";

MessageBox.Show("Данный преподаватель не ведет ни один предмет");

}

}

else

{

MessageBox.Show("Преподаватель не выбран");

}

}

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

{

pr = 0;

//фильтр ассоциативной таблицы

if ((sender as DataGridView).SelectedCells.Count > 0)

{

int pred = Convert.ToInt32(dataGridView2.Rows[dataGridView2.SelectedCells[0].RowIndex].Cells[0].Value);

prepodavBindingSource.Filter = "pnum = " + pred;

//Создание строки фильтра для таблицы преподавателей и фильтрация

int count = prepodavBindingSource.Count;

string filt;

if (count != 0)// если выбранный предмет кто-то преподает

{

filt = "(";

for (int i = 0; i < kazakovDataSet.Prepodav.Rows.Count; i++)

{

if (kazakovDataSet.Prepodav[i][1].ToString() == pred.ToString())

{

filt += Convert.ToInt32(kazakovDataSet.Prepodav[i][0]) + ", ";

}

}

filt = filt.Remove(filt.Length - 1 - 1);

teachersBindingSource.Filter = "tnum in " + filt + ")";

}

else

{

teachersBindingSource.Filter = "tnum = 0";

MessageBox.Show("Данный предмет никто не преподает");

}

}

else

{

MessageBox.Show("Предмет не выбран");

}

}

private void dataGridView3\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

pr = 0;

//Фильтр таблицы преподавателей по кабинету

if ((sender as DataGridView).SelectedCells.Count > 0)

{

int prep = Convert.ToInt32(dataGridView3.Rows[dataGridView3.SelectedCells[0].RowIndex].Cells[2].Value);

teachersBindingSource.Filter = "tnum = " + prep;

if (teachersBindingSource.Count == 0)

{

MessageBox.Show("За выбранный кабинет никто не отвечает");

}

}

else

{

MessageBox.Show("Кабинет не выбран");

}

}

private void button5\_Click(object sender, EventArgs e)

{

pr = 0;

//Сброс всех созданных фильтров

teachersBindingSource.RemoveFilter();

predmetBindingSource.RemoveFilter();

kabsBindingSource.RemoveFilter();

prepodavBindingSource.RemoveFilter();

}

#endregion

#region Поиск методом ввода

private void textBox1\_TextChanged(object sender, EventArgs e)

{

pr = 0;

//Очистка фильтра

teachersBindingSource.RemoveFilter();

if ((sender as TextBox).Text != "")

{

//Создание фильтра с кодами найденных преподавателей

string filt = "(";

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView1.Rows[i].Cells[1].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()))

{

filt += dataGridView1.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены преподаватели

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода преподавателя, которого точно не будет

}

//Создание фильтра

teachersBindingSource.Filter = "tnum in " + filt + ")";

if (teachersBindingSource.Count == 0)

{

MessageBox.Show("Преподаватели, содержащие в фамилии '" + (sender as TextBox).Text + "', не найдены");

}

}

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

pr = 0;

//Очистка фильтра

kabsBindingSource.RemoveFilter();

if ((sender as TextBox).Text != "")

{

//Создание фильтра с кодами найденных кабинетов

string filt = "(";

for (int i = 0; i < dataGridView3.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView3.Rows[i].Cells[1].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()))

{

filt += dataGridView3.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены преподаватели

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода кабинета, которого точно не будет

}

//Создание фильтра

kabsBindingSource.Filter = "knum in " + filt + ")";

if (kabsBindingSource.Count == 0)

{

MessageBox.Show("Кабинеты, содержащие '" + (sender as TextBox).Text + "', не найдены");

}

}

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

pr = 0;

//Очистка фильтра

predmetBindingSource.RemoveFilter();

if ((sender as TextBox).Text != "")

{

if (textBox4.Text == "")

{

//Создание фильтра с кодами найденных предметов

string filt = "(";

for (int i = 0; i < dataGridView2.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView2.Rows[i].Cells[1].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()))

{

filt += dataGridView2.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены предметы

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода предмета, которого точно не будет

}

//Создание фильтра

predmetBindingSource.Filter = "pnum in " + filt + ")";

if (predmetBindingSource.Count == 0)

{

MessageBox.Show("Предметы, содержащие '" + (sender as TextBox).Text + "', не найдены");

}

}

else

{

//Создание фильтра с кодами найденных предметов

string filt = "(";

for (int i = 0; i < dataGridView2.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView2.Rows[i].Cells[1].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()) && dataGridView2.Rows[i].Cells[3].FormattedValue.ToString().ToLower().StartsWith(textBox4.Text.ToLower()))

{

filt += dataGridView2.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены предметы

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода предмета, которого точно не будет

}

//Создание фильтра

predmetBindingSource.Filter = "pnum in " + filt + ")";

if (predmetBindingSource.Count == 0)

{

MessageBox.Show("Предметы, содержащие '" + (sender as TextBox).Text + "' и в специальности, содержащей '" + textBox4.Text + "', не найдены");

}

}

}

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

pr = 0;

//Очистка фильтра

predmetBindingSource.RemoveFilter();

if ((sender as TextBox).Text != "")

{

if (textBox3.Text == "")

{

//Создание фильтра с кодами найденных предметов

string filt = "(";

for (int i = 0; i < dataGridView2.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView2.Rows[i].Cells[3].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()))

{

filt += dataGridView2.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены предметы

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода предмета, которого точно не будет

}

//Создание фильтра

predmetBindingSource.Filter = "pnum in " + filt + ")";

if (predmetBindingSource.Count == 0)

{

MessageBox.Show("Предметы, содержащие '" + (sender as TextBox).Text + "', не найдены");

}

}

else

{

//Создание фильтра с кодами найденных предметов

string filt = "(";

for (int i = 0; i < dataGridView2.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView2.Rows[i].Cells[3].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()) && dataGridView2.Rows[i].Cells[1].FormattedValue.ToString().ToLower().StartsWith(textBox4.Text.ToLower()))

{

filt += dataGridView2.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены предметы

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода предмета, которого точно не будет

}

//Создание фильтра

predmetBindingSource.Filter = "pnum in " + filt + ")";

if (predmetBindingSource.Count == 0)

{

MessageBox.Show("Предметы, содержащие '" + textBox3.Text + "' и в специальности, содержащей '" + (sender as TextBox).Text + "', не найдены");

}

}

}

}

#endregion

#region Изменение

private void button2\_Click(object sender, EventArgs e)

{

//Вызов формы изменения выбранного предмета с переносом данных

if (dataGridView2.SelectedCells.Count > 0)

{

string[] pnum = new string[dataGridView2.Columns.Count];

for (int i = 0; i < dataGridView2.Columns.Count; i++)

{

pnum[i] = dataGridView2.Rows[dataGridView2.SelectedCells[0].RowIndex].Cells[i].FormattedValue.ToString();

}

Change\_Prep\_Form\_pred f = new Change\_Prep\_Form\_pred(pnum, connectionstring);

f.Show();

pr++;

}

else

{

MessageBox.Show("Предмет не выбран");

}

}

private void button9\_Click(object sender, EventArgs e)

{

//Вызов формы изменения выбранного преподавателя с переносом данных

if (dataGridView1.SelectedCells.Count > 0)

{

string[] tnum = new string[dataGridView1.Columns.Count];

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

tnum[i] = dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[i].FormattedValue.ToString();

}

Change\_Prep\_Form\_prep f = new Change\_Prep\_Form\_prep(tnum, connectionstring);

f.Show();

pr++;

}

else

{

MessageBox.Show("Преподаватель не выбран");

}

}

private void button10\_Click(object sender, EventArgs e)

{

//Вызов формы изменения выбранного кабинета с переносом данных

if (dataGridView3.SelectedCells.Count > 0)

{

string[] kab = new string[dataGridView3.Columns.Count];

for (int i = 0; i < dataGridView3.Columns.Count; i++)

{

kab[i] = dataGridView3.Rows[dataGridView3.SelectedCells[0].RowIndex].Cells[i].FormattedValue.ToString();

}

Change\_Prep\_Form\_kabs f = new Change\_Prep\_Form\_kabs(kab, connectionstring);

f.Show();

pr++;

}

else

{

MessageBox.Show("Кабинет не выбран");

}

}

#endregion

#region Экспорт

private void button14\_Click(object sender, EventArgs e)

{

if (dataGridView1.Rows.Count > 0)

{

Word.Application application = new Word.Application();

Word.Document document = application.Documents.Add();

document.Paragraphs[1].Range.Text = "Отчет по преподавателю: " + dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[1].Value + " " + dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[2].Value + " " + dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[3].Value;

document.Paragraphs.Add();

document.Paragraphs.Add();

document.Paragraphs.Add();

document.Paragraphs.Add();

string s = "";

kabsBindingSource.Filter = "teach = " + Convert.ToInt32(dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[0].Value);

for (int i = 0; i < kabsBindingSource.Count; i++)

{

s += dataGridView3.Rows[i].Cells[1].Value.ToString() + ", ";

}

document.Paragraphs[2].Range.Text = "Кабинеты: " + s;

document.Paragraphs[3].Range.Text = "Предметы:";

document.Paragraphs.Add();

Word.Range range = document.Paragraphs[4].Range;

prepodavBindingSource.Filter = "tnum = " + Convert.ToInt32(dataGridView1.Rows[dataGridView1.SelectedCells[0].RowIndex].Cells[0].Value);

int row = prepodavBindingSource.Count;

int col = kazakovDataSet.Predmet.Columns.Count;

document.PageSetup.Orientation = Word.WdOrientation.wdOrientLandscape;

range.Tables.Add(range, row + 1, col);

Word.Table table = range.Tables[1];

table.Borders.OutsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;

table.Borders.InsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;

for (int i = 0; i < col; i++)

{

table.Rows[1].Cells[i + 1].Range.Text = dataGridView2.Columns[i].HeaderText;

}

string filt = "(";

for (int i = 0; i < dataGridView4.Rows.Count; i++)

{

filt += Convert.ToInt32(dataGridView4[1, i].Value) + ", ";

}

filt = filt.Remove(filt.Length - 1 - 1);

predmetBindingSource.Filter = "pnum in " + filt + ")";

for (int i = 0; i < col; i++)

{

for (int j = 0; j < row; j++)

{

table.Rows[j + 2].Cells[i + 1].Range.Text = dataGridView2.Rows[j].Cells[i].Value.ToString();

}

}

predmetBindingSource.RemoveFilter();

prepodavBindingSource.RemoveFilter();

application.Visible = true;

}

else

{

MessageBox.Show("Преподаватель не выбран");

}

}

#endregion

#endregion

#region Подсистема "Программное обеспечение"

#region Добавление

private void button3\_Click(object sender, EventArgs e)

{

//Вызов формыдобавления записей, относящихся к подсистеме "Программное обеспечение"

Add\_Prog\_Form f = new Add\_Prog\_Form();

f.Show();

pr++;

}

#endregion

#region Поиск методом выбора

private void dataGridView5\_CellClick(object sender, DataGridViewCellEventArgs e)

{

//Фильтр ассоциативных таблиц

pr = 0;

if ((sender as DataGridView).SelectedCells.Count > 0)

{

licenseBindingSource.Filter = "lnum = " + Convert.ToInt32(dataGridView5.Rows[dataGridView5.SelectedCells[0].RowIndex].Cells[3].Value);

if (licenseBindingSource.Count == 0)

{

licenseBindingSource.Filter = "lnum = 0";

MessageBox.Show("У данной программы отсутствует лицензия");

}

int prog = Convert.ToInt32(dataGridView5.Rows[dataGridView5.SelectedCells[0].RowIndex].Cells[0].Value);

installBindingSource.Filter = "prog = " + prog;

//Создание строки фильтра для таблицы ПК и фильтрация

int count = installBindingSource.Count;

string filt;

if (count != 0)//Если программа установлена на ПК

{

filt = "(";

for (int i = 0; i < dataGridView7.Rows.Count; i++)

{

filt += Convert.ToInt32(dataGridView7[1, i].Value) + ", ";

}

filt = filt.Remove(filt.Length - 1 - 1);

pC1BindingSource.Filter = "PCnum in " + filt + ")";

}

else

{

MessageBox.Show("Данная программа нигде не установлена");

}

//Создание строки фильтра для таблицы предметов и фильтрация

progPredBindingSource.Filter = "prog = " + prog;

count = progPredBindingSource.Count;

if (count != 0)//Если программа используется в предмете

{

filt = "(";

for (int i = 0; i < dataGridView8.Rows.Count; i++)

{

filt += dataGridView8[1, i].FormattedValue + ", ";

}

filt = filt.Remove(filt.Length - 1 - 1);

predmetBindingSource.Filter = "pnum in " + filt + ")";

}

else

{

predmetBindingSource.Filter = "pnum = 0";

MessageBox.Show("Данная программа нигде не используется");

}

}

else

{

MessageBox.Show("Программа не выбрана");

}

}

private void dataGridView6\_CellClick(object sender, DataGridViewCellEventArgs e)

{

pr = 0;

//Фильтр ассоциативной таблицы

if ((sender as DataGridView).SelectedCells.Count > 0)

{

int pc = Convert.ToInt32(dataGridView6.Rows[dataGridView6.SelectedCells[0].RowIndex].Cells[0].Value);

installBindingSource.Filter = "pc = " + pc;

//Создание строки фильтра для таблицы программ и фильтрация

int count = installBindingSource.Count;

string filt;

if (count != 0)//Если программа находится на данном ПК

{

filt = "(";

for (int i = 0; i < dataGridView7.Rows.Count; i++)

{

filt += Convert.ToInt32(dataGridView7[0, i].Value) + ", ";

}

filt = filt.Remove(filt.Length - 1 - 1);

programsBindingSource.Filter = "Prnum in " + filt + ")";

}

else

{

programsBindingSource.Filter = "prnum = 0";

MessageBox.Show("На данном ПК нет установленных программ");

}

}

else

{

MessageBox.Show("ПК не выбран");

}

}

private void dataGridView9\_CellClick(object sender, DataGridViewCellEventArgs e)

{

pr = 0;

//Фильтр ассоциативной таблицы

if ((sender as DataGridView).SelectedCells.Count > 0)

{

int pred = Convert.ToInt32(dataGridView9.Rows[dataGridView9.SelectedCells[0].RowIndex].Cells[0].Value);

progPredBindingSource.Filter = "predm = " + pred;

//Создание строки фильтра для таблицы программ и фильтрация

int count = progPredBindingSource.Count;

string filt;

if (count != 0)//Если программы используются в данном предмете

{

filt = "(";

for (int i = 0; i < dataGridView8.Rows.Count; i++)

{

filt += Convert.ToInt32(dataGridView8[0, i].Value) + ", ";

}

filt = filt.Remove(filt.Length - 1 - 1);

programsBindingSource.Filter = "Prnum in " + filt + ")";

}

else

{

programsBindingSource.Filter = "prnum = 0";

MessageBox.Show("У данного предмета нет установленных программ");

}

}

else

{

MessageBox.Show("Предмет не выбран");

}

}

private void dataGridView10\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

pr = 0;

//Фильтр таблицы ПО по лицензии

if ((sender as DataGridView).SelectedCells.Count > 0)

{

int prog = Convert.ToInt32(dataGridView10.Rows[dataGridView10.SelectedCells[0].RowIndex].Cells[0].Value);

programsBindingSource.Filter = "license = " + prog;

if (programsBindingSource.Count == 0)

{

MessageBox.Show("Нет программ с данной лицензией");

}

}

else

{

MessageBox.Show("Лицензия не выбрана");

}

}

private void button6\_Click(object sender, EventArgs e)

{

pr = 0;

//Сброс всех установленных фильтров

predmetBindingSource.RemoveFilter();

progPredBindingSource.RemoveFilter();

installBindingSource.RemoveFilter();

pC1BindingSource.RemoveFilter();

programsBindingSource.RemoveFilter();

licenseBindingSource.RemoveFilter();

}

#endregion

#region Поичк методом ввода

private void textBox5\_TextChanged(object sender, EventArgs e)

{

pr = 0;

//Очистка фильтра

programsBindingSource.RemoveFilter();

if ((sender as TextBox).Text != "")

{

//Создание фильтра с кодами найденных кабинетов

string filt = "(";

for (int i = 0; i < dataGridView5.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView5.Rows[i].Cells[1].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()))

{

filt += dataGridView5.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены преподаватели

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода кабинета, которого точно не будет

}

//Создание фильтра

programsBindingSource.Filter = "prnum in " + filt + ")";

if (programsBindingSource.Count == 0)

{

MessageBox.Show("Программы, содержащие '" + (sender as TextBox).Text + "', не найдены");

}

}

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

pr = 0;

//Очистка фильтра

licenseBindingSource.RemoveFilter();

if ((sender as TextBox).Text != "")

{

//Создание фильтра с кодами найденных кабинетов

string filt = "(";

for (int i = 0; i < dataGridView10.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView10.Rows[i].Cells[1].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()))

{

filt += dataGridView10.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены преподаватели

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода кабинета, которого точно не будет

}

//Создание фильтра

licenseBindingSource.Filter = "lnum in " + filt + ")";

if (licenseBindingSource.Count == 0)

{

MessageBox.Show("Лицензии, содержащие '" + (sender as TextBox).Text + "', не найдены");

}

}

}

private void textBox9\_TextChanged(object sender, EventArgs e)

{

pr = 0;

//Очистка фильтра

pC1BindingSource.RemoveFilter();

if ((sender as TextBox).Text != "")

{

//Создание фильтра с кодами найденных кабинетов

string filt = "(";

for (int i = 0; i < dataGridView6.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView6.Rows[i].Cells[3].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()))

{

filt += dataGridView6.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены преподаватели

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода кабинета, которого точно не будет

}

//Создание фильтра

pC1BindingSource.Filter = "pcnum in " + filt + ")";

if (pC1BindingSource.Count == 0)

{

MessageBox.Show("ПК, содержащие '" + (sender as TextBox).Text + "', не найдены");

}

}

}

private void textBox7\_TextChanged(object sender, EventArgs e)

{

pr = 0;

//Очистка фильтра

predmetBindingSource.RemoveFilter();

if ((sender as TextBox).Text != "")

{

if (textBox8.Text == "")

{

//Создание фильтра с кодами найденных предметов

string filt = "(";

for (int i = 0; i < dataGridView9.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView9.Rows[i].Cells[1].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()))

{

filt += dataGridView9.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены предметы

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода предмета, которого точно не будет

}

//Создание фильтра

predmetBindingSource.Filter = "pnum in " + filt + ")";

if (predmetBindingSource.Count == 0)

{

MessageBox.Show("Предметы, содержащие '" + (sender as TextBox).Text + "', не найдены");

}

}

else

{

//Создание фильтра с кодами найденных предметов

string filt = "(";

for (int i = 0; i < dataGridView9.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView9.Rows[i].Cells[1].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()) && dataGridView9.Rows[i].Cells[3].FormattedValue.ToString().ToLower().StartsWith(textBox8.Text.ToLower()))

{

filt += dataGridView9.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены предметы

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода предмета, которого точно не будет

}

//Создание фильтра

predmetBindingSource.Filter = "pnum in " + filt + ")";

if (predmetBindingSource.Count == 0)

{

MessageBox.Show("Предметы, содержащие '" + (sender as TextBox).Text + "' и в специальности, содержащей '" + textBox8.Text + "', не найдены");

}

}

}

}

private void textBox8\_TextChanged(object sender, EventArgs e)

{

pr = 0;

//Очистка фильтра

predmetBindingSource.RemoveFilter();

if ((sender as TextBox).Text != "")

{

if (textBox7.Text == "")

{

//Создание фильтра с кодами найденных предметов

string filt = "(";

for (int i = 0; i < dataGridView9.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView9.Rows[i].Cells[3].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()))

{

filt += dataGridView9.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены предметы

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода предмета, которого точно не будет

}

//Создание фильтра

predmetBindingSource.Filter = "pnum in " + filt + ")";

if (predmetBindingSource.Count == 0)

{

MessageBox.Show("Предметы, содержащие '" + (sender as TextBox).Text + "', не найдены");

}

}

else

{

//Создание фильтра с кодами найденных предметов

string filt = "(";

for (int i = 0; i < dataGridView9.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView9.Rows[i].Cells[3].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()) && dataGridView9.Rows[i].Cells[1].FormattedValue.ToString().ToLower().StartsWith(textBox7.Text.ToLower()))

{

filt += dataGridView9.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены предметы

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода предмета, которого точно не будет

}

//Создание фильтра

predmetBindingSource.Filter = "pnum in " + filt + ")";

if (predmetBindingSource.Count == 0)

{

MessageBox.Show("Предметы, содержащие '" + textBox7.Text + "' и в специальности, содержащей '" + (sender as TextBox).Text + "', не найдены");

}

}

}

}

#endregion

#region Изменение

private void button7\_Click(object sender, EventArgs e)

{

//Вызов формы изменения выбранной программы с переносом данных

if (dataGridView5.SelectedCells.Count > 0)

{

string[] prog = new string[dataGridView5.Columns.Count];

for (int i = 0; i < dataGridView5.Columns.Count; i++)

{

prog[i] = dataGridView5.Rows[dataGridView5.SelectedCells[0].RowIndex].Cells[i].FormattedValue.ToString();

}

Change\_Prog\_Form\_prog f = new Change\_Prog\_Form\_prog(prog, connectionstring);

f.Show();

pr++;

}

else

{

MessageBox.Show("Программа не выбрана");

}

}

private void button4\_Click(object sender, EventArgs e)

{

//Вызов формы изменения выбранного предмета с переносом данных

if (dataGridView9.SelectedCells.Count > 0)

{

string[] pred = new string[dataGridView9.Columns.Count];

for (int i = 0; i < dataGridView9.Columns.Count; i++)

{

pred[i] = dataGridView9.Rows[dataGridView9.SelectedCells[0].RowIndex].Cells[i].FormattedValue.ToString();

}

Change\_Prog\_Form\_pred f = new Change\_Prog\_Form\_pred(pred, connectionstring);

f.Show();

pr++;

}

else

{

MessageBox.Show("Предмет не выбран");

}

}

private void button8\_Click(object sender, EventArgs e)

{

//Вызов формы изменения выбранного ПК с переносом данных

if (dataGridView6.SelectedCells.Count > 0)

{

string[] pc = new string[dataGridView6.Columns.Count];

for (int i = 0; i < dataGridView6.Columns.Count; i++)

{

pc[i] = dataGridView6.Rows[dataGridView6.SelectedCells[0].RowIndex].Cells[i].FormattedValue.ToString();

}

Change\_Prog\_Form\_pc f = new Change\_Prog\_Form\_pc(pc, connectionstring);

f.Show();

pr++;

}

else

{

MessageBox.Show("ПК не выбран");

}

}

private void button11\_Click(object sender, EventArgs e)

{

//Вызов формы изменения выбранной лицензии с переносом данных

if (dataGridView10.SelectedCells.Count > 0)

{

string[] lice = new string[dataGridView10.Columns.Count];

for (int i = 0; i < dataGridView10.Columns.Count; i++)

{

lice[i] = dataGridView10.Rows[dataGridView10.SelectedCells[0].RowIndex].Cells[i].FormattedValue.ToString();

}

Change\_Prog\_Form\_lice f = new Change\_Prog\_Form\_lice(lice, connectionstring);

f.Show();

pr++;

}

else

{

MessageBox.Show("Лицензия не выбрана");

}

}

#endregion

#region Обновление по

private void button12\_Click(object sender, EventArgs e)

{

//Вызов формы Обновления выбранной программы с переносом данных

if (dataGridView5.SelectedCells.Count > 0)

{

string[] prog = new string[dataGridView5.Columns.Count];

for (int i = 0; i < dataGridView5.Columns.Count; i++)

{

prog[i] = dataGridView5.Rows[dataGridView5.SelectedCells[0].RowIndex].Cells[i].FormattedValue.ToString();

}

DateTime s = Convert.ToDateTime(prog[5]);

if ((s - DateTime.Today).Days > 30)

{

MessageBox.Show("Программу обновлять не нужно");

}

else

{

Update\_Form f = new Update\_Form(prog, connectionstring);

f.Show();

pr++;

}

}

else

{

MessageBox.Show("Программа не выбрана");

}

}

#endregion

#region Проверка ПО

private void tabControl1\_SelectedIndexChanged\_1(object sender, EventArgs e)

{

if (tabControl1.SelectedIndex == 1)

{

for (int i = 0; i < dataGridView5.Rows.Count; i++)

{

if (Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Year == DateTime.Today.Year)

{

if (Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Month - DateTime.Today.Month < 2 && Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Month - DateTime.Today.Month >= 0)

{

dataGridView5.Rows[i].DefaultCellStyle.BackColor = Color.Red;

}

else

if (Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Month - DateTime.Today.Month < 0)

{

dataGridView5.Rows[i].DefaultCellStyle.BackColor = Color.Gray;

}

}

else

if (Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Year < DateTime.Today.Year)

{

dataGridView5.Rows[i].DefaultCellStyle.BackColor = Color.Gray;

}

}

}

}

#endregion

#region Экспорт

private void button15\_Click(object sender, EventArgs e)

{

if (dataGridView5.Rows.Count > 0)

{

Word.Application application = new Word.Application();

Word.Document document = application.Documents.Add();

document.Paragraphs[1].Range.Text = "Отчет по ПО: " + dataGridView5.Rows[dataGridView5.SelectedCells[0].RowIndex].Cells[1].Value + " " + dataGridView5.Rows[dataGridView5.SelectedCells[0].RowIndex].Cells[2].Value;

document.Paragraphs.Add();

document.Paragraphs.Add();

document.Paragraphs.Add();

document.Paragraphs.Add();

string s = "";

licenseBindingSource.Filter = "lnum = " + Convert.ToInt32(dataGridView5.Rows[dataGridView5.SelectedCells[0].RowIndex].Cells[3].Value);

for (int i = 0; i < licenseBindingSource.Count; i++)

{

s += dataGridView10.Rows[i].Cells[1].Value.ToString();

}

document.Paragraphs[2].Range.Text = "Лицензия: " + s;

document.Paragraphs[3].Range.Text = "Предметы:";

document.Paragraphs.Add();

Word.Range range = document.Paragraphs[4].Range;

progPredBindingSource.Filter = "prog = " + Convert.ToInt32(dataGridView5.Rows[dataGridView5.SelectedCells[0].RowIndex].Cells[0].Value);

int row = progPredBindingSource.Count;

int col = kazakovDataSet.Predmet.Columns.Count;

document.PageSetup.Orientation = Word.WdOrientation.wdOrientLandscape;

range.Tables.Add(range, row + 1, col);

Word.Table table = range.Tables[1];

table.Borders.OutsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;

table.Borders.InsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;

for (int i = 0; i < col; i++)

{

table.Rows[1].Cells[i + 1].Range.Text = dataGridView9.Columns[i].HeaderText;

}

string filt = "(";

for (int i = 0; i < dataGridView8.Rows.Count; i++)

{

filt += Convert.ToInt32(dataGridView8[1, i].Value) + ", ";

}

if (filt.Length >= 2)

{

filt = filt.Remove(filt.Length - 1 - 1);

}

else

{

filt = "(0";

}

predmetBindingSource.Filter = "pnum in " + filt + ")";

for (int i = 0; i < col; i++)

{

for (int j = 0; j < row; j++)

{

table.Rows[j + 2].Cells[i + 1].Range.Text = dataGridView9.Rows[j].Cells[i].Value.ToString();

}

}

predmetBindingSource.RemoveFilter();

progPredBindingSource.RemoveFilter();

installBindingSource.Filter = "prog = " + Convert.ToInt32(dataGridView5.Rows[dataGridView5.SelectedCells[0].RowIndex].Cells[0].Value);

row = installBindingSource.Count;

col = kazakovDataSet.PC1.Columns.Count;

document.Paragraphs.Add();

document.Paragraphs.Add();

Word.Range range1 = document.Paragraphs.Last.Range;

range1.Text = "ПК";

document.Paragraphs.Add();

range1 = document.Paragraphs.Last.Range;

range.Tables.Add(range1, row + 1, col - 1);

table = range1.Tables[1];

table.Borders.OutsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;

table.Borders.InsideLineStyle = Word.WdLineStyle.wdLineStyleSingle;

for (int i = 0; i < col - 1; i++)

{

if (i != 2)

{

table.Rows[1].Cells[i + 1].Range.Text = dataGridView6.Columns[i].HeaderText;

}

else

{

table.Rows[1].Cells[i + 1].Range.Text = dataGridView6.Columns[i + 1].HeaderText;

}

}

filt = "(";

for (int i = 0; i < dataGridView7.Rows.Count; i++)

{

filt += Convert.ToInt32(dataGridView7[1, i].Value) + ", ";

}

if (filt.Length >= 2)

{

filt = filt.Remove(filt.Length - 1 - 1);

}

else

{

filt = "(0";

}

pC1BindingSource.Filter = "pcnum in " + filt + ")";

for (int i = 0; i < col - 1; i++)

{

for (int j = 0; j < row; j++)

{

if (i != 2)

{

table.Rows[j + 2].Cells[i + 1].Range.Text = dataGridView6.Rows[j].Cells[i].Value.ToString();

}

else

{

table.Rows[j + 2].Cells[i + 1].Range.Text = dataGridView6.Rows[j].Cells[i + 1].Value.ToString();

}

}

}

application.Visible = true;

}

else

{

MessageBox.Show("Программа не выбрана");

}

}

#endregion

#endregion

#region Подсистема "Сотрудники"

#region Регистрация

private void button13\_Click(object sender, EventArgs e)

{

Add\_Sotr\_Form f = new Add\_Sotr\_Form();

f.Show();

}

#endregion

#endregion

#region МЕНЮ

#region МЕНЮ

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

{

Stat\_form f = new Stat\_form();

f.Show();

}

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

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Install". При необходимости она может быть перемещена или удалена.

installTableAdapter.Fill(kazakovDataSet.Install);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.ProgPred". При необходимости она может быть перемещена или удалена.

progPredTableAdapter.Fill(kazakovDataSet.ProgPred);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Sotr". При необходимости она может быть перемещена или удалена.

sotrTableAdapter.Fill(kazakovDataSet.Sotr);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.PC1". При необходимости она может быть перемещена или удалена.

pC1TableAdapter.Fill(kazakovDataSet.PC1);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.PC". При необходимости она может быть перемещена или удалена.

pCTableAdapter.Fill(kazakovDataSet.PC);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.License". При необходимости она может быть перемещена или удалена.

licenseTableAdapter.Fill(kazakovDataSet.License);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Programs". При необходимости она может быть перемещена или удалена.

programsTableAdapter.Fill(kazakovDataSet.Programs);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Prepodav". При необходимости она может быть перемещена или удалена.

prepodavTableAdapter.Fill(kazakovDataSet.Prepodav);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Predmet". При необходимости она может быть перемещена или удалена.

predmetTableAdapter.Fill(kazakovDataSet.Predmet);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Kabs". При необходимости она может быть перемещена или удалена.

kabsTableAdapter.Fill(kazakovDataSet.Kabs);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Teachers". При необходимости она может быть перемещена или удалена.

teachersTableAdapter.Fill(kazakovDataSet.Teachers);

for (int i = 0; i < dataGridView5.Rows.Count; i++)

{

if (Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Year == DateTime.Today.Year)

{

if (Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Month - DateTime.Today.Month < 2 && Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Month - DateTime.Today.Month >= 0)

{

dataGridView5.Rows[i].DefaultCellStyle.BackColor = Color.Red;

}

else

{

if (Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Month - DateTime.Today.Month < 0)

{

dataGridView5.Rows[i].DefaultCellStyle.BackColor = Color.Gray;

}

}

}

else

{

if (Convert.ToDateTime(dataGridView5.Rows[i].Cells[5].Value).Year < DateTime.Today.Year)

{

dataGridView5.Rows[i].DefaultCellStyle.BackColor = Color.Gray;

}

}

}

}

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

{

User = "";

Close();

}

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

{

Application.Exit();

}

#endregion

#region О программе

private void оПрограммеToolStripMenuItem1\_Click(object sender, EventArgs e)

{

MessageBox.Show("Курсовой проект МДК 4:" + Environment.NewLine + "Информационная система учета программного обеспечения цикловой комиссии" + Environment.NewLine + "Автор: Казаков Артем, студент группы ИП-41");

}

#endregion

#endregion

#region Закрытие формы

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

{

pr = 0;

string a = MessageBox.Show("Вы действительно хотите выйти?", "Выход", MessageBoxButtons.YesNo).ToString();

if (a == "No")

{

e.Cancel = true;

}

}

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

{

Login f = new Login();

f.Show();

}

#endregion

}

}

**Вход:**

using System;

using System.Windows.Forms;

namespace Uchet

{

public partial class Login : Form

{

public Login()

{

InitializeComponent();

}

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

{

int k = 0;

if (comboBox1.SelectedIndex < 0 || textBox1.Text == "")

{

MessageBox.Show("Введите данные для входа");

}

else

{

for (int i = 0; i < kazakovDataSet.Sotr.Rows.Count; i++)

{

if (kazakovDataSet.Sotr.Rows[i][1].ToString() == comboBox1.Text.ToString() && kazakovDataSet.Sotr.Rows[i][2].ToString() == textBox1.Text.ToString())

{

Main f = new Main(kazakovDataSet.Sotr.Rows[i][3].ToString());

Hide();

f.Show();

k++;

break;

}

}

if (k != 1)

{

MessageBox.Show("Пароль не верен");

}

}

}

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

{

Application.Exit();

}

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

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Sotr". При необходимости она может быть перемещена или удалена.

sotrTableAdapter.Fill(kazakovDataSet.Sotr);

}

}

}

**Добавление подсистемы «Преподаватели»:**

using System;

using System.Windows.Forms;

namespace Uchet

{

public partial class Add\_Prep\_Form : Form

{

public Add\_Prep\_Form()

{

InitializeComponent();

}

private void Add\_Form\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Prepodav". При необходимости она может быть перемещена или удалена.

prepodavTableAdapter.Fill(kazakovDataSet.Prepodav);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Teachers1". При необходимости она может быть перемещена или удалена.

teachers1TableAdapter.Fill(kazakovDataSet.Teachers1);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Teachers". При необходимости она может быть перемещена или удалена.

teachersTableAdapter.Fill(kazakovDataSet.Teachers);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Predmet". При необходимости она может быть перемещена или удалена.

predmetTableAdapter.Fill(kazakovDataSet.Predmet);

kabsTableAdapter.Fill(kazakovDataSet.Kabs);

}

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

{

//Добавление преподавателя

if (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "" || comboBox2.SelectedIndex < 0)

{

MessageBox.Show("Заполните обязательные поля");

}

else

{

if (textBox1.Text.Length > kazakovDataSet.Teachers.tfamColumn.MaxLength || textBox2.Text.Length > kazakovDataSet.Teachers.tnamColumn.MaxLength || textBox3.Text.Length > kazakovDataSet.Teachers.totchColumn.MaxLength)

{

MessageBox.Show("Произошло переполнение полей");

}

else

{

int m = 0;

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

if (Convert.ToBoolean(dataGridView1.Rows[i].Cells[0].FormattedValue) == true)

{

//Занесение связей в ассоциативную таблицу

m++;

}

}

if (m > 0)

{

if (teachersTableAdapter.Insert(0, textBox1.Text, textBox2.Text, textBox3.Text, comboBox2.SelectedItem.ToString()) > 1)

{

teachersTableAdapter.Fill(kazakovDataSet.Teachers);

//Выбор преподаваемых им предметов с использованием Checkbox в datagridview

int k;

k = kazakovDataSet.Teachers.Count;

k = kazakovDataSet.Teachers[k - 1].tnum;

if (predmetBindingSource.Count > 0)

{

for (int i = 0; i < predmetBindingSource.Count; i++)

{

if (Convert.ToBoolean(dataGridView1.Rows[i].Cells[0].FormattedValue) == true)

{

//Занесение связей в ассоциативную таблицу

prepodavTableAdapter.Insert(k, Convert.ToInt32(dataGridView1.Rows[i].Cells[1].Value));

}

}

//Обновление ассоциативной таблицы

prepodavTableAdapter.Fill(kazakovDataSet.Prepodav);

}

//Обновление виртуальной таблицы

teachers1TableAdapter.Fill(kazakovDataSet.Teachers1);

MessageBox.Show("Запись добавлена");

//Очистка

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

}

else

{

MessageBox.Show("Запись уже существует");

}

}

else

{

MessageBox.Show("Выберите предметы, которые ведет преподаватель");

}

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

if (textBox5.Text == "" || textBox6.Text == "" || comboBox3.SelectedIndex < 0)

{

MessageBox.Show("Заполните обязательные поля");

}

else

{

if (textBox5.Text.Length > kazakovDataSet.Predmet.pnamColumn.MaxLength)

{

MessageBox.Show("Произошло переполнение полей");

}

else

{

int k = 0;

for (int i = 0; i < textBox6.Text.Length; i++)

{

if (char.IsDigit(textBox6.Text[i]))

{

k++;

}

}

if (k == textBox6.Text.Length)

{

try

{

Convert.ToInt32(textBox6.Text);

}

catch

{

MessageBox.Show("Слишком большое число");

k = -5;

}

finally

{

if (k > 0)

{

if (Convert.ToInt32(textBox6.Text) == 0)

{

MessageBox.Show("Число слишком мало");

}

else

{

//Внесение данных в таблицу Предмет

if (predmetTableAdapter.Insert(0, textBox5.Text, Convert.ToInt32(textBox6.Text), comboBox3.SelectedItem.ToString()) > 1)

{

predmetTableAdapter.Fill(kazakovDataSet.Predmet);

MessageBox.Show("Запись добавлена");

//Очистка

textBox5.Text = "";

textBox6.Text = "";

}

else

{

MessageBox.Show("Запись уже существует");

}

}

}

}

}

else

{

MessageBox.Show("Введены неверные данные");

}

}

}

}

private void button3\_Click(object sender, EventArgs e)

{

if (textBox8.Text == "" || comboBox1.SelectedIndex < 0)

{

MessageBox.Show("Заполните поля");

}

else

{

if (textBox8.Text.Length > kazakovDataSet.Kabs.knamColumn.MaxLength)

{

MessageBox.Show("Присутствует переполнение полей");

}

else

{

//Занесение данных в таблицу Кабинет

if (kabsTableAdapter.Insert(0, textBox8.Text, Convert.ToInt32(comboBox1.SelectedValue)) > 1)

{

kabsTableAdapter.Fill(kazakovDataSet.Kabs);

MessageBox.Show("Запись добавлена");

//Очистка

textBox8.Text = "";

comboBox1.SelectedIndex = -1;

}

else

{

MessageBox.Show("Запись уже существует");

}

}

}

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

label5.Text = "Наименование, " + textBox5.Text.Length + "/" + kazakovDataSet.Predmet.pnamColumn.MaxLength;

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

label1.Text = "Фамилия, " + textBox1.Text.Length + "/" + kazakovDataSet.Teachers.tfamColumn.MaxLength;

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

label2.Text = "Имя, " + textBox2.Text.Length + "/" + kazakovDataSet.Teachers.tnamColumn.MaxLength;

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

label3.Text = "Отчество, " + textBox3.Text.Length + "/" + kazakovDataSet.Teachers.totchColumn.MaxLength;

}

private void textBox8\_TextChanged(object sender, EventArgs e)

{

label8.Text = "Кабинет, " + textBox8.Text.Length + "/" + kazakovDataSet.Kabs.knamColumn.MaxLength;

}

}

}

**Добавление подсистемы «Программное обеспечение»:**

using System;

using System.Windows.Forms;

namespace Uchet

{

public partial class Add\_Prog\_Form : Form

{

public Add\_Prog\_Form()

{

InitializeComponent();

}

private void Add\_Prog\_Form\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.PC1". При необходимости она может быть перемещена или удалена.

pC1TableAdapter.Fill(kazakovDataSet.PC1);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Programs". При необходимости она может быть перемещена или удалена.

programsTableAdapter.Fill(kazakovDataSet.Programs);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Install". При необходимости она может быть перемещена или удалена.

installTableAdapter.Fill(kazakovDataSet.Install);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.PC". При необходимости она может быть перемещена или удалена.

pCTableAdapter.Fill(kazakovDataSet.PC);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Kabs". При необходимости она может быть перемещена или удалена.

kabsTableAdapter.Fill(kazakovDataSet.Kabs);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.ProgPred". При необходимости она может быть перемещена или удалена.

progPredTableAdapter.Fill(kazakovDataSet.ProgPred);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Predmet". При необходимости она может быть перемещена или удалена.

predmetTableAdapter.Fill(kazakovDataSet.Predmet);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.License". При необходимости она может быть перемещена или удалена.

licenseTableAdapter.Fill(kazakovDataSet.License);

}

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

{

if (textBox1.Text == "" || textBox2.Text == "")

{

MessageBox.Show("Заполните обязательные поля");

}

else

{

if (textBox1.Text.Length > kazakovDataSet.License.lnamColumn.MaxLength)

{

MessageBox.Show("Присутствует переполнение полей");

}

else

{

int k = 0;

for (int i = 0; i < textBox2.Text.Length; i++)

{

if (char.IsDigit(textBox2.Text[i]))

{

k++;

}

}

if (k == textBox2.Text.Length)

{

try

{

Convert.ToInt32(textBox2.Text);

}

catch

{

MessageBox.Show("Слишком большое число");

k = -5;

}

finally

{

if (k > 0)

{

if (Convert.ToInt32(textBox2.Text) == 0)

{

MessageBox.Show("Число слишком мало");

}

else

{

if (licenseTableAdapter.Insert(0, textBox1.Text, Convert.ToInt32(textBox2.Text)) > 1)

{

licenseTableAdapter.Fill(kazakovDataSet.License);

MessageBox.Show("Запись добавлена");

//Очистка

textBox1.Text = "";

textBox2.Text = "";

}

else

{

MessageBox.Show("Запись уже существует");

}

}

}

}

//Занесение данных в таблицу лицензий

}

else

{

MessageBox.Show("Введены неверные данные");

}

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

if (textBox6.Text == "" || comboBox2.SelectedIndex < 0)

{

MessageBox.Show("Заполните обязательные поля");

}

else

{

if (textBox6.Text.Length > kazakovDataSet.PC.PCnetColumn.MaxLength)

{

MessageBox.Show("Присутствует переполнение полей");

}

else

{

//Занесение данных в таблицу ПК

if (pCTableAdapter.Insert(0, textBox6.Text, Convert.ToInt32(comboBox2.SelectedValue)) > 1)

{

pCTableAdapter.Fill(kazakovDataSet.PC);

MessageBox.Show("Запись добавлена");

//Очистка

textBox6.Text = "";

comboBox2.SelectedIndex = -1;

}

else

{

MessageBox.Show("Запись уже существует");

}

}

}

}

private void button3\_Click(object sender, EventArgs e)

{

if (textBox3.Text == "" || textBox4.Text == "" || comboBox1.SelectedIndex < 0 || textBox5.Text == "")

{

MessageBox.Show("Заполните обязательные поля");

}

else

{

if (textBox3.Text.Length > kazakovDataSet.Programs.prnamColumn.MaxLength || textBox4.Text.Length > kazakovDataSet.Programs.versionColumn.MaxLength)

{

MessageBox.Show("Присутствует переполнение полей");

}

else

{

int k = 0;

for (int i = 0; i < textBox5.Text.Length; i++)

{

if (char.IsDigit(textBox5.Text[i]))

{

k++;

}

}

if (k == textBox5.Text.Length)

{

try

{

Convert.ToInt32(textBox5.Text);

}

catch

{

MessageBox.Show("Слишком большое число");

k = -5;

}

finally

{

if (k > 0)

{

if (Convert.ToInt32(textBox5.Text) == 0)

{

MessageBox.Show("Число слишком мало");

}

else

{

int m = 0;

int n = 0;

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

if (Convert.ToBoolean(dataGridView1.Rows[i].Cells[0].FormattedValue) == true)

{

m++;

}

}

for (int i = 0; i < dataGridView3.Rows.Count; i++)

{

if (Convert.ToBoolean(dataGridView3.Rows[i].Cells[0].FormattedValue) == true)

{

n++;

}

}

if (m > 0 && n > 0)

{

//Занесение данных в таблицу программ

if (programsTableAdapter.Insert(0, textBox3.Text, textBox4.Text, Convert.ToInt32(comboBox1.SelectedValue), dateTimePicker1.Value, dateTimePicker2.Value, Convert.ToInt32(textBox5.Text)) > 1)

{

programsTableAdapter.Fill(kazakovDataSet.Programs);

int prnum = Convert.ToInt32(kazakovDataSet.Programs[kazakovDataSet.Programs.Count - 1][0]);

//Добавление ассоциативных связей с выбранными предметами

if (predmetBindingSource.Count > 0)

{

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

if (Convert.ToBoolean(dataGridView1.Rows[i].Cells[0].FormattedValue) == true)

{

progPredTableAdapter.Insert(prnum, Convert.ToInt32(dataGridView1.Rows[i].Cells[1].FormattedValue));

}

}

progPredTableAdapter.Fill(kazakovDataSet.ProgPred);

}

//Добавление ассоциативных связей с выбранными ПК

if (pCBindingSource.Count > 0)

{

for (int i = 0; i < dataGridView3.Rows.Count; i++)

{

if (Convert.ToBoolean(dataGridView3.Rows[i].Cells[0].FormattedValue) == true)

{

installTableAdapter.Insert(prnum, Convert.ToInt32(dataGridView3.Rows[i].Cells[1].FormattedValue));

}

}

installTableAdapter.Fill(kazakovDataSet.Install);

}

MessageBox.Show("Запись добавлена");

//Очистка

textBox3.Text = "";

textBox4.Text = "";

comboBox1.SelectedIndex = -1;

textBox5.Text = "";

}

else

{

MessageBox.Show("Запись уже существует");

}

}

else

{

MessageBox.Show("Выберите предмет и ПК");

}

}

}

}

}

else

{

MessageBox.Show("Введены неверные данные");

}

}

}

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (comboBox1.SelectedIndex > -1)

{

for (int i = 0; i < kazakovDataSet.License.Rows.Count; i++)

{

if (kazakovDataSet.License.Rows[i][0].ToString() == comboBox1.SelectedValue.ToString())

{

dateTimePicker2.Value = DateTime.Now;

dateTimePicker2.Value = dateTimePicker2.Value.AddYears(Convert.ToInt32(kazakovDataSet.License.Rows[i][2].ToString()));

}

}

}

}

private void tabControl1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (tabControl1.SelectedIndex == 2)

{

comboBox1.SelectedIndex = -1;

}

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

label1.Text = "Тип лицензии, " + textBox1.Text.Length + "/" + kazakovDataSet.License.lnamColumn.MaxLength;

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

label9.Text = "Сетевое имя ПК, " + textBox6.Text.Length + "/" + kazakovDataSet.PC.PCnetColumn.MaxLength;

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

label3.Text = "Наименование, " + textBox3.Text.Length + "/" + kazakovDataSet.Programs.prnamColumn.MaxLength;

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

label4.Text = "Версия, " + textBox4.Text.Length + "/" + kazakovDataSet.Programs.versionColumn.MaxLength;

}

}

}

**Добавление сотрудника:**

using System;

using System.Windows.Forms;

namespace Uchet

{

public partial class Add\_Sotr\_Form : Form

{

public Add\_Sotr\_Form()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

Random r = new Random();

textBox2.Text = "";

string[] s = new string[62];

string k = "";

for (int i = 65; i < 91; i++)

{

k += char.ConvertFromUtf32(i);

}

for (int i = 97; i < 123; i++)

{

k += char.ConvertFromUtf32(i);

}

for (int i = 0; i < k.Length; i++)

{

s[i] = k[i].ToString();

}

for (int i = 0; i < 10; i++)

{

s[52 + i] = i.ToString();

}

k = "";

for (int i = 0; i < 16; i++)

{

textBox2.Text += s[r.Next(0, 62)];

}

}

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

{

if (textBox1.Text == "" || textBox2.Text == "" || comboBox1.SelectedIndex < 0)

{

MessageBox.Show("Заполните обязательные поля");

}

else

{

if (textBox1.Text.Length > kazakovDataSet.Sotr.snamColumn.MaxLength)

{

MessageBox.Show("Присутствует переполнение полей");

}

if (sotrTableAdapter.Insert(0, textBox1.Text, textBox2.Text, comboBox1.SelectedItem.ToString()) == 2)

{

MessageBox.Show("Пользователь зарегистрирован");

textBox1.Text = "";

textBox2.Text = "";

comboBox1.SelectedIndex = -1;

}

else

{

MessageBox.Show("Пользователя уже зарегистрирован");

}

}

}

private void Add\_Sotr\_Form\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Sotr". При необходимости она может быть перемещена или удалена.

sotrTableAdapter.Fill(kazakovDataSet.Sotr);

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

label1.Text = "Логин, " + textBox1.Text.Length + "/" + kazakovDataSet.Sotr.snamColumn.MaxLength;

}

}

}

**Изменение кабинета:**

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Uchet

{

public partial class Change\_Prep\_Form\_kabs : Form

{

//Создание временного подключения к БД

private readonly SqlConnection con;

private readonly string k;

public Change\_Prep\_Form\_kabs(string[] kab, string connect)

{

InitializeComponent();

con = new SqlConnection(connect);

//Занесение перенесенных данных в соответствующие поля

textBox10.Text = kab[0].ToString();

textBox11.Text = kab[1].ToString();

k = kab[2];

}

private void Change\_Prep\_Form\_kabs\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Kabs". При необходимости она может быть перемещена или удалена.

kabsTableAdapter.Fill(kazakovDataSet.Kabs);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Teachers1". При необходимости она может быть перемещена или удалена.

teachers1TableAdapter.Fill(kazakovDataSet.Teachers1);

//Изменение текущего значения combobox

comboBox1.SelectedValue = k;

label11.Text = "Кабинет, " + textBox11.Text.Length + "/" + kazakovDataSet.Kabs.knamColumn.MaxLength;

}

private void button3\_Click(object sender, EventArgs e)

{

//Обновление выбранной записи измененными данными

if (textBox11.Text == "" || comboBox1.SelectedIndex < 0)

{

MessageBox.Show("Заполните поля");

}

else

{

if (textBox11.Text.Length > kazakovDataSet.Kabs.knamColumn.MaxLength)

{

MessageBox.Show("Присутствует переполнение полей");

}

else

{

SqlCommand com = new SqlCommand("Update Kabs set knam = '" + textBox11.Text + "', teach = " + Convert.ToInt32(comboBox1.SelectedValue) + " where knum = " + Convert.ToInt32(textBox10.Text), con);

com.Connection.Open();

if (com.ExecuteNonQuery() == 1)

{

MessageBox.Show("Запись уже существует");

}

else

{

com.Connection.Close();

MessageBox.Show("Запись изменена");

}

}

}

}

private void textBox11\_TextChanged(object sender, EventArgs e)

{

label11.Text = "Кабинет, " + textBox11.Text.Length + "/" + kazakovDataSet.Kabs.knamColumn.MaxLength;

}

}

}

**Изменение предмета:**

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Uchet

{

public partial class Change\_Prep\_Form\_pred : Form

{

//Создание временного подключения к БД

private readonly SqlConnection con;

private readonly string k;

public Change\_Prep\_Form\_pred(string[] pnum, string connect)

{

InitializeComponent();

con = new SqlConnection(connect);

//Занесение данных в соответствующие поля

textBox1.Text = pnum[0].ToString();

textBox2.Text = pnum[1].ToString();

textBox3.Text = pnum[2].ToString();

k = pnum[3].ToString();

}

private void Change\_Form\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Predmet". При необходимости она может быть перемещена или удалена.

predmetTableAdapter.Fill(kazakovDataSet.Predmet);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Prepodav". При необходимости она может быть перемещена или удалена.

prepodavTableAdapter.Fill(kazakovDataSet.Prepodav);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Teachers". При необходимости она может быть перемещена или удалена.

teachersTableAdapter.Fill(kazakovDataSet.Teachers);

comboBox1.SelectedItem = k;

label2.Text = "Наименование, " + textBox2.Text.Length + "/" + kazakovDataSet.Predmet.pnamColumn.MaxLength;

//Фильтр ассоциативной связи по предмету

prepodavBindingSource.Filter = "pnum = " + Convert.ToInt32(textBox1.Text);

if (prepodavBindingSource.Count > 0)

{

//Занесение в массив кодов преподавателей из ассоциативной таблицы

int t = prepodavBindingSource.Count;

int[] tnum = new int[t];

for (int i = 0; i < t; i++)

{

tnum[i] = Convert.ToInt32(dataGridView3[0, i].Value);

}

//Установка флажка в таблице DataGridView для преподавателей, которые ведут данный предмет

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

for (int j = 0; j < t; j++)

{

if (dataGridView1[1, i].FormattedValue.ToString() == tnum[j].ToString())

{

dataGridView1[0, i].Value = 1;

}

}

}

}

}

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

{

if (textBox2.Text == "" || textBox3.Text == "" || comboBox1.SelectedIndex < 0)

{

MessageBox.Show("Заполните поля");

}

else

{

if (textBox2.Text.Length > kazakovDataSet.Predmet.pnamColumn.MaxLength)

{

MessageBox.Show("присутствует переполнение полей");

}

else

{

int k = 0;

for (int i = 0; i < textBox3.Text.Length; i++)

{

if (char.IsDigit(textBox3.Text[i]))

{

k++;

}

}

if (k == textBox3.Text.Length)

{

try

{

Convert.ToInt32(textBox3.Text);

}

catch

{

MessageBox.Show("Слишком большое число");

k = -5;

}

finally

{

if (k > 0)

{

if (Convert.ToInt32(textBox3.Text) == 0)

{

MessageBox.Show("Число слишком мало");

}

else

{

SqlCommand com = new SqlCommand("Update Predmet Set pnam = '" + textBox2.Text + "', lab\_hours = " + Convert.ToInt32(textBox3.Text) + ", spec = '" + comboBox1.SelectedItem.ToString() + "' where pnum = " + Convert.ToInt32(textBox1.Text), con);

com.Connection.Open();

if (com.ExecuteNonQuery() == 1)

{

MessageBox.Show("Запись уже существует");

}

else

{

//Удаление всех записей предмета из ассоциации

com = new SqlCommand("Delete from Prepodav where pnum = " + Convert.ToInt32(textBox1.Text), con);

com.ExecuteNonQuery();

com.Connection.Close();

//Занесение в ассоциативную связь всех отмеченных преподавателей

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

if (Convert.ToBoolean(dataGridView1.Rows[i].Cells[0].FormattedValue) == true)

{

prepodavTableAdapter.Insert(Convert.ToInt32(dataGridView1.Rows[i].Cells[1].Value), Convert.ToInt32(textBox1.Text));

}

}

prepodavTableAdapter.Fill(kazakovDataSet.Prepodav);

MessageBox.Show("Запись изменена");

}

}

}

}

}

else

{

MessageBox.Show("Введены неверные данные");

}

}

}

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

label2.Text = "Наименование, " + textBox2.Text.Length + "/" + kazakovDataSet.Predmet.pnamColumn.MaxLength;

}

}

}

**Изменение преподавателя:**

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Uchet

{

public partial class Change\_Prep\_Form\_prep : Form

{

//Создание временного подключения к БД

private readonly SqlConnection con;

private readonly string k;

public Change\_Prep\_Form\_prep(string[] tnum, string connect)

{

InitializeComponent();

con = new SqlConnection(connect);

//Вставка данных в соответствующие поля

textBox5.Text = tnum[0].ToString();

textBox6.Text = tnum[1].ToString();

textBox7.Text = tnum[2].ToString();

textBox8.Text = tnum[3].ToString();

k = tnum[4].ToString();

}

private void Change\_Prep\_Form\_prep\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Teachers". При необходимости она может быть перемещена или удалена.

teachersTableAdapter.Fill(kazakovDataSet.Teachers);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Prepodav". При необходимости она может быть перемещена или удалена.

prepodavTableAdapter.Fill(kazakovDataSet.Prepodav);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Predmet". При необходимости она может быть перемещена или удалена.

predmetTableAdapter.Fill(kazakovDataSet.Predmet);

comboBox1.SelectedItem = k;

label6.Text = "Фамилия, " + textBox6.Text.Length + "/" + kazakovDataSet.Teachers.tfamColumn.MaxLength;

label7.Text = "Имя, " + textBox7.Text.Length + "/" + kazakovDataSet.Teachers.tnamColumn.MaxLength;

label8.Text = "Отчество, " + textBox8.Text.Length + "/" + kazakovDataSet.Teachers.totchColumn.MaxLength;

//Фильтр ассоциации по данному преподавателю

prepodavBindingSource.Filter = "tnum = " + Convert.ToInt32(textBox5.Text);

if (prepodavBindingSource.Count > 0)

{

//Занесение в массив кодов предметов из ассоциативной таблицы

int p = prepodavBindingSource.Count;

int[] pnum = new int[p];

for (int i = 0; i < p; i++)

{

pnum[i] = Convert.ToInt32(dataGridView4[1, i].Value);

}

//Установка флажка в таблице DataGridView для предметов, которые ведет данный преподаватель

for (int i = 0; i < dataGridView2.Rows.Count; i++)

{

for (int j = 0; j < p; j++)

{

if (dataGridView2[1, i].FormattedValue.ToString() == pnum[j].ToString())

{

dataGridView2[0, i].Value = 1;

}

}

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

if (textBox6.Text == "" || textBox7.Text == "" || textBox8.Text == "" || comboBox1.SelectedIndex < 0)

{

MessageBox.Show("Заполните поля");

}

else

{

if (textBox6.Text.Length > kazakovDataSet.Teachers.tfamColumn.MaxLength || textBox7.Text.Length > kazakovDataSet.Teachers.tnamColumn.MaxLength || textBox8.Text.Length > kazakovDataSet.Teachers.totchColumn.MaxLength)

{

MessageBox.Show("Присутствует переполнение полей");

}

else

{

//Обновление преподавателя

SqlCommand com = new SqlCommand("Update Teachers Set tfam = '" + textBox6.Text + "', tnam = '" + textBox7.Text + "', totch = '" + textBox8.Text + "', dolg = '" + comboBox1.SelectedItem.ToString() + "' where tnum = " + Convert.ToInt32(textBox5.Text), con);

com.Connection.Open();

if (com.ExecuteNonQuery() == 1)

{

MessageBox.Show("Запись уже существует");

}

else

{

//Удаление всех записей преподавателя из ассоциации

com = new SqlCommand("Delete from Prepodav where tnum = " + Convert.ToInt32(textBox5.Text), con);

com.ExecuteNonQuery();

com.Connection.Close();

//Занесение в ассоциативную связь все отмеченные предметы

for (int i = 0; i < dataGridView2.Rows.Count; i++)

{

if (Convert.ToBoolean(dataGridView2.Rows[i].Cells[0].FormattedValue) == true)

{

prepodavTableAdapter.Insert(Convert.ToInt32(textBox5.Text), Convert.ToInt32(dataGridView2.Rows[i].Cells[1].Value));

}

}

prepodavTableAdapter.Fill(kazakovDataSet.Prepodav);

MessageBox.Show("Запись изменена");

}

}

}

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

label6.Text = "Фамилия, " + textBox6.Text.Length + "/" + kazakovDataSet.Teachers.tfamColumn.MaxLength;

}

private void textBox7\_TextChanged(object sender, EventArgs e)

{

label7.Text = "Имя, " + textBox7.Text.Length + "/" + kazakovDataSet.Teachers.tnamColumn.MaxLength;

}

private void textBox8\_TextChanged(object sender, EventArgs e)

{

label8.Text = "Отчество, " + textBox8.Text.Length + "/" + kazakovDataSet.Teachers.totchColumn.MaxLength;

}

}

}

**Изменение лицензии:**

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Uchet

{

public partial class Change\_Prog\_Form\_lice : Form

{

//Создание временного подключения к БД

private readonly SqlConnection con;

public Change\_Prog\_Form\_lice(string[] lice, string connect)

{

InitializeComponent();

con = new SqlConnection(connect);

//Вставка данных в соответствующие поля

textBox1.Text = lice[0];

textBox2.Text = lice[1];

textBox3.Text = lice[2];

}

private void Change\_Prog\_Form\_lice\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.License". При необходимости она может быть перемещена или удалена.

licenseTableAdapter.Fill(kazakovDataSet.License);

label2.Text = "Тип лицензии, " + textBox2.Text.Length + "/" + kazakovDataSet.License.lnamColumn.MaxLength;

}

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

{

if (textBox2.Text == "" || textBox3.Text == "")

{

MessageBox.Show("Заполните поля");

}

else

{

if (textBox2.Text.Length > kazakovDataSet.License.lnamColumn.MaxLength)

{

MessageBox.Show("Присутствует переполнение полей");

}

else

{

int k = 0;

for (int i = 0; i < textBox3.Text.Length; i++)

{

if (char.IsDigit(textBox3.Text[i]))

{

k++;

}

}

if (k == textBox3.Text.Length)

{

try

{

Convert.ToInt32(textBox3.Text);

}

catch

{

MessageBox.Show("Слишком большое число");

k = -5;

}

finally

{

if (k > 0)

{

if (Convert.ToInt32(textBox3.Text) == 0)

{

MessageBox.Show("Число слишком мало");

}

else

{

//Обновление Лицензии

SqlCommand com = new SqlCommand("Update License Set lnam = '" + textBox2.Text + "', dlit = '" + Convert.ToInt32(textBox3.Text) + "' where pcnum = " + Convert.ToInt32(textBox1.Text), con);

com.Connection.Open();

if (com.ExecuteNonQuery() == 1)

{

MessageBox.Show("Записьуже существует");

}

else

{

com.Connection.Close();

MessageBox.Show("Запись изменена");

}

}

}

}

}

else

{

MessageBox.Show("Введены неверные данные");

}

}

}

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

label2.Text = "Тип лицензии, " + textBox2.Text.Length + "/" + kazakovDataSet.License.lnamColumn.MaxLength;

}

}

}

**Изменение ПК:**

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Uchet

{

public partial class Change\_Prog\_Form\_pc : Form

{

//Создание временного подключения к БД

private readonly SqlConnection con;

private readonly string k;

public Change\_Prog\_Form\_pc(string[] pc, string connect)

{

InitializeComponent();

con = new SqlConnection(connect);

//Вставка данных в соответствующие поля

textBox1.Text = pc[0];

textBox2.Text = pc[1];

k = pc[2];

}

private void Change\_Prog\_Form\_pc\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.PC". При необходимости она может быть перемещена или удалена.

pCTableAdapter.Fill(kazakovDataSet.PC);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Install". При необходимости она может быть перемещена или удалена.

installTableAdapter.Fill(kazakovDataSet.Install);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Programs". При необходимости она может быть перемещена или удалена.

programsTableAdapter.Fill(kazakovDataSet.Programs);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Kabs". При необходимости она может быть перемещена или удалена.

kabsTableAdapter.Fill(kazakovDataSet.Kabs);

comboBox1.SelectedValue = k;

label2.Text = "Сетевое имя, " + textBox2.Text.Length + "/" + kazakovDataSet.PC.PCnetColumn.MaxLength;

//Фильтр ассоциации по данному ПК

installBindingSource.Filter = "pc = " + Convert.ToInt32(textBox1.Text);

if (installBindingSource.Count > 0)

{

int p = installBindingSource.Count;

int[] pnum = new int[p];

for (int i = 0; i < p; i++)

{

pnum[i] = Convert.ToInt32(dataGridView3[0, i].Value);

}

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

for (int j = 0; j < p; j++)

{

if (dataGridView1[1, i].FormattedValue.ToString() == pnum[j].ToString())

{

dataGridView1[0, i].Value = 1;

}

}

}

}

}

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

{

if (textBox2.Text == "" || comboBox1.SelectedIndex < 0)

{

MessageBox.Show("Заполните поля");

}

else

{

if (textBox2.Text.Length > kazakovDataSet.PC.PCnetColumn.MaxLength)

{

MessageBox.Show("Присутствует переполнение полей");

}

else

{

//Обновление ПК

SqlCommand com = new SqlCommand("Update PC Set pcnet = '" + textBox2.Text + "', kab = " + Convert.ToInt32(comboBox1.SelectedValue) + " where pcnum = " + Convert.ToInt32(textBox1.Text), con);

com.Connection.Open();

if (com.ExecuteNonQuery() == 1)

{

MessageBox.Show("Запись уже существует");

}

else

{

//Удаление всех записей ПК из ассоциации

com = new SqlCommand("Delete from Install where pc = " + Convert.ToInt32(textBox1.Text), con);

com.ExecuteNonQuery();

com.Connection.Close();

//Занесение в ассоциативную связь все отмеченные программы

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

if (Convert.ToBoolean(dataGridView1.Rows[i].Cells[0].FormattedValue) == true)

{

installTableAdapter.Insert(Convert.ToInt32(dataGridView1.Rows[i].Cells[1].Value), Convert.ToInt32(textBox1.Text));

}

}

installTableAdapter.Fill(kazakovDataSet.Install);

MessageBox.Show("Запись изменена");

}

}

}

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

label2.Text = "Сетевое имя, " + textBox2.Text.Length + "/" + kazakovDataSet.PC.PCnetColumn.MaxLength;

}

}

}

**Изменение предмета:**

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Uchet

{

public partial class Change\_Prog\_Form\_pred : Form

{

//Создание временного подключения к БД

private readonly SqlConnection con;

private readonly string k;

public Change\_Prog\_Form\_pred(string[] pred, string connect)

{

InitializeComponent();

con = new SqlConnection(connect);

//Вставка данных в соответствующие поля

textBox1.Text = pred[0];

textBox2.Text = pred[1];

textBox3.Text = pred[2];

k = pred[3];

}

private void Change\_Prog\_Form\_pred\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Predmet". При необходимости она может быть перемещена или удалена.

predmetTableAdapter.Fill(kazakovDataSet.Predmet);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.ProgPred". При необходимости она может быть перемещена или удалена.

progPredTableAdapter.Fill(kazakovDataSet.ProgPred);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Programs". При необходимости она может быть перемещена или удалена.

programsTableAdapter.Fill(kazakovDataSet.Programs);

comboBox1.SelectedItem = k;

label2.Text = "Наименование, " + textBox2.Text.Length + "/" + kazakovDataSet.Predmet.pnamColumn.MaxLength;

//Фильтр ассоциации по данному предмету

progPredBindingSource.Filter = "predm = " + Convert.ToInt32(textBox1.Text);

if (progPredBindingSource.Count > 0)

{

//Занесение в массив кодов программ из ассоциативной таблицы

int p = progPredBindingSource.Count;

int[] prnum = new int[p];

for (int i = 0; i < p; i++)

{

prnum[i] = Convert.ToInt32(dataGridView3[0, i].Value);

}

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

for (int j = 0; j < p; j++)

{

if (dataGridView1[1, i].FormattedValue.ToString() == prnum[j].ToString())

{

dataGridView1[0, i].Value = 1;

}

}

}

}

}

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

{

if (textBox2.Text == "" || textBox3.Text == "" || comboBox1.SelectedIndex < 0)

{

MessageBox.Show("Заполните поля");

}

else

{

if (textBox2.Text.Length > kazakovDataSet.Predmet.pnamColumn.MaxLength)

{

MessageBox.Show("Присутствует переполнение полей");

}

else

{

int k = 0;

for (int i = 0; i < textBox3.Text.Length; i++)

{

if (char.IsDigit(textBox3.Text[i]))

{

k++;

}

}

if (k == textBox3.Text.Length)

{

try

{

Convert.ToInt32(textBox3.Text);

}

catch

{

MessageBox.Show("Слишком большое число");

k = -5;

}

finally

{

if (k > 0)

{

if (Convert.ToInt32(textBox3.Text) == 0)

{

MessageBox.Show("Число слишком мало");

}

else

{

//Обновление предмета

SqlCommand com = new SqlCommand("Update Predmet Set pnam = '" + textBox2.Text + "', lab\_hours = " + Convert.ToInt32(textBox3.Text) + ", spec = '" + comboBox1.SelectedItem.ToString() + "' where pnum = " + Convert.ToInt32(textBox1.Text), con);

com.Connection.Open();

if (com.ExecuteNonQuery() == 1)

{

MessageBox.Show("Запись уже существует");

}

else

{

//Удаление всех записей предмета из ассоциации

com = new SqlCommand("Delete from ProgPred where predm = " + Convert.ToInt32(textBox1.Text), con);

com.ExecuteNonQuery();

com.Connection.Close();

//Занесение в ассоциативную связь все отмеченные программы

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

if (Convert.ToBoolean(dataGridView1.Rows[i].Cells[0].FormattedValue) == true)

{

progPredTableAdapter.Insert(Convert.ToInt32(dataGridView1.Rows[i].Cells[1].Value), Convert.ToInt32(textBox1.Text));

}

}

progPredTableAdapter.Fill(kazakovDataSet.ProgPred);

MessageBox.Show("Запись изменена");

}

}

}

}

}

else

{

MessageBox.Show("Введены неверные данные");

}

}

}

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

label2.Text = "Наименование, " + textBox2.Text.Length + "/" + kazakovDataSet.Predmet.pnamColumn.MaxLength;

}

}

}

**Изменение программы:**

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Uchet

{

public partial class Change\_Prog\_Form\_prog : Form

{

//Создание временного подключения к БД

private readonly SqlConnection con;

private readonly string k;

public Change\_Prog\_Form\_prog(string[] prog, string connect)

{

InitializeComponent();

con = new SqlConnection(connect);

//Вставка данных в соответствующие поля

textBox1.Text = prog[0];

textBox3.Text = prog[1];

textBox4.Text = prog[2];

k = prog[3];

dateTimePicker1.Value = Convert.ToDateTime(prog[4]);

dateTimePicker2.Value = Convert.ToDateTime(prog[5]);

textBox5.Text = prog[6];

}

private void Change\_Prog\_Form\_prog\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Install". При необходимости она может быть перемещена или удалена.

installTableAdapter.Fill(kazakovDataSet.Install);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.ProgPred". При необходимости она может быть перемещена или удалена.

progPredTableAdapter.Fill(kazakovDataSet.ProgPred);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Programs". При необходимости она может быть перемещена или удалена.

programsTableAdapter.Fill(kazakovDataSet.Programs);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.PC". При необходимости она может быть перемещена или удалена.

pCTableAdapter.Fill(kazakovDataSet.PC);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.PC1". При необходимости она может быть перемещена или удалена.

pC1TableAdapter.Fill(kazakovDataSet.PC1);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Predmet". При необходимости она может быть перемещена или удалена.

predmetTableAdapter.Fill(kazakovDataSet.Predmet);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.License". При необходимости она может быть перемещена или удалена.

licenseTableAdapter.Fill(kazakovDataSet.License);

comboBox1.SelectedValue = k;

label3.Text = "Наименование, " + textBox3.Text.Length + "/" + kazakovDataSet.Programs.prnamColumn.MaxLength;

label4.Text = "Версия, " + textBox4.Text.Length + "/" + kazakovDataSet.Programs.versionColumn.MaxLength;

//Фильтр ассоциации по данной программе

progPredBindingSource.Filter = "prog = " + Convert.ToInt32(textBox1.Text);

if (progPredBindingSource.Count > 0)

{

//Занесение в массив кодов предметов из ассоциативной таблицы

int p = progPredBindingSource.Count;

int[] pnum = new int[p];

for (int i = 0; i < p; i++)

{

pnum[i] = Convert.ToInt32(dataGridView2[1, i].Value);

}

//Установка флажка в таблице DataGridView для предметов, в которых используется данная программа

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

for (int j = 0; j < p; j++)

{

if (dataGridView1[1, i].FormattedValue.ToString() == pnum[j].ToString())

{

dataGridView1[0, i].Value = 1;

}

}

}

}

//Фильтр ассоциации по данной программе

installBindingSource.Filter = "prog = " + Convert.ToInt32(textBox1.Text);

if (installBindingSource.Count > 0)

{

//Занесение в массив кодов ПК из ассоциативной таблицы

int pc = installBindingSource.Count;

int[] pcnum = new int[pc];

for (int i = 0; i < pc; i++)

{

pcnum[i] = Convert.ToInt32(dataGridView4[1, i].Value);

}

//Установка флажка в таблице DataGridView для ПК, на которых установлена программа

for (int i = 0; i < dataGridView3.Rows.Count; i++)

{

for (int j = 0; j < pc; j++)

{

if (dataGridView3[1, i].FormattedValue.ToString() == pcnum[j].ToString())

{

dataGridView3[0, i].Value = 1;

}

}

}

}

}

private void button3\_Click(object sender, EventArgs e)

{

if (textBox3.Text == "" || textBox4.Text == "" || textBox5.Text == "" || comboBox1.SelectedIndex < 0)

{

MessageBox.Show("Заполните поля");

}

else

{

if (textBox3.Text.Length > kazakovDataSet.Programs.prnamColumn.MaxLength || textBox4.Text.Length > kazakovDataSet.Programs.versionColumn.MaxLength)

{

MessageBox.Show("Присутствует переполнение полей");

}

else

{

int k = 0;

for (int i = 0; i < textBox5.Text.Length; i++)

{

if (char.IsDigit(textBox5.Text[i]))

{

k++;

}

}

if (k == textBox5.Text.Length)

{

try

{

Convert.ToInt32(textBox5.Text);

}

catch

{

MessageBox.Show("Слишком большое число");

k = -5;

}

finally

{

if (k > 0)

{

if (Convert.ToInt32(textBox5.Text) == 0)

{

MessageBox.Show("Число слишком мало");

}

else

{

//Обновление программы

SqlCommand com = new SqlCommand("Update Programs Set prnam = '" + textBox3.Text + "', version = '" + textBox4.Text + "', license = " + Convert.ToInt32(comboBox1.SelectedValue) + ", inst\_dat = '" + dateTimePicker1.Value + "', expr\_dat = '" + dateTimePicker2.Value + "', value = " + Convert.ToInt32(textBox5.Text) + " where prnum = " + Convert.ToInt32(textBox1.Text), con);

com.Connection.Open();

if (com.ExecuteNonQuery() == 1)

{

MessageBox.Show("Такая запись уже существует");

}

else

{

//Удаление всех записей программы из ассоциации

com = new SqlCommand("Delete from ProgPred where prog = " + Convert.ToInt32(textBox1.Text), con);

com.ExecuteNonQuery();

//Занесение в ассоциативную связь все отмеченные предметы

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

if (Convert.ToBoolean(dataGridView1.Rows[i].Cells[0].FormattedValue) == true)

{

progPredTableAdapter.Insert(Convert.ToInt32(textBox1.Text), Convert.ToInt32(dataGridView1.Rows[i].Cells[1].Value));

}

}

progPredTableAdapter.Fill(kazakovDataSet.ProgPred);

//Удаление всех записей программы из ассоциации

com = new SqlCommand("Delete from Install where prog = " + Convert.ToInt32(textBox1.Text), con);

com.ExecuteNonQuery();

com.Connection.Close();

//Занесение в ассоциативную связь все отмеченные ПК

for (int i = 0; i < dataGridView3.Rows.Count; i++)

{

if (Convert.ToBoolean(dataGridView3.Rows[i].Cells[0].FormattedValue) == true)

{

installTableAdapter.Insert(Convert.ToInt32(textBox1.Text), Convert.ToInt32(dataGridView3.Rows[i].Cells[1].Value));

}

}

installTableAdapter.Fill(kazakovDataSet.Install);

MessageBox.Show("Запись изменена");

}

}

}

}

}

else

{

MessageBox.Show("Введены неверные данные");

}

}

}

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

label3.Text = "Наименование, " + textBox3.Text.Length + "/" + kazakovDataSet.Programs.prnamColumn.MaxLength;

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

label4.Text = "Версия, " + textBox4.Text.Length + "/" + kazakovDataSet.Programs.versionColumn.MaxLength;

}

}

}

**Обновление ПО:**

using System;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Uchet

{

public partial class Update\_Form : Form

{

private readonly SqlConnection con;

private readonly string k;

public Update\_Form(string[] prog, string connect)

{

InitializeComponent();

con = new SqlConnection(connect);

textBox4.Text = prog[0];

textBox1.Text = prog[1];

textBox2.Text = prog[2];

k = prog[3];

dateTimePicker1.Value = Convert.ToDateTime(prog[4]);

dateTimePicker2.Value = Convert.ToDateTime(prog[5]);

textBox3.Text = prog[6];

}

private void Update\_Form\_Load(object sender, System.EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.Programs". При необходимости она может быть перемещена или удалена.

programsTableAdapter.Fill(kazakovDataSet.Programs);

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.License". При необходимости она может быть перемещена или удалена.

licenseTableAdapter.Fill(kazakovDataSet.License);

comboBox1.SelectedValue = k;

if (comboBox1.SelectedIndex > -1)

{

for (int i = 0; i < kazakovDataSet.License.Rows.Count; i++)

{

if (kazakovDataSet.License.Rows[i][0].ToString() == comboBox1.SelectedValue.ToString())

{

dateTimePicker3.Value = DateTime.Today.AddYears(Convert.ToInt32(kazakovDataSet.License.Rows[i][2].ToString()));

}

}

}

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (comboBox1.SelectedIndex > -1)

{

for (int i = 0; i < kazakovDataSet.License.Rows.Count; i++)

{

if (kazakovDataSet.License.Rows[i][0].ToString() == comboBox1.SelectedValue.ToString())

{

dateTimePicker3.Value = DateTime.Today.AddYears(Convert.ToInt32(kazakovDataSet.License.Rows[i][2].ToString()));

}

}

}

}

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

{

if (comboBox1.SelectedIndex < 0)

{

MessageBox.Show("Выберите лицензию");

}

else

{

SqlCommand com = new SqlCommand("Update Programs set license = " + Convert.ToInt32(comboBox1.SelectedValue) + ", expr\_dat = '" + dateTimePicker3.Value + "' where prnum = " + Convert.ToInt32(textBox4.Text) + "", con);

com.Connection.Open();

com.ExecuteNonQuery();

com = new SqlCommand("Update ProgUpd set upd = upd + 1 where Years = " + dateTimePicker2.Value.Year, con);

com.ExecuteNonQuery();

com = new SqlCommand("if not exists(select \* from progupd where years = " + dateTimePicker3.Value.Year + ") insert into progupd values (" + dateTimePicker3.Value.Year + ",1,0) else update progupd set all\_upd = all\_upd+1 where years = " + dateTimePicker3.Value.Year, con);

com.ExecuteNonQuery();

com.Connection.Close();

button1.Enabled = false;

MessageBox.Show("Программа обновлена");

}

}

}

}

**Статистика:**

using System;

using System.Windows.Forms;

using exel = Microsoft.Office.Interop.Excel;

namespace Uchet

{

public partial class Stat\_form : Form

{

public Stat\_form()

{

InitializeComponent();

}

private void Stat\_form\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "kazakovDataSet.ProgUpd". При необходимости она может быть перемещена или удалена.

progUpdTableAdapter.Fill(kazakovDataSet.ProgUpd);

if (progUpdBindingSource.Count == 0)

{

button3.Enabled = false;

}

else

{

button3.Enabled = true;

}

}

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

{

if (textBox1.Text == "" || textBox2.Text == "")

{

MessageBox.Show("Определите границы поиска");

}

else

{

int m = 0;

int n = 0;

for (int i = 0; i < textBox1.Text.Length; i++)

{

if (char.IsDigit(textBox1.Text[i]))

{

m++;

}

}

for (int i = 0; i < textBox2.Text.Length; i++)

{

if (char.IsDigit(textBox2.Text[i]))

{

n++;

}

}

if (m < textBox1.Text.Length || n < textBox2.Text.Length)

{

MessageBox.Show("Введены неверные года");

}

else

{

try

{

Convert.ToInt32(textBox2.Text);

Convert.ToInt32(textBox1.Text);

}

catch

{

MessageBox.Show("Слишком большое число");

m = -5;

}

finally

{

if (m > 0)

{

chart1.Series[0].Points.Clear();

chart1.Series[1].Points.Clear();

for (int i = 0; i < kazakovDataSet.ProgUpd.Rows.Count; i++)

{

if (Convert.ToInt32(kazakovDataSet.ProgUpd.Rows[i][0]) >= Convert.ToInt32(textBox1.Text) && Convert.ToInt32(kazakovDataSet.ProgUpd.Rows[i][0]) <= Convert.ToInt32(textBox2.Text))

{

chart1.Series[0].Points.AddXY(Convert.ToInt32(kazakovDataSet.ProgUpd.Rows[i][0]), Convert.ToInt32(kazakovDataSet.ProgUpd.Rows[i][1]));

chart1.Series[1].Points.AddXY(Convert.ToInt32(kazakovDataSet.ProgUpd.Rows[i][0]), Convert.ToInt32(kazakovDataSet.ProgUpd.Rows[i][2]));

}

}

if (chart1.Series[0].Points.Count == 0 && chart1.Series[1].Points.Count == 0)

{

button3.Enabled = false;

MessageBox.Show("В этих границах информация отсутствует");

}

}

}

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

chart1.Series[0].Points.Clear();

chart1.Series[1].Points.Clear();

for (int i = 0; i < kazakovDataSet.ProgUpd.Rows.Count; i++)

{

chart1.Series[0].Points.AddXY(Convert.ToInt32(kazakovDataSet.ProgUpd.Rows[i][0]), Convert.ToInt32(kazakovDataSet.ProgUpd.Rows[i][1]));

chart1.Series[1].Points.AddXY(Convert.ToInt32(kazakovDataSet.ProgUpd.Rows[i][0]), Convert.ToInt32(kazakovDataSet.ProgUpd.Rows[i][2]));

}

if (chart1.Series[0].Points.Count == 0)

{

button3.Enabled = false;

}

else

{

button3.Enabled = true;

}

}

private void button3\_Click(object sender, EventArgs e)

{

exel.Application application = new exel.Application();

exel.Sheets sheets;

exel.Worksheet worksheet;

exel.Range cells;

application.SheetsInNewWorkbook = 1;

application.Workbooks.Add(Type.Missing);

sheets = application.ActiveWorkbook.Sheets;

worksheet = (exel.Worksheet)sheets.get\_Item(1);

worksheet.Name = "Статистика обновления";

worksheet.Activate();

int row = chart1.Series[0].Points.Count;

int col = chart1.Series.Count;

for (int i = 0; i < row; i++)

{

cells = (exel.Range)worksheet.Cells[i + 1, 1];

cells.Value2 = chart1.Series[0].Points[i].XValue + " г";

}

for (int i = 0; i < col; i++)

{

for (int j = 0; j < row; j++)

{

cells = (exel.Range)worksheet.Cells[j + 1, i + 2];

cells.Value2 = chart1.Series[i].Points[j].YValues;

}

}

string cur = "c" + (row);

cells = worksheet.get\_Range("a1", cur.ToString());

cells.Select();

exel.Chart chart = (exel.Chart)application.Charts.Add(Type.Missing, Type.Missing, Type.Missing, Type.Missing);

chart.Activate();

chart.Select(Type.Missing);

application.ActiveChart.PlotBy = exel.XlRowCol.xlColumns;

application.ActiveChart.HasTitle = true;

application.ActiveChart.ChartTitle.Text = "Обновления";

application.ActiveChart.HasLegend = true;

application.ActiveChart.Legend.Position = exel.XlLegendPosition.xlLegendPositionBottom;

application.ActiveChart.SeriesCollection(1).Name = "Требуется обновить";

application.ActiveChart.SeriesCollection(2).Name = "Обновлено";

application.ActiveChart.ChartType = exel.XlChartType.xlColumnClustered;

application.ActiveChart.Location(exel.XlChartLocation.xlLocationAsObject, "Статистика обновления");

application.ActiveChart.ApplyDataLabels(exel.XlDataLabelsType.xlDataLabelsShowValue, false, true, false, false, false, false, true, false, true);

application.Visible = true;

}

}

}