Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
филиал «Минский радиотехнический колледж»

ПРОГРАММНОЕ СРЕДТСВО ФАЙЛОВЫЙ МЕНЕДЖЕР

Пояснительная записка
к курсовому проекту по дисциплине
«Основы алгоритмизации и программирования»

КП9К.939314.102.081 ПЗ

Руководитель С.А. Апанасевич

Учащаяся гр. 9К9393 В. Л. Жук

2021

Содержание

Введение

Файловый менеджер — компьютерная программа, предоставляющая интерфейс пользователя для работы с файловой системой и файлами. Файловый менеджер позволяет выполнять наиболее частые операции над файлами — создание, открытие/проигрывание/просмотр, редактирование, перемещение, переименование, копирование, удаление, изменение атрибутов и свойств, поиск файлов и назначение прав. Помимо основных функций, многие файловые менеджеры включают ряд дополнительных возможностей, например, таких как работа с сетью, резервное копирование, управление принтерами и прочее.

Ни одна операционная система на сегодняшний день не может обойтись без удобного и надежного файлового менеджера. Огромное количество нарастающих с каждым днем данных нуждается в грамотной структуризации и разделении. Не зря все современные операционные системы, как правило, включают в свой состав, в первую очередь, именно файловый менеджер, как неотъемлемую часть самой ОС и необходимый компонент для реализации всех возможностей по доступу к файловой системе. При этом такой доступ должен удовлетворять многим, зачастую противоположным условиям, к которым относятся: возможность быстрого поиска и отображения нужной информации, полнота операций над этими данными, гарантированное исключение ошибок при этих операциях, простота и т. д.

Выделяют различные типы файловых менеджеров, например:

- навигационные и пространственные — иногда поддерживается переключение между этими режимами;
- двупанельные — в общем случае имеют две равноценных панели для списка файлов, дерева каталогов.

Обычный Проводник Windows является представителем навигационных (пространственных) файловых менеджеров. Это не самая лучшая и удобная программа. Она не всегда справляется с массовыми операциями с файлами, например, когда надо перенести или переименовать сотню фотографий. При работе с очень большим количеством файлов Проводник значительно заменяет работу системы. Но самый существенный недостаток - это, все же, отсутствие второй панели для копирования или перемещения файлов.

В менеджерах второго типа (двупанельных) для таких операций окно программы разделено на две части. При этом работа с файлами становится более удобной и быстрой. А если учесть, что большинство файловых менеджеров позволяют управлять всеми действиями с клавиатуры, то скорость работы повышается в несколько раз.

Принимая во внимание выше сказанное, становится ясно, что построение трехмерных объектов и сцен является очень перспективным направлением деятельности. Задачами данного курсового проекта являются:

- анализ актуальности файлового менеджера;
- обзор программ для реализации цели;
- создание двухмерных объектов для заполнения сцены;

- построение трехмерной сцены.

Цель данной работы - это реализовать программное средство с простым и удобным интерфейсом, который поможет в и упростит работу с файлами.

При реализации курсового проекта программное средство для работы с файлами были поставлены следующие задачи:

- составить требования;
- проанализировать предметную область;
- спроектировать программный продукт;
- реализовать программный продукт;
- протестировать программу.

Данная программа создавалась для пользователей, которые нуждаются в упрощенном управлении с файлами.

Программа была написана на Visual Studio, среда разработки является профессиональной и удобной в использовании. Помимо стандартного редактора и отладчика, которые существуют в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства авто завершения кода, графические конструкторы, встроенные модули и многие другие функции для упрощения процесса разработки.

Курсовой проект актуален, так как прост в использовании, чтобы быстро просмотреть, изменить, сохранить свой проект. Одной из главных особенностей пакета программы считается его маленький размер по сравнению с прочими популярными программами для создания дизайна.

Данная программа подходит для любого пользователя, поскольку не трудна в использовании.

1 Постановка задачи

Описание предметной области

Большинство алгоритмов зависит от того, каким образом организованы данные поэтому начинать проектирование программы следует не с алгоритмов, а с разработки структуры, необходимых для представления входных, выходных и промежуточных данных.

Структура данных - программная единица, позволяющая хранить и просматривать множество однотипных или логически связанных данных в вычислительной технике. Структура данных часто является реализацией, какого либо, абстрактного типа данных.

Все данные необходимые для решения практических задач подразделяются на несколько типов, причем понятие тип связывается не только с представлением данных в адресном пространстве, но и со способом их обработки.

Любые данные могут быть отнесены к одному из двух типов: основному (простому), форма представления которого определяется архитектурой компьютера, или сложному, конструируемому пользователем для решения конкретных задач.

Данные простого типа это - символы, числа и т.п. элементы, дальнейшее дробление которых не имеет смысла. Из элементарных данных формируются структуры (сложные типы) данных.

Массив (функция с конечной областью определения) - простая совокупность элементов данных одного типа, средство оперирования группой данных одного типа. Отдельный элемент массива задается индексом. Массив может быть одномерным, двумерным и т.д. Разновидностями одномерных массивов переменной длины являются структуры типа кольцо, стек, очередь и двухсторонняя очередь.

Запись - совокупность элементов данных разного типа. В простейшем случае запись содержит постоянное количество элементов, которые называют полями. Совокупность записей одинаковой структуры называется файлом. Файлом называют также набор данных во внешней памяти, например, на магнитном диске.

Такие структуры данных как массив или запись занимают в памяти ЭВМ постоянный объем, поэтому их называют статическими структурами. К статическим структурам относится также множество.

Имеется ряд структур, которые могут изменять свою длину - так называемые динамические структуры. К ним относятся дерево, список.

В программном продукте будут использоваться файлы системы, над которыми программа «Файловый менеджер» будет выполнять следующие действия: копирование файлов, перемещение файлов, удаление файлов или каталогов, создание новой каталогов, запуск файлов. Из этого следует, что программа будет работать с сложными данными файлами системы и их свойствами. Также будут использоваться данные простых типов символы, числа для промежуточных вычислений.

Для разработки данного программного продукта, необходимо спроектировать и разработать окно программы. Которое должно обеспечит удобное визуальное отображение результатов работы программы «Файловый менеджер» и эргономичное взаимодействие с пользователем персонального компьютера.

Файловый менеджер - один из самых востребованных программных продуктов. Самые разнообразные версии от большого круга производителей можно обнаружить на рабочем столе персонального компьютера практически любого пользователя и выбор пользователей, за частую, падает на те программные продукты, в которых наиболее эргономичен пользовательский интерфейс.

Основной принцип, обеспечивший популярность данных программ - это наличие двух панелей, каждая из которых показывает содержание одной из папок файловой системы. По каждой из панелей можно при помощи стрелок на клавиатуре перемещать курсор. Переход между панелями осуществляется клавишей табуляции. Клавиша Enter позволяет открыть файл, на котором установлен курсор. При этом открытие производится при помощи той же программы, которая ассоциирована с файлом данного типа в Windows. Например, исполнимые файлы (jpg, gif и др.) будут запущены на выполнение, картинки показаны при помощи программы просмотра графических файлов и т.п. Если же это папка, то в текущей панели появится содержание этой папки.

Поэтому определение пользовательского интерфейса программного приложения является неотъемлемой частью проектирования приложения «Файловый менеджер».

Программный интерфейс - система унифицированных связей, предназначенных для обмена информацией между компонентами вычислительной системы. Программный интерфейс задает набор необходимых процедур, их параметров и способов обращения.

Интерфейс пользователя - это элементы и компоненты программы, которые способны оказывать влияние на взаимодействие пользователя с программным обеспечением.

В разрабатываемом программном приложении будем использовать объектно-ориентированные интерфейс со свободной навигацией, так как он даёт возможность наиболее эффективно и удобно взаимодействовать пользователю с программным продуктом.

1.2 Обзор существующих аналогов

Проводник Windows (см. рисунок 1.1) — приложение, реализующее графический интерфейс доступа пользователя к файлам в операционной системе Microsoft Windows.

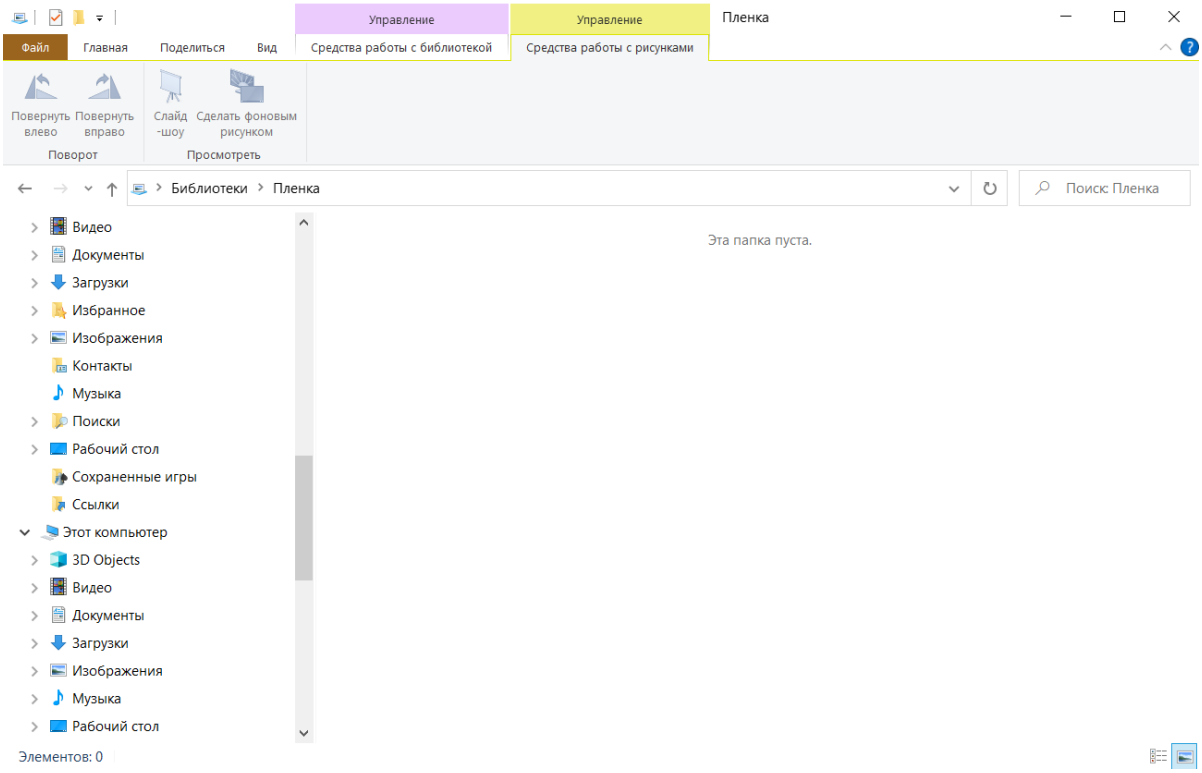

Рисунок 1.1 – Интерфейс файлового менеджера «Проводник»

Данный графический редактор обладает рядом таких возможностей, как:

- открытие любого количества окон.
- фильтр.
- поиск.
- быстрый доступ.

Несмотря на это, у «Проводника» существуют свои недостатки. Среди них можно выделить:

- маленькая скорость копирования;
- неудобный интерфейс.

Multi Commander (см. рисунок 1.2) - двухпанельный файловый менеджер, доступный на всех популярных системах– Windows, Linux, macOS и FreeBSD. Некоторое время назад окончательно перешёл на свободную модель распространения.

По задумке разработчиков менеджер должен был представлять из себя этакий «комбайн», содержащий в себе архиваторы, видео и аудио кодеки, редактор реестра, надстройки для работы с серверами FTP и прочие дополнения. Весь функционал настраивается при установке, но в процессе работы вы сможете отменить подключение ненужных вам опций.

Основные возможности Multi Commander:

- свободное распространение;
- есть официальная русификация;
- крайне широкий функционал, который настраивается при установке;
- очень гибкая настройка всех параметров программы.

функция автоматического обновления (включается/отключается в настройках).

Решение «всё в одном»: простой и удобный аналог «Проводника» для любителей настраивать ПО под свои нужды.

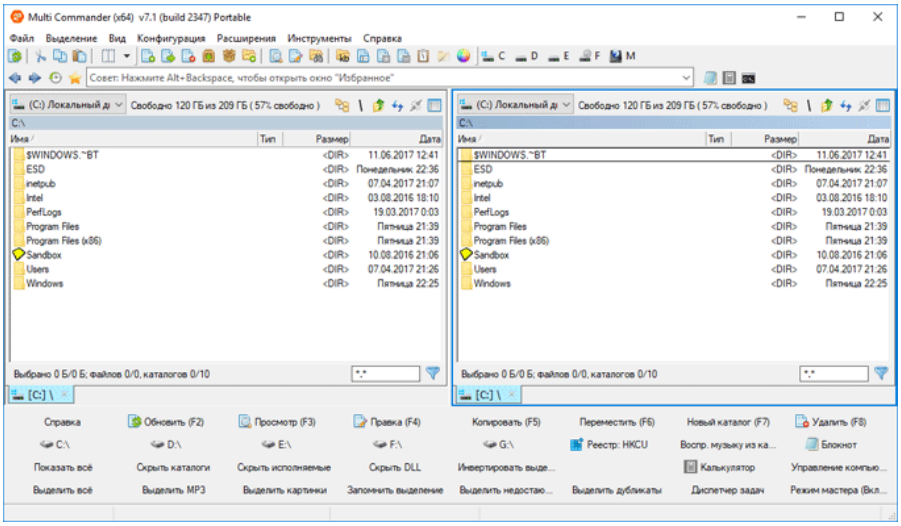

Рисунок 1.2 – Интерфейс файлового менеджера «Multi Commander»

Far Manager (см. рисунок 1.3) – программа является прямым наследником классического Norton Commander.

Крайне простой интерфейс. Двухпанельное отображение папок и применимые функции в нижней части экрана.

Наличие официальной русификации. Для активации перейдём по следующему маршруту: F9 – Options – Languages – Русский.

Поддержка плагинов. Бесплатные дополнения можно загрузить на официальной странице программы в разделе «PlugRing».

Полностью бесплатен.

Интересный выбор для любителей DOS и эпохи псевдографических изображений. По функционалу – на уровне именитых аналогов, но

особенности интерфейса могут отпугнуть некоторых пользователей.

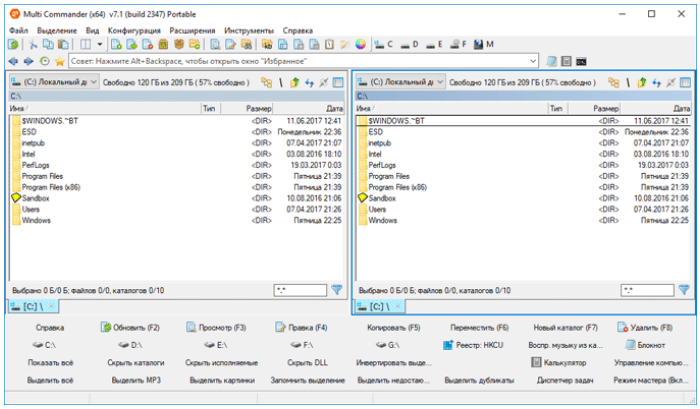

Рисунок 1.3 – Интерфейс файлового менеджера «Far Manager»

1.3 Функциональное назначение

Для работы в программном средстве для дизайна дома, оно должно иметь базовый набор основных функций:

- просмотр файлов;
- возможность просматривать файлы, переходить по путям к файлам;
- возможность редактировать, удалять, создавать файлы, или папки;
- визуализация, представить папки и файлы в виде иконок;
- прямой переход, ручной ввод пути к файлу, папке;
- возможность запуска файлов.

Функция «поиск» осуществляет первичный поиск носителей и осуществляет загрузку их номеров в глобальную переменную.

Функция добавления записи в дерево осуществляет вставку потомка для переданного в функцию узла. Осуществляется настройка свойств вставляемого элемента: пути к нему значка имени.

Функция добавления записи в список извлекает из переданной структуры необходимые элементы – дату, название, а также сама осуществляет поиск дополнительной информации – название типа и значок, соответствующий типу файла. В итоге функция осуществляет полное формирование строки и возвращает указатель на нее.

Функция «Delete» выполняет обратную операцию – удаляет файлы или папки в находящемся пути.

Функция «Copy» выполняет операцию – согласно полученным данным: символьному массиву «откуда копировать» массиву «куда копировать».

2 Проектирование задачи

2.1 Алгоритмы работы программы

При запуске программы создается главная форма, которая запускает окно файлового менеджера, затем приложение ожидает действия пользователя. Каждому предусмотренному действию сопоставлена какая-то функция-обработчик.

При нажатии на правую кнопку мыши программа выводит на экран контекстное меню, где так же есть предусмотренные действия пользователя с объектами на этих формах.

После выполнения какого-либо действия, программа возвращается в режим ожидания пользовательских действий.

Для завершения работы с файловым менеджером пользователю достаточно нажать на кнопку «Закрыть» системного меню, находящуюся в правом верхнем углу окна программы.

В Приложении А приведены исходные тексты всех модулей программы, а также исходный текст заголовочного файла второго модуля (т.к. он, в отличие от заголовочных файлов других модулей не был сгенерирован автоматически).

Главная форма не является окном, ее основная задача – не дать закрыться одновременно всем дочерним формам, при закрытии одной из дочерних форм, поэтому главная форма не является окном. Главная форма закрывается, когда закрыты все дочерние формы.

2.2 Выбор и обоснование инструментов разработки

Для того чтобы начать работу, необходимо выбрать для этого подходящую среду, ведь от неё тоже будет зависеть качество продукта, в данном случае – курсового проекта. Среда в компьютере – программные средства, которые представляют собой совокупность программ, которые имеют определенное назначение и предназначены для работы на персональном компьютере.

Программным средством для разработки любой программы является операционная система, среда разработки программы, язык программирования, предназначенный для данной среды разработки программы. При выполнении данного проекта были использованы следующие программные средства:

— операционная система Windows 10;

— визуальная среда разработки программ Visual Studio;

— язык программирования C++;

— текстовый редактор Microsoft Word.

Преимущества языка программирования C++.

C++ разработан как универсальный язык со статическими типами данных - переменная не может менять свой тип данных. Это значит, что при автодополнении интегрированная среда разработки будет предлагать исключительно те методы, которые применимы к данному типу данных. Статическая типизация также значительно уменьшает количество ошибок; многие из них исключаются уже на стадии компилирования приложения. Тем не менее, применение статической типизации влечет за собой определенные трудности с реализацией алгоритмов и читаемостью кода.

C++ - компилируемый, статически типизированный язык программирования общего назначения. Он универсален, так как поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщенное программирование. Имеет богатую стандартную библиотеку, которая включает в себя распространенные контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности.

C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков. Основная черта высокоуровневых языков – это абстракция, то есть введение смысловых конструкций, кратко описывающих такие структуры данных и операции над ними, описания которых на машинном коде очень длинны и сложны для понимания. Низкоуровневый же язык программирования близок к программированию непосредственно в машинных кодах используемого реального или виртуального процессора.

C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения обширна: это и создание операционных систем, и разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр.

Для курсового проектирования был выбран данный язык, так как он содержит все необходимые инструменты для реализации программы, а также отвечает всем поставленным требованиям.

3 Программная реализация

3.1 Физическая структура

Физическая структура программного средства включает в себя множество различных файлов. Данные файлы выполняют свои определенные функции. Все файлы представляют собой одно целостное программное средство для дизайна дома. Ознакомиться с данным набором файлов можно в таблице 3.1.

Таблица 3.1 – Название и назначение файлов, составляющих физическую структуру программы.

Наименование	Назначение
Головной файл проекта (.cpp) main.cpp	Visual Studio C++ создает файл .cpp для головной функции main, инициирующей приложение и запускающей его на выполнение.
Заголовочный файл модуля (.h)	Каждой создаваемой вами форме соответствует не только файл реализации модуля, но и его заголовочный файл с описанием класса формы. Вы можете и сами создавать необходимые заголовочные файлы.
Файлы резервных копий (.~cpp, .~h)	Это соответственно файлы резервных копий для файлов реализации модуля, заголовочного, проекта и формы. Если вы что-то безнадежно испортили в своем проекте, можете соответственно изменить расширение этих файлов и таким образом вернуться к предыдущему не испорченному варианту.

Разрабатываемое ПС имеет 3 головных файла Main.cpp, AmmountBaze.cpp, MyForm.spp, 3 заголовочных файла .h, 1. В головных файлах представлен основной код проекта, который подключает все остальные и является главным. Во всех остальных файлах .h и .dat охарактеризованы базовые функции. Подробная информация о файлах и их функциях располагается в приложении А.

4 Тестирование

Тестирование программного обеспечения – это:

– процесс исследования ПО с целью получения информации о качестве продукта;

– процесс проверки соответствия заявленных к продукту требований и реально реализованной функциональности, осуществляемый путем наблюдения за его работой в искусственно созданных ситуациях и на ограниченном наборе тестов, выбранных определенным образом;

– оценка системы с тем, чтобы найти различия между тем, какой система должна быть и какой она есть.

В широком смысле, тестирование – это одна из техник контроля качества (Quality Control), которая включает планирование, составление тестов, непосредственно выполнение тестирования и анализ полученных результатов.

Некоторые ошибки появляются после первого же запуска программы на выполнение, либо же при выполнении рядовых команд. Для их обнаружения не надо прибегать ни к каким специальным средствам. Некоторые же ошибки появляются в случайные моменты работы программы. С такими ошибками справиться труднее всего, ведь если не зафиксировать условия возникновения ошибки, будет очень сложно понять причину и устранить ее.

Всевозможные ошибки обычно подразделяют на три группы:

– синтаксические ошибки;

– ошибки времени выполнения программы;

– смысловые (логические) ошибки.

Рассмотрим каждую из выделенных групп по порядку.

Синтаксические ошибки являются самыми простыми, они устраняются уже на этапе компиляции. Причина их проста и однообразна – неправильная запись служебных слов, операторов.

Намного больше неприятностей доставляют ошибки времени выполнения. Они дают о себе знать прекращением выполнения программы. Чаще всего ошибка времени выполнения является признаком смысловой ошибки.

Смысловые (логические) ошибки – самые сложные и трудноуловимые. Они проявляются в том, что программа выполняет не то, что от нее предполагалось. Последствия смысловых ошибок могут быть самыми разными: это может быть что-то безобидное, например, неправильное содержимое окна, невыполнение или неверное выполнение команд

пользователя, неправильное содержимое выходной информации и прочее, а может быть и достаточно серьезными – программа может досрочно завершиться с ошибкой времени выполнения и многое другое.

При исправлении ошибки самое главное – это не внести в программу новых ошибок.

Таблица 4.1 – Тестирование программного средства

Тестируемая операция	Требуемый результат	Полученный результат
Запуск программы.	Появление дочерней формы для работы с файлами.	После загрузки главной формы – появление дочерней формы.
Работа функций.	Правильная работа функций.	Все функции работают корректно.
Попытка изменить размеры окон.	Размеры окон не зафиксированы, поэтому объекты формы должны перемещаться в положенное место от зависимости размера окна.	При успешной попытке изменить размеры окон - их размеры изменятся, а объекты переместятся.
Завершение программы.	Закрытие программы.	Программа успешно завершила свою работу.

Тестирование программы наглядно показывает таблица 4.1. В результате тестирования программы ошибок, которые могли бы привести к некорректной работе, обнаружено не было.

После тестирования можно сделать вывод, что программный продукт справляется со всеми поставленными задачами и выполняет обозначенные функции.

5 Применение

5.1 Руководство пользователя

Программа предназначена для любой аудитории. Открытие и просмотр файлов, а также все манипуляции с ними реализованы таким образом, чтобы пользователям было удобно и комфортно пользоваться программой.

Программное средство кроссплатформенное, поэтому для работы подходит любая операционная система. Данное приложение не нуждается в установке.

Программное средство не нуждается в дополнительных файлах.

После запуска на экране пользователя появится главное окно программы. (см. рисунок 5.1)

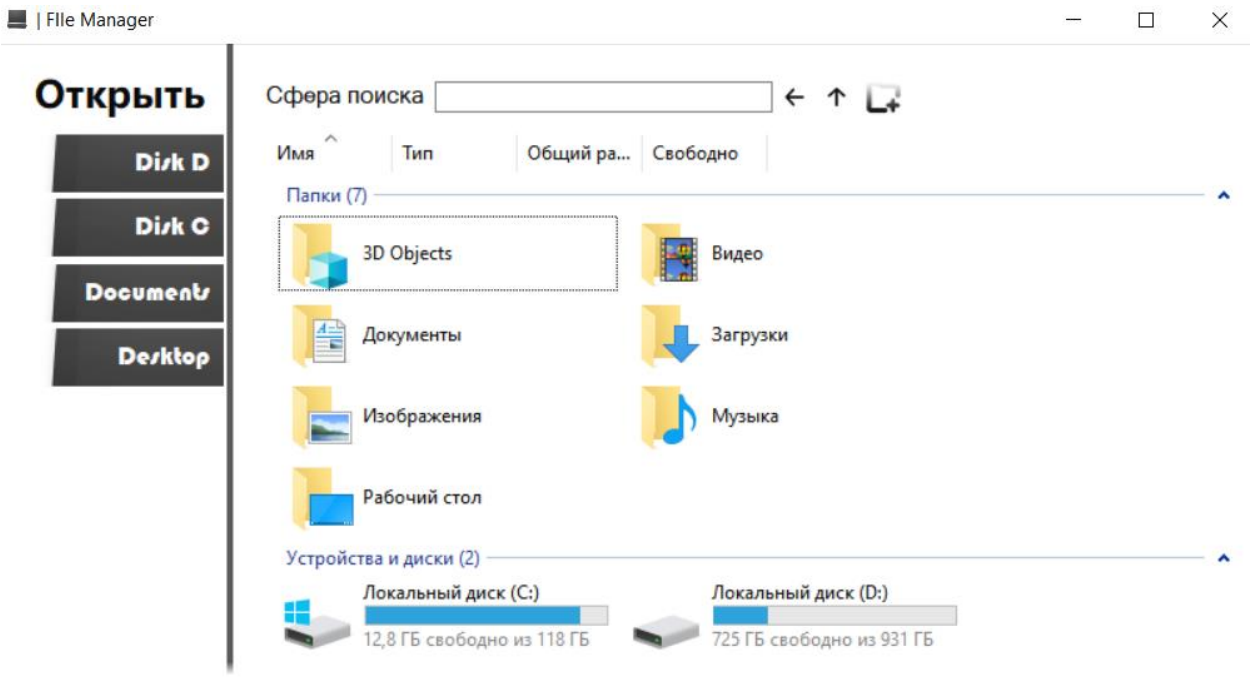

Рисунок 5.1 – Интерфейс программного средства

Заключение

В ходе курсового проекта была достигнута цель проектирования – создание программного средства для дизайна дома.

При разработке программы были использованы стандартные компоненты среды разработки, которые были и использованы при создании средств управления, редактирования или отображения. Курсовой проект реализован максимально удобно как для пользователей несмотря на то, что разработка данной программы оказалась достаточно сложной.

Выполненная программа не нуждается в установке. Программному средству необходимы малые требования к устройству. Работа с программным средством максимально комфортна, удобна и понятна, к тому же доставляет удовольствие. Программа вполне может сравниться с современными аналогами, так как имеет хороший набор функций.

В ходе выполнения курсового проекта была создана программа «Файловый менеджер». В ней были реализованы следующие функции:

- просмотр файлов и каталогов;
- копирование файлов и каталогов;
- удаление файлов и каталогов; перемещение файлов и каталогов;
- создание новой директории.

В программе было создано 2 формы: главная форма программы, форма файлового менеджера. В данную программу при необходимости можно внести изменения. А именно улучшение дизайна главной формы и добавление новых функций и возможностей.

В результате выполнения курсового проекта были усовершенствованы знания по работе с языком программирования C++. Были получены знания по работе с различными модулями. Данный проект оказался полезным и точно сможет помочь в дальнейшем изучении данной среды программирования. Благодаря реализации данного курсового проекта были приобретены практические и закреплены теоретические навыки и знания, и получено умение самостоятельного анализа поставленной задачи.

Список используемых источников

1. Берлянт А. М. Картографический словарь. / А. М. Берлянт – Москва: Научный мир, 2016. – 424 с.
2. Лафоре, Р. В. Объектно-ориентированное программирование в C++ / Р. В. Лафоре. – Питер: ПГГА, 2019. – 928 с.
3. Поляков, А. Методы и алгоритмы компьютерной графики в примерах на Visual C++ / А. Поляков. – Москва: БХВ-Петербург, 2017. - 416 с.

Интернет-ресурсы

4. compress.ru [Электронный портал]. – Режим доступа: https://compress.ru/article.aspx?id=14449. – Дата доступа: 23.05.2021
5. ru.wikipedia.org [Электронный портал]. – Режим доступа: https://ru.wikipedia.org/wiki/Файловый_менеджер. – Дата доступа: 26.05.2021
6. dic.academic.ru [Электронный портал]. – Режим доступа: https://dic.academic.ru/dic.nsf/ruwiki/1160654. – Дата доступа: 27.05.2021
7. programmsfree.com [Электронный портал]. – Режим доступа: https://www.programmsfree.com/file-manageri/2-file-manager.html. – Дата доступа: 05.06.2021
8. studbooks.net [Электронный портал]. – Режим доступа: https://studbooks.net. – Дата доступа: 05.06.2021

ПРИЛОЖЕНИЕ А

Код программы

```
#pragma once
#include <Windows.h>
#include "AmmountBaze.h"
#include <Lmcons.h>
#include <string>

namespace ManagerVib {

	using namespace System;
	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Windows::Forms;
	using namespace System::Data;
	using namespace System::Drawing;
	using namespace std;
	/// <summary>
	/// Сводка для MyForm
	/// </summary>
	public ref class MyForm : public System::Windows::Forms::Form
	{
	public:
		MyForm(void)
		{
			InitializeComponent();
			//
			//TODO: добавьте код конструктора
			//
		}

	protected:
		/// <summary>
		/// Освободить все используемые ресурсы.
		/// </summary>
		~MyForm()
		{
			if (components)
			{
				delete components;
			}
		}
	private: System::Windows::Forms::Button^ button1;
	protected:



	private: System::Windows::Forms::Label^ label1;
	private: System::Windows::Forms::Button^ button2;
	private: System::Windows::Forms::Button^ button3;

	private: System::Windows::Forms::TextBox^ textBox1;
	public:
	private: System::Windows::Forms::Label^ label2;
	private: System::Windows::Forms::PictureBox^ pictureBox1;


	private: System::Windows::Forms::Button^ button6;
	private: System::Windows::Forms::Button^ button4;
	private: System::Windows::Forms::Button^ button5;
```

```
      public: System::Windows::Forms::WebBrowser^ webBrowser2;
      private: System::Windows::Forms::Button^ button7;
      public:
      private:

      private:

      private:


      public:

      public:

      public:
      private:

      private:
            /// <summary>
            /// Обязательная переменная конструктора.
            /// </summary>
            System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
            /// <summary>
            /// Требуемый метод для поддержки конструктора — не изменяйте
            /// содержимое этого метода с помощью редактора кода.
            /// </summary>
            void InitializeComponent(void)
            {
                  System::ComponentModel::ComponentResourceManager^ resources
= (gcnew System::ComponentModel::ComponentResourceManager(MyForm::typeid));
                  this->button1 = (gcnew System::Windows::Forms::Button());
                  this->label1 = (gcnew System::Windows::Forms::Label());
                  this->button2 = (gcnew System::Windows::Forms::Button());
                  this->button3 = (gcnew System::Windows::Forms::Button());
                  this->textBox1 = (gcnew System::Windows::Forms::TextBox());
                  this->label2 = (gcnew System::Windows::Forms::Label());
                  this->pictureBox1 = (gcnew
System::Windows::Forms::PictureBox());
                  this->button6 = (gcnew System::Windows::Forms::Button());
                  this->button4 = (gcnew System::Windows::Forms::Button());
                  this->button5 = (gcnew System::Windows::Forms::Button());
                  this->webBrowser2 = (gcnew
System::Windows::Forms::WebBrowser());
                  this->button7 = (gcnew System::Windows::Forms::Button());

      (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox1))->BeginInit();
                  this->SuspendLayout();
                  //
                  // button1
                  //
                  this->button1->FlatAppearance->BorderSize = 0;
                  this->button1->FlatStyle =
System::Windows::Forms::FlatStyle::Flat;
                  this->button1->Font = (gcnew
System::Drawing::Font(L"Bauhaus 93", 12, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                        static_cast<System::Byte>(0)));
                  this->button1->ForeColor = System::Drawing::Color::White;
```

```
                this->button1->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"button1.Image")));
                this->button1->ImageAlign =
System::Drawing::ContentAlignment::MiddleLeft;
                this->button1->Location = System::Drawing::Point(0, 71);
                this->button1->Name = L"button1";
                this->button1->Size = System::Drawing::Size(198, 45);
                this->button1->TabIndex = 0;
                this->button1->Text = L"Disk D";
                this->button1->TextAlign =
System::Drawing::ContentAlignment::MiddleRight;
                this->button1->UseVisualStyleBackColor = true;
                this->button1->Click += gcnew System::EventHandler(this,
&MyForm::button1_Click);
                //
                // label1
                //
                this->label1->AutoSize = true;
                this->label1->BackColor = System::Drawing::Color::White;
                this->label1->Cursor =
System::Windows::Forms::Cursors::Default;
                this->label1->Font = (gcnew
System::Drawing::Font(L"Microsoft YaHei", 18,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
                      static_cast<System::Byte>(204)));
                this->label1->ForeColor = System::Drawing::Color::Black;
                this->label1->Location = System::Drawing::Point(28, 20);
                this->label1->Name = L"label1";
                this->label1->Size = System::Drawing::Size(154, 40);
                this->label1->TabIndex = 4;
                this->label1->Text = L"Открыть";
                //
                // button2
                //
                this->button2->FlatAppearance->BorderSize = 0;
                this->button2->FlatStyle =
System::Windows::Forms::FlatStyle::Flat;
                this->button2->Font = (gcnew
System::Drawing::Font(L"Bauhaus 93", 12, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                      static_cast<System::Byte>(0)));
                this->button2->ForeColor = System::Drawing::Color::White;
                this->button2->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"button2.Image")));
                this->button2->ImageAlign =
System::Drawing::ContentAlignment::MiddleLeft;
                this->button2->Location = System::Drawing::Point(0, 122);
                this->button2->Name = L"button2";
                this->button2->Size = System::Drawing::Size(198, 45);
                this->button2->TabIndex = 5;
                this->button2->Text = L"Disk C";
                this->button2->TextAlign =
System::Drawing::ContentAlignment::MiddleRight;
                this->button2->UseVisualStyleBackColor = true;
                this->button2->Click += gcnew System::EventHandler(this,
&MyForm::button2_Click);
                //
                // button3
                //
                this->button3->FlatAppearance->BorderSize = 0;
```

```
                this->button3->FlatStyle =
System::Windows::Forms::FlatStyle::Flat;
                this->button3->Font = (gcnew
System::Drawing::Font(L"Bauhaus 93", 12, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                        static_cast<System::Byte>(0)));
                this->button3->ForeColor = System::Drawing::Color::White;
                this->button3->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"button3.Image")));
                this->button3->ImageAlign =
System::Drawing::ContentAlignment::MiddleLeft;
                this->button3->Location = System::Drawing::Point(0, 173);
                this->button3->Name = L"button3";
                this->button3->Size = System::Drawing::Size(198, 45);
                this->button3->TabIndex = 6;
                this->button3->Text = L" Documents";
                this->button3->TextAlign =
System::Drawing::ContentAlignment::MiddleRight;
                this->button3->UseVisualStyleBackColor = true;
                this->button3->Click += gcnew System::EventHandler(this,
&MyForm::button3_Click);
                //
                // textBox1
                //
                this->textBox1->Location = System::Drawing::Point(377, 30);
                this->textBox1->Name = L"textBox1";
                this->textBox1->Size = System::Drawing::Size(287, 22);
                this->textBox1->TabIndex = 8;
                this->textBox1->TextChanged += gcnew
System::EventHandler(this, &MyForm::textBox1_TextChanged);
                //
                // label2
                //
                this->label2->AutoSize = true;
                this->label2->Font = (gcnew
System::Drawing::Font(L"Microsoft Sans Serif", 10.8F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,
                        static_cast<System::Byte>(204)));
                this->label2->Location = System::Drawing::Point(230, 28);
                this->label2->Name = L"label2";
                this->label2->Size = System::Drawing::Size(136, 24);
                this->label2->TabIndex = 9;
                this->label2->Text = L"Сфера поиска";
                //
                // pictureBox1
                //
                this->pictureBox1->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::None;
                this->pictureBox1->Cursor =
System::Windows::Forms::Cursors::Default;
                this->pictureBox1->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"pictureBox1.Image")));
                this->pictureBox1->Location = System::Drawing::Point(200, -
4);
                this->pictureBox1->MaximumSize = System::Drawing::Size(5,
500);
                this->pictureBox1->Name = L"pictureBox1";
                this->pictureBox1->Size = System::Drawing::Size(5, 500);
                this->pictureBox1->SizeMode =
System::Windows::Forms::PictureBoxSizeMode::StretchImage;
                this->pictureBox1->TabIndex = 10;
```

```
                this->pictureBox1->TabStop = false;
                this->pictureBox1->Click += gcnew
System::EventHandler(this, &MyForm::pictureBox1_Click);
                //
                // button6
                //
                this->button6->AutoSizeMode =
System::Windows::Forms::AutoSizeMode::GrowAndShrink;
                this->button6->BackColor = System::Drawing::Color::White;
                this->button6->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"button6.BackgroundImage")));
                this->button6->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Zoom;
                this->button6->FlatAppearance->BorderSize = 0;
                this->button6->FlatStyle =
System::Windows::Forms::FlatStyle::Flat;
                this->button6->Font = (gcnew System::Drawing::Font(L"Times
New Roman", 20, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::World));
                this->button6->Location = System::Drawing::Point(705, 27);
                this->button6->Margin = System::Windows::Forms::Padding(3,
2, 3, 2);
                this->button6->Name = L"button6";
                this->button6->RightToLeft =
System::Windows::Forms::RightToLeft::No;
                this->button6->Size = System::Drawing::Size(30, 30);
                this->button6->TabIndex = 13;
                this->button6->UseMnemonic = false;
                this->button6->UseVisualStyleBackColor = false;
                this->button6->Click += gcnew System::EventHandler(this,
&MyForm::button6_Click);
                //
                // button4
                //
                this->button4->AutoSizeMode =
System::Windows::Forms::AutoSizeMode::GrowAndShrink;
                this->button4->BackColor = System::Drawing::Color::White;
                this->button4->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"button4.BackgroundImage")));
                this->button4->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Zoom;
                this->button4->FlatAppearance->BorderSize = 0;
                this->button4->FlatStyle =
System::Windows::Forms::FlatStyle::Flat;
                this->button4->Font = (gcnew System::Drawing::Font(L"Times
New Roman", 20, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::World));
                this->button4->Location = System::Drawing::Point(670, 27);
                this->button4->Margin = System::Windows::Forms::Padding(3,
2, 3, 2);
                this->button4->Name = L"button4";
                this->button4->RightToLeft =
System::Windows::Forms::RightToLeft::No;
                this->button4->Size = System::Drawing::Size(30, 30);
                this->button4->TabIndex = 12;
                this->button4->UseMnemonic = false;
                this->button4->UseVisualStyleBackColor = false;
                this->button4->Click += gcnew System::EventHandler(this,
&MyForm::button5_Click);
                //
                // button5
```

```
                    //
                    this->button5->AutoSizeMode =
System::Windows::Forms::AutoSizeMode::GrowAndShrink;
                    this->button5->BackColor = System::Drawing::Color::White;
                    this->button5->BackgroundImage =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"button5.BackgroundImage")));
                    this->button5->BackgroundImageLayout =
System::Windows::Forms::ImageLayout::Zoom;
                    this->button5->FlatAppearance->BorderSize = 0;
                    this->button5->FlatStyle =
System::Windows::Forms::FlatStyle::Flat;
                    this->button5->Font = (gcnew System::Drawing::Font(L"Times
New Roman", 20, System::Drawing::FontStyle::Bold,
System::Drawing::GraphicsUnit::World));
                    this->button5->Location = System::Drawing::Point(745, 27);
                    this->button5->Margin = System::Windows::Forms::Padding(3,
2, 3, 2);
                    this->button5->Name = L"button5";
                    this->button5->RightToLeft =
System::Windows::Forms::RightToLeft::No;
                    this->button5->Size = System::Drawing::Size(30, 30);
                    this->button5->TabIndex = 14;
                    this->button5->UseMnemonic = false;
                    this->button5->UseVisualStyleBackColor = false;
                    this->button5->Click += gcnew System::EventHandler(this,
&MyForm::button5_Click_1);
                    //
                    // webBrowser2
                    //
                    this->webBrowser2->Anchor =
static_cast<System::Windows::Forms::AnchorStyles>((((System::Windows::Forms::
AnchorStyles::Top | System::Windows::Forms::AnchorStyles::Bottom)
                          | System::Windows::Forms::AnchorStyles::Left)
                          | System::Windows::Forms::AnchorStyles::Right));
                    this->webBrowser2->Location = System::Drawing::Point(234,
71);
                    this->webBrowser2->Margin =
System::Windows::Forms::Padding(3, 2, 3, 2);
                    this->webBrowser2->Name = L"webBrowser2";
                    this->webBrowser2->Size = System::Drawing::Size(836, 470);
                    this->webBrowser2->TabIndex = 15;
                    this->webBrowser2->Url = (gcnew System::Uri(L"D:\\",
System::UriKind::Absolute));
                    this->webBrowser2->DocumentCompleted += gcnew
System::Windows::Forms::WebBrowserDocumentCompletedEventHandler(this,
&MyForm::webBrowser2_DocumentCompleted);
                    //
                    // button7
                    //
                    this->button7->FlatAppearance->BorderSize = 0;
                    this->button7->FlatStyle =
System::Windows::Forms::FlatStyle::Flat;
                    this->button7->Font = (gcnew
System::Drawing::Font(L"Bauhaus 93", 12, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                          static_cast<System::Byte>(0)));
                    this->button7->ForeColor = System::Drawing::Color::White;
                    this->button7->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"button7.Image")));
                    this->button7->ImageAlign =
System::Drawing::ContentAlignment::MiddleLeft;
```

```
                this->button7->Location = System::Drawing::Point(0, 224);
                this->button7->Name = L"button7";
                this->button7->Size = System::Drawing::Size(198, 45);
                this->button7->TabIndex = 16;
                this->button7->Text = L"Desktop";
                this->button7->TextAlign =
System::Drawing::ContentAlignment::MiddleRight;
                this->button7->UseVisualStyleBackColor = true;
                this->button7->Click += gcnew System::EventHandler(this,
&MyForm::button7_Click);
                //
                // MyForm
                //
                this->AutoScaleDimensions = System::Drawing::SizeF(8, 16);
                this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
                this->BackColor = System::Drawing::Color::White;
                this->ClientSize = System::Drawing::Size(1082, 553);
                this->Controls->Add(this->button7);
                this->Controls->Add(this->webBrowser2);
                this->Controls->Add(this->button5);
                this->Controls->Add(this->button6);
                this->Controls->Add(this->button4);
                this->Controls->Add(this->pictureBox1);
                this->Controls->Add(this->label2);
                this->Controls->Add(this->textBox1);
                this->Controls->Add(this->button3);
                this->Controls->Add(this->button2);
                this->Controls->Add(this->label1);
                this->Controls->Add(this->button1);
                this->Icon =
(cli::safe_cast<System::Drawing::Icon^>(resources-
>GetObject(L"$this.Icon")));
                this->MinimumSize = System::Drawing::Size(800, 600);
                this->Name = L"MyForm";
                this->Opacity = 0.96;
                this->Text = L"| FIle Manager";
                this->FormClosing += gcnew
System::Windows::Forms::FormClosingEventHandler(this,
&MyForm::MyForm_FormClosing);
                this->Load += gcnew System::EventHandler(this,
&MyForm::MyForm_Load);

      (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>pictureBox1))->EndInit();
                this->ResumeLayout(false);
                this->PerformLayout();

            }
#pragma endregion
      private: System::Void pictureBox1_Click(System::Object^ sender,
System::EventArgs^ e) {
      }
private: System::Void MyForm_Load(System::Object^ sender, System::EventArgs^
e) {
}
private: System::Void button5_Click(System::Object^ sender,
System::EventArgs^ e) {

      // go Back
      if (webBrowser2->CanGoBack) {
            webBrowser2->GoBack();
      }
```

```
}
private: System::Void button6_Click(System::Object^ sender,
System::EventArgs^ e) {
      // go Forward

      if (webBrowser2->CanGoForward) {
            webBrowser2->GoForward();
      }



}
private: System::Void webBrowser2_DocumentCompleted(System::Object^ sender,
System::Windows::Forms::WebBrowserDocumentCompletedEventArgs^ e) {

      /*String^ urle = System::Convert::ToString(webBrowser2->Url);
      String^ gotovo = urle->*/
}
private: System::Void webBrowser2_DocumentCompleted_1(System::Object^ sender,
System::Windows::Forms::WebBrowserDocumentCompletedEventArgs^ e) {

}
private: System::Void webBrowser2_DocumentCompleted_2(System::Object^ sender,
System::Windows::Forms::WebBrowserDocumentCompletedEventArgs^ e) {
}
private: System::Void textBox1_TextChanged(System::Object^ sender,
System::EventArgs^ e) {
      if (textBox1->Text != "") {
            try {

                  webBrowser2->Url = (gcnew System::Uri(textBox1->Text,
System::UriKind::Absolute));

            }
            catch (...){

                  return;




            }

}
};
private: System::Void button5_Click_1(System::Object^ sender,
System::EventArgs^ e) {
      AmmountBaze::form++;
      MyForm^ file = gcnew MyForm();
      file->Show();


}
private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {
      // disk C
      webBrowser2->Url = gcnew System::Uri("C:\\",
System::UriKind::Absolute);
}
```

```
      private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) {
            // disk D
      webBrowser2->Url = gcnew System::Uri("D:\\",
System::UriKind::Absolute);

      }
private: System::Void button3_Click(System::Object^ sender,
System::EventArgs^ e) {
      // documents

      //String^ usern =
System::Security::Principal::WindowsIdentity::GetCurrent()->Name;
      String^ usern = Environment::UserName;
      String^ url = "C:\\Users\\"+ usern + "\\OneDrive\\Документы";
      webBrowser2->Url = gcnew System::Uri(url, System::UriKind::Absolute);



}
private: System::Void MyForm_FormClosing(System::Object^ sender,
System::Windows::Forms::FormClosingEventArgs^ e) {
      // form closing
      AmmountBaze::form--;
      if (AmmountBaze::form <= 0) {
            Application::Exit();
      }

}
private: System::Void button7_Click(System::Object^ sender,
System::EventArgs^ e) {
      String^ usern = Environment::UserName;
      String^ url = "C:\\Users\\" + usern + "\\OneDrive\\Рабочий стол";
      webBrowser2->Url = gcnew System::Uri(url, System::UriKind::Absolute);

}
};
}
```