**ЛАБОРАТОРНОЯ РАБОТА 5**

**«Приложение банк»**

**Цель:** Разработать приложение по вкладам

**Задание:** Разработать…

1. Главное окно программы
2. Калькулятор вкладов
3. Сравнение вкладов
4. Авторизация

# **Комплексное задание**

# **«Банк»**

**Задача** - разработка модуля демо-версии настольного приложения банка для открытия вкладов в автоматическом режиме под операционную систему Windows. Модуль должен позволять зарегистрированным клиентам осуществлять открытие вклада, прогнозировать доход и формировать договор. Более подробное описание представлено в последующих сладах.

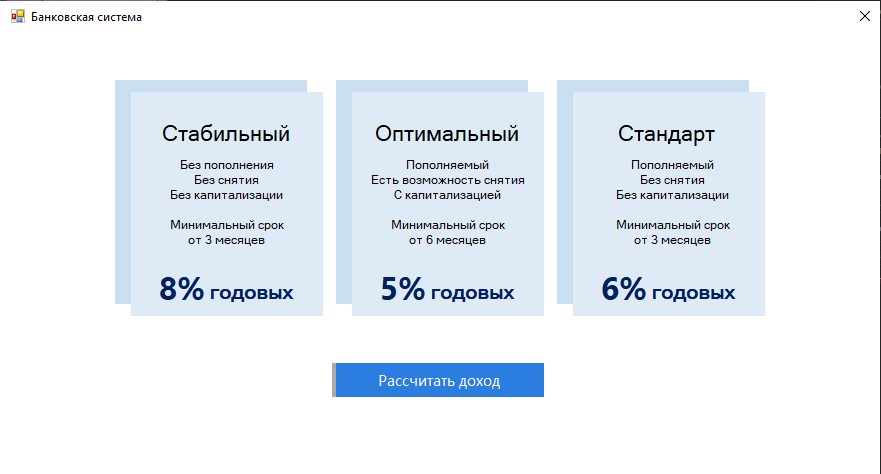


Рисунок 1 – Главное окно программы(Form1)

**1. Главное окно программы** - В банке «Деньги в каждый карман» имеется три тарифа вкладов: «Стабильный», «Оптимальный» и «Стандарт». Каждый тариф имеет отличия по процентной ставке, наличии капитализации, возможности снятия или пополнения вклада. Условия каждого вида вклада указаны справа. Капитализация процентов – это прибавление начисленных за период процентов к основной сумме вклада и последующее начисление дохода на сумму вклада и сумму прибавленных к нему процентов. Иными словами – начисление процентов на проценты. При запуске клиентского приложения пользователю должно открываться окно в соответствии с макетом. Данное окно служит для ознакомления клиента с текущими тарифами и условиями. Кнопка «Рассчитать доход» должна осуществлять переход на форму «Калькулятор». Переход должен осуществляться перелистыванием снизу вверх, как «Якорь».

using System;

using System.Windows.Forms;

namespace Bank

{

public partial class Form1 : Form

{

Base sql = new Base();

public Form1()

{

InitializeComponent();

//sql.InitializeDatabase();

}

private void nextSlide1\_Click(object sender, EventArgs e)

{

var form = new Form2();

form.Closed += (s, args) => this.Close();

form.Show();

this.Hide();

}

private void label3\_Click(object sender, EventArgs e)

{

}

}

}

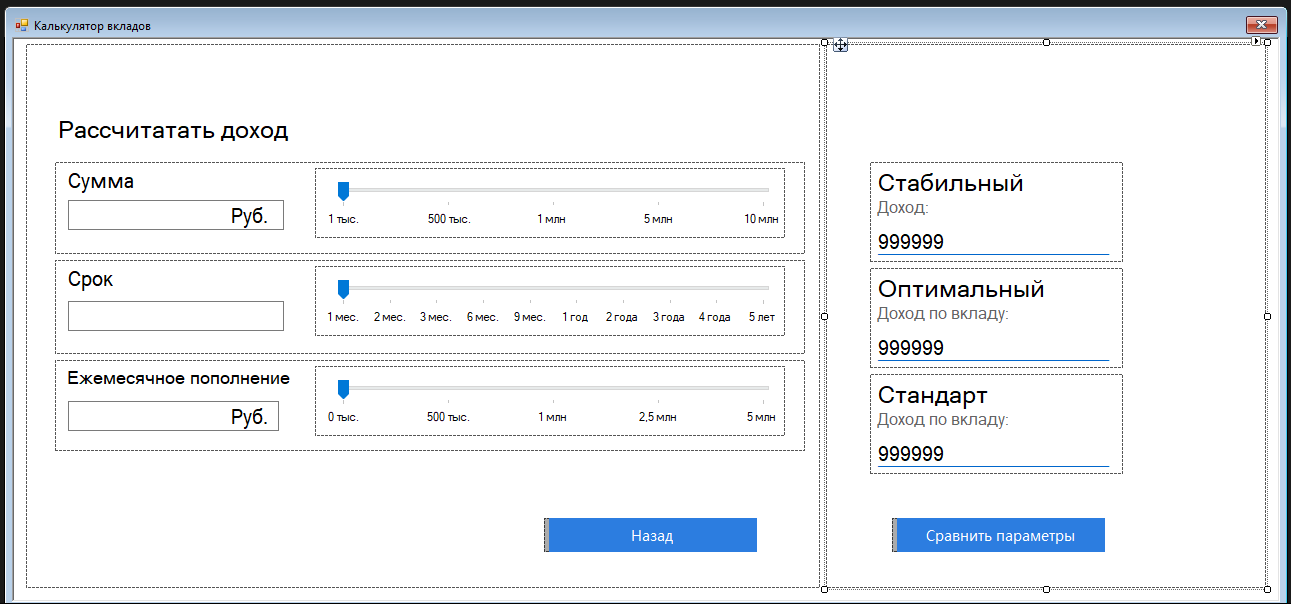


Рисунок 2 – Калькулятор вкладов(Form2)

**2. Калькулятор вкладов** - Данное окно позволят осуществить расчёт ожидаемого дохода по завершению сроков действия вкладов. Пользователю предоставляется возможность ввести сумму вклада с клавиатуры или с помощью полосы прокрутки. Также пользователь должен ввести срок действия вклада в днях и указать предполагаемую сумму для ежемесячного пополнения (в зависимости от тарифа). Справой части окна должен автоматически выводиться подсчитанный результат по каждому тарифу. В окне «Калькулятор вкладов» обязательно требуется предусмотреть защиту от некорректного ввода данных. По нажатию на кнопку «Сравнить параметры» программа должно перелистываться на окно «Сравнение вкладов».

using System;

using System.Windows.Forms;

namespace Bank

{

public partial class Form2 : Form

{

public int day;

public int mounth;

int sum;

float[] save = new float[6];

public Form2()

{

InitializeComponent();

textBox1.Text = "1000";

textBox2.Text = "1";

textBox3.Text = "0";

day = 30;

Timer tmr = new Timer();

tmr.Interval = 100;

tmr.Tick += Tmr\_Tick;

tmr.Start();

}

private void Tmr\_Tick(object sender, EventArgs e)

{

sum = Convert.ToInt32(textBox1.Text);

{ //1

float time = day;

float buff = Convert.ToInt32(textBox3.Text);

float procent = 0.08f;

float f = ((sum \* (1 + ((procent \* time) / (365)))) - sum);

if (f < 10)

{

finalCashText.Text = f.ToString("0") + " Руб.";

}

else if (f > 10)

{

finalCashText.Text = f.ToString("0,0") + " Руб.";

}

save[0] = f;

save[3] = (sum+f);

}

{ //2

float time = day;

float buff = Convert.ToInt32(textBox3.Text);

float procent = 5f;

float firstCash = sum \* (procent / 100)/365 \* time;

float nextCash = (sum + buff + firstCash) \* (procent / 100) / 365 \* (time);

float f = nextCash;

if (f < 10)

{

finalCashText2.Text = f.ToString("0") + " Руб.";

}

else if (f > 10)

{

finalCashText2.Text = f.ToString("0,0") + " Руб.";

}

save[1] = f;

save[4] = (sum +f+ buff \* (time/30));

}

{ //3

float time = day;

float buff = Convert.ToInt32(textBox3.Text);

float procent = 0.06f;

float f = (((sum+buff) \* (1 + ((procent \* time) / (365)))) - (sum+buff));

if (f < 10)

{

finalCashText3.Text = f.ToString("0") + " Руб.";

}

else if (f > 10)

{

finalCashText3.Text = f.ToString("0,0") + " Руб.";

}

save[2] = f;

save[5] = (sum +f+ buff \* (time/30));

}

}

public void UpdateDay(int i)

{

string[] stringBox = { "1 месяц", "2 месяца", "3 месяца", "6 месяцев", "9 месяцев", "1 год", "2 года", "3 года", "4 года", "5 лет" };

int[] dayBox = { 30, 60, 90, 180, 270, 365, 2 \* 365, 3 \* 365, 4 \* 365, 5 \* 365 };

int[] mounthBox = {1, 2, 3, 6, 9, 12, 24, 36, 48, 60};

textBox2.Text = stringBox[i];

day = dayBox[i]; mounth = mounthBox[i];

}

public void UpdateCash(int i)

{

string[] moneyBox = { "0", "500000" , "1000000" , "2500000", "5000000" };

textBox3.Text = moneyBox[i];

}

public void UpdateSum(int i)

{

string[] moneyBox = { "1000", "500000", "1000000", "5000000", "10000000" };

textBox1.Text = moneyBox[i];

}

private void sumTrackBar1\_Scroll(object sender, EventArgs e)

{

UpdateSum(sumTrackBar1.Value);

}

private void dayTrackBar1\_Scroll(object sender, EventArgs e)

{

UpdateDay(dayTrackBar1.Value);

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

UpdateCash(trackBar1.Value);

}

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

{

int sum = Convert.ToInt32(textBox1.Text);

if (sum < 1000)

{

sumTrackBar1.Value = 1;

textBox1.Text = 1000.ToString();

sum = 1\_000;

}

else if(sum >= 1000 && sum < 249\_999)

{

sumTrackBar1.Value = 1;

}

else if (sum >= 250\_000 && sum <= 500\_000)

{

sumTrackBar1.Value = 2;

}

else if (sum >= 500\_000 && sum <= 749\_999)

{

sumTrackBar1.Value = 2;

}

else if (sum >= 750\_000 && sum <= 1\_000\_000)

{

sumTrackBar1.Value = 3;

}

else if (sum >= 1\_000\_000 && sum <= 2\_499\_999)

{

sumTrackBar1.Value = 3;

}

else if (sum >= 2\_500\_000 && sum <= 5\_000\_000)

{

sumTrackBar1.Value = 4;

}

else if (sum >= 5\_000\_000 && sum <= 7\_499\_999)

{

sumTrackBar1.Value = 4;

}

else if (sum >= 7\_500\_000 && sum <= 10\_000\_000)

{

sumTrackBar1.Value = 5;

}

else if (sum >= 10\_000\_000)

{

sumTrackBar1.Value = 5;

textBox1.Text = 10000000.ToString();

sum = 10\_000\_000;

}

}

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

{

int sum = Convert.ToInt32(textBox3.Text);

if (sum < 0)

{

trackBar1.Value = 1;

textBox1.Text = 0.ToString();

sum = 0;

}

else if (sum >= 0 && sum < 249\_999)

{

trackBar1.Value = 1;

}

else if (sum >= 250\_000 && sum <= 500\_000)

{

trackBar1.Value = 2;

}

else if (sum >= 500\_000 && sum <= 749\_999)

{

trackBar1.Value = 2;

}

else if (sum >= 750\_000 && sum <= 1\_000\_000)

{

trackBar1.Value = 3;

}

else if (sum >= 1\_000\_000 && sum <= 1\_249\_999)

{

trackBar1.Value = 3;

}

else if (sum >= 1\_250\_000 && sum <= 2\_499\_999)

{

trackBar1.Value = 4;

}

else if (sum >= 2\_500\_000 && sum <= 3\_749\_999)

{

trackBar1.Value = 4;

}

else if (sum >= 5\_000\_000)

{

trackBar1.Value = 5;

textBox3.Text = 5\_000\_000.ToString();

sum = 5\_000\_000;

}

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))

{

e.Handled = true;

}

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))

{

e.Handled = true;

}

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))

{

e.Handled = true;

}

}

private void backButton1\_Click(object sender, EventArgs e)

{

var form = new Form1();

form.Closed += (s, args) => this.Close();

form.Show();

this.Hide();

}

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

{

mounth = day / 30;

Form3 form = new Form3(save[0], save[1], save[2], save[3], save[4], save[5], mounth, sum);

form.Show();

}

}

}

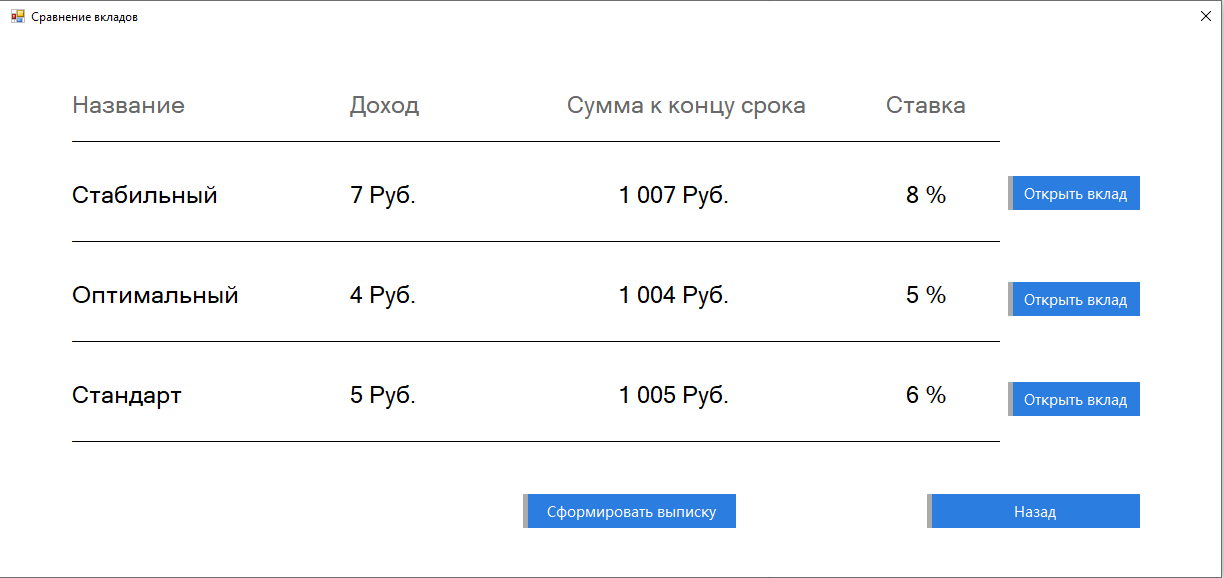


Рисунок 3 – Сравнение вкладов(Form3)

**3. Сравнение вкладов -** Данное окно осуществляет вывод результата расчётов вкладов с окна «Калькулятор вкладов». Вывод информации должен быть в табличном виде. Нажатие пользователем на кнопу «Открыть вклад» позволяет пользователю приложения перейти на форму «Авторизация», также нажатие на данную кнопку осуществляется выбор одного из 3 тарифов для формирования договора. По нажатию на кнопку «Сформировать выписку» должен формироваться документ в формате pdf, дублирующий информацию с текущей формы.

using System;

using System.Drawing;

using System.IO;

using System.Windows.Forms;

using iTextSharp.text.pdf;

using iTextSharp.text;

using System.Diagnostics;

namespace Bank

{

public partial class Form3 : Form

{

int sum;

int mounth;

public Form3(float a, float b, float c, float a1, float b1, float c1, int \_mounth, int \_sum)

{

InitializeComponent();

if(a < 10)

cash1.Text = a.ToString("0") + " Руб.";

else

cash1.Text = a.ToString("0,0") + " Руб.";

if (b < 10)

cash2.Text = b.ToString("0") + " Руб.";

else

cash2.Text = b.ToString("0,0") + " Руб.";

if (c < 10)

cash3.Text = c.ToString("0") + " Руб.";

else

cash3.Text = c.ToString("0,0") + " Руб.";

finalCash1.Text = a1.ToString("0,0") + " Руб.";

finalCash2.Text = b1.ToString("0,0") + " Руб.";

finalCash3.Text = c1.ToString("0,0") + " Руб.";

sum = \_sum;

mounth = \_mounth;

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))

{

e.Handled = true;

}

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))

{

e.Handled = true;

}

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))

{

e.Handled = true;

}

}

private void backButton1\_Click(object sender, EventArgs e)

{

this.Close();

}

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

{

//Генерация pdf файла

string path = AppContext.BaseDirectory + @"\jpg.jpg";

string path2 = AppContext.BaseDirectory + @"\pdf.pdf";

Bitmap printscreen = GetControlScreenshot(panel1);

printscreen.Save(path, System.Drawing.Imaging.ImageFormat.Jpeg);

Document document = new Document();

using (var stream = new FileStream(path2, FileMode.Create, FileAccess.Write, FileShare.None))

{

using (var imageStream = new FileStream(path, FileMode.Open, FileAccess.Read, FileShare.ReadWrite))

{

var image = iTextSharp.text.Image.GetInstance(imageStream);

PdfWriter writer = PdfWriter.GetInstance(document, stream);

document.SetPageSize(new iTextSharp.text.Rectangle(panelMain.Size.Width + document.LeftMargin + document.RightMargin, panelMain.Size.Height + document.TopMargin + document.BottomMargin));

document.Open();

document.Add(image);

document.Close();

}

}

Process.Start(path2);

}

//Получение картинки для pdf файла

public Bitmap GetControlScreenshot(Control control)

{

Size szCurrent = control.Size;

control.AutoSize = true;

Bitmap bmp = new Bitmap(control.Width, control.Height);

control.DrawToBitmap(bmp, control.ClientRectangle);

control.AutoSize = false;

control.Size = szCurrent;

return bmp;

}

private void contrbutionButton1\_Click(object sender, EventArgs e)

{

string procent = "8%";

Form4 form = new Form4(sum, mounth, procent);

form.Show();

}

private void contrbutionButton2\_Click(object sender, EventArgs e)

{

string procent = "5%";

Form4 form = new Form4(sum, mounth, procent);

form.Show();

}

private void contrbutionButton3\_Click(object sender, EventArgs e)

{

string procent = "6%";

Form4 form = new Form4(sum, mounth, procent);

form.Show();

}

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

{

var form = new Form2();

form.Closed += (s, args) => this.Close();

form.Show();

this.Hide();

}

}

}

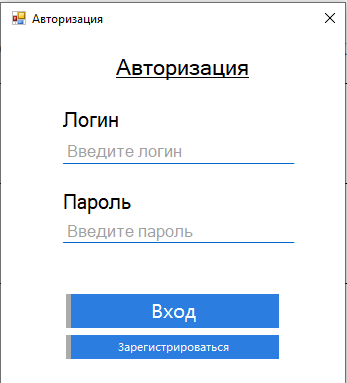


Рисунок 4 – Авторизация(Form4)

**4. Авторизация** - Данное окно позволяет осуществить подтверждение открытия вклада с помощью авторизации. Список логинов и паролей проверяется с базой данных. При успешной авторизации должен формироваться договор по шаблону в формате Word. Шаблон представлен в папке «Ресурсы». Необходимые заполняемые значения в документе выделены желтым цветом. Информация для заполнения должна браться из базе данных.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Bank

{

public partial class Form4 : Form

{

Base \_SQLBase = new Base();

WordsGeneration \_WordsGeneration = new WordsGeneration();

public Form4(int \_sum, int \_date, string \_procent)

{

InitializeComponent();

sum = \_sum;

date = \_date;

procent = \_procent;

}

string procent; int sum, cash, date;

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

{

if(loginBox.Text != null)

{

visOff1.Text = "";

}

else

{

}

}

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

{

if (passwordBox.Text != null)

{

visOff2.Text = "";

}

}

private void visOff1\_Click(object sender, EventArgs e)

{

}

private void visOff2\_Click(object sender, EventArgs e)

{

}

private void label9\_Click(object sender, EventArgs e)

{

}

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

{

Form5 form = new Form5();

form.Show();

}

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

{

\_SQLBase.SearchProduct(loginBox.Text, passwordBox.Text);

string login = \_SQLBase.LoginReturn();

string password = \_SQLBase.PasswordReturn();

string fio = \_SQLBase.FIOReturn();

if (login != "" && password != "")

{

\_WordsGeneration.WordGeneration(fio, sum, cash, procent, date);

}

\_SQLBase.DataClear();

}

}

}

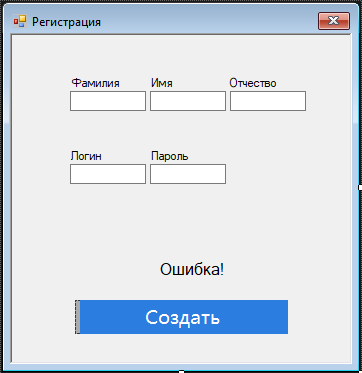


Рисунок 5 – Регистрация(Form5)

**5. Регистрация -** Это окно предназначено для регистрации пользователей, желающих открыть вклад. При вводе логина и пароля происходит проверка с базой данных на уникальность данных. После успешной регистрации данные пользователя сохраняются в базе.

Регистрация обеспечивает возможность последующей авторизации для доступа к функционалу открытия вклада. Она является первым шагом для получения доступа к услугам банка

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Bank

{

public partial class Form5 : Form

{

Base sql = new Base();

public Form5()

{

InitializeComponent();

sql.InitializeDatabase();

errorBox.Text = "";

}

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

{

}

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

{

}

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

{

}

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

{

}

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

{

}

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

{

string cheak = "";

if (loginBox.Text != "" && passwordBox.Text != "" && fBox.Text != "" && iBox.Text != "" && oBox.Text != "")

{

sql.AddNewUser(loginBox.Text, passwordBox.Text, fBox.Text +"\_"+ iBox.Text + "\_" + oBox.Text);

cheak = sql.ErrorInfo();

if (cheak == "")

{

this.Close();

}

else

{

errorBox.Text = cheak;

}

}

else

{

errorBox.Text = "Ошибка при заполнении данных";

}

}

}

}

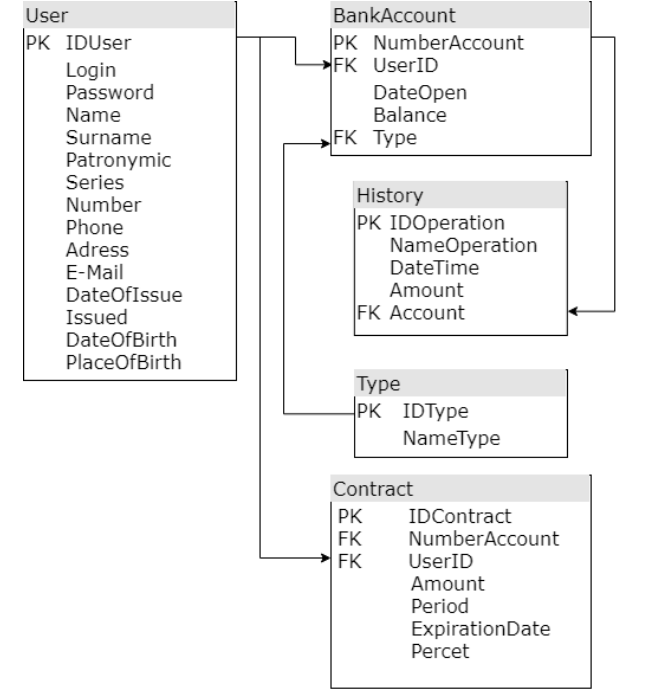


Рисунок 6 – ERD модель базы данных

using System;

using System.Data.SQLite;

namespace Bank

{

internal class Base

{

static string connectionString = "Data Source=accounts.db;Version=3;";

string errorText;

string \_login;

string \_password;

string \_fio;

public void InitializeDatabase()

{

using (SQLiteConnection connection = new SQLiteConnection(connectionString))

{

connection.Open();

string createTable = "CREATE TABLE \"Base\" (\r\n\t\"Login\"\tTEXT NOT NULL UNIQUE,\r\n\t\"Password\"\tTEXT NOT NULL,\r\n\t\"FIO\"\tTEXT NOT NULL,\r\n\t\"Sum\"\tINTEGER,\r\n\t\"Day\"\tINTEGER,\r\n\t\"AddSum\"\tINTEGER,\r\n\t\"Procent\"\tINTEGER\r\n);";

using (SQLiteCommand command = new SQLiteCommand(createTable, connection))

{

try

{

command.ExecuteNonQuery();

}

catch (SQLiteException)

{

}

}

}

}

public void AddNewUser(string login, string password, string fio)

{

errorText = "";

using (SQLiteConnection connection = new SQLiteConnection(connectionString))

{

connection.Open();

// string insertProductQuery = "INSERT INTO Base (Login, Password, FIO, Sum, Day, AddSum, Procent) VALUES (@Login, @Password, @FIO, @Sum, @Day, @AddSum, @Procent)";

string insertQuery = "INSERT INTO Base (Login, Password, FIO) VALUES (@Login, @Password, @FIO)";

using (SQLiteCommand command = new SQLiteCommand(insertQuery, connection))

{

command.Parameters.AddWithValue("@Login", login);

command.Parameters.AddWithValue("@Password", password);

command.Parameters.AddWithValue("@FIO", fio);

//, float sum, int day, int addSum, float procent

//command.Parameters.AddWithValue("@Sum", sum);

//command.Parameters.AddWithValue("@Day", day);

//command.Parameters.AddWithValue("@AddSum", addSum);

//command.Parameters.AddWithValue("@Procent", procent);

try

{

command.ExecuteNonQuery();

}

catch (SQLiteException)

{

errorText = "Пользователь уже существует!";

}

}

}

}

public string ErrorInfo()

{

return errorText;

}

public void SearchProduct(string login, string password)

{

string userLogin = login;

string userPassword = password;

using (SQLiteConnection connection = new SQLiteConnection(connectionString))

{

connection.Open();

string searchUser = "SELECT \* FROM Base WHERE Login = @Login AND Password = @Password";

using (SQLiteCommand command = new SQLiteCommand(searchUser, connection))

{

command.Parameters.AddWithValue("@Login", userLogin);

command.Parameters.AddWithValue("@Password", password);

using (SQLiteDataReader reader = command.ExecuteReader())

{

while (reader.Read())

{

\_login = $"{reader["Login"]}";

\_password = $"{reader["Password"]}";

\_fio = $"{reader["FIO"]}";

}

}

}

}

}

public string LoginReturn()

{

return \_login;

}

public string PasswordReturn()

{

return \_password;

}

public string FIOReturn()

{

return \_fio;

}

public void DataClear()

{

\_login = "";

\_password = "";

\_fio = "";

}

}

}