**Компьютерная академия «ШАГ»**

**Днепропетровский филиал**

**Кафедра Разработки программного обеспечения**

**КУРСОВОЙ ПРОЕКТ**

**“Разработка программного обеспечения**

**для банковской системы”**

**Студента группы 14-П-2**

**Лыски Л.Ю.**

**Руководитель:  
Стражко Е.А.**

**Днепропетровск 2015**

**Содержание**

Введение……………………………………………………………………………….…..3

Техническое задание……………………………………………………………………...4

Технологии………………………………………………………………………………...5

Схема взаимодействия классов…………………………………………………………..6

Разработка алгоритмов системы…………………………………………………..……..7

Руководство пользователя и интерфейс..………………………………………..………8

Выводы…………………………………………………………………………………...20

**Введение**

Для написания данной программы использован язык программирования С++, использование WinAPI для разработки оконного интерфейса. Для того чтобы выбрать C++ в качестве языка программирования существует четыре причины:

1. Компилируемый язык со статической типизацией.
2. Сочетание высокоуровневых и низкоуровневых средств.
3. Реализация ООП.
4. STL.
5. Поддержка WinAPI.

Данная программа написана для использования в банковских учреждениях. В программе хранится вся информация о клиентах и она значительно упрощает работу с клиентскими счетами.

**Техническое задание**

Написать программу с оконным интерфейсом для использования банковскими структурами.

Используют программу менеджеры и руководители банковской структуры.

Программа предназначена для:

1. Просмотра счетов клиентов.
2. Добавление и удаление клиентов из базы данных.
3. Поиск, просмотр и редактирование информации о клиенте.
4. Открытие новых счетов у клиентов.
5. Провести платеж по счетам, а так же полное его погашение и закрытие.
6. Просмотр истории платежей по всем счетам и по конкретным отдельно.
7. Добавление и удаление администраторов из базы.

Данная программа помогает в хранении информации о клиентах банка, а также значительно упрощает работу с их счетами. Так же программу могут использовать клиенты, чтобы отслеживать свои платежи и проверять данные по-своему аккаунту.

**Технологии**

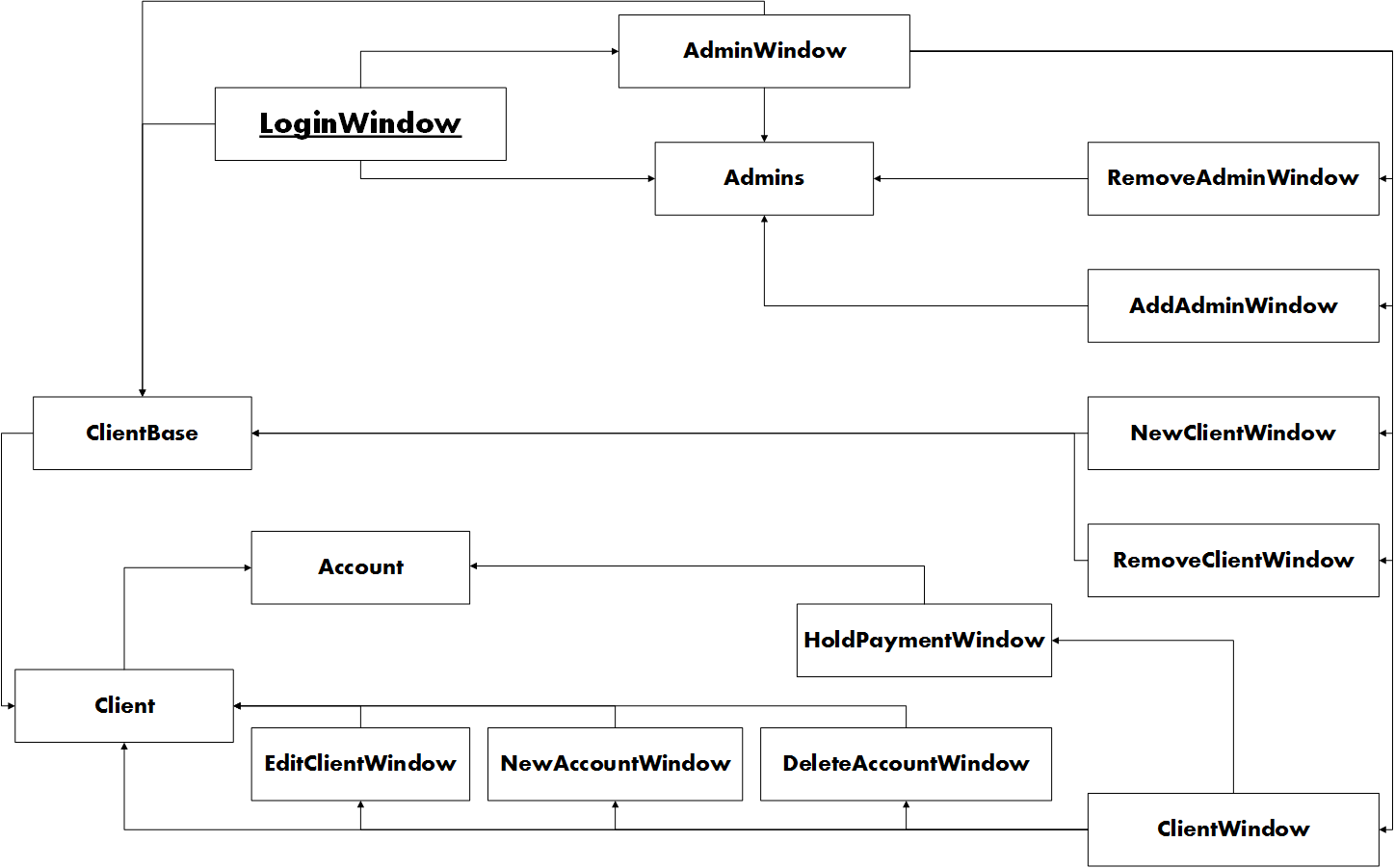
Для написания программы использовалась STL - стандартная библиотека шаблонов, а именно классы vector, string и wstring.

Vector - это динамический массив.

String и Wstring - заголовочные файлы с классами, функциями, и переменными, для организации работы со строками в языке программирования C++

В программе реализовано переключение между библиотеками, в зависимости от кодировки, при помощи макроса TCHAR, а так же собственном макросе STRING, который подключает библиотеку string.h или wstring.h, в зависимости от текущей кодировки.

**Схема взаимодействия классов**

****

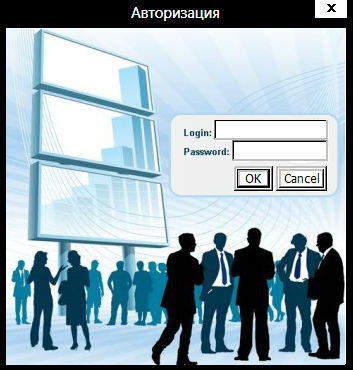
**Разработка алгоритма системы**



**Руководство пользователя и интерфейс**

Эта программа предназначена для использования сотрудниками банка, поэтому для авторизации необходимо ввести логин и пароль администратора (рис 1).

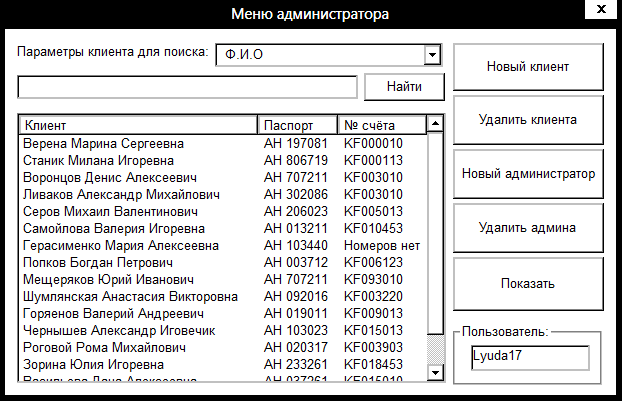
У пользователя есть 3 попытки ввода данных, если все они неверны, программа закрывается.

****

*Рис 1.*

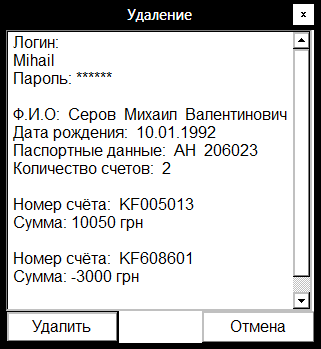
После успешной авторизации пользователь попадает в главное меню – «Меню администратора» (рис 2). В правом, нижнем углу окна, указывается логин текущего пользователя – админа.

В таблице указаны данные всех клиентов в базе данных на данный момент времени. Вы можете выбрать из списка нужно клиента и просмотреть данные о нём, выбрав кнопку «Показать».



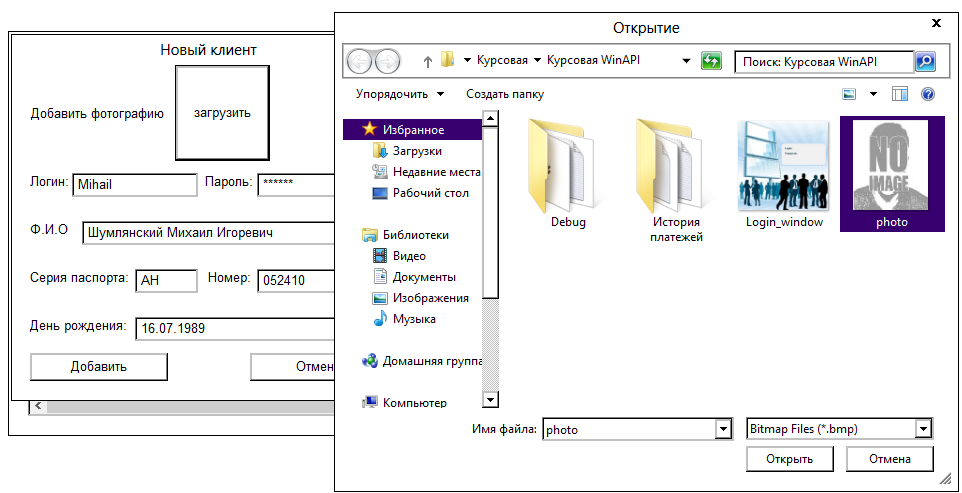
*Рис 2.*

Также, что бы удалить клиента из таблицы и базы данных, пользователю требуется выделить клиента, нажать кнопку «удалить» и подтвердить удаление (рис 3.).



*Рис 3.*

При нажатии на кнопку «новый клиент», администратору будет предложено запомнить личные данные человека, для регистрации его в базе данных, так же загрузить его фотографию (в случае, если фото будет не загружено, будет добавлена стандартная иконка). Для подтверждения операции необходимо нажать «добавить», если вы передумали - «отмена» (рис 4).

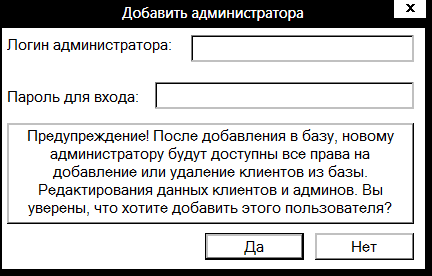


*Рис 4*

Хотим обратить внимание на то, что: если вы присвоите клиенту логин, который уже есть в базе, программа сообщит вам о невозможности добавить его в базу и предложит ввести другой логин.

Текущий пользователь программы может, так же, править базу администраторов, путём добавления или удаления из неё аккаунтов.

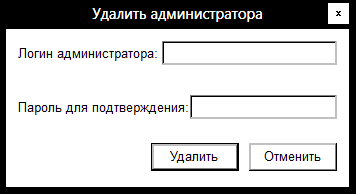
Для добавления нового администратора требуется всего лишь ввести логин (который не совпадёт с уже текущими пользователями) и пароль (рис 5).



*Рис 5*

Чтобы удалить администратора из базы данных, требуется ввести его логин и пароль для подтверждения (рис 6).

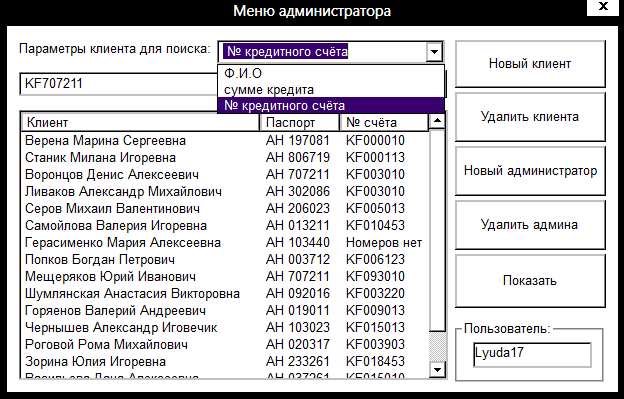
! Обратите внимание, что удаление безвозвратное! Подтвердив его, отменить действие уже невозможно.



*Рис 6*

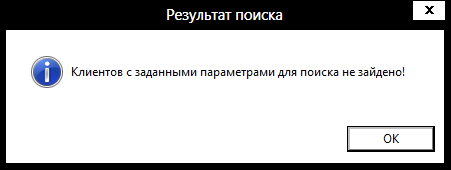
Если текущий пользователь удалит из базы свои же данные, программа вернётся к окну авторизации (рис 1).

Чтобы облегчить пользователю работу с таблицей базы данных, на главном окне есть поиск клиентов. В выпадающем списке, вам необходимо выбрать удобный для способ поиска: по Ф.И.О, по номеру счёта или сумме кредита. Затем заполнить данные для поиска и подтвердить выбор, нажав кнопку «найти» (рис 7).



*Рис 7*

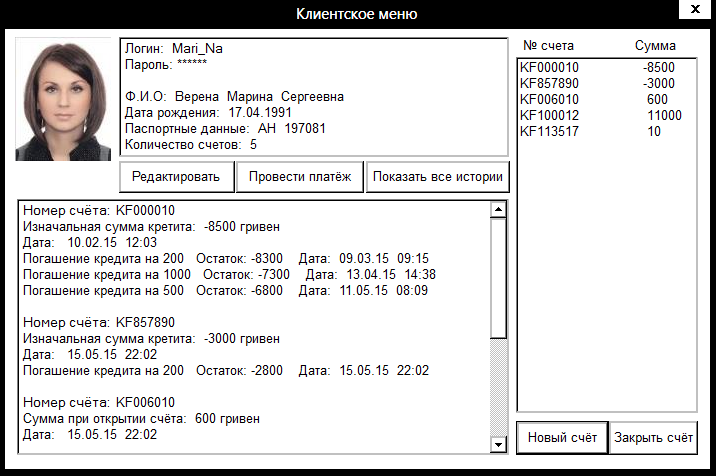
В случае, если клиент с такими данными найден не будет, программа сообщит вам, о том, что данный пользователь не найден (рис 8).

**

*Рис 8*

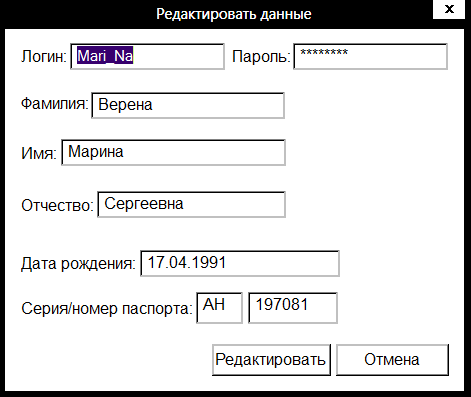
Если же клиент будет найден, вы будете перенаправлены в «клиентское меню».

«Клиентское меню» (рис 9) содержит всю информацию о клиенте – его личные данные, список счетов и платежи по ним. В этом окне пользователь может редактировать личные данные клиента, а так же просматривать информацию о его счетах. Провести платеж, открыть и закрыт номер счёта.



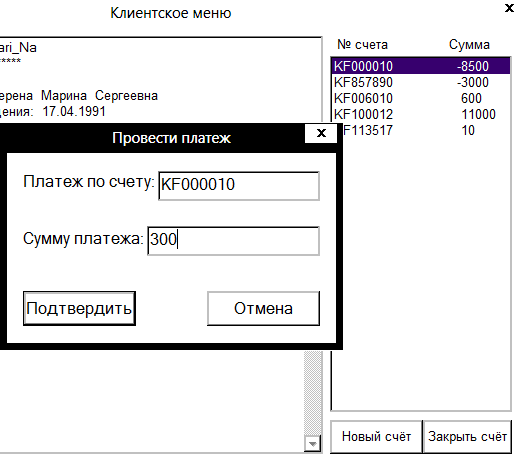
*Рис 9*

При редактировании данных, вам будут предложены для редактирования прошлые данные в базе. Если вы всё подправили, для сохранения нажмите «редактировать», если изменять данные не стоит – «отмена» (рис 10).



*Рис 10*

Для проведения платежа, администратору следует выбрать номер счёта из списка. Затем, нажав «провести платеж» (рис 11), пользователю выдаст информацию с какого счёта, будут списаны или зачислены средства.

**

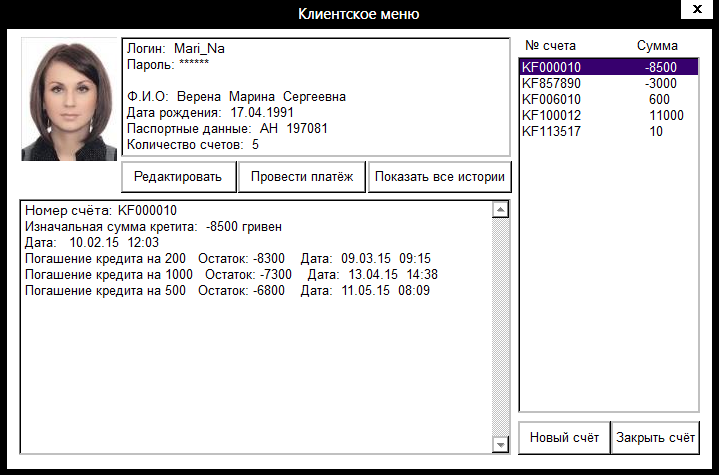
*Рис 11*

В графу «сумма», при проведении платежа по «кредитному» счёту, стоит вводить только сумму, без знаков -/+. Если номер счёта кредитный, то сумма будет зачислена на погашение. Снятие средств со счёта невозможно. Если номер счёта «накопительный», то сумма вводится со знаками -/+ в зависимости от того, нужно средства списать или зачислить на счёт.

При выборе в списке счётов определённого номера, на экране внизу, будет выводиться история платежей по нему (рис 12).

Если номер счёта не выбран, на экране отобразится информация по всем кредитным счетам в порядке очереди – сначала кредитные, затем накопительные, как и в самом списке (рис 8).

Так же, при нажатии кнопки «показать все истории», выделение текущего номера в списке будет снято и показана история по всем платежам.

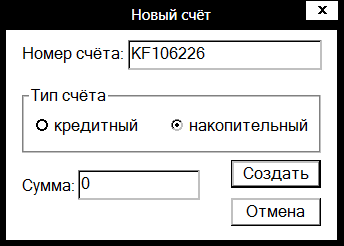


*Рис 12*

В истории платежей номера указываются данные:

* Номер счёта;
* Изначальная сумма кредита или сумма при открытии счёта;
* Затем построчно выводится информация по каждому платежу: сумма зачисления или погашения, остаток средств на счету, дата и время проведения платежа.

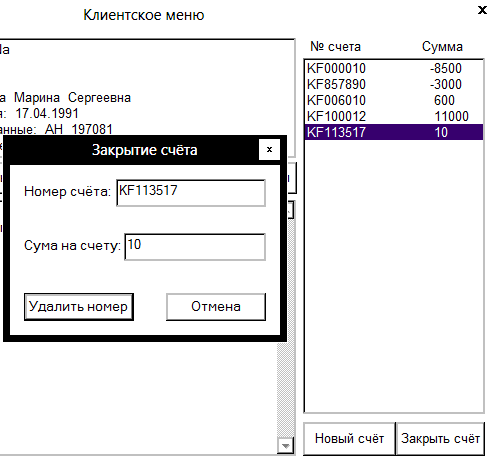
Для открытия нового счёта выбираем пункт «новый счёт» (рис 13). После этого в новом окне вам следует заполнить данные. При открытии этого окна в базе данных генерируется индивидуальный, новый номер счёта и выводится вам на экран, поэтому вводить вручную его не нужно.



*Рис 13*

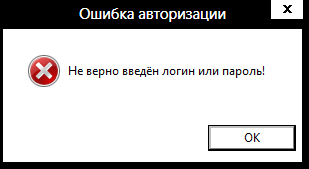
Всё, что необходимо сделать, это выбрать тип счёта – накопительный или кредитный и ввести сумму. Для подтверждения открытия – «создать», для отмены – «отмена».

При закрытии счёта, вам так же необходимо выбрать номер в списке. Только тогда программа откроет вам окно для удаления счёта из базы (рис 14). Проверите все данные на экране и убедитесь, то номер счёта выбран правильно, и вы действительно хотите его удалить. Затем подтвердите или отмените действие.

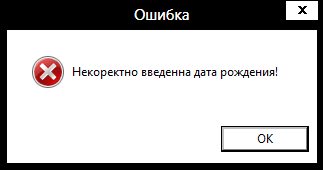


*Рис 14*

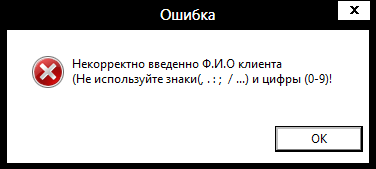
Настоятельно рекомендуем, тщательно следить за корректностью ввода данных в поля ввода, чтобы избежать ошибок в будущем. Однако, если вы в чём то ошибётесь – забудете заполнить необходимые поля ввода или же введётся не корректно номер, сумму или дату, тогда программы выдаст вам предупреждения и попросить отредактировать/ввести новые данные (рис 8, 15, 16, 17).

**

*Рис 15*

**

*Рис 16*

**

*Рис 17*

Все важные данные программы хранятся в корне программы в текстовых файлах и папках. При любом изменении данных база автоматически обновляется, дабы избежать потери данных.

**Хранение информации в программе**

**!**Не рекомендуется изменять данные в файлах вручную – это может привести к потере данных или ошибке при их считывании.

1. База данных клиентов (рис 18).

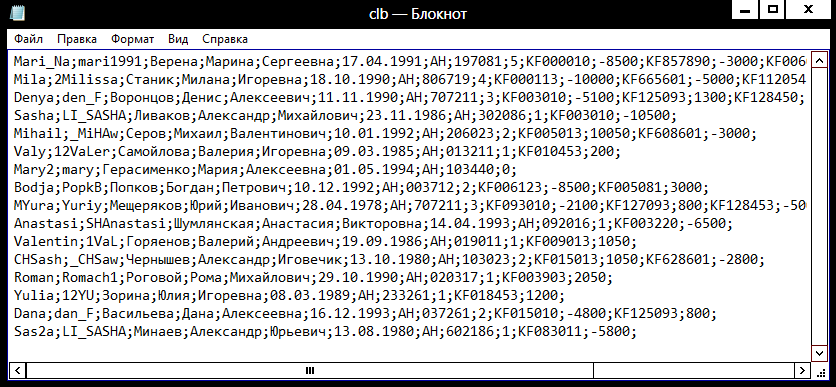
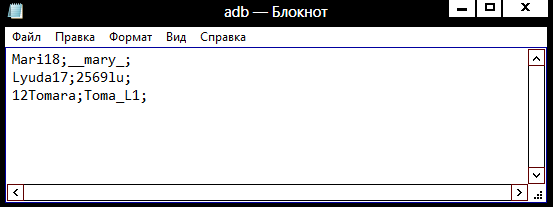


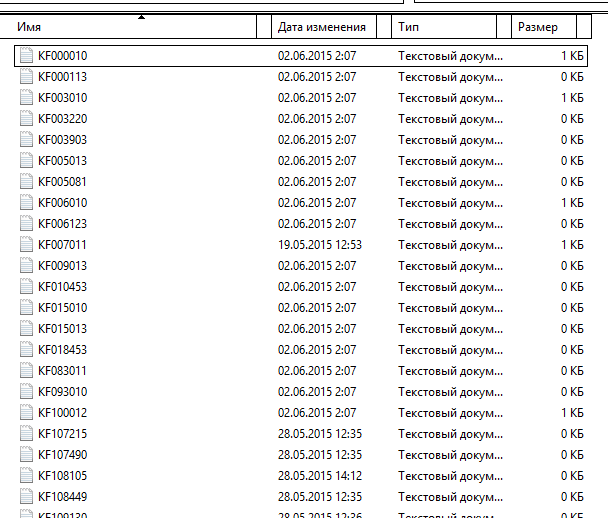
Рис 18

1. База администраторов системы (рис 19).



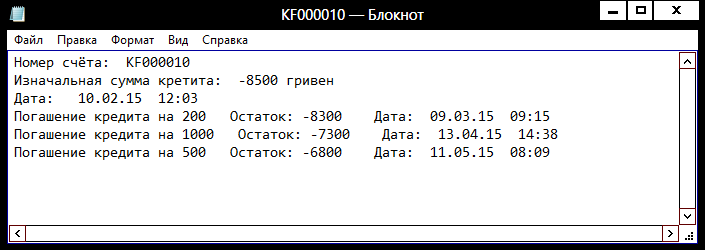
*Рис 19*

1. Папка для хранения всех данных о платежах (имя файла – номер счёта) (рис 20).



*Рис 20*

1. Хранение истории платежей, начиная с открытия счёта (рис 21).

**

*Рис 21*

**Выводы**

В процессе написания программы, закреплены знания по ООП, разработка оконного интерфейса при помощи средств WinAPI. Изучено использование файлов, запись их и считывание.

Использован класс std::string и wstring и такие методы как compare() – сравнивает 2 строки, atoi() – преобразование строки в число, \_itoi\_s() – преобразование числа в строку, перегрузка операторов =, +, +=.

Использован класс std::vector и такие методы как push\_back() – добавить элемент в конец массива, size() – возвращает количество элементов в векторе, erase() – удаляет элемент из вектора по итератору, begin() – возвращает итератор, указывающий на начало вектора.

Для создания оконного интерфейса были использованы такие элементы WinAPI:

* EditControl;
* ComboBox;
* ListBox;
* ListView;
* GroupBox;
* RadioButton;
* Button;
* StaticText;
* PictureControl.

Задача безопасности в программе не стояла, поэтому в будущем планируется развитие.