МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники (ФИТР)

Кафедра программного обеспечения информационных систем и технологий

**КУРСОВАЯ РАБОТА**

**по дисциплине «Базы данных»**

на тему: «Банк – Информационная система учета вкладов»

Выполнил: ст. гр. 10701319 ЧжанЦзыхан

Принял: доцент Куприянов А. Б.

Минск 2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники (ФИТР)

Кафедра программного обеспечения информационных систем и технологий

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовому проекту (работе)**

**по дисциплине «Базы данных»**

на тему: «Банк – Информационная система учета вкладов»

**Исполнитель**: ЧжанЦзыхан (фамилия, инициалы)

(подпись)

**Студент 3 курса 10701319 группы**

**Руководитель**: Куприянов А.Б. (фамилия, инициалы)

(подпись)

Минск 2022

**Оглавление**

[Введение 4](file:///C:\\Users\\HP\\Downloads\\CourseWork.docx" \l "_Toc103019202)

[1. Проектирование базы данных 5](file:///C:\\Users\\HP\\Downloads\\CourseWork.docx" \l "_Toc103019203)

[1.1. Построение концептуальной модели 5](file:///C:\\Users\\HP\\Downloads\\CourseWork.docx" \l "_Toc103019204)

[1.2. Построение логической модели 6](file:///C:\\Users\\HP\\Downloads\\CourseWork.docx" \l "_Toc103019205)

[1.3. Построение физической модели 8](file:///C:\\Users\\HP\\Downloads\\CourseWork.docx" \l "_Toc103019206)

[2. Перечень элементов управления 9](file:///C:\\Users\\HP\\Downloads\\CourseWork.docx" \l "_Toc103019211)

[3. Руководство пользователя 10](file:///C:\\Users\\HP\\Downloads\\CourseWork.docx" \l "_Toc103019212)

[4. Графический материал 15](file:///C:\\Users\\HP\\Downloads\\CourseWork.docx" \l "_Toc103019213)

[Заключение 16](file:///C:\\Users\\HP\\Downloads\\CourseWork.docx" \l "_Toc103019214)

[Список литературы 17](file:///C:\\Users\\HP\\Downloads\\CourseWork.docx" \l "_Toc103019215)

[ПРИЛОЖЕНИЕ 18](file:///C:\\Users\\HP\\Downloads\\CourseWork.docx" \l "_Toc103019216)

# Введение

Проект создания программы для учёта вкладов и услуг является универсальным для любого банка. Ведь, добавив параметры, а также переименовав некоторые названия и имена, можно с лёгкостью переделать данный проект для других банков.

Из плюсов данного проекта можно учесть:

* Точный и быстрый просмотр информации из базы данных;
* Столь же точное и быстрое изменение информации в базе;
* Красивый и лаконичный дизайн;
* Универсальность;
* Удобная фильтрация таблиц.

В настоящее время всё эти плюсы являются незаменимыми условиями для любых проектов в экономической среде, ведь чем меньше времени тратиться на поиск и подбор информации, тем довольнее клиент, и тем больше времени можно уделить для общения с ним. Тем самым экономя деньги банку.

Основой для написания данного проекта является декларативный язык программирования SQL. Для работы с ним и базами данных используется рабочая среда Microsoft SQL Server. А для компоновки работоспособности всего проекта и внешнего вида ответственен язык программирования C# в среде разработки Microsoft Visual Studio 2019. Для оформления визуального стиля программного продукта используется GUNA Framework и её GUNA UI X C# библиотека.

Целью курсового проекта является разработка информационной системы для банка – Информационной системы учета вкладов. Согласно данной цели, выведем соответствующие задачи:

1. Разработка Базы данных. Проектирование базы данных в соответствии с рабочей областью информационной системы.
2. Разработка формы. Проектирование формы приложения для ввода данных, отчетов, запросов, элементы управления, свойства, методы и события.
3. Разработка руководства пользователя. Создание текстового документа для описания всего процесса работы с данным программным продуктом.

**1. Проектирование базы данных**

Процесс проектирования базы данных состоит из следующих этапов:

* концептуальное проектирование;
* логическое проектирование;
* физическое проектирование.

**1.1. Построение концептуальной модели**

Концептуальная модель базы данных – это диаграмма, показывающая связь между объектами и их характеристиками.

На этапе концептуального проектирования определяются все сущности предметной области, их основные атрибуты и связи. Результатом концептуального проектирования является концептуальная модель базы данных.  
 在概念设计阶段，确定了主题领域的所有实体，它们的主要属性和关系。 概念设计的结果是概念数据库模型。

При проектировании базы данных определяем сущности:

* Deposits – список всех операций с депозитами в банке: определяет ID, id клиента, дату открытия вклада, начальная сумму вклада, количество месяцев, id работника, id вклада, дата закрытия вклада.
* 存款 - 银行存款的所有操作列表：定义 ID、客户 ID、存款开户日期、初始存款金额、月数、员工 ID、存款 ID、存款关闭日期。
* Clients – список всех клиентов банка: определяет ID, Имя клиента, Фамилию клиента, и его паспортный номер.

客户 - 所有银行客户的列表：定义 ID、客户的名字、客户的姓氏和他的护照号码。

* Employees – список работников: определяет ID, Фамилию работника, Имя работника, номер телефона.

-员工 - 员工列表：定义 ID、员工的姓氏、员工的名字、电话号码。

* DepositTypes – список вкладов: определяет ID, название вклада, процент вклада.
* 存款类型 – 存款列表：定义 ID、存款名称、存款百分比。
* Withdrawals – список отзывов: определяет ID, время отзыва вклада, id вклада, сумму денег.
* -取款 - 取款列表：定义ID、取款取款时间、存款ID、金额。



Рисунок 1.1 – концептуальная модель проекта базы данных

**1.2. Построение логической модели**

Логическое проектированиезаключается в определении числа и структуры таблиц, включая названия столбцов, типы данных и ограничения в столбцах, первичные и внешние ключи, а также свойства связей.

逻辑设计包括确定表的数量和结构，包括列名、数据类型和列约束、主键和外键以及关系属性。

Логическая схема представлена ниже в таблицах 1 – 5.

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| Clients | | | |
| Column name | Data type | Allow Nulls | Constraints |
| ID | int | No | PK, IDENTITY(1,1) |
| Name | nchar(10) | No | FK, Name.ID |
| Surname | nchar(10) | No | FK, Surname.ID |
| Passport | nchar(10) | No | FK, Passport.ID |

Таблица 2

|  |  |  |  |
| --- | --- | --- | --- |
| Employees | | | |
| Column name | Data type | Allow Nulls | Constraints |
| ID | int | No | PK, IDENTITY(1,1) |
| Surname | nchar(10) | No | FK, Surname.ID |
| Name | nchar(10) | No | FK, Name.ID |
| PhoneNumber | nchar(10) | No | FK, PhoneNumber.ID |

Таблица 3

|  |  |  |  |
| --- | --- | --- | --- |
| Deposits | | | |
| Column name | Data type | Allow Nulls | Constraints |
| ID | int | No | PK,IDENTITY(1,1) |
| ClientID | int | No | FK, ClientID.ID |
| OpenDate | date | No |  |
| StartSum | money | No | FK, StertSum.ID |
| MonthAmm | int | No | FK, MonthAmm.ID |
| EmployeeID | int | No | FK,EmployeeID.ID |
| DepositType | int | No | FK,DepositType.ID |
| CloseDate | date | Yes |  |

Таблица 4

|  |  |  |  |
| --- | --- | --- | --- |
| DepositTypes | | | |
| Column name | Data type | Allow Nulls | Constraints |
| ID | int | No | PK, IDENTITY(1,1) |
| Title | nchar(10) | No | FK, Title.ID |
| Percentage | int | No | FK, Percentage.ID |

Таблица 5

|  |  |  |  |
| --- | --- | --- | --- |
| Services | | | |
| Column name | Data type | Allow Nulls | Constraints |
| ID | int | No | PK, IDENTITY(1,1) |
| Date | datetime | No |  |
| DepositID | int | No | FK, DepositID.ID |
| MoneyAmm | money | No |  |

**1.3. Построение физической модели**

**1. Имя базы данных: Bank.**

2. Имя файла базы данных: Bank.

3. Путь файла базы данных:

D:\SQLServerExpress\MSSQL15.SQLEXPRESS\MSSQL\DATA\

4. Начальный размер файла базы данных (МБ): 8.

5. Шаг приращения файла базы данных (МБ): 64.

6. Максимальный размер файла базы данных: без ограничений.

7. Имя файла журнала транзакции: Bank\_log.

8. Путь файла журнала транзакции:

D:\SQLServerExpress\MSSQL15.SQLEXPRESS\MSSQL\DATA

9. Начальный размер файла журнала транзакции (МБ): 8.

10. Шаг приращения файла журнала транзакции (МБ): 64.

11. Максимальный размер файла журнала транзакций (МБ): 2097152.

**2. Перечень элементов управления**

Главная форма Main. Перечень элементов управления: nameDataGridViewTextBoxColumn, phoneNumberDataGridViewTextBoxColumn, postIDDataGridViewTextBoxColumn, postsDataGridView, postsBindingNavigator, postsBindingSource, postsConfirmButton, postsTableAdapter, postsTitleTextBox, PostTitle, serviceIDDataGridViewTextBoxColumn, servicesBindingNavigator, servicesConfirmButton, servicesDataGridView, servicesFilterEnableButton, servicesFilterGroupBox, servicesFilterRemoveButton, servicesFilterSwitch, servicesFilterTitleTextBox, servicesNameTextBox, servicesTableAdapter, serviceTitleDataGridViewTextBoxColumn, sevicesSource, surnameDataGridViewTextBoxColumn, surnameDataGridViewTextBoxColumn1, tabPage1 (начиная с 1 по 5), textBox1, titleDataGridViewTextBoxColumn, toolStripButton1 (начиная с 1 по 18), toolStripLabel1 (начиная с 1 по 3), toolStripSeparator1 (начиная с 1 по 9), toolStripTextBox1 (начиная с 1 по 3). И многие другие… На главное форме расположено вся рабочая сторона программы, начиная с отображения элементов таблицы, и заканчивая кнопками изменения.

Форма-дубликат Form12. Содержит в большую часть элементов функционала с главной формы, кроме GUI связанных элементов.

Форма DepositReport – форма для отображения отчета по таблице депозитов

Форма WithdrawalsReport – форма для отображения отчета по таблице списаний

Файл packages.config – файл позволяющий пользоваться Фреймворком GUNA.

Файл App.config – файл связывающий базу данных и проект посредством соединения программы к серверу и постоянном подключении к нему.

Файлы Report1.rdlc и Report2.rdlc – файлы мастера отчетов таблиц депозитов и списаний соответственно

**3. Руководство пользователя**

Программа написана на языке программирования C# в среде MS Visual Studio 2019. Функционал реализован в соответствии с требованиями к проекту.

После запуска приложения пользователь попадает в главное меню информационной системы учета вкладов (Рисунок 2.1).

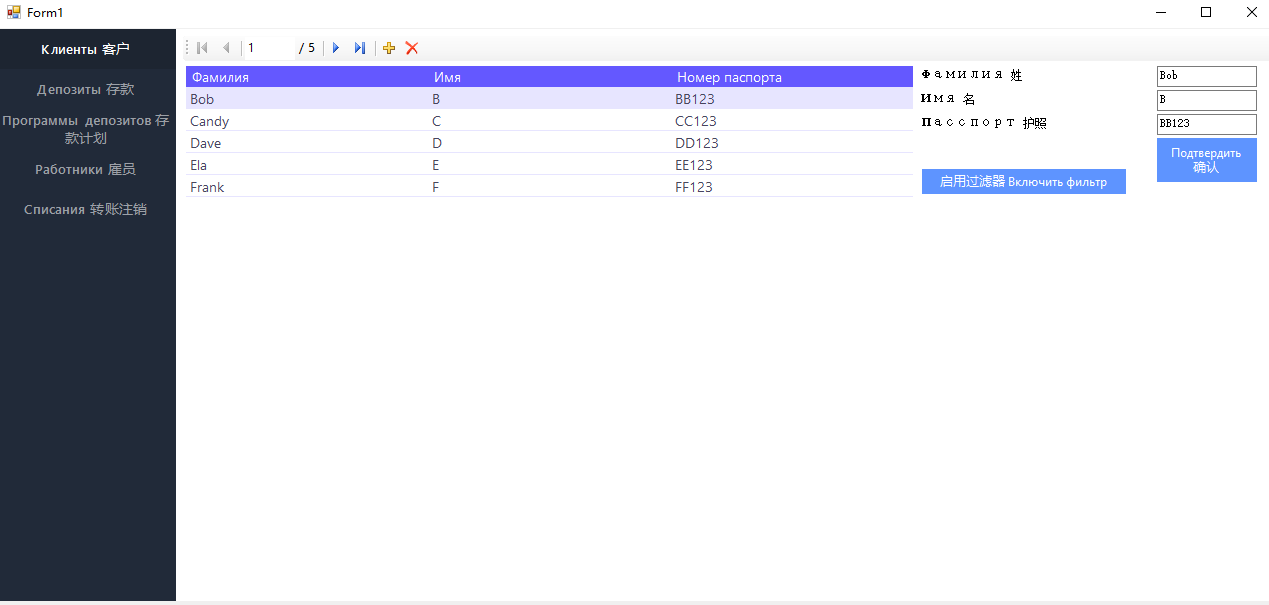


Рисунок 2.1 – стартовое меню программы

В меню слева находятся кнопки перехода между таблицами в базе данных, в самом центре – таблица Клиенты со всей ей сопутствующей информацией. Над таблицей находятся кнопки: «Переместить к началу», «Переместить назад», «Текущее расположение», «Общее число элементов», «Переместить вперёд», «Переместить в конец», «Добавить», «Удалить». У каждый кнопки свой функционал, соответствующий её названию.

在左侧的菜单中，有用于在数据库中的表之间移动的按钮，在最中心的是 Clients 表及其所有相关信息。 表格上方有按钮：“移至开头”、“后移”、“当前位置”、“元素总数”、“前移”、“移至末尾”、“添加”、“删除” . 每个按钮都有与其名称对应的自己的功能。

Справа в углу находятся строки с информацией, соответствующей выбранной ранее строке из таблицы. При нажатии кнопки «Подтвердить» все внесённые изменения в строки сверху будут сохранены в базу данных. Ниже находиться кнопка переключения режима фильтра, при нажатии на который откроется поле для фильтрации таблицы (Рисунок 2.2)

在右上角有与表中先前选择的行相对应的信息的行。 当您单击“确认”按钮时，对上述行所做的所有更改都将保存到数据库中。 下面有一个用于切换过滤模式的按钮，点击后会打开一个过滤表格的字段（图2.2）

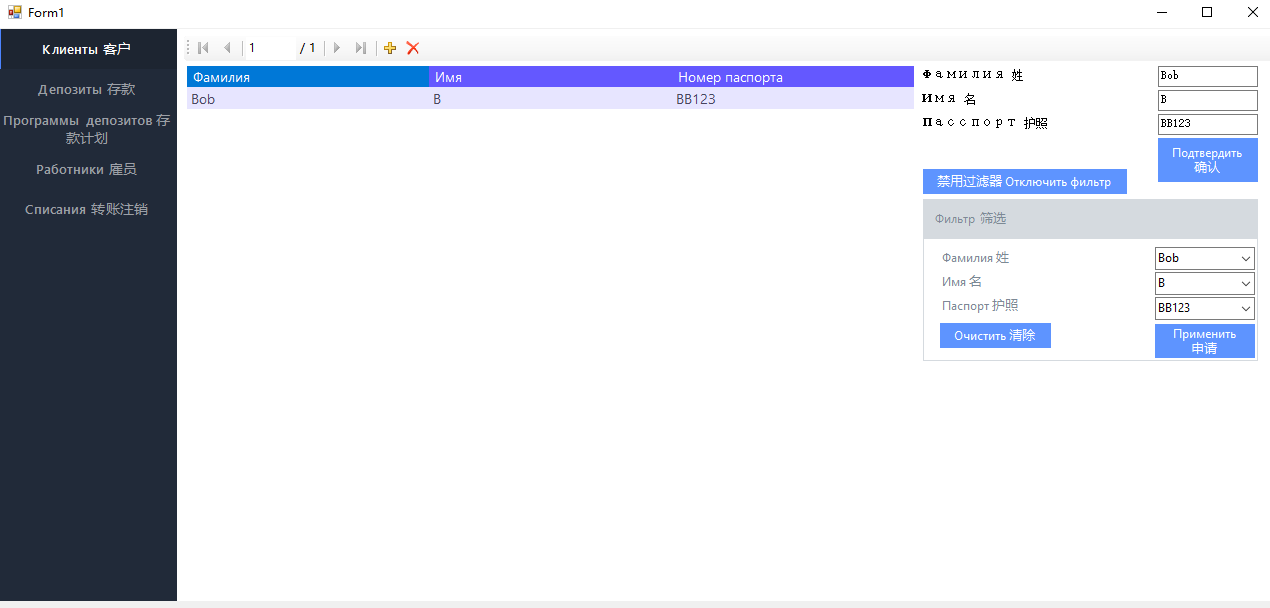


Рисунок 2.2 – стартовое меню программы с полем фильтрации

В поле фильтрации можно заметить пустые поля фильтрации, при заполнении которых таблица слева будет отображена в соответствии с предпочтениями пользователя. В полях напротив условий «Фамилия», «Имя» и «Паспорт» из выпадающего меню следует выбрать условие для фильтрации. Ниже находиться кнопка «Сбросить», что опустошает выбор фильтра, а также кнопка «Применить», что применяет фильтр к таблице.

在过滤字段中，您可以看到空的过滤字段，填写后，左侧的表格将根据用户的喜好显示。 在下拉菜单中与条件“姓氏”、“名字”和“护照”相对的字段中，选择过滤条件。 其下方是清除过滤器选择的重置按钮，以及将过滤器应用于表格的应用按钮。

Рассмотрим другие таблицы, при нажатии на левом поле приложения по кнопке «Депозиты» откроется окно со схожим интерфейсов, как и у начального окна, однако таблица будет соответствующая (Рисунок 2.3). Так же измениться и набор элементов справа от таблицы. Появятся кнопки «к клиентам», «к работникам», «к программам» при нажатии на которые откроются таблицы Клиенты (Рисунок 2.4), Работники (Рисунок 2.5), Депозиты (Рисунок 2.6) соответственно в форме Form12.

让我们考虑其他表格，当您单击应用程序左侧字段的“存款”按钮时，将打开一个具有类似界面的窗口，与初始窗口一样，但表格将对应（图 2.3）。 表格右侧的元素集也会发生变化。 点击后会出现“给客户”、“给员工”、“给程序”按钮，分别在Form12中打开客户（图2.4）、雇员（图2.5）、存款（图2.6）表。

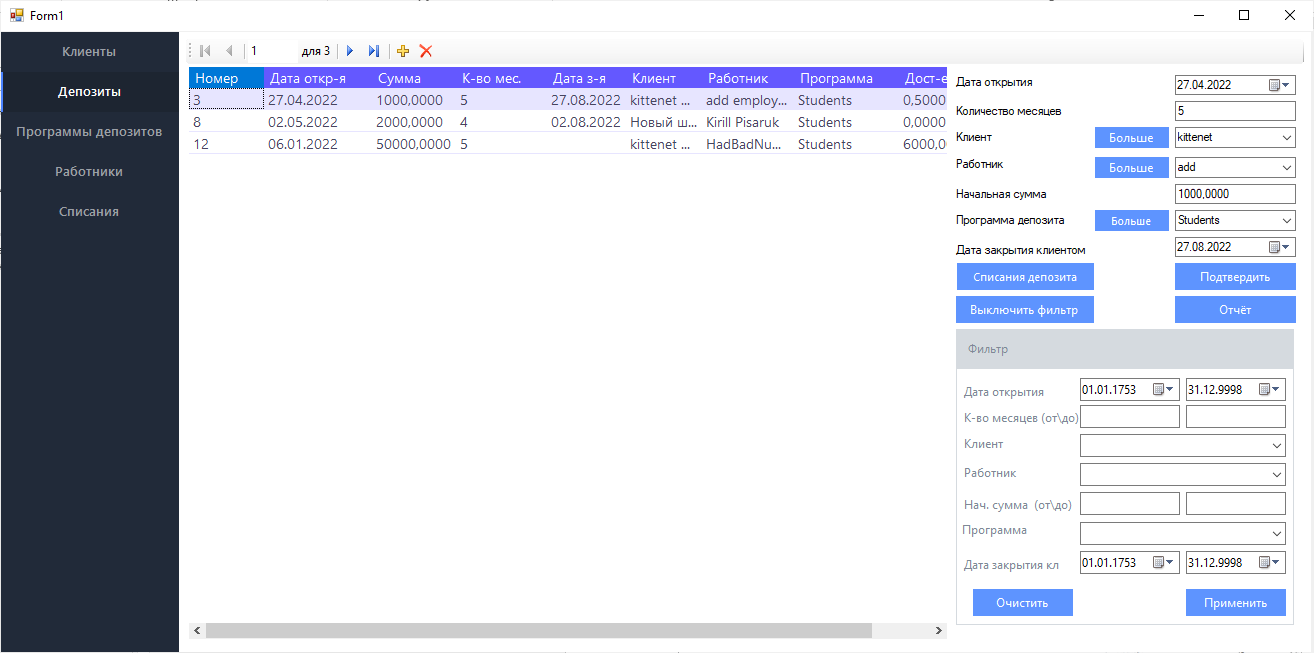


Рисунок 2.3 – меню программы с выбранным полем «Депозиты».

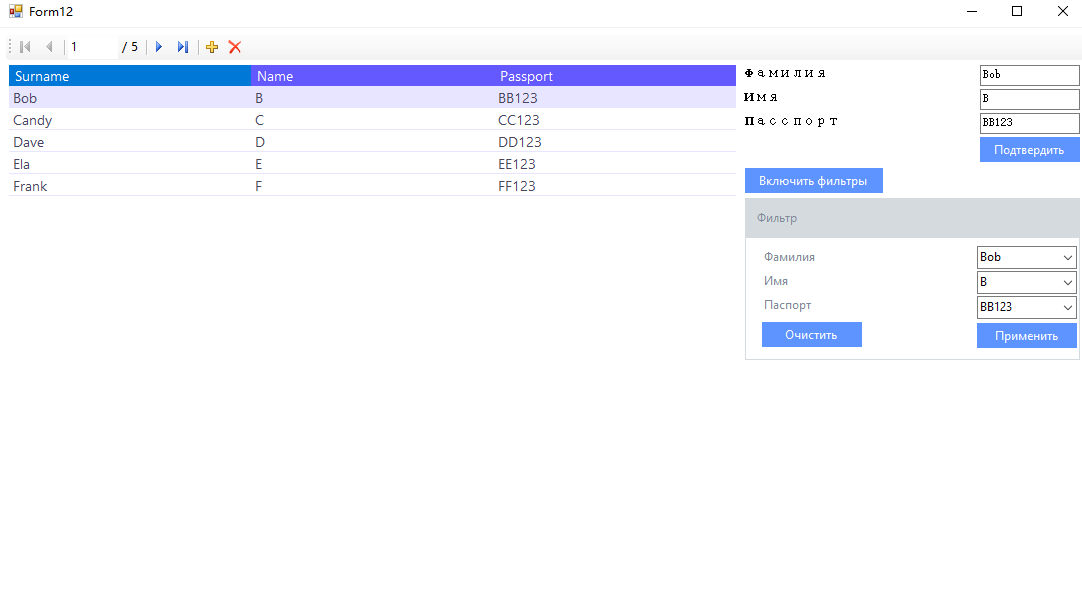


Рисунок 2.4 – форма Form1, таблица «Клиенты»

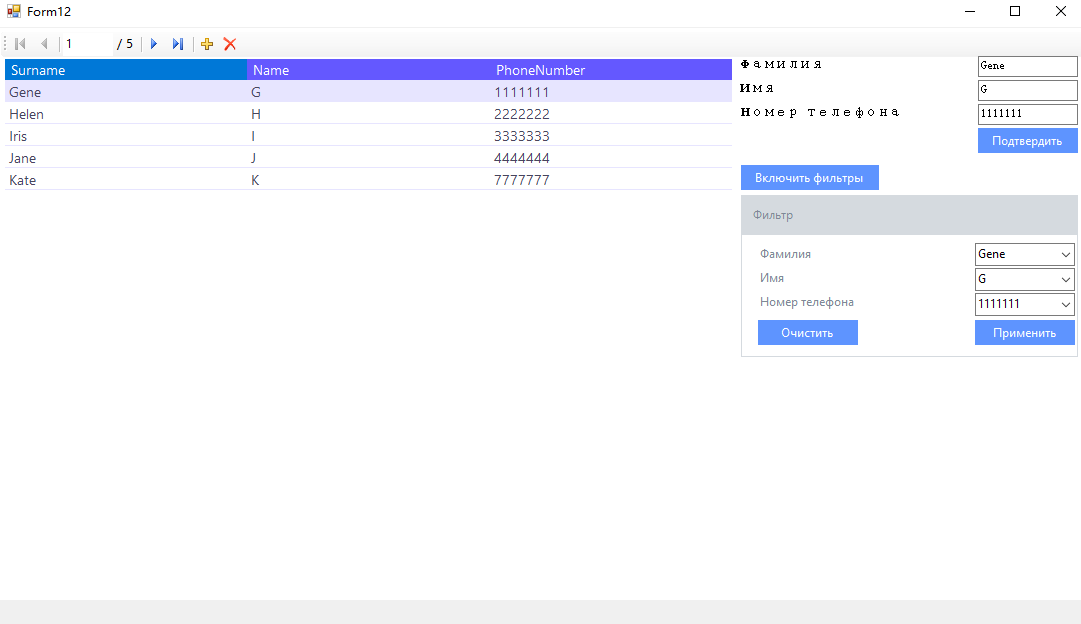


Рисунок 2.5 – форма Form1, таблица «Работники»

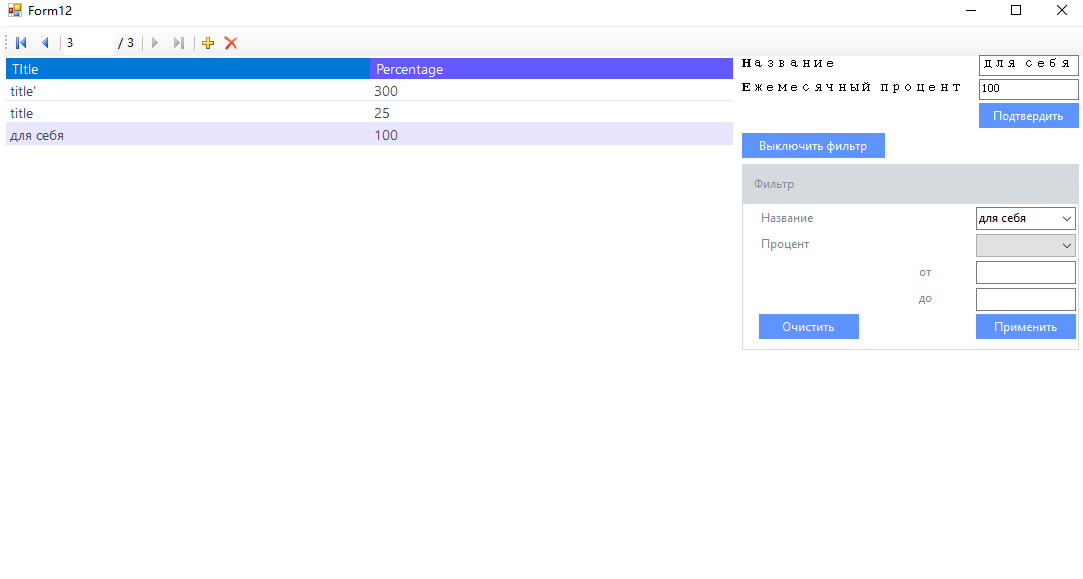


Рисунок 2.6 – форма Form1, таблица «Программы депозитов»

В выбранном поле «Программы депозитов» так же изменяется таблица в центре меню на соответствующую, а также поля справа на «Название» и «Ежемесячный процент», условия для фильтрации можно выбрать из выпадающего списка для названия, а для процентов нежно выбрать будет ли список равен указанному в поле, или промежуток от и до (Рисунок 2.7).

在选中的“存款计划”栏目中，菜单中心的表格也改为对应的栏目，右侧栏目为“名称”和“月利息”，过滤条件可从名称和百分比的下拉列表，您可以轻柔地选择列表是否等于字段中指定的值，或从和到的间隔（图 2.7）。

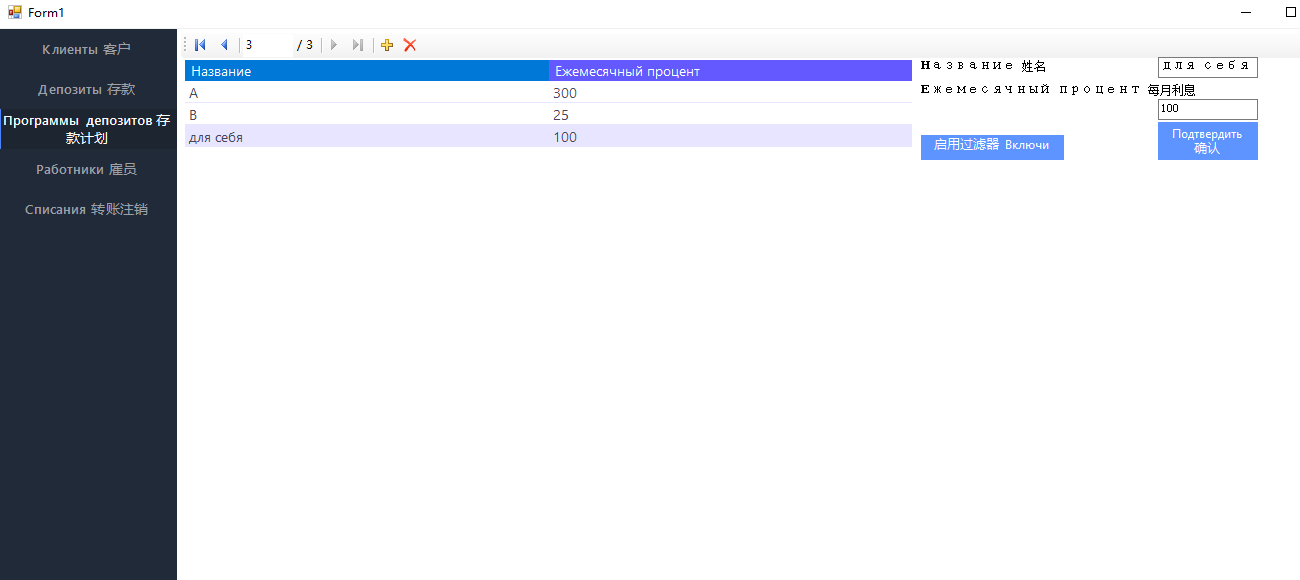


Рисунок 2.7 – меню программы с выбранным полем Программы депозитов

В поле «Работники» таблица меняемся в соответствии с названием, меняются поля справа на «Имя», «Фамилия», «Номер телефона». Фильтр измениться в соответствием с полями сверху, следует выбирать из выпадающих полей.

在“Employees”字段中，按照姓名更改表格，右侧字段更改为“First Name”、“Last Name”、“Phone Number”。 过滤器会根据上面的字段而变化，您应该从下拉字段中进行选择。

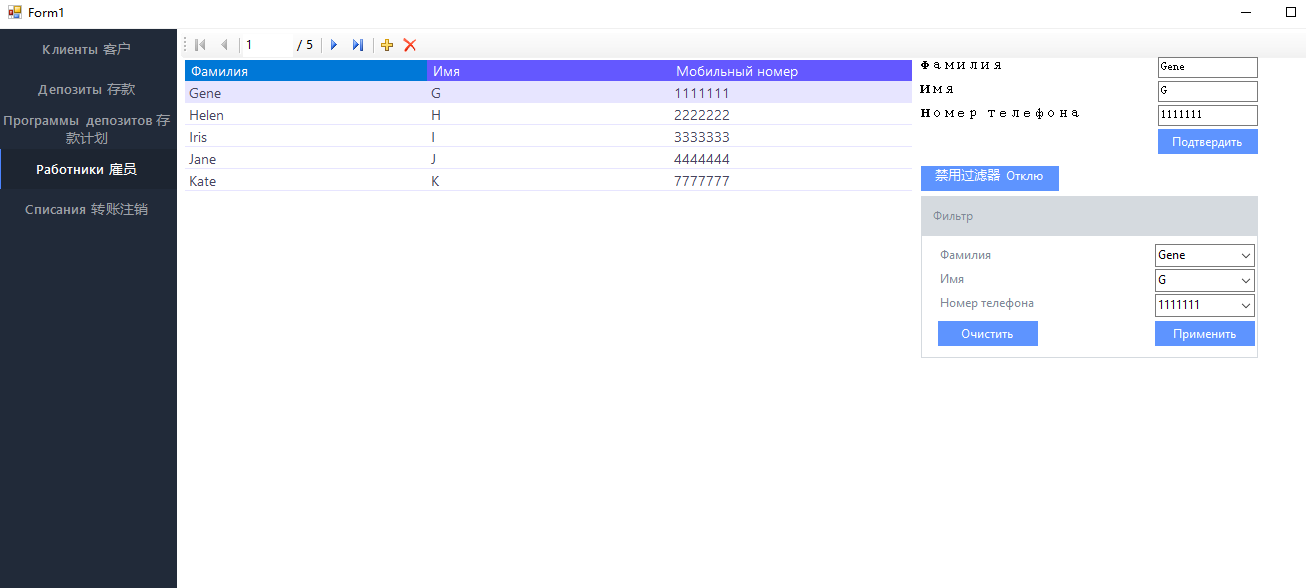


Рисунок 2.8 – меню программы с выбранным полем Работники

При выборе поля «Списания» таблица измениться на соответствующую таблицу. Поле справа, как и поле в фильтрах будет «Сумма списания», «Дата списания», «Идентификатор депозита», а в фильтрах Напротив поля «Дата списания» указывается промежуток времени, напротив «Сумма (от/до)» указывается промежуток суммы, а поле «Депозит» - выпадающее меню (Рисунок 2.9).

当您选择“核销”字段时，表格将更改为相应的表格。 右侧的字段，以及过滤器中的字段，将是“借记金额”、“借记日期”、“存款 ID”，并在过滤器中与“借记日期”字段相对，指示时间间隔，对面的“Amount（from/to）”表示金额区间，“Deposit”字段为下拉菜单（图2.9）。

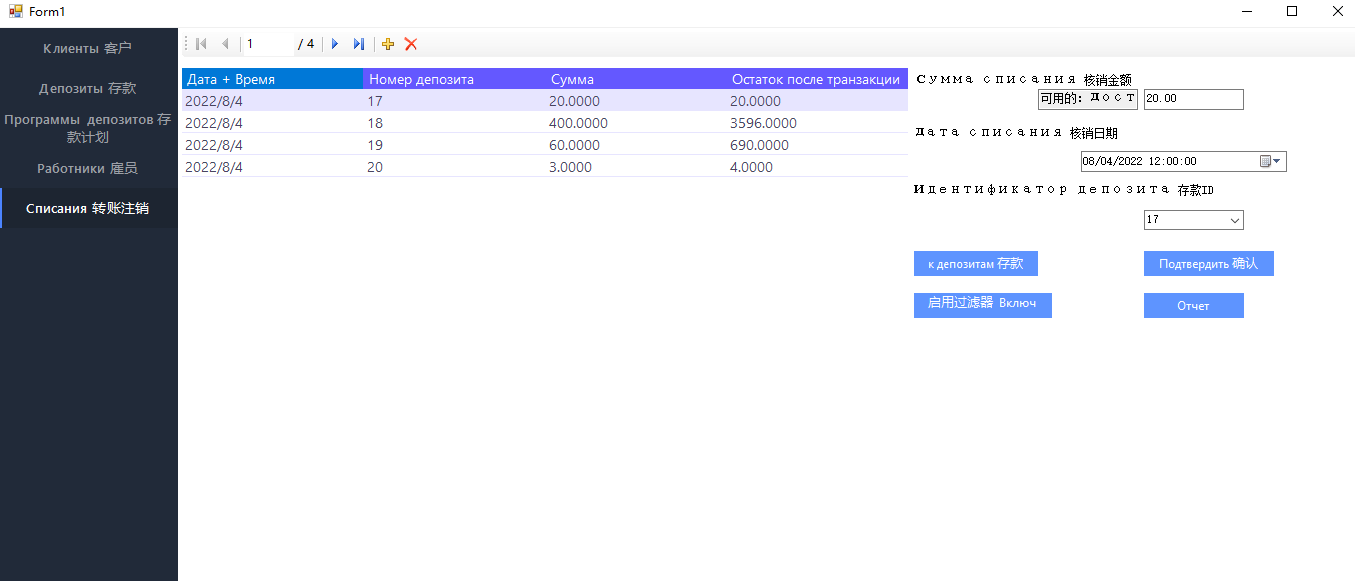


Рисунок 2.8 – меню программы с выбранным полем Списания

Важно отметить, что на странице «Депозиты» находиться кнопка «Отчёт». При нажатии на кнопку будет произведёт Report таблицы «Deposits» с примененными фильтрами (Рисунок 2.9).

需要注意的是，在“存款”页面上有一个“报告”按钮。 当您单击该按钮时，它将生成一个应用了过滤器的“存款”表的报告（图 2.9）。

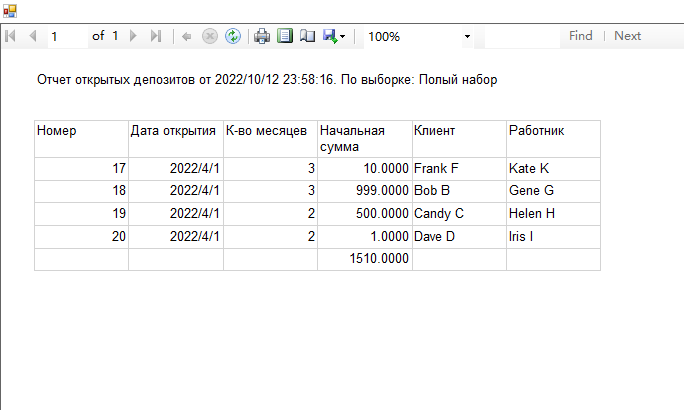


Рисунок 2.9 – отчёт по таблице «Deposits»

Также механизм формирования отчетов (Рисунок 2.10) реализован для таблицы Withdrawals. Кнопку для формирования и отображения отчета можно найти на вкладке «Списания».

此外，还为取款表实施了报告机制（图 2.10）。 用于生成和显示报告的按钮可以在“注销”选项卡上找到。

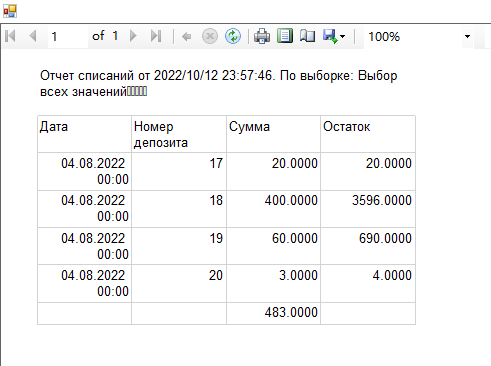


Рисунок 2.10 – отчёт по таблице «Withdrawals»

**4. Графический материал**

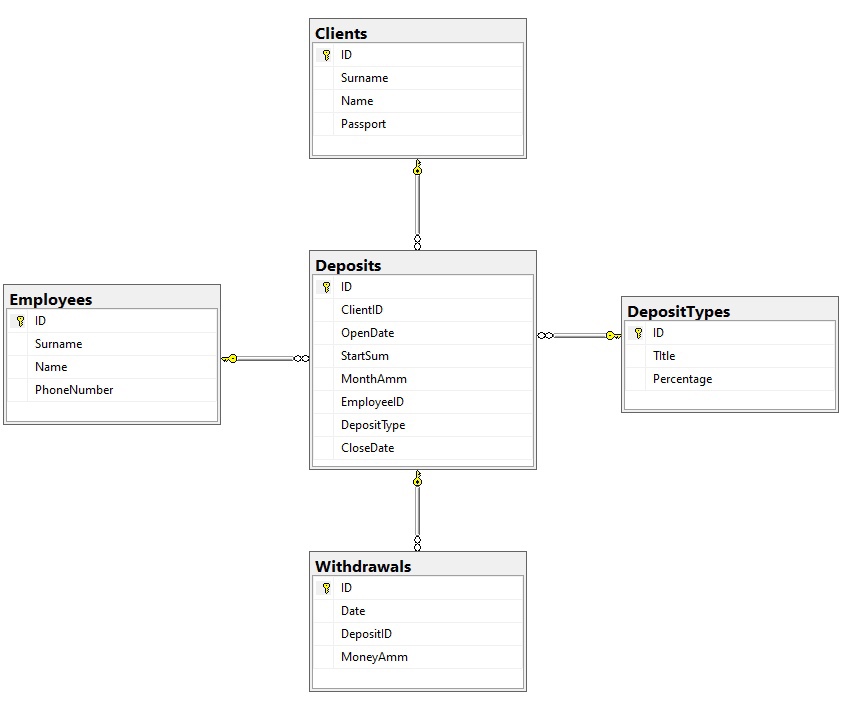


Рисунок 3.1 – Диаграмма базы данных Bank

**Заключение**

В ходе выполнения данного курсового проекта была спроектирована и создана база данных для банка, а также разработано приложение для взаимодействия с этой базой данных. Условием проекта было создать приложение и базу данных для Банка – Информационной системе учета вкладов. В базе данных было созданы таблицы для учёта всех типов вкладов, предоставляемые банком, список всех сотрудников и клиентов, список списаний и выдачей, а также таблица всех операций с заказдепозитами. Всё это удобно расположено для быстрого и удобного доступа в приложении. Следовательно – условие было выполнено.

Сверх этого приложение имеет приятный минималистичный визуальный стиль на основе GUNA Framework и её GUNA UI X C# библиотеки, на языке С# в визуальной среде программирования Microsoft Visual Studio 2019. База данных написана на языке SQL. Для работы с ним и базами данных используется рабочая среда Microsoft SQL Server. Их этих составляющих можно сделать вывод, что данное приложение можно будет запустить на практически любом устройстве, а если устройство поддерживает С# и базу данных, то полноценное и удобное пользование приложением гарантировано.

**Список литературы**

1. «Курсовое проектирование. Общие правила и требования оформления. СТП БНТУ 3.01 – 2003.» — Минск, 2003. – Яз. Русск.
2. Петкович Д. Microsoft® SQL Server™ 2012. Руководство для начинающих: Пер. с англ. — СПб.: БХВ-Петербург, 2013. — 816 с.: ил.
3. «Изучаем SQL» — Алан Бьюли. – Яз. Русск.
4. Wikipedia [Электронный ресурс]. – Wikipedia, 01.05.2022. –https://ru.wikipedia.org/wiki/SQL, свободный, - Загл. С экрана. – Яз. Русск..
5. sql-language [Электронный ресурс]. -sql-language, 23.04.2022 – Язык запросов SQL https://sql-language.ru/query-select.html, свободный, - Загл. С экрана. – Яз. Русск.

ПРИЛОЖЕНИЕ

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Windows.Forms;

namespace CourseWork

{

public partial class Main : Form

{

DataRowView clientsCurrentRow;

DataRowView depositsCurrentRow;

DataRowView depositTypesCurrentRow;

DataRowView employeesCurrentRow;

DataRowView withdrawalsCurrentRow;

Main2 secondForm;

public Main()

{

InitializeComponent();

}

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

{

this.depositsTableAdapter.Fill(this.theDataSet1.Deposits);

this.clientsTableAdapter.Fill(this.theDataSet1.Clients);

this.depositTypesTableAdapter.Fill(this.theDataSet1.DepositTypes);

this.employeesTableAdapter1.Fill(this.theDataSet1.Employees);

this.withdrawalsTableAdapter1.Fill(this.theDataSet1.Withdrawals);

employeesFilterEnableButton\_Click(null, null);

clientsFilterEnableButton\_Click(null, null);

guna2Button9\_Click(null, null);

guna2Button11\_Click(null, null);

guna2Button13\_Click(null, null);

}

private void guna2TabControl1\_SelectedIndexChanged(object sender, EventArgs e)

{

clientsBindingSource.RemoveFilter();

employeesBindingSource.RemoveFilter();

depositsBindingSource.RemoveFilter();

withdrawalsBindingSource.RemoveFilter();

depositTypesBindingSource.RemoveFilter();

}

#region Clients

bool isClientsFilterEnabled = true;

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

{

isClientsFilterEnabled = !isClientsFilterEnabled;

clientsFilterGroupBox.Visible = isClientsFilterEnabled;

if (isClientsFilterEnabled)

{

clientsFilterEnableButton.Text = "Отключить фильтр";

}

else

{

clientsFilterEnableButton.Text = "Включить фильтр";

}

clientsBindingSource.RemoveFilter();

}

private void clientsFiltersDropValuesButtonClick(object sender, EventArgs e)

{

clientsBindingSource.RemoveFilter();

clientsFilterSurnameCombobox.Text = "";

clientsFilterNameComboBox.Text = "";

clientsFilterPassportComboBox.Text = "";

}

private void clientsFilterApplyButtonClick(object sender, EventArgs e)

{

clientsBindingSource.RemoveFilter();

string name = clientsFilterNameComboBox.Text.Trim();

string surname = clientsFilterSurnameCombobox.Text.Trim();

string passport = clientsFilterPassportComboBox.Text.Trim();

var lst = new List<string>();

if (!string.IsNullOrWhiteSpace(name)) lst.Add($"Name like '%{name}%'");

if (!string.IsNullOrWhiteSpace(surname)) lst.Add($"Surname like '%{surname}%'");

if (!string.IsNullOrWhiteSpace(passport)) lst.Add($"Passport like '%{passport}%'");

clientsBindingSource.Filter = lst.Count > 0 ? string.Join(" and ", lst) : "";

}

private void ClientsBindingSource\_CurrentChanged(object sender, EventArgs e)

{

if (clientsCurrentRow != null) clientsCurrentRow.CancelEdit();

if (clientsBindingSource.Current == null) return;

clientsCurrentRow = (DataRowView)clientsBindingSource.Current;

clientsSurnameTextBox.Text = clientsCurrentRow["Surname"].ToString().Trim();

clientsNameTextBox.Text = clientsCurrentRow["Name"].ToString().Trim();

clientsPassportTextBox.Text = clientsCurrentRow["Passport"].ToString().Trim();

}

private void guna2Button1\_Click(object sender, EventArgs e)

{

if (clientsCurrentRow == null) return;

try

{

if (String.IsNullOrWhiteSpace(clientsSurnameTextBox.Text))

{

MessageBox.Show("Пустое значение поля 'Фамилия'");

return;

}

if (String.IsNullOrWhiteSpace(clientsNameTextBox.Text))

{

MessageBox.Show("Пустое значение поля 'Имя'");

return;

}

if (String.IsNullOrWhiteSpace(clientsPassportTextBox.Text))

{

MessageBox.Show("Пустое значение поля 'Пасспорт'");

return;

}

clientsCurrentRow["Name"] = clientsNameTextBox.Text.Trim();

clientsCurrentRow["Surname"] = clientsSurnameTextBox.Text.Trim();

clientsCurrentRow["Passport"] = clientsPassportTextBox.Text.Trim();

clientsCurrentRow.EndEdit();

clientsTableAdapter.Update(theDataSet1);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

this.clientsTableAdapter.Fill(this.theDataSet1.Clients);

}

private void tabPage1\_Enter(object sender, EventArgs e)

{

this.clientsTableAdapter.Fill(this.theDataSet1.Clients);

}

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

{

this.clientsTableAdapter.Update(this.theDataSet1.Clients);

}

#endregion

#region Employees

bool isEmployeesFilterEnabled = true;

private void guna2Button4\_Click(object sender, EventArgs e)

{

employeesFilterNameComboBox.Text = "";

employeesFilterSurnameComboBox.Text = "";

employeesFilterPhoneNumberComboBox.Text = "";

}

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

{

isEmployeesFilterEnabled = !isEmployeesFilterEnabled;

employeesFilterGroupBox.Visible = isEmployeesFilterEnabled;

if (isEmployeesFilterEnabled)

{

employeesFilterActivateButton.Text = "Отключить фильтр";

}

else

{

employeesFilterActivateButton.Text = "Включить фильтр";

}

employeesBindingSource.RemoveFilter();

}

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

{

employeesBindingSource.RemoveFilter();

string name = employeesFilterNameComboBox.Text.Trim();

string surname = employeesFilterSurnameComboBox.Text.Trim();

string phoneNumber = employeesFilterPhoneNumberComboBox.Text.Trim();

var lst = new List<string>();

if (!string.IsNullOrWhiteSpace(name)) lst.Add($"Name like '%{name}%'");

if (!string.IsNullOrWhiteSpace(surname)) lst.Add($"Surname like '%{surname}%'");

if (!string.IsNullOrWhiteSpace(phoneNumber)) lst.Add($"PhoneNumber like '%{phoneNumber}%'");

employeesBindingSource.Filter = string.Join(" and ", lst);

}

private void guna2Button2\_Click(object sender, EventArgs e)

{

if (employeesCurrentRow == null) return;

if (String.IsNullOrWhiteSpace(employeesSurnameTextBox.Text))

{

MessageBox.Show("Пустое значение поля 'Фамилия'");

return;

}

if (String.IsNullOrWhiteSpace(employeesNameTextBox.Text))

{

MessageBox.Show("Пустое значение поля 'Имя'");

return;

}

if (String.IsNullOrWhiteSpace(employeesPhoneNumberTextBox.Text))

{

MessageBox.Show("Пустое значение поля 'Номер телефона'");

return;

}

employeesCurrentRow["Surname"] = employeesSurnameTextBox.Text.Trim();

employeesCurrentRow["Name"] = employeesNameTextBox.Text.Trim();

employeesCurrentRow["PhoneNumber"] = employeesPhoneNumberTextBox.Text.Trim();

employeesCurrentRow.EndEdit();

try

{

employeesTableAdapter1.Update(theDataSet1);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

employeesTableAdapter1.Fill(theDataSet1.Employees);

}

private void employeesBindingSource\_CurrentChanged(object sender, EventArgs e)

{

if (employeesCurrentRow != null && employeesCurrentRow.IsNew)

{

for (int i = 0; i < employeesCurrentRow.Row.Table.Columns.Count; i++)

{

if (employeesCurrentRow[i] == null) return;

}

employeesCurrentRow.CancelEdit();

}

if (employeesBindingSource.Current == null) return;

employeesCurrentRow = (DataRowView)employeesBindingSource.Current;

employeesSurnameTextBox.Text = employeesCurrentRow["Surname"].ToString().Trim();

employeesNameTextBox.Text = employeesCurrentRow["Name"].ToString().Trim();

employeesPhoneNumberTextBox.Text = employeesCurrentRow["PhoneNumber"].ToString().Trim();

}

private void bindingNavigatorDeleteItem3\_Click(object sender, EventArgs e)

{

employeesTableAdapter1.Update(theDataSet1.Employees);

}

private void tabPage4\_Enter(object sender, EventArgs e)

{

employeesTableAdapter1.Fill(theDataSet1.Employees);

}

#endregion