Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

Лабораторная работа №8

По дисциплине

“Объектно-ориентированное программирование”

# Тема “Работа с файлами”

Вариант 15

Выполнил работу

студент группы РИС-19-1б

Миннахметов Э.Ю.

Проверила

доцент кафедры ИТАС

Викентьева О.Л.

Работу выполнил:

Пермь 2020

**1 АНАЛИЗ ЗАДАЧИ**

**1.1 Постановка задачи**

Написать Windows-приложение для работы с простой базой данных, хранящей информацию об объекте на внешнем носителе. Приложение должно выполнять следующие функции:

1. Создание базы данных, содержащей записи указанного формата.
2. Просмотр базы данных.
3. Удаление элементов из базы данных (по ключу/ по номеру).
4. Корректировка элементов в базе данных (по ключу / по номеру).
5. Добавление элементов в базу данных (в начало / в конец/ с заданным номером).
6. Выполнение задания, указанного в варианте.

**Вариант 15**

Типизированный файл содержит данные о клиентах банка, получивших кредит (ФИО клиента, сумма кредита, вид кредита, срок, на который выдан кредит). Клиент может взять несколько кредитов.

* 1. Программа должна
     1. Добавлять, удалять, корректировать, позволять просматривать записи файлов.
  2. Выполнять:
     1. поиск клиентов, взявших кредит на N месяцев;
     2. поиск клиентов по шаблону (3 первые буквы фамилии);

**1.2 Требуемые инструменты**

Для реализации графического интерфейса будет использован Microsoft Windows Presentation Foundation (WPF), поскольку является в настоящее время более актуальной технологией.

**2 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ**

**2.1 Client.cs – класс хранения данных о конкретном клиенте**

|  |  |
| --- | --- |
| public string Name | Автосвойство для хранения имени |
| public int Summ | Автосвойство суммы кредита |
| public string Type | Автосвойство типа кредита |
| public string Period | Автосвойство периода кредита |
| public bool IsBeginOn(string name) | Проверка совпадения начальных символов автосвойства Name со всеми символами параметра name |

**2.2 ClientList.cs – класс-коллекция для множества клиентов**

|  |  |
| --- | --- |
| public ClientList GetListByName(string name) | Получение списка клиентов, чьи имена начинаются на name |
| public ClientList GetListByPeriod(string period) | Получение списка клиентов, чьи сроки кредитов равны period |

**2.3 MainWindow.xaml.cs – обработчик главного окна**

|  |  |
| --- | --- |
| protected override void OnKeyDown(KeyEventArgs e) | Обработка нажатий на клавиши |
| private void MoveClient(int oldIndex, int newIndex) | Перемещение поля в списке |
| private void ClearFields() | Очистка полей ввода |
| private void ButtonToUpdate(int index) | Перевод обработчика нажатия на кнопку в режим обновления |
| private void ButtonToAdd() | Перевод обработчика нажатия на кнопку в режим добавления |
| private void UpdateClient(int index) | Обновление по индексу |
| private void DeleteClient(int index) | Удаление по индексу |
| private void ButtonAddClient(object sender, RoutedEventArgs e) | Обработчик кнопки на обновление данных о клиенте |
| private void ButtonUpdateClient(object sender, RoutedEventArgs e) | Обработчик кнопки на добавление клиента |
| private void ClickSave(object sender, RoutedEventArgs e) | Обработка кнопки сохранения |
| private void ClickLoad(object sender, RoutedEventArgs e) | Обработка кнопки загрузки |
| private void ClickFindName(object sender, RoutedEventArgs e) | Обработка кнопки поиска по имени |
| private void ClickFindPeriod(object sender, RoutedEventArgs e) | Обработка кнопки поиска по сроку |
| private void ClickReset(object sender, RoutedEventArgs e) | Обработка кнопки сброса поиска |

**3 ТЕСТИРОВАНИЕ**

**ап**

**4 ИСХОДНЫЙ КОД ПРОГРАММЫ**

**4.1 Client.cs**

using System;

namespace Lab8

{

[Serializable]

public class Client

{

public string Name { get; set; }

public int Summ { get; set; }

public string Type { get; set; }

public string Period { get; set; }

public bool IsBeginOn(string name)

{

int N = Name.Length;

int n = name.Length;

if (N >= n)

{

bool flag = true;

for (int i = 0; i < n && flag; i++)

flag = Name[i] == name[i];

return flag;

}

else

return false;

}

}

}

**4.2 ClientList.cs**

using System;

using System.Collections.Generic;

using System.Collections.ObjectModel;

namespace Lab8

{

[Serializable]

public class ClientList : ObservableCollection<Client>

{

public ClientList()

: base() { }

public ClientList(List<Client> collection)

: base(collection) { }

public ClientList(IEnumerable<Client> collection)

: base(collection) { }

public ClientList GetListByName(string name)

{

ClientList list = new ClientList();

for (int i = 0, n = Count; i < n; ++i)

if (this[i].IsBeginOn(name))

list.Add(this[i]);

return list;

}

public ClientList GetListByPeriod(string period)

{

ClientList list = new ClientList();

for (int i = 0, n = Count; i < n; ++i)

if (this[i].Period == period)

list.Add(this[i]);

return list;

}

}

}

**4.3 MainWindow.xaml.cs**

using System.IO;

using System.Windows;

using Microsoft.Win32;

using System.Windows.Controls;

using System.Runtime.Serialization.Formatters.Binary;

using System.Windows.Input;

namespace Lab8

{

public partial class MainWindow : Window

{

private ClientList clients { get; set; }

private int updateIndex = -1;

public MainWindow()

{

InitializeComponent();

clients = new ClientList();

ListClient.ItemsSource = clients;

}

protected override void OnKeyDown(KeyEventArgs e)

{

int index = ListClient.SelectedIndex;

if (index != -1)

{

switch (e.Key)

{

case Key.W:

MoveClient(index, index - 1);

break;

case Key.S:

MoveClient(index, index + 1);

break;

case Key.E:

UpdateClient(index);

break;

case Key.D:

case Key.Delete:

DeleteClient(index);

break;

}

}

}

private void MoveClient(int oldIndex, int newIndex)

{

if (newIndex >= 0 && newIndex < clients.Count)

{

ButtonToAdd();

clients.Move(oldIndex, newIndex);

ListClient.SelectedIndex = newIndex;

}

}

private void ClearFields()

{

ClientName.Text = "";

ClientSumm.Text = "";

ClientType.Text = "Микрозайм";

ClientPeriod.Text = "3 м";

}

private void ButtonToUpdate(int index)

{

if (updateIndex == -1)

{

ButtonClient.Click -= ButtonAddClient;

ButtonClient.Click += ButtonUpdateClient;

ButtonClient.Content = "Обновить";

}

updateIndex = index;

}

private void ButtonToAdd()

{

if (updateIndex != -1)

{

updateIndex = -1;

ClearFields();

ButtonClient.Click -= ButtonUpdateClient;

ButtonClient.Click += ButtonAddClient;

ButtonClient.Content = "Добавить";

}

}

private void UpdateClient(int index)

{

ButtonToUpdate(index);

Client client = clients[updateIndex];

ClientName.Text = client.Name;

ClientSumm.Text = client.Summ.ToString();

ClientType.Text = client.Type;

ClientPeriod.Text = client.Period;

}

private void DeleteClient(int index)

{

ButtonToAdd();

clients.RemoveAt(index);

ListClient.ItemsSource = clients;

}

private void ButtonAddClient(object sender, RoutedEventArgs e)

{

if (int.TryParse(ClientSumm.Text, out int summ) && summ > 0)

{

clients.Add(new Client

{

Name = ClientName.Text,

Summ = summ,

Type = (ClientType.SelectedItem as ComboBoxItem).Content.ToString(),

Period = (ClientPeriod.SelectedItem as ComboBoxItem).Content.ToString()

});

ClearFields();

}

else

MessageBox.Show("Некорректное значение поля суммы!");

}

private void ButtonUpdateClient(object sender, RoutedEventArgs e)

{

if (int.TryParse(ClientSumm.Text, out int summ) && summ > 0)

{

Client client = clients[updateIndex];

client.Name = ClientName.Text;

client.Summ = summ;

client.Type = (ClientType.SelectedItem as ComboBoxItem).Content.ToString();

client.Period = (ClientPeriod.SelectedItem as ComboBoxItem).Content.ToString();

ButtonToAdd();

ListClient.Items.Refresh();

}

else

MessageBox.Show("Некорректное значение поля суммы!");

}

private void ClickSave(object sender, RoutedEventArgs e)

{

SaveFileDialog dialog = new SaveFileDialog();

dialog.Filter = "Документ CLI (\*.cli)|\*.cli";

if (dialog.ShowDialog() == true)

{

using (FileStream fstream = new FileStream(dialog.FileName, FileMode.OpenOrCreate))

{

BinaryFormatter buffer = new BinaryFormatter();

buffer.Serialize(fstream, clients);

fstream.Close();

}

}

}

private void ClickLoad(object sender, RoutedEventArgs e)

{

OpenFileDialog dialog = new OpenFileDialog();

dialog.Filter = "Документ CLI (\*.cli)|\*.cli";

if (dialog.ShowDialog() == true)

{

using (FileStream fstream = new FileStream(dialog.FileName, FileMode.Open))

{

BinaryFormatter buffer = new BinaryFormatter();

clients = (ClientList)buffer.Deserialize(fstream);

fstream.Close();

ListClient.ItemsSource = clients;

}

}

}

private void ClickFindName(object sender, RoutedEventArgs e)

=> ListClient.ItemsSource = clients.GetListByName(BoxFindName.Text);

private void ClickFindPeriod(object sender, RoutedEventArgs e)

=> ListClient.ItemsSource = clients.GetListByPeriod(

(BoxFindPeriod.SelectedItem as ComboBoxItem).Content.ToString());

private void ClickReset(object sender, RoutedEventArgs e)

=> ListClient.ItemsSource = clients;

}

}

**4.4 MainWindow.xaml**

<Window x:Class="Lab8.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d"

Title="Лабораторная работа №8"

Height="450" Width="800">

<ScrollViewer VerticalScrollBarVisibility="Auto">

<StackPanel>

<Menu>

<MenuItem Header="Сохранить" Click="ClickSave" />

<MenuItem Header="Загрузить" Click="ClickLoad" />

</Menu>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<Label Grid.Row="0" Grid.Column="0" HorizontalAlignment="Right">ФИО</Label>

<TextBox Grid.Row="0" Grid.Column="1" Grid.ColumnSpan="3" Name="BoxFindName" />

<Button Grid.Row="0" Grid.Column="4" Click="ClickFindName">Поиск</Button>

<Label Grid.Row="0" Grid.Column="5" HorizontalAlignment="Right">Срок</Label>

<ComboBox Grid.Row="0" Grid.Column="6" Name="BoxFindPeriod">

<ComboBoxItem IsSelected="True" Content="3 м" />

<ComboBoxItem Content="6 м" />

<ComboBoxItem Content="9 м" />

<ComboBoxItem Content="1 г" />

<ComboBoxItem Content="1,5 г" />

<ComboBoxItem Content="2 г" />

<ComboBoxItem Content="3 г" />

<ComboBoxItem Content="5 г" />

</ComboBox>

<Button Grid.Row="0" Grid.Column="7" Click="ClickFindPeriod">Поиск</Button>

<Button Grid.Row="0" Grid.Column="9" Click="ClickReset">Сбросить</Button>

</Grid>

<ListView Name="ListClient">

<ListView.View>

<GridView>

<GridViewColumn DisplayMemberBinding="{Binding Path=Name}">ФИО</GridViewColumn>

<GridViewColumn DisplayMemberBinding="{Binding Path=Summ}">Сумма</GridViewColumn>

<GridViewColumn DisplayMemberBinding="{Binding Path=Type}">Тип</GridViewColumn>

<GridViewColumn DisplayMemberBinding="{Binding Path=Period}">Срок</GridViewColumn>

</GridView>

</ListView.View>

<ListView.ItemTemplate>

<DataTemplate>

<StackPanel Margin="5">

<TextBlock FontSize="16" Text="{Binding Path=Name}" HorizontalAlignment="Center" />

<TextBlock FontSize="16" Text="{Binding Path=Summ}" HorizontalAlignment="Center" />

<TextBlock FontSize="16" Text="{Binding Path=Type}" HorizontalAlignment="Center" />

<TextBlock FontSize="16" Text="{Binding Path=Period}" HorizontalAlignment="Center" />

</StackPanel>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<Grid Width="250">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="50"></ColumnDefinition>

<ColumnDefinition Width="200"></ColumnDefinition>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<Label Grid.Column="0" Grid.Row="0">ФИО</Label>

<TextBox Grid.Column="1" Grid.Row="0" Name="ClientName"></TextBox>

<Label Grid.Column="0" Grid.Row="1">Сумма</Label>

<TextBox Grid.Column="1" Grid.Row="1" Name="ClientSumm"></TextBox>

<Label Grid.Column="0" Grid.Row="2">Тип</Label>

<ComboBox Grid.Column="1" Grid.Row="2" Name="ClientType">

<ComboBoxItem IsSelected="True" Content="Микрозайм" />

<ComboBoxItem Content="Потребительский" />

<ComboBoxItem Content="Автокредит" />

<ComboBoxItem Content="Ипотечный" />

</ComboBox>

<Label Grid.Column="0" Grid.Row="3">Срок</Label>

<ComboBox Grid.Column="1" Grid.Row="3" Name="ClientPeriod">

<ComboBoxItem IsSelected="True" Content="3 м" />

<ComboBoxItem Content="6 м" />

<ComboBoxItem Content="9 м" />

<ComboBoxItem Content="1 г" />

<ComboBoxItem Content="1,5 г" />

<ComboBoxItem Content="2 г" />

<ComboBoxItem Content="3 г" />

<ComboBoxItem Content="5 г" />

</ComboBox>

<Button x:Name="ButtonClient" Grid.Column="0" Grid.Row="4" Grid.ColumnSpan="2" Click="ButtonAddClient">Добавить</Button>

</Grid>

</StackPanel>

</ScrollViewer>

</Window>