

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

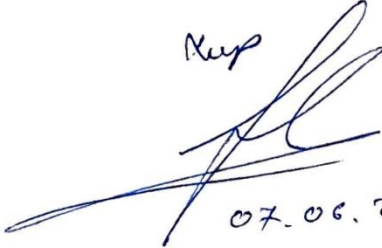
К курсовой работе
по курсу «Программирование»
на тему «Разработка программы «просмотр текстовых файлов
(Viewer).»»

Выполнил:

студент группы 21ВВ2
Тюрин В.И

Принял:

Голотенков Н.О.

Кур

07.06.22

Объем работы по курсу

1. Расчетная часть

Разработка программы.

2. Графическая часть

Схема данных, схема ресурсов системы, схема работы системы, иерархическая структура программы, схема взаимодействия программы.

3. Экспериментальная часть

Отладка программы.

Срок выполнения проекта по разделам

В соответствии с графиком.

Дата выдачи задания "22" февраля

Дата защиты проекта "07" 06.2022

Руководитель Талочников Н.О.

Задание получил "22" февраля 2022г.

Студент Тюрин Владислав Игоревич

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет Вычислительной техники
Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"

Зав. кафедрой ВТ

 М.А. Митрохин

« ____ » ____ 20 ____ г

ЗАДАНИЕ

на курсовое проектирование по курсу

Программирование

Студенту Тершицу Вячеславу Игоревичу Группа 21ВВ2

Тема проекта: *Разработка программы сложной структуры методом нисходящего программирования.*

Исходные данные (технические требования) на проектирование

*Приложение: Просмотр текстовых файлов
Функции приложения: Просмотр и редактирование
Текстового файла, сокращение его по заданной
длине, поиск нужного фрагмента текста*

Обязательные требования к программе:

1. Многомодульность.
2. Использование сложных типов данных – структур, списков, файлов.
3. Режим работы видеосистемы – текстовый / графический.
4. Устройство ввода информации – клавиатура, мышь.
5. Пользовательский интерфейс должен быть построен на основе меню и панели инструментов.
6. Наличие заставки.
7. Операционная система MS Windows.
8. Язык программирования – Си и Ассемблер.
9. Среда разработки ПО – Microsoft Visual Studio.

Содержание

Введение	4
1. Постановка задачи	5
2. Выбор решения	6
3. Описание разработки программы	7
4. Отладка и тестирование	8
5. Описание программы. Функции	10
1. 5.1 Иерархическая структура	10
6. Руководство пользователя	11
Заключение	17
Список используемых источников	18
Приложение А Листинг программы	19
2. Приложение А.1 Файл «MyForm.cpp»	19
3. Приложение А.2 Файл «MyForm.h»	20
4. Приложение А.3 Файл «MyForm1.cpp»	26
5. Приложение А.4 Файл «MyForm1.h»	27
6. Приложение А.5 Файл «MyForm2.cpp»	31
7. Приложение А.6 Файл «MyForm2.h»	32
Приложение В. Схема взаимодействия программы	35

Введение

Компьютеры в современном обществе стали неотъемлемой частью жизни многих людей. Для комфортного использования компьютера существуют разные программы на разных языках программирования.

Язык C++ один из наиболее используемых языков программирования в мире. Он прост, но на нем написано множество приложений, которые используют миллионы людей по всему миру.

В данной работе продемонстрированы возможности создавать приложения для Windows на языке C++.

1. Постановка задачи

Необходимо разработать программу «Просмотр текстовых файлов (Viewer).».

Программа должна выполнять базовые функции Блокнота.

Программа должна предоставить возможность просмотра текстовых файлов, редактировать текстовые файлы, сохранение его под другим именем. Для реализации этих требований нужно разработать интуитивно понятный интерфейс пользователя.

1. Многомодульность.
2. Использование сложных типов данных – списков, структур, файлов.
3. Режим работы видеосистемы- текстовый/графический.
4. Устройство ввода информации – клавиатура, мышь.
5. Пользовательский интерфейс должен быть построен на основе меню и панели инструментов.
6. Наличие заставки
7. Операционная система – MS Windows
8. Язык программирования – Си и Ассемблер.
9. Среда разработки ПО – Microsoft Visual Studio

2. Выбор решения

Приложение «Notepad» состоит из трех модулей:

1. **MyForm.cpp**
2. **MyForm1.cpp**
3. **MyForm2.cpp**

Файл **MyForm.cpp** является основным модулем данной курсовой работы, контролирующим управляющие элементы интерфейса и вызывающем различные функции из других файлов. А также содержит функции, необходимые для реализации меню просмотрщика файлов. Он связан с файлами MyForm1.cpp, MyForm2.cpp и с заголовочным файлом MyForm.h.

Файл **MyForm1.cpp** содержит все функции необходимые для реализации работы интерфейса и функционала программы. Он связан с заголовочными файлами MyForm1.h

Файл **MyForm2.cpp** используется для представления заставки программы.

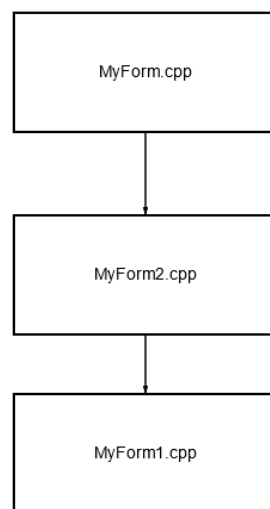


Рис.1 - Структура модулей программы

Всё вышеперечисленное удовлетворяет требованию многомодульности данной курсовой работы. Управление программой осуществляется с помощью контекстного меню. При запуске выводится заставка, и, после нажатия на кнопку «начать», программа готова к работе. Программа предназначена для работы с текстом, текст вводится при помощи клавиатуры, после ввода можно сохранить данные в отдельном файле.

3. Описание разработки программы

Для написания данной программы будет использован язык программирования C++. C++ компилируемый язык программирования общего назначения, сочетает свойства как высокоуровневых, так и низкоуровневых языков программирования. В сравнении с его предшественником, языком программирования Си, наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования. Название «язык программирования C++» происходит от языка программирования С, в котором унарный оператор ++ обозначает инкремент переменной.

Язык программирования C++ широко используется для разработки программного обеспечения. А именно, создание разнообразных прикладных программ, разработка операционных систем, драйверов устройств, а также видео игр и многое другое. В качестве среды программирования был выбран программный продукт Visual Studio 2021. Интерфейс текстового редактора будет представлен в виде окна Windows Form. Управление программой осуществляется выпадающим списком, открывающимся по нажатию ПКМ.

При попытке закрыть окно, не сохранив данные, пользователь увидит сообщение, уведомляющее его о наличии несохраненных данных.

Разработка программы началась с реализации набора функций, необходимых для работы с текстом.

Основная программа **MyForm.cpp** выполняется функцией **main**. В функции **main** выполняется запрос на вывод заставки и на запуск родительского окна.

Был разработан алгоритм функции **main**. Функция отвечает за инициализацию окна Windows Forms, за запрос на вывод заставки.

4. Отладка и тестирование

Отладчик Visual Studio — очень эффективное средство. Прежде чем приступать к его использованию, следует ознакомиться с базовыми терминами, такими как *отладчик*, *отладка* и *режим отладки*. Когда позднее мы будем вести речь о поиске и устранении ошибок, мы будем иметь в виду то же самое.

Отладчик — важнейший инструмент для поиска и устранения ошибок в приложениях. Однако большое значение имеет контекст. Важно использовать все средства, имеющиеся в вашем распоряжении, чтобы быстро устранять ошибки. Зачастую лучшим "средством" являются правильные методики написания кода. Зная, когда лучше использовать отладчик, а когда — другие средства.

В качестве среды разработки была выбрана программа Visual Studio 2021. Программа предоставляет все средства, необходимые при разработке и отладке модулей и программ. В ходе отладки применялись такие возможности, как точка остановки, выполнение кода по шагам, анализ содержимого локальных и глобальных переменных, анализ содержимого памяти.

Тестирование проводилось в рабочем порядке, в процессе разработки, после завершения написания программы.

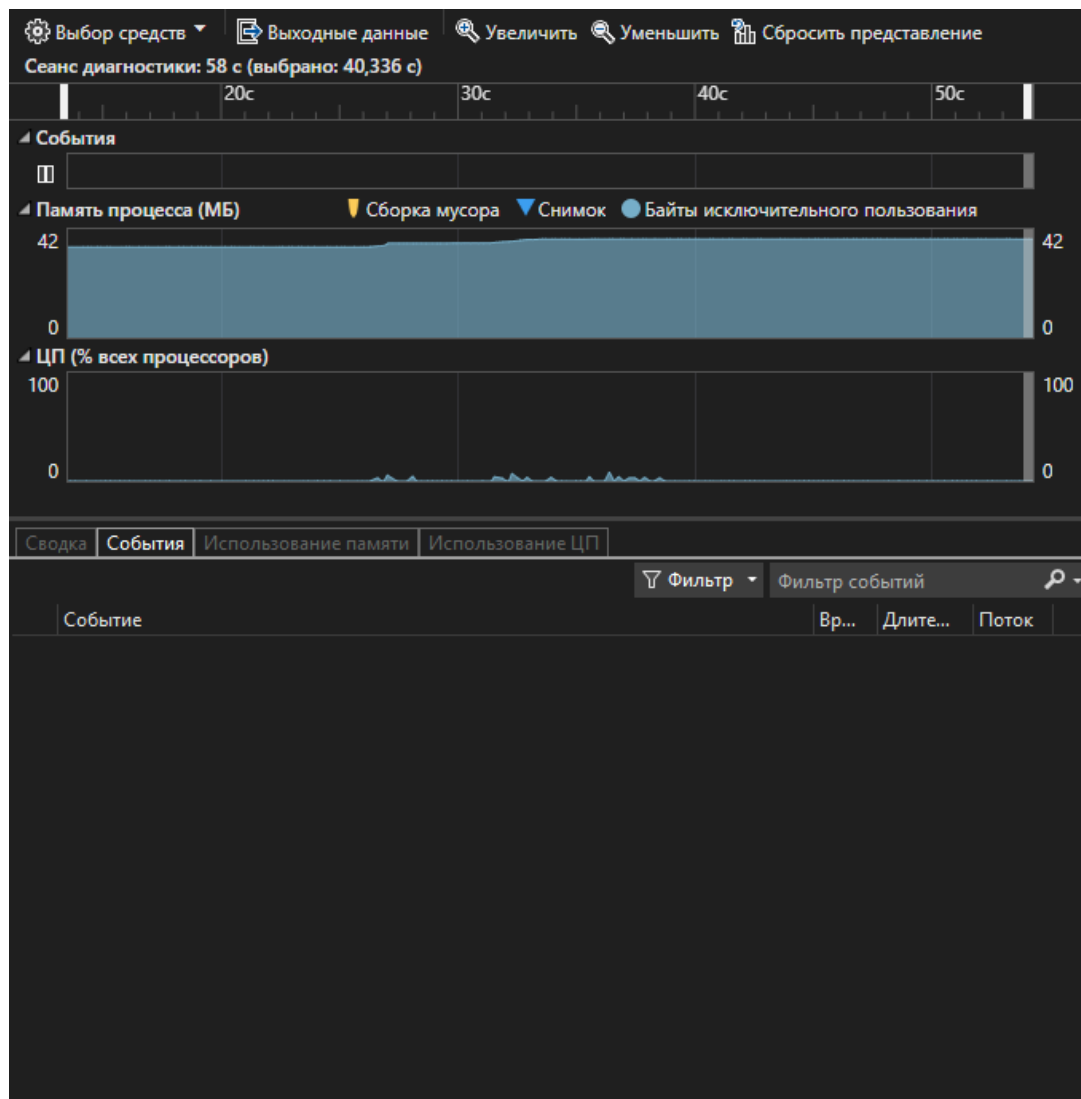


Рис.2 – Использование памяти, и процент загрузки процессора

5. Описание программы. Функции

Приложение **MyForm.cpp** является основным модулем программы. После запуска программы выводится заставка. После инициализации окна Windows Forms пользователь может создать окно с текстовым файлом, набрать в нем текст, сохранить, создать или открыть новый файл с разрешением RTF (Rich Text Format). В данной программе представлен обширный функционал возможностей.

Имеется возможность набора текста, копирование выделенного текста, экспортирование заранее копированного текста из буфера обмена, копирование всего набранного текста, также создание текстового файла, сохранение, открытие уже существующего файла, вставка текущей даты и времени.

Главной особенностью данного текстового редактора является возможность открывать несколько окон для удобной работы с несколькими файлами, и для удобной сортировки была разработана возможность укладывать открытые окна каскадом.

5.1 Иерархическая структура

Иерархическая структура программы представлена на рисунке 2.

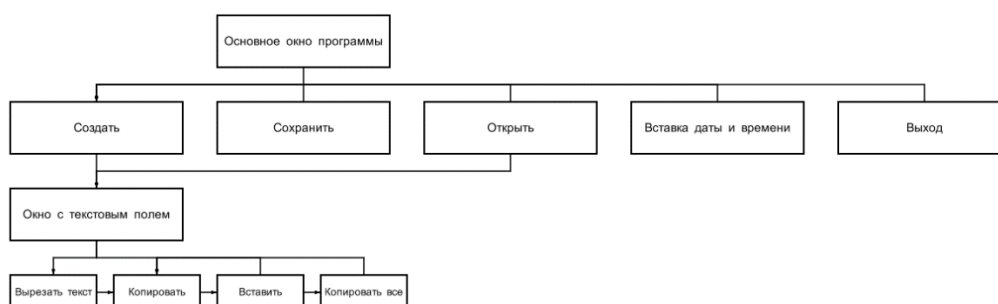


Рис.3 – Иерархическая структура программы

На данной блок схеме представлена иерархическая структура программы Notepad. При запуске мы попадаем в основное окно программы, где мы можем выбрать несколько пунктов: создать, сохранить, открыть файл, Вставка даты и времени и выход из данной программы. При выборе пункта создания файла, открывается окно с текстовым полем, где мы можем писать любой текст. В процессе набора текста можно воспользоваться функционалом данного

текстового редактора: вырезать выделенный текст, копировать в буфер обмена выделенным текст, вставить. Такая структура позволяет работать с несколькими окнами и копировать и вставлять из одного окна в другое текст.

6. Руководство пользователя

Программа Notepad предназначена для просмотра и редактирования текстовых файлов. Она имеет понятный интерфейс и поддерживает множество описанных выше операций, а также интуитивно понятное управление.

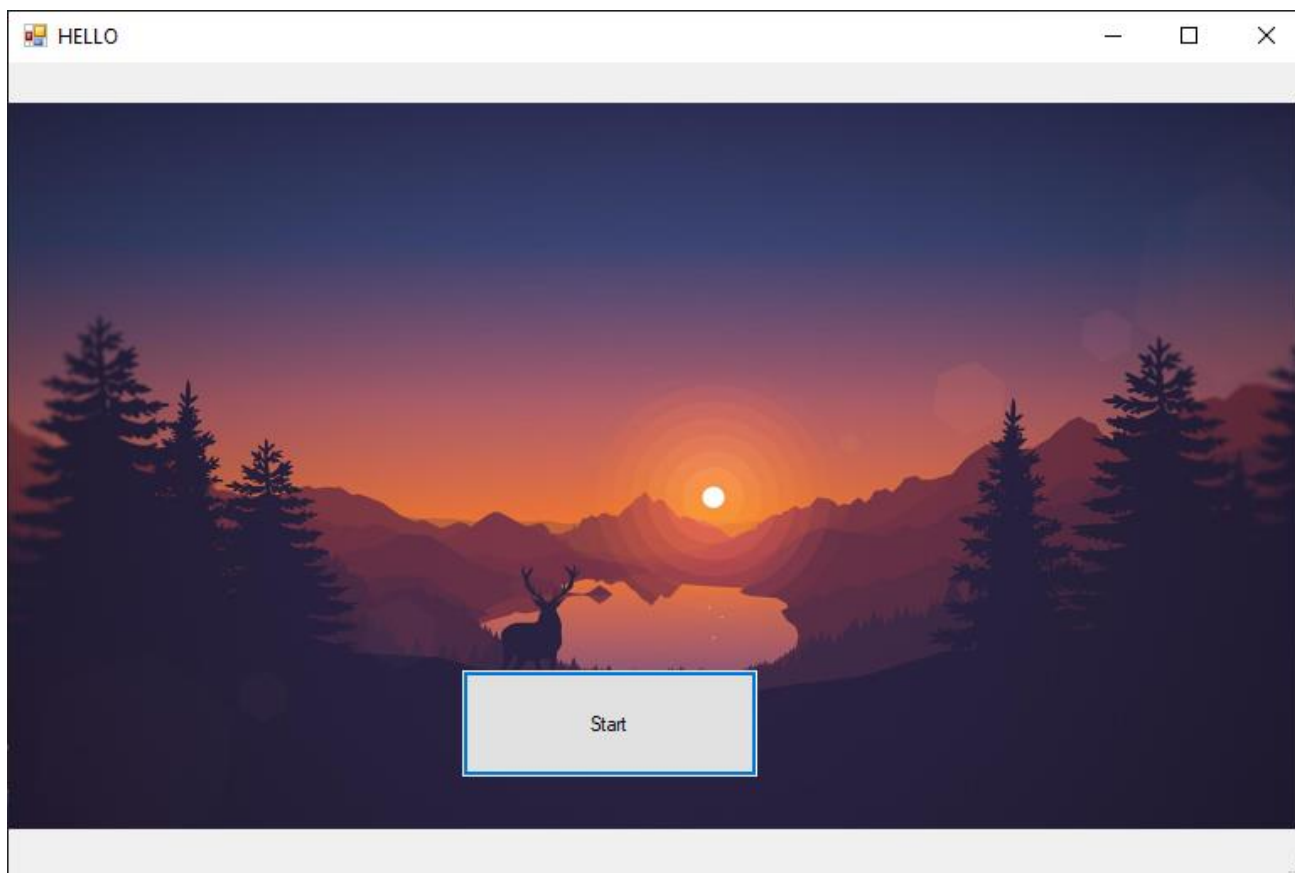


Рисунок 4.1 – заставка

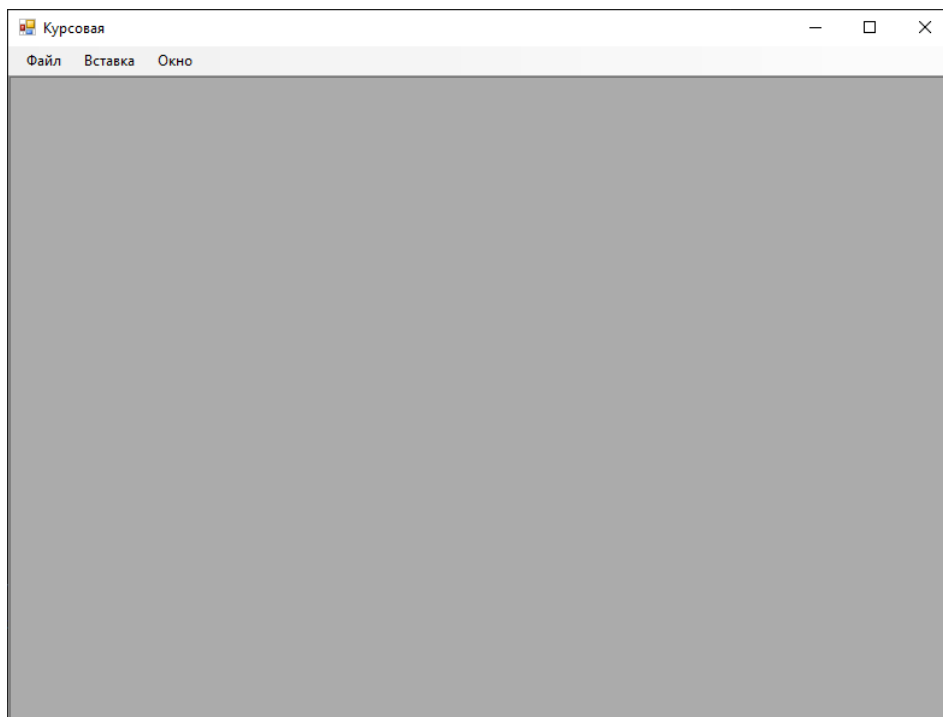


Рисунок 4.2 – основное окно программы

Создать файл можно в меню «Файл» кнопкой «Создать».

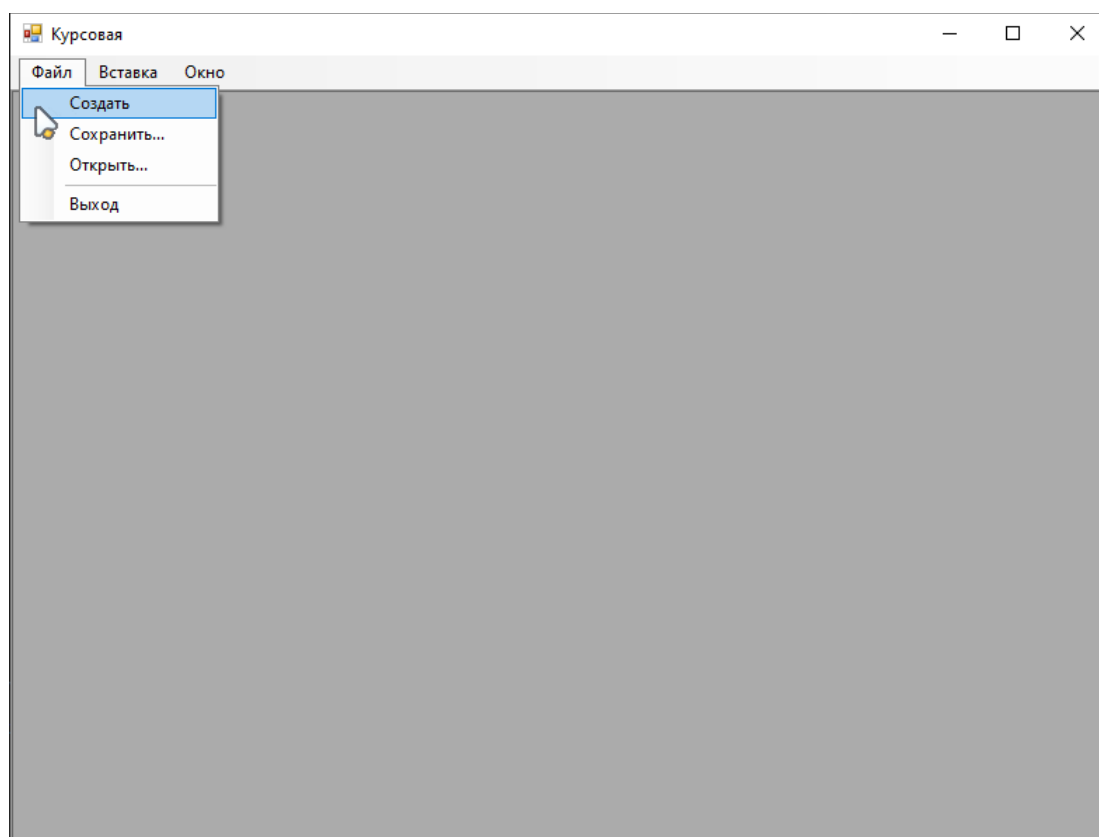


Рисунок 4.3 – создание файла

Пример сохранения написанного текста

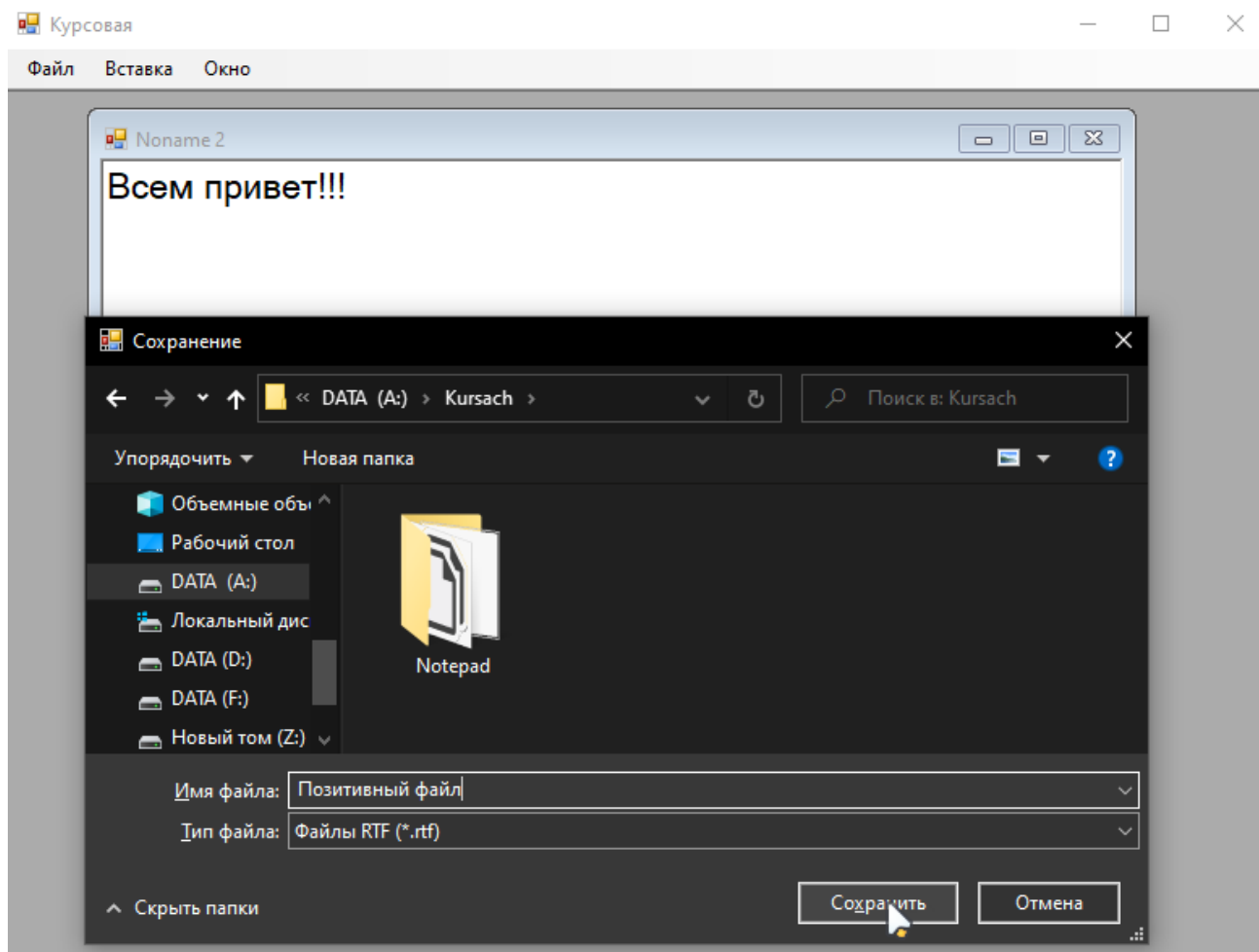


Рисунок 4.4 – сохранение файла

После попытки закрыть файл с несохраненными данными программа предложит сохранить их в файле.

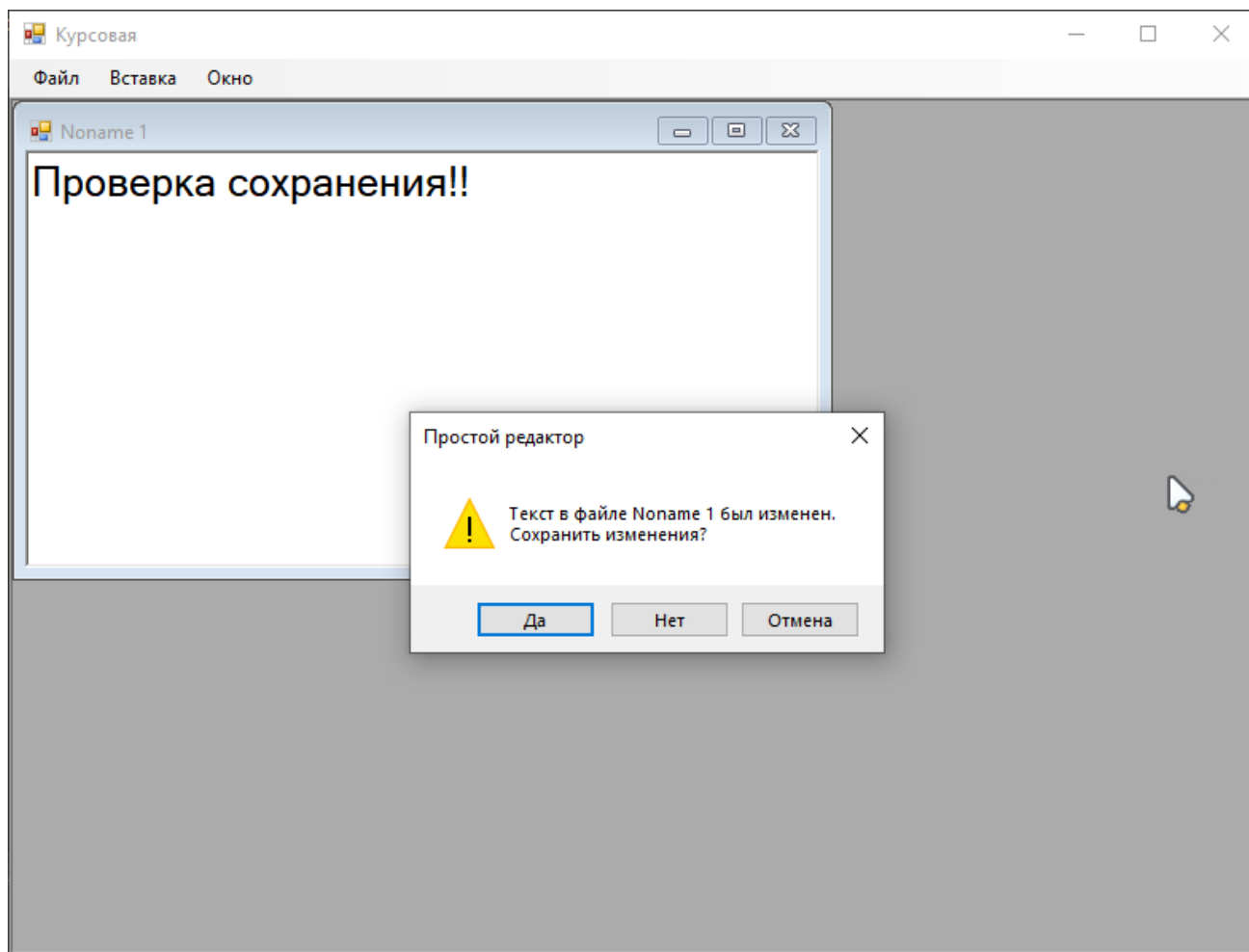


Рисунок 4.5 – выход без сохранения

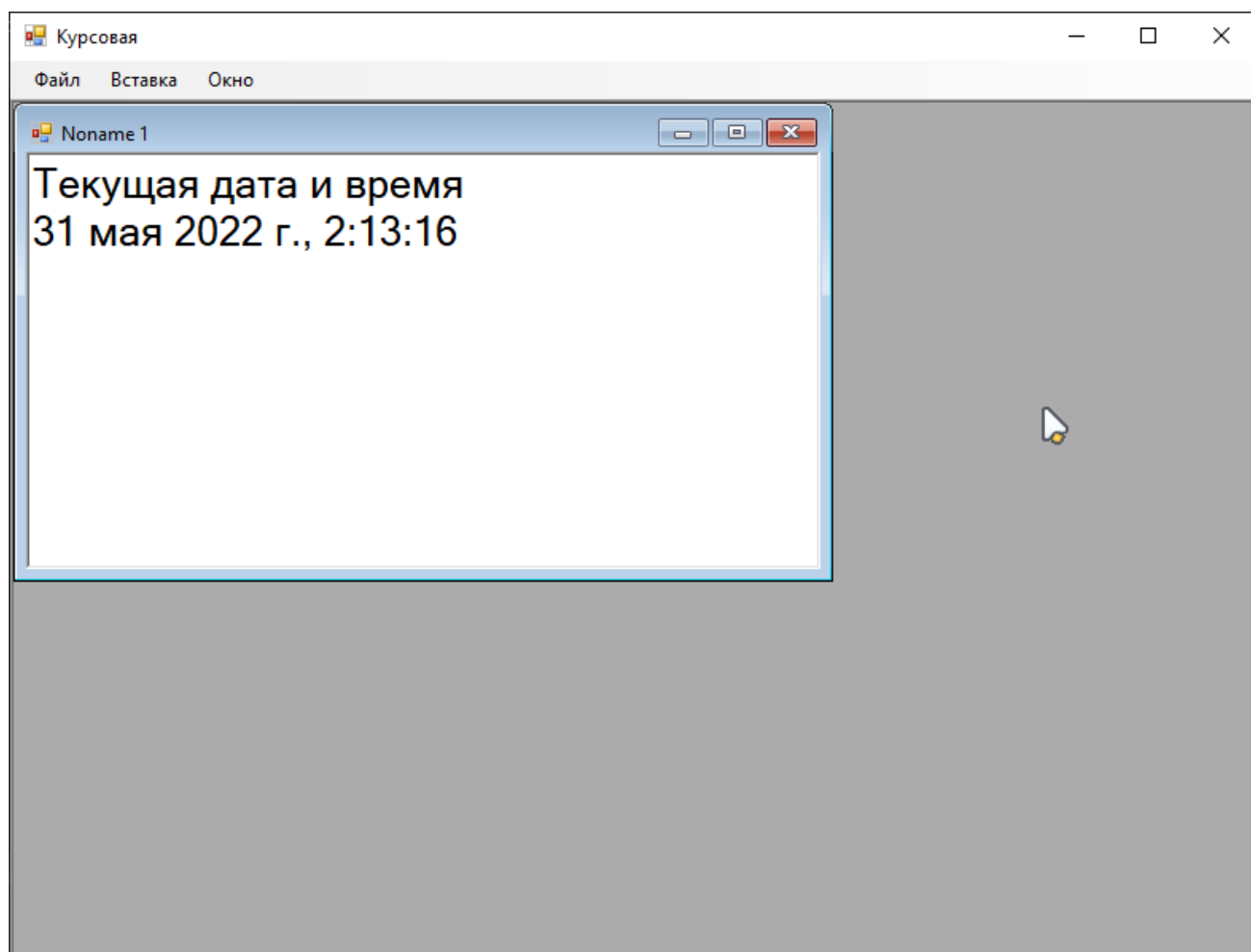


Рисунок 4.6 – Дата и время

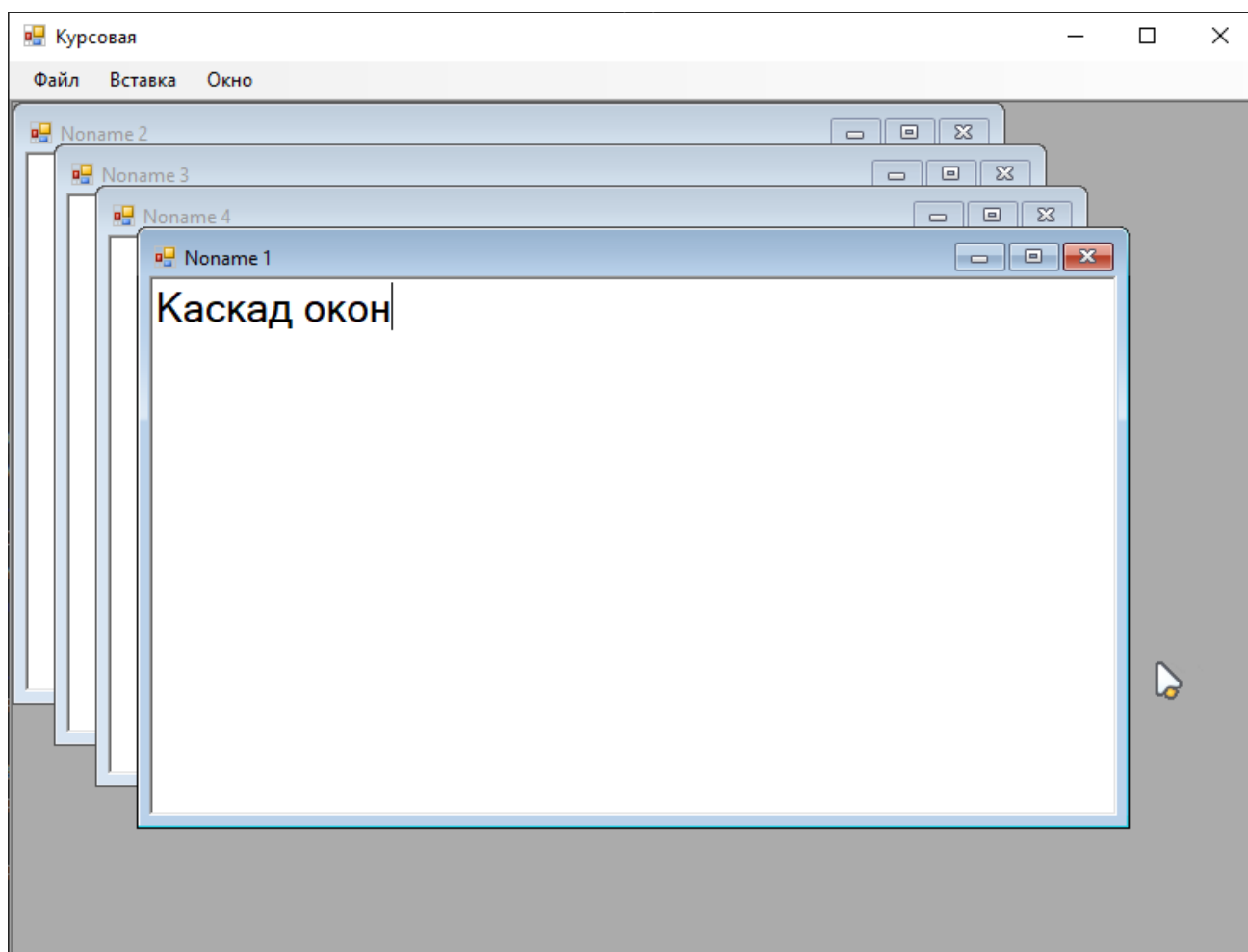


Рисунок 4.7 – Каскад окон

Заключение

При выполнении данной курсовой работы были получены навыки разработки многомодульных программ и приложения с интерфейсом MDI. Были освоены способы обработки событий от клавиатуры. Усвоены принципы реализации интерфейса. Получены базовые навыки программирования на языке C++. Изучены возможности среды программирования VisualStudio 2021. Получены навыки отладки и тестирования программ.

В рамках выполнения курсовой работы была написана программа, выполняющая функции Блокнота. Программа не требует большого количества системных ресурсов, обеспечивает быструю обработку данных и мгновенный показ результата пользователю.

Программу можно улучшить, оптимизировав код для более быстрого выполнения функций.

Список используемых источников

1. Практическое руководство Microsoft по созданию MDI приложений

URL: [HTTPS://DOCS.MICROSOFT.COM/RU-](https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/advanced/how-to-create-mdi-child-forms?view=netframeworkdesktop-4.8)

[RU/DOTNET/DESKTOP/WINFORMS/ADVANCED/HOW-TO-CREATE-MDI-CHILD-FORMS?VIEW=NETFRAMEWORKDESKTOP-4.8](https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/advanced/how-to-create-mdi-child-forms?view=netframeworkdesktop-4.8)

(дата обращения 15.04.2022)

2. Руководство по программированию в Windows Forms

URL: <https://metanit.com/> (дата обращения 15.04.2022)

Приложение А

Листинг программы

Приложение А.1

Файл «MyForm.cpp»

```
#include "MyForm.h"
#include "MyForm2.h"
using namespace System;
using namespace System::Windows::Forms;
int iplus(int i) {
    __asm {
        inc i
        ret
    }
}

[STAThread]
int main(array<String^>^ arg) {
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);

    Notepad::MyForm form;
    Notepad::MyForm2 form2;

    Application::Run(% form2);

    return 0;
}
```

Приложение A.2

Файл «MyForm.h»

```
#pragma once
#include "MyForm1.h" /* прописал класс потомка в предке */

namespace Notepad {
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

    /// <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::MenuStrip^ menuStrip1;
    protected:
    private: System::Windows::Forms::ToolStripMenuItem^ файлToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ создатьToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ выходToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ сохранитьToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ открытьToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ окноToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ каскадToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ правкаToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ датаИВремяToolStripMenuItem;
    private: System::Windows::Forms::OpenFileDialog^ openFileDialog1;
    private: System::Windows::Forms::SaveFileDialog^ saveFileDialog1;
    private: System::Windows::Forms::ToolStripSeparator^ toolStripMenuItem1;

    protected:

    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)
    {
        this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());
        this->файлToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->создатьToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->сохранитьToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->открытьToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->toolStripMenuItem1 = (gcnew
System::Windows::Forms::ToolStripSeparator());
        this->выходToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->правкаToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->датаИВремяToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->окноToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->каскадToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->openFileDialog1 = (gcnew System::Windows::Forms::OpenFileDialog());
        this->saveFileDialog1 = (gcnew System::Windows::Forms::SaveFileDialog());
        this->menuStrip1->SuspendLayout();
        this->SuspendLayout();
        //
        // menuStrip1
        //
        this->menuStrip1->ImageScalingSize = System::Drawing::Size(20, 20);
        this->menuStrip1->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(3) {
            this->файлToolStripMenuItem,
            this->правкаToolStripMenuItem, this->окноToolStripMenuItem
        });
        this->menuStrip1->Location = System::Drawing::Point(0, 0);
        this->menuStrip1->Name = L"menuStrip1";
        this->menuStrip1->Size = System::Drawing::Size(784, 24);
        this->menuStrip1->TabIndex = 1;
        this->menuStrip1->Text = L"menuStrip1";
        //
        // файлToolStripMenuItem
        //
        this->файлToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(5) {
            this->создатьToolStripMenuItem,
            this->сохранитьToolStripMenuItem, this->открытьToolStripMenuItem,
this->toolStripMenuItem1, this->выходToolStripMenuItem
        });
        this->файлToolStripMenuItem->Name = L"файлToolStripMenuItem";
        this->файлToolStripMenuItem->Size = System::Drawing::Size(48, 20);
        this->файлToolStripMenuItem->Text = L"&Файл";
        //
        // создатьToolStripMenuItem
        //
        this->создатьToolStripMenuItem->Name = L"создатьToolStripMenuItem";
        this->создатьToolStripMenuItem->Size = System::Drawing::Size(142, 22);
    }

```

```

        this->создатьToolStripMenuItem->Text = L"Создать";
        this->создатьToolStripMenuItem->Click += gcnew System::EventHandler(this,
&MyForm::создатьToolStripMenuItem_Click);
        //
        // сохранитьToolStripMenuItem
        //
        this->сохранитьToolStripMenuItem->Name = L"сохранитьToolStripMenuItem";
        this->сохранитьToolStripMenuItem->Size = System::Drawing::Size(142, 22);
        this->сохранитьToolStripMenuItem->Text = L"Сохранить...";
        this->сохранитьToolStripMenuItem->Click += gcnew System::EventHandler(this,
&MyForm::сохранитьToolStripMenuItem_Click);
        //
        // открытьToolStripMenuItem
        //
        this->открытьToolStripMenuItem->Name = L"открытьToolStripMenuItem";
        this->открытьToolStripMenuItem->Size = System::Drawing::Size(142, 22);
        this->открытьToolStripMenuItem->Text = L"Открыть...";
        this->открытьToolStripMenuItem->Click += gcnew System::EventHandler(this,
&MyForm::открытьToolStripMenuItem_Click);
        //
        // toolStripMenuItem1
        //
        this->toolStripMenuItem1->Name = L"toolStripMenuItem1";
        this->toolStripMenuItem1->Size = System::Drawing::Size(139, 6);
        //
        // выходToolStripMenuItem
        //
        this->выходToolStripMenuItem->Name = L"выходToolStripMenuItem";
        this->выходToolStripMenuItem->Size = System::Drawing::Size(142, 22);
        this->выходToolStripMenuItem->Text = L"Выход";
        this->выходToolStripMenuItem->Click += gcnew System::EventHandler(this,
&MyForm::выходToolStripMenuItem_Click);
        //
        // правкаToolStripMenuItem
        //
        this->правкаToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(1) { this->датаИВремяToolStripMenuItem });
        this->правкаToolStripMenuItem->Name = L"правкаToolStripMenuItem";
        this->правкаToolStripMenuItem->Size = System::Drawing::Size(61, 20);
        this->правкаToolStripMenuItem->Text = L"Вставка";
        //
        // датаИВремяToolStripMenuItem
        //
        this->датаИВремяToolStripMenuItem->Name = L"датаИВремяToolStripMenuItem";
        this->датаИВремяToolStripMenuItem->Size = System::Drawing::Size(146, 22);
        this->датаИВремяToolStripMenuItem->Text = L"Дата и время";
        this->датаИВремяToolStripMenuItem->Click += gcnew System::EventHandler(this,
&MyForm::датаИВремяToolStripMenuItem_Click);
        //
        // окноToolStripMenuItem
        //
        this->окноToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(1) { this->каскадToolStripMenuItem });
        this->окноToolStripMenuItem->Name = L"окноToolStripMenuItem";
        this->окноToolStripMenuItem->Size = System::Drawing::Size(48, 20);
        this->окноToolStripMenuItem->Text = L"Окно";
        //
        // каскадToolStripMenuItem
        //
        this->каскадToolStripMenuItem->Name = L"каскадToolStripMenuItem";
        this->каскадToolStripMenuItem->Size = System::Drawing::Size(111, 22);
        this->каскадToolStripMenuItem->Text = L"Каскад";
        this->каскадToolStripMenuItem->Click += gcnew System::EventHandler(this,
&MyForm::каскадToolStripMenuItem_Click);
        //

```

```

        // openFileDialog1
        //
        this->openFileDialog1->Filter = L"Файлы RTF|*.rtf";
        this->openFileDialog1->FileOk += gcnew
System::ComponentModel::CancelEventHandler(this, &MyForm::openFileDialog1_FileOk);
        //
        // saveFileDialog1
        //
        this->saveFileDialog1->Filter = L"Файлы RTF|*.rtf";
        //
        // MyForm
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
        this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
        this->ClientSize = System::Drawing::Size(784, 562);
        this->Controls->Add(this->menuStrip1);
        this->IsMdiContainer = true;
        this->MainMenuStrip = this->menuStrip1;
        this->Name = L"MyForm";
        this->RightToLeftLayout = true;
        this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Text = L"Курсовая";
        this->FormClosing += gcnew
System::Windows::Forms::FormClosingEventHandler(this, &MyForm::MyForm_FormClosing);
        this->Load += gcnew System::EventHandler(this, &MyForm::MyForm_Load);
        this->menuStrip1->ResumeLayout(false);
        this->menuStrip1->PerformLayout();
        this->ResumeLayout(false);
        this->PerformLayout();

    }
#pragma endregion
    public:
        static int winCounter; //Просто счётчик окон
    private: void createNewWindow() { //Метод для создания нового окна
        MyForm1^ newForm = gcnew MyForm1(); //Вверху файла есть #include "MyForm1.h"
        newForm->MdiParent = this; //Указали новой форме, кто её родитель
        newForm->Text = "Noname " + ++winCounter; //Номер нового окна - в заголовок
        newForm->Show(); //Показываем новую форму
    }

    private: System::Void создатьToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
        createNewWindow();
    }
    private: System::Void выходToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
        this->Close();
    }

    private: System::Void сохранитьToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
        Form^ activeChild = this->ActiveMdiChild;
        if (activeChild != nullptr) {
            RichTextBox^ theBox = (RichTextBox^)activeChild->ActiveControl;
            if (!activeChild->Text->Empty) saveFileDialog1->FileName = activeChild->Text;
            if (saveFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK) {
                try {
                    theBox->SaveFile(saveFileDialog1->FileName);
                }
                catch (...) {
                    MessageBox::Show(String::Format(L"Ошибка сохранения файла {0}",
saveFileDialog1->FileName));
                }
            }
        }
    }

```

```

        }
        activeChild->Text = saveFileDialog1->FileName;
    }
}
else {
    MessageBox::Show(L"Не выбрано ни одно окно");
}
}

private: System::Void открытьToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    createNewWindow(); //Открываем всегда в новом окне
    Form^ activeChild = this->ActiveMdiChild;
    if (activeChild != nullptr) {
        RichTextBox^ theBox = (RichTextBox^)activeChild->ActiveControl;
        if (!activeChild->Text->Empty) openFileDialog1->FileName = activeChild->Text;
        if (openFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK) {
            if (openFileDialog1->FileName == nullptr) return;
            try {
                theBox->LoadFile(openFileDialog1->FileName);
                theBox->Modified = false;
            }
            catch (IO::FileNotFoundException^ e) {
                MessageBox::Show(e->Message + "\nНет такого файла", "Ошибка",
                    MessageBoxButtons::OK, MessageBoxIcon::Exclamation);
            }
            catch (Exception^ e) { // Отчет о других ошибках
                MessageBox::Show(e->Message, "Ошибка", MessageBoxButtons::OK,
MessageBoxIcon::Exclamation);
            }
            activeChild->Text = openFileDialog1->FileName;
        }
    }
    else {
        MessageBox::Show(L"Не выбрано ни одно окно");
    }
}

private: System::Void MyForm_FormClosing(System::Object^ sender,
System::Windows::Forms::FormClosingEventArgs^ e) {

    e->Cancel = false; //Отказ в потомке не приводит к отмене закрытия родительского
окна
}

private: System::Void датаИВремяToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    Form^ activeChild = this->ActiveMdiChild;
    if (activeChild != nullptr) {
        try {
            RichTextBox^ theBox = (RichTextBox^)activeChild->ActiveControl;
            if (theBox != nullptr) {
                DateTime^ d = gcnew DateTime(DateTime::Now.Year,
DateTime::Now.Month, DateTime::Now.Day,
                DateTime::Now.Hour, DateTime::Now.Minute,
DateTime::Now.Second);
                theBox->Text += String::Format("\n{0}, {1}\n", d-
>ToLongDateString(), d->ToLongTimeString());
            }
        }
        catch (...) {
            MessageBox::Show(L"Ошибка вставки даты и времени");
        }
    }
}

```

```

        else {
            MessageBox::Show(L"Не выбрано ни одно окно");
        }
    }

    private: System::Void каскадToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
        //Уложить дочерние окна каскадом
        this->LayoutMdi(MdiLayout::Cascade);
    }

    //////////////////////////////////////
    private: System::Void openFileDialog1_FileOk(System::Object^ sender,
System::ComponentModel::CancelEventArgs^ e) {
    }
    private: System::Void MyForm_Load(System::Object^ sender, System::EventArgs^ e) {
    }
};
}

```

Приложение А.3
Файл «MyForm1.cpp»

```
#include "MyForm1.h"
```

Приложение А.4

Файл «MyForm1.h»

```
#pragma once

namespace Notepad {
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

    /// <summary>
    /// Сводка для MyForm1
    /// </summary>
    public ref class MyForm1 : public System::Windows::Forms::Form
    {
    public:
        MyForm1()
        {
            InitializeComponent();
            //
            //TODO: добавьте код конструктора
            //
        }
    protected:
        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        ~MyForm1()
        {
            if (components)
            {
                delete components;
            }
        }
    private: System::Windows::Forms::RichTextBox^ richTextBox1;
    private: System::Windows::Forms::ContextMenuStrip^ contextMenuStrip1;
    private: System::Windows::Forms::ToolStripMenuItem^ вырезатьToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ копироватьToolStripMenuItem;
    копироватьВсёToolStripMenuItem;
    private: System::Windows::Forms::ToolStripMenuItem^ вставитьToolStripMenuItem;
    private: System::Windows::Forms::SaveFileDialog^ saveFileDialog1;
    private: System::ComponentModel::IContainer^ components;
    protected:

    private:
        /// <summary>
        /// Обязательная переменная конструктора.
        /// </summary>

#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)
        {
            this->components = (gcnew System::ComponentModel::Container());
            this->richTextBox1 = (gcnew System::Windows::Forms::RichTextBox());
```

```

        this->contextMenuStrip1 = (gcnew
System::Windows::Forms::ContextMenuStrip(this->components));
        this->вырезатьToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->копироватьToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->копироватьВсёToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->вставитьToolStripMenuItem = (gcnew
System::Windows::Forms::ToolStripMenuItem());
        this->saveFileDialog1 = (gcnew
System::Windows::Forms::SaveFileDialog());
        this->contextMenuStrip1->SuspendLayout();
        this->SuspendLayout();
        //
        // richTextBox1
        //
        this->richTextBox1->ContextMenuStrip = this->contextMenuStrip1;
        this->richTextBox1->Dock = System::Windows::Forms::DockStyle::Fill;
        this->richTextBox1->Location = System::Drawing::Point(0, 0);
        this->richTextBox1->Margin = System::Windows::Forms::Padding(4);
        this->richTextBox1->Name = L"richTextBox1";
        this->richTextBox1->Size = System::Drawing::Size(379, 322);
        this->richTextBox1->TabIndex = 0;
        this->richTextBox1->Text = L"";
        this->richTextBox1->SelectionChanged += gcnew System::EventHandler(this,
&MyForm1::richTextBox1_SelectionChanged);
        this->richTextBox1->TextChanged += gcnew System::EventHandler(this,
&MyForm1::richTextBox1_TextChanged);
        //
        // contextMenuStrip1
        //
        this->contextMenuStrip1->ImageScalingSize = System::Drawing::Size(20,
20);
        this->contextMenuStrip1->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(4) {
            this->вырезатьToolStripMenuItem,
            this->копироватьToolStripMenuItem, this->
копироватьВсёToolStripMenuItem, this->вставитьToolStripMenuItem
        });
        this->contextMenuStrip1->Name = L"contextMenuStrip1";
        this->contextMenuStrip1->Size = System::Drawing::Size(196, 108);
        //
        // вырезатьToolStripMenuItem
        //
        this->вырезатьToolStripMenuItem->Enabled = false;
        this->вырезатьToolStripMenuItem->Name = L"вырезатьToolStripMenuItem";
        this->вырезатьToolStripMenuItem->Size = System::Drawing::Size(195, 26);
        this->вырезатьToolStripMenuItem->Text = L"Вырезать";
        this->вырезатьToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MyForm1::вырезатьToolStripMenuItem_Click);
        //
        // копироватьToolStripMenuItem
        //
        this->копироватьToolStripMenuItem->Enabled = false;
        this->копироватьToolStripMenuItem->Name =
L"копироватьToolStripMenuItem";
        this->копироватьToolStripMenuItem->Size = System::Drawing::Size(195,
26);
        this->копироватьToolStripMenuItem->Text = L"Копировать";
        this->копироватьToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MyForm1::копироватьToolStripMenuItem_Click);
        //
        // копироватьВсёToolStripMenuItem
        //
        this->копироватьВсёToolStripMenuItem->Enabled = false;

```

```

        this->копироватьВсёToolStripMenuItem->Name =
L"копироватьВсёToolStripMenuItem";
        this->копироватьВсёToolStripMenuItem->Size = System::Drawing::Size(195,
26);
        this->копироватьВсёToolStripMenuItem->Text = L"Копировать всё";
        this->копироватьВсёToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MyForm1::копироватьВсёToolStripMenuItem_Click);
        //
        // вставитьToolStripMenuItem
        //
        this->вставитьToolStripMenuItem->Enabled = false;
        this->вставитьToolStripMenuItem->Name = L"вставитьToolStripMenuItem";
        this->вставитьToolStripMenuItem->Size = System::Drawing::Size(195, 26);
        this->вставитьToolStripMenuItem->Text = L"Вставить";
        this->вставитьToolStripMenuItem->Click += gcnew
System::EventHandler(this, &MyForm1::вставитьToolStripMenuItem_Click);
        //
        // saveFileDialog1
        //
        this->saveFileDialog1->Filter = L"Файлы RTF|*.rtf";
        //
        // MyForm1
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(8, 16);
        this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
        this->ClientSize = System::Drawing::Size(379, 322);
        this->ContextMenuStrip = this->contextMenuStrip1;
        this->Controls->Add(this->richTextBox1);
        this->Margin = System::Windows::Forms::Padding(4);
        this->Name = L"MyForm1";
        this->Text = L"MyForm1";
        this->FormClosing += gcnew
System::Windows::Forms::FormClosingEventHandler(this, &MyForm1::MyForm1_FormClosing);
        this->Load += gcnew System::EventHandler(this, &MyForm1::MyForm1_Load);
        this->contextMenuStrip1->ResumeLayout(false);
        this->ResumeLayout(false);

    }

#pragma endregion

private: System::Void checkForClipboardFormat() {
    if (Clipboard::GetDataObject()->GetDataPresent(DataFormats::Rtf))
        contextMenuStrip1->Items[3]->Enabled = true; //Включить пункт Вставить
    else contextMenuStrip1->Items[3]->Enabled = false;
}

private: System::Void MyForm1_Load(System::Object^ sender, System::EventArgs^ e) {
    checkForClipboardFormat();
}

private: System::Void richTextBox1_SelectionChanged(System::Object^ sender,
System::EventArgs^ e) {
    if (richTextBox1->SelectionLength == 0)
        for (int i = 0; i < 2; i++) contextMenuStrip1->Items[i]->Enabled =
false;
    else
        for (int i = 0; i < 2; i++) contextMenuStrip1->Items[i]->Enabled = true;
    //Включить Вырезать, Копировать
    checkForClipboardFormat();
}

private: System::Void richTextBox1_TextChanged(System::Object^ sender,
System::EventArgs^ e) {
    if (richTextBox1->Text->Length < 1) {
        richTextBox1->Modified = false; //Пустой текст не предлагать сохранить
    }
}

```

```

        for (int i = 1; i < 3; i++) contextMenuStrip1->Items[2]->Enabled =
false; //и копировать
    }
    else //Включить Копировать, Копировать все
        for (int i = 1; i < 3; i++) contextMenuStrip1->Items[2]->Enabled = true;
    checkForClipboardFormat();
}

private: System::Void вырезатьToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    if (richTextBox1->SelectionLength > 0) richTextBox1->Cut();
}
private: System::Void копироватьToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    if (richTextBox1->SelectionLength > 0) richTextBox1->Copy();
}
private: System::Void копироватьВсёToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    richTextBox1->SelectAll(); richTextBox1->Copy(); richTextBox1->DeselectAll();
}
private: System::Void вставитьToolStripMenuItem_Click(System::Object^ sender,
System::EventArgs^ e) {
    if (Clipboard::GetDataObject()->GetDataPresent(DataFormats::Rtf)) { //Есть
Rtf в буфере
        if (richTextBox1->SelectionLength > 0) { //И что-то выделено,
//спросим, как вставлять – поверх выделенного или в конец?
            if (MessageBox::Show(L"Вставить поверх выделения?", L"Сообщение",
MessageBoxButtons::YesNo) ==
System::Windows::Forms::DialogResult::No)
                richTextBox1->SelectionStart = richTextBox1->Text->Length;
        }
        richTextBox1->Paste();
    }
}

private: System::Void MyForm1_FormClosing(System::Object^ sender,
System::Windows::Forms::FormClosingEventArgs^ e) {
    if (richTextBox1->Modified == false) return;
    saveFileDialog1->FileName = this->Text;
    auto MBox = MessageBox::Show(
        String::Format(L"Текст в файле {0} был изменен.\nСохранить изменения?",
saveFileDialog1->FileName),
        L"Простой редактор", MessageBoxButtons::YesNoCancel,
MessageBoxIcon::Exclamation);
    if (MBox == System::Windows::Forms::DialogResult::No) return;
    if (MBox == System::Windows::Forms::DialogResult::Cancel) e->Cancel = true;
    if (MBox == System::Windows::Forms::DialogResult::Yes) {
        if (saveFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK) {
            try {
                richTextBox1->SaveFile(saveFileDialog1->FileName);
                richTextBox1->Modified = false;
            }
            catch (Exception^ e) {
                MessageBox::Show(L"Ошибка сохранения файла");
            }
            return;
        }
        else e->Cancel = true; // Передумал закрывать
    }
}

//////////
};
}

```

Приложение А.5
Файл «MyForm2.cpp»

```
#include "MyForm2.h"
```

Приложение A.6

Файл «MyForm2.h»

```
#pragma once
#include "MyForm.h"
namespace Notepad {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

    /// <summary>
    /// Сводка для MyForm2
    /// </summary>
    public ref class MyForm2 : public System::Windows::Forms::Form
    {
    public:
        MyForm2(void)
        {
            InitializeComponent();
            //
            //TODO: добавьте код конструктора
            //
        }

    protected:
        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        ~MyForm2()
        {
            if (components)
            {
                delete components;
            }
        }

    private: System::Windows::Forms::Button^ button1;
    private: System::Windows::Forms::PictureBox^ pictureBox1;

    protected:

    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(MyForm2::typeid));
            this->button1 = (gcnew System::Windows::Forms::Button());
            this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());
            (cli::safe_cast<System::ComponentModel::ISupportInitialize>(this-
>pictureBox1))->BeginInit();
```

```

        this->SuspendLayout();
        //
        // button1
        //
        this->button1->DialogResult =
System::Windows::Forms::DialogResult::Cancel;
        this->button1->Location = System::Drawing::Point(269, 360);
        this->button1->Name = L"button1";
        this->button1->Size = System::Drawing::Size(175, 63);
        this->button1->TabIndex = 0;
        this->button1->Text = L"Start";
        this->button1->TextImageRelation =
System::Windows::Forms::TextImageRelation::TextAboveImage;
        this->button1->UseVisualStyleBackColor = true;
        this->button1->Click += gcnew System::EventHandler(this,
&MyForm2::button1_Click);
        //
        // pictureBox1
        //
        this->pictureBox1->Image =
(cli::safe_cast<System::Drawing::Image^>(resources->GetObject(L"pictureBox1.Image"))));
        this->pictureBox1->Location = System::Drawing::Point(0, 0);
        this->pictureBox1->Name = L"pictureBox1";
        this->pictureBox1->Size = System::Drawing::Size(766, 479);
        this->pictureBox1->SizeMode =
System::Windows::Forms::PictureBoxSizeMode::Zoom;
        this->pictureBox1->TabIndex = 1;
        this->pictureBox1->TabStop = false;
        this->pictureBox1->Click += gcnew System::EventHandler(this,
&MyForm2::pictureBox1_Click_2);
        //
        // MyForm2
        //
        this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
        this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
        this->AutoSize = true;
        this->AutoSizeMode =
System::Windows::Forms::AutoSizeMode::GrowAndShrink;
        this->ClientSize = System::Drawing::Size(766, 479);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->pictureBox1);
        this->Name = L"MyForm2";
        this->SizeGripStyle = System::Windows::Forms::SizeGripStyle::Show;
        this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Text = L"HELLO";
        this->Load += gcnew System::EventHandler(this, &MyForm2::MyForm2_Load);
(cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this->
pictureBox1))->EndInit();
        this->ResumeLayout(false);

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

    }
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{
    MyForm^ f2 = gcnew MyForm;
    f2->Show();
    this->Hide();
}
private: System::Void pictureBox1_Click_1(System::Object^ sender,
System::EventArgs^ e) {

```

```
    }  
    private: System::Void MyForm2_Load(System::Object^ sender, System::EventArgs^ e) {  
    }  
    private: System::Void pictureBox1_Click_2(System::Object^ sender,  
System::EventArgs^ e) {  
    }  
};  
}
```

Приложение В. Схема взаимодействия программы

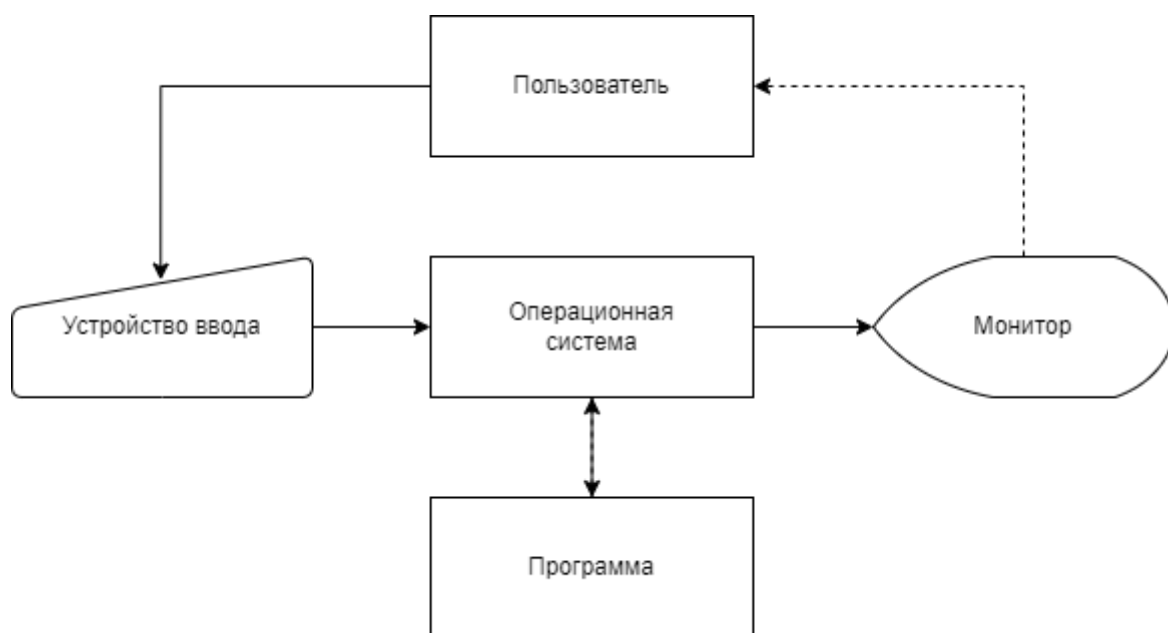


Рисунок 4 – Схема взаимодействия программы