

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 17.12.2021 11:04:46
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eab73e943df4a4851fda56d089

МИНОБРАЗОВАНИЯ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ

Проректор по учебной работе

О. Г. Локтионова

« 1 » 02



ИНТЕРФЕЙСЫ СО СВОБОДНОЙ НАВИГАЦИЕЙ
В СРЕДЕ C++BUILDER

Методические указания по выполнению
лабораторных работ по дисциплине
«Технологии и методы программирования»
для студентов укрупненной группы специальностей и
направлений подготовки бакалавров 10.00.00

Курск 2018

УДК 004

Составитель: К.А. Тезик

Рецензент

Кандидат технических наук, доцент А. Л. Марухленко

Интерфейсы со свободной навигацией в среде C++ Builder: методические указания по выполнению лабораторных работ / Юго-Зап. гос. ун-т; сост.: К. А. Тезик, Курск, 2018. 21 с.: ил. 4, Библиогр.: с. 21.

Содержат краткие теоретические положения о типах пользовательских интерфейсов. Рассмотрены практические задачи разработки графических пользовательских интерфейсов.

Методические указания соответствуют требованиям программы по направлению подготовки бакалавров: информационная безопасность.

Предназначены для студентов укрупненной группы специальностей и направления подготовки бакалавров 10.00.00 дневной формы обучения.

Текст печатается в авторской редакции

Подписано в печать 11.02.18 Формат 60x84 1/16.

Усл. печ. л. 1,13. Уч. – изд. л. 1,11. Тираж 100 экз. Заказ. Бесплатно. 252

Юго - Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

Введение

C++ Builder — программный продукт, инструмент быстрой разработки приложений (RAD), интегрированная среда программирования (IDE), система, используемая программистами для разработки программного обеспечения на языке программирования C++. Изначально разрабатывался компанией Borland Software, а затем её подразделением CodeGear, ныне принадлежащим компании Embarcadero Technologies.

C++ Builder объединяет в себе комплекс объектных библиотек (STL, VCL, CLX, MFC и др.), компилятор, отладчик, редактор кода и многие другие компоненты. Цикл разработки аналогичен Delphi. Большинство компонентов, разработанных в Delphi, можно использовать и в C++ Builder без модификации, но обратное утверждение неверно. C++ Builder содержит инструменты, которые при помощи drag-and-drop действительно делают разработку визуальной, упрощает программирование благодаря встроенному WYSIWYG — редактору интерфейса и пр.

C++Builder первоначально создавалась только для платформы Microsoft Windows. Поздние версии, содержащие кроссплатформенную компонентную библиотеку Borland, поддерживают и Windows, и Linux. В 2003 году Borland выпустила C++BuilderX (CBX), написанный при помощи той же инфраструктуры, что и JBuilder, который при этом был мало похож на C++ Builder или Delphi. Этот продукт предназначался для разработки больших программ для крупных предприятий, но коммерческого успеха не достиг. В конце 2004 года Borland объявила, что продолжит развитие классического C++ Builder и объединит его со средой разработки Delphi, прекратив, таким образом, разработку C++ BuilderX. Спустя примерно год после этого объявления, Borland выпустила Borland Developer Studio 2006, который включал в себя Borland C++Builder 2006, предлагавший улучшенное управление конфигурацией и отладкой. Borland Developer Studio 2006 — единственный полноценный комплект, содержащий Delphi, C++ Builder и C# Builder. В 2007 году CodeGear

выпустила C++Builder 2007, в котором реализовала полную поддержку API Microsoft Windows Vista, увеличила полноту соответствия стандарту ANSI C++, увеличила скорость компиляции и сборки до 500 %, включила поддержку MSBuild, архитектур баз данных DBX4 и «VCL для Web», поддерживающий AJAX. Поддержка API Microsoft Windows Vista включила в себя приложения, изначально оформленные в стиле Vista, и естественную поддержку VCL для Aero и Vista Desktop. CodeGear RAD Studio 2007 содержит C++Builder 2007 и Delphi. Также в 2007 году CodeGear «воскресила» марку «Turbo» и выпустила две «Turbo» версии C++Builder: Turbo C++ Professional и Turbo C++ Explorer (бесплатный), основанных на Borland C++ Builder 2006. В конце 2008 года компания CodeGear выпустила новую версию RAD Studio, в которую вошли Delphi 2009 и C++Builder 2009. В 2009 году в составе RAD Studio вышел C++Builder 2010, а в 2010 году в состав RAD Studio XE включена версия C++ Builder XE. В 2012 году Embarcadero выпустила C++ Builder XE3, совместимый с Windows 8. В 2013 году был выпущен C++ Builder XE4.

Лабораторная работа

Интерфейсы со свободной навигацией в среде C++ Builder

Краткие теоретические положения

Пользовательский интерфейс представляет собой совокупность программных и аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером. Основу такого взаимодействия составляют диалоги. Под *диалогом* в данном случае понимают *регламентированный* обмен информацией между человеком и компьютером, осуществляемый в реальном масштабе времени и направленный на совместное решение конкретной задачи: обмен информацией и координация действий. Каждый диалог состоит из отдельных процессов ввода-вывода, которые физически обеспечивают связь пользователя и компьютера.

Типы интерфейсов. По аналогии с процедурным и объектным подходом к программированию различают процедурно-ориентированный и объектно-ориентированный подходы к разработке интерфейсов (рис. 1)

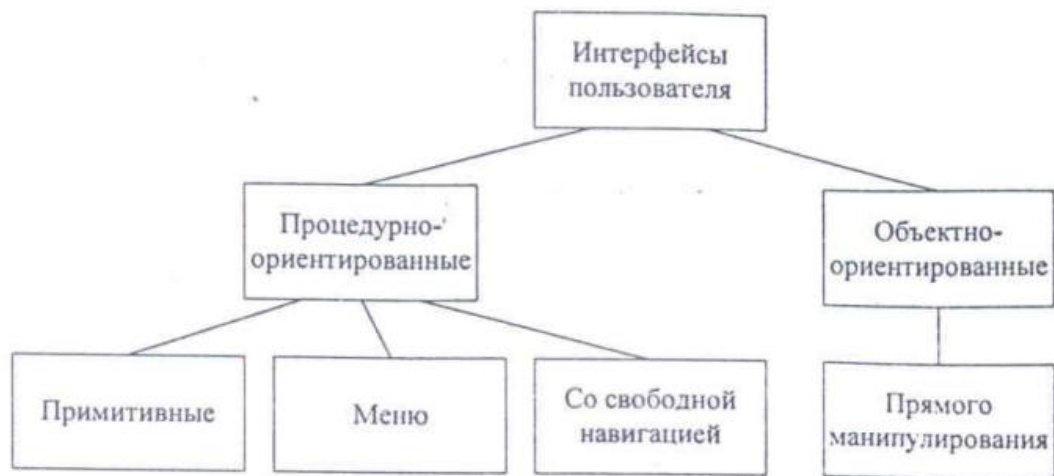


Рисунок. 1. Типы интерфейсов

Процедурно-ориентированные интерфейсы используют традиционную модель взаимодействия с пользователем, основанную на понятиях «процедура» и «операция». В рамках этой модели программное обеспечение предоставляет пользователю возможность выполнения некоторых *действий*, для которых пользователь определяет соответствующие данные и следствием выполнения которых является получение желаемых результатов.

Объектно-ориентированные интерфейсы используют несколько иную модель взаимодействия с пользователем, ориентированную на манипулирование *объектами* предметной области. В рамках этой модели пользователю предоставляется возможность напрямую взаимодействовать с каждым объектом и инициировать выполнение операций, в процессе которых взаимодействуют несколько объектов. Задача пользователя формулируется как целенаправленное изменение некоторого объекта, имеющего внутреннюю структуру, определенное содержание и внешнее символьное или графическое представление. Объект при этом понимается в широком смысле слова, например, модель реальной системы или процесса, база данных, текст и т. п. Пользователю предоставляется возможность создавать объекты, изменять их параметры и связи с другими объектами, а также инициировать взаимодействие этих объектов. Элементы интерфейсов данного типа включены в пользовательский интерфейс Windows, например, пользователь может «взять» файл и «переместить» его в другую папку. Таким образом, он инициирует выполнение операции перемещения файла.

Различают п р о ц е д у р н о – о р и е н т и р о в а н н ы е интерфейсы трех типов: «примитивные», меню и со свободной навигацией.

Примитивным называют интерфейс, который организует взаимодействие с пользователем в консольном режиме. Обычно такой интерфейс реализует конкретный сценарий работы программного обеспечения, например: ввод данных - решение задачи - вывод результата. Подобные интерфейсы в настоящее время используют только в процессе обучения программированию

или в тех случаях, когда вся программа реализует одну функцию, например, в некоторых системных утилитах.

Интерфейс-меню в отличие от примитивного интерфейса позволяет пользователю выбирать необходимые операции из специального списка, выводимого ему программой. Эти интерфейсы предполагают реализацию множества сценариев работы, последовательность действий в которых определяется пользователем.

Различают одноуровневые и иерархические меню. Первые используют для сравнительно простого управления вычислительным процессом, когда вариантов немного (не более 5-7), и они включают операции одного типа, например, Создать, Открыть, Закрыть и т. п. Вторые – при большом количестве вариантов или их очевидных различиях, например, операции с файлами и операции с данными, хранящимися в этих файлах. Интерфейсы данного типа несложно реализовать в рамках структурного подхода к программированию.

Алгоритм программы с многоуровневым меню обычно строится по уровням, причем выбор команды на каждом уровне осуществляется так же, как для одноуровневого меню.

Интерфейс-меню предполагает, что программа находится либо в состоянии **Уровень меню**, либо в состоянии **Выполнение операции**. В состоянии Уровень меню осуществляется вывод меню соответствующего уровня и выбор нужного пункта меню, а в состоянии Выполнение операции реализуется сценарий выбранной операции.

Интерфейсы-меню в настоящее время также используют редко и только для сравнительно простого программного обеспечения или в разработках, которые должны быть выполнены по структурной технологии и без использования специальных библиотек.

Интерфейсы со свободной навигацией также называют *графическими пользовательскими интерфейсами* (GUI – Graphic User Interface) или интерфейсами WYSIWYG (What You See Is What You Get - что видишь, то и получишь, т. е., что

пользователь видит на экране, то он и получит при печати). Эти названия подчеркивают, что интерфейсы данного типа ориентированы на использование экрана в графическом режиме с высокой разрешающей способностью.

Графические интерфейсы поддерживают концепцию интерактивного взаимодействия с программным обеспечением, осуществляя визуальную обратную связь с пользователем и возможность прямого манипулирования объектами и информацией на экране. Кроме того, интерфейсы данного типа поддерживают концепцию совместимости программ, позволяя перемещать между ними информацию (технология OLE). В отличие от интерфейса-меню интерфейс со свободной навигацией обеспечивает возможность осуществления любых допустимых в конкретном состоянии операций, доступ к которым возможен через различные интерфейсные компоненты. Например, окна программ, реализующих интерфейс Windows, обычно содержат:

- меню различных типов: ниспадающее, кнопочное, контекстное;
- разного рода компоненты ввода данных.

Причем выбор следующей операции в меню осуществляется как мышью, так и с помощью клавиатуры.

Существенной особенностью интерфейсов данного типа является способность изменяться в процессе взаимодействия с пользователем, предлагая выбор только тех операций, которые имеют смысл в конкретной ситуации. Реализуют интерфейсы со свободной навигацией, используют событийное программирование и объектно-ориентированные библиотеки, что предполагает применение визуальных сред разработки программного обеспечения.

Объектно-ориентированные интерфейсы пока представлены только *интерфейсом прямого манипулирования*. Этот тип интерфейса предполагает, что взаимодействие пользователя с программным обеспечением осуществляется посредством выбора и перемещения *пиктограмм*, соответствующих объектам предметной области. Для реализации таких интерфейсов также используют

событийное программирование и объектно-ориентированные библиотеки.

В данной лабораторной работе рассмотрены три примера решения задач разработки интерфейсов со свободной навигацией:

1) задача выбора и расчета листа материала с помощью компонента `RadioButton` (переключателя);

2) задача выбора и загрузки фотографии с помощью компонентов `ListBox` и `Image`;

3) запуска обозревателя `Internet Explorer` и вывода сетевого документа с помощью компонента `CppWebBrowser`.

Задание 1

Цель работы: изучить методику применения компонента Radio Button (переключателя) при программировании на языке C++ Builder.

Назначение программы: в программе выбирается вид материала (алюминий или пластик), вводится ширина и длина листа материала, по нажатию кнопки Ok рассчитывается стоимость листа материала.

Форма имеет вид (см. рисунок 2)

Порядок выполнения работы

1) Создать интерфейс в соответствии с рисунком.

Необходимые компоненты находятся на вкладке Standart.

2) Щелкнуть по кнопке ОК и набрать текст программы обработчика нажатия кнопки

3) Запустить программу на исполнение (F9).

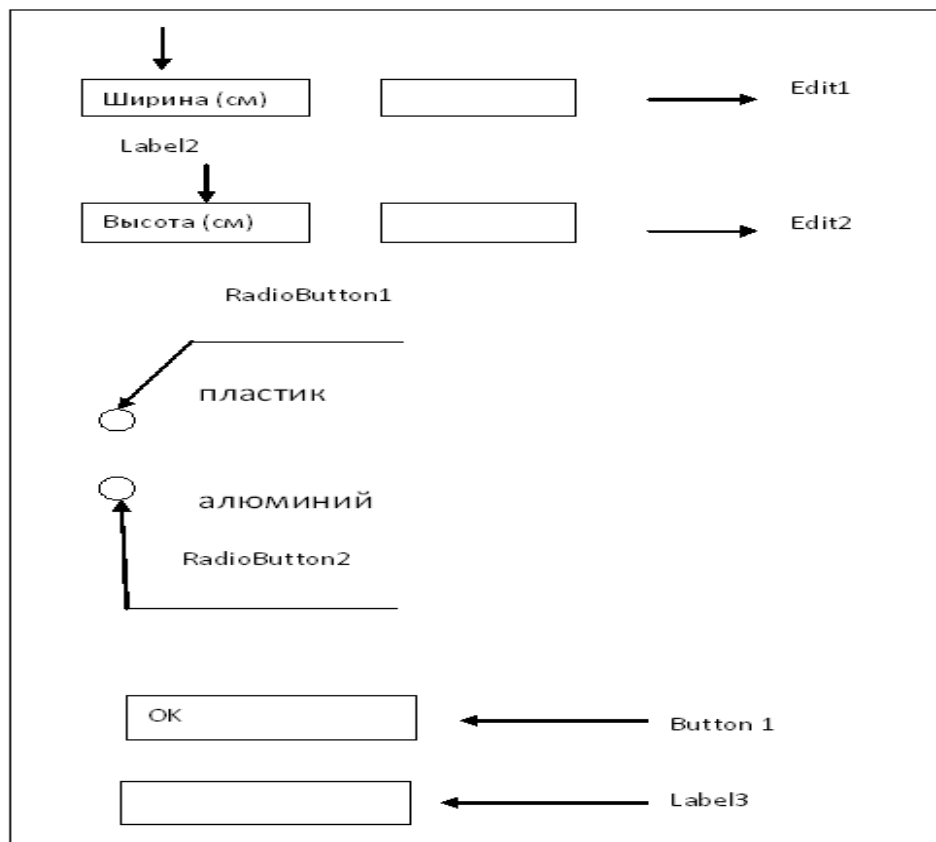


Рисунок. 2. Интерфейс программы «Выбор и расчет стоимости листа материала»

Код программы (сpp - файл)

```
#include <vcl.h>
#pragma hdrstop

#include "Unit2.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall TForm1::Button1Click(TObject *Sender)
{
float w, h, s;
float c;
float sum;
AnsiString st;
w=StrToFloat(Edit1->Text);
h=StrToFloat(Edit2->Text);
s=w*h/10000;
if (RadioButton1->Checked)
c=360;
else
c=180;
sum=s*c;
st="Размер: " + Edit1->Text + "x" + Edit2->Text +
"см\nМатериал";
if (RadioButton1->Checked)
st=st + " алюминий";
else
st=st + " пластик";
st=st+ "\nСумма:" + FloatToStrF(sum, ffCurrency, 6,2);
```

Label3->Caption=st;

}

//-----

Задание 2

Цель работы: Изучить применение компонента `ListBox` (списка, предназначенного для выбора нужного элемента) при программировании на языке `C++Builder`.

Назначение программы: программа выбирает фотографию (файл с расширением `JPG`), загруженного на диск компьютера и выводит ее в окно (компонент `Image`). Имена файлов просматриваемой папки отображаются списком в компоненте `ListBox`.

Форма имеет вид:

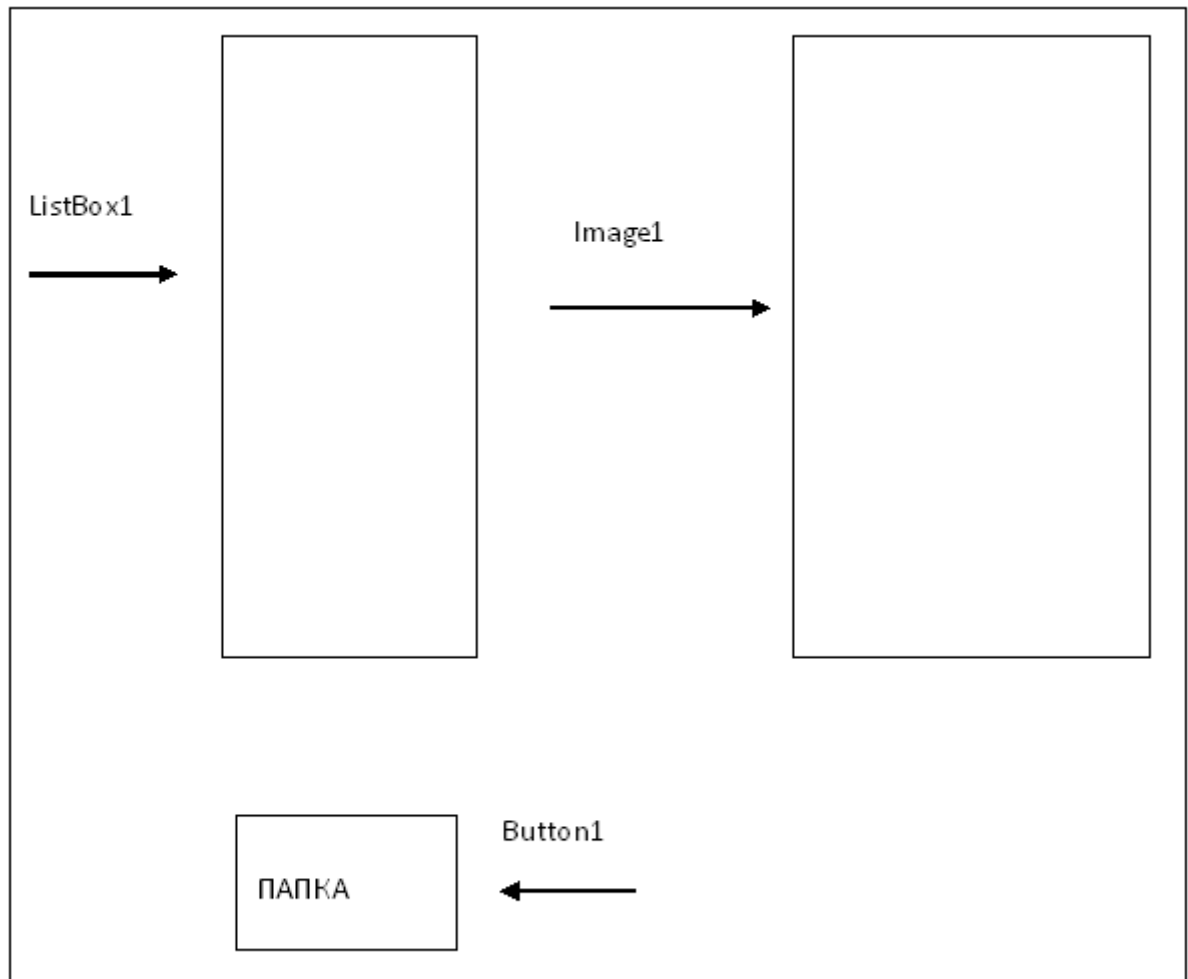


Рисунок 3. Интерфейс программы «Выбор и загрузка фотографии из списка»

Код программы (h-файл)

```
//-----  
  
#ifndef Unit4H  
#define Unit4H  
//-----  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
#include <ExtCtrls.hpp>  
//-----  
class TForm1 : public TForm  
{  
    __published:    // IDE-managed Components  
        TListBox *ListBox1;  
        TButton *Button1;  
        TImage *Image1;  
        void __fastcall ListBox1Click(TObject *Sender);  
        void __fastcall Button1Click(TObject *Sender);  
private:    // User declarations  
    __fastcall int FillListBox();  
    AnsiString Path;  
public:    // User declarations  
    __fastcall TForm1(TComponent* Owner);  
};  
//-----  
extern PACKAGE TForm1 *Form1;  
//-----  
#endif
```

Код программы (сpp – файл)

```
//-----  
  
#include <vcl.h>
```

```

#pragma hdrstop
#include "Unit4.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
#include <jpeg.hpp>
#include <filectrl.hpp>
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    this->FillListBox( );
}
//-----
__fastcall TForm1::FillListBox( )
{
    TSearchRec SearchRec; // результат поиска файла
    int r;

    r = FindFirst(Path + "*.jpg",faAnyFile,SearchRec);
    if (r == 0)
    {
        // в каталоге Path есть по крайней мере один JPG-файл
        ListBox1->Items->Clear();
        ListBox1->Items->Add(SearchRec.Name);
        while ( FindNext(SearchRec) == 0 )
        {
            ListBox1->Items->Add(SearchRec.Name);
        };

        ListBox1->ItemIndex = 0;

        // отобразить иллюстрацию
        Image1->Picture->LoadFromFile(Path +
            ListBox1->Items->Strings[ListBox1->ItemIndex]);
    }
}

```

```

        return ListBox1->Items->Count;
    }
    else
        return -1;
}
void __fastcall TForm1::ListBox1Click(TObject *Sender)
{
    AnsiString FileName;

    FileName = Path + ListBox1->Items->Strings[ListBox1-
>ItemIndex];
    Image1->Picture->LoadFromFile(FileName);
}
//-----

void __fastcall TForm1::Button1Click(TObject *Sender)
{
    if (SelectDirectory("Выберите каталог", "", Path ))
    {
        Path = Path + "\\";
        Form1->Caption="Просмотр иллюстраций - " + Path;
        FillListBox();
    }
}

//-----

```

Задание 3

Цель работы: изучить применение компонента CppWebBrowser, предназначенного для запуска обозревателя Internet Explorer

Назначение программы: программа запускает Internet Explorer для вывода сетевого документа.

Форма имеет вид:

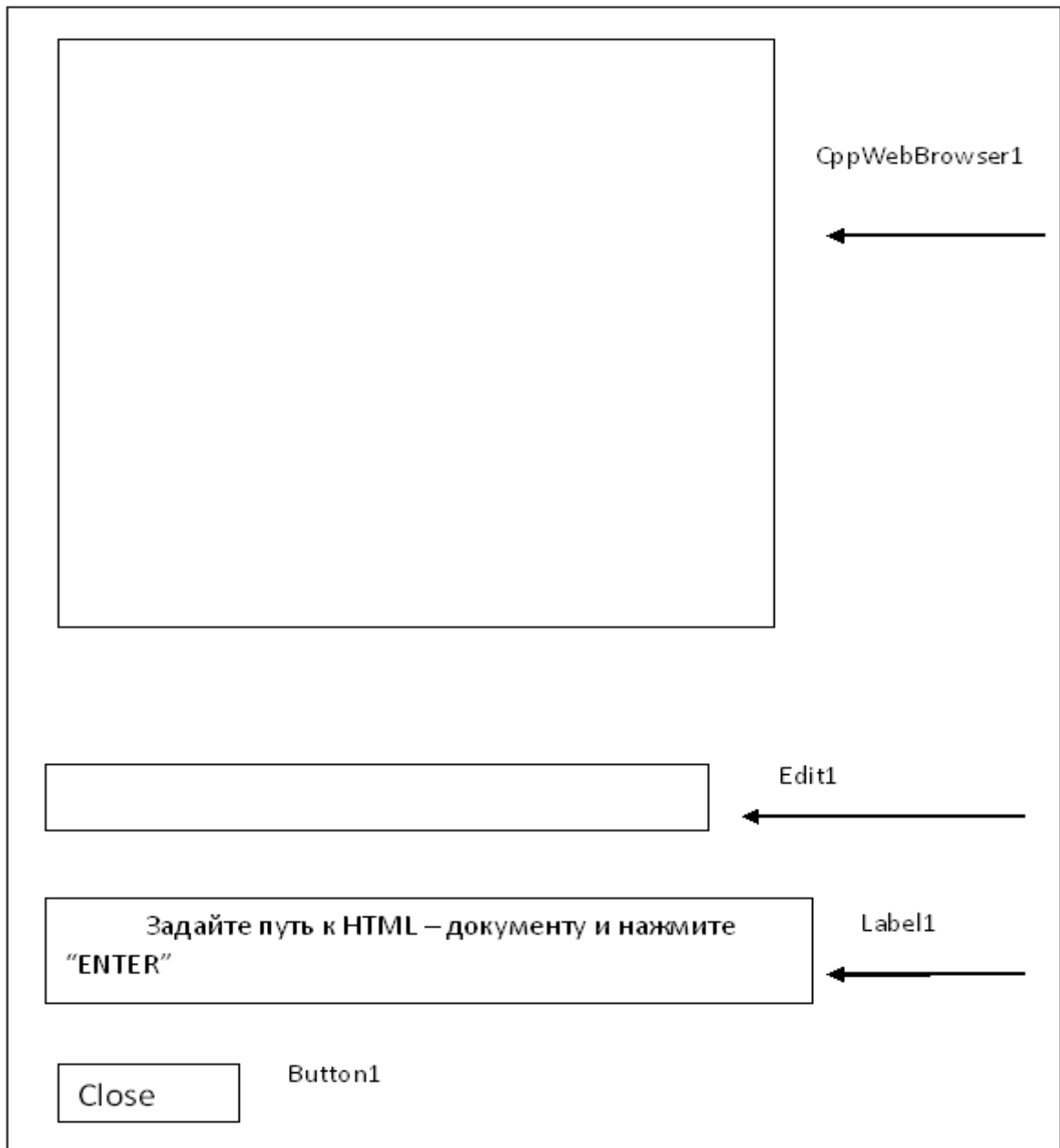


Рисунок 4. Интерфейс программы «Вывод сетевого документа»

Код программы (h - файл)

```
//-----  
  
#ifndef Unit11H  
#define Unit11H  
//-----  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
#include "SHDocVw_OCX.h"  
#include <OleCtrls.hpp>  
//-----  
class TForm1 : public TForm  
{  
  __published:    // IDE-managed Components  
    TCppWebBrowser *CppWebBrowser1;  
    TEdit *Edit1;  
    TButton *Button1;  
    TLabel *Label1;  
    void __fastcall Edit1KeyDown(TObject *Sender, WORD  
&Key,  
        TShiftState Shift);  
    void __fastcall Button1Click(TObject *Sender);  
private:    // User declarations  
public:    // User declarations  
    __fastcall TForm1(TComponent* Owner);  
};  
//-----  
extern PACKAGE TForm1 *Form1;  
//-----  
#endif
```

Код программы (cpp - файл)

```
//-----
```



```

#include <vcl.h>
#pragma hdrstop

#include "Unit11.h"
//-----
#pragma package(smart_init)
#pragma link "SHDocVw_OCX"
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::Edit1KeyDown(TObject *Sender, WORD
&Key,
    TShiftState Shift)
{
if(Key==VK_RETURN)
{
WideString URL=Edit1->Text;
Variant flag=1;
CppWebBrowser1->Navigate(URL,flag,Null,Null,Null);

}
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
Form1->Close();
}
//-----

```

Список контрольных вопросов

- 1) Какие типы интерфейсов существуют? Дайте их краткую характеристику.
- 2) К какому типу пользовательского интерфейса можно отнести рабочий стол Windows?
- 3) Какие модели интерфейсов существуют? Каким образом данные модели должны быть согласованы между собой?
- 4) Какие типы диалогов используются при разработке интерфейсов? В чем их основное различие?
- 5) Какие формы диалогов используются при разработке интерфейсов? В чем их основное различие?
- 6) Каким образом должны быть согласованы между собой типы диалогов и формы диалогов?
- 7) Перечислите основные этапы разработки пользовательского интерфейса.
- 8) Перечислите файлы проекта ИСР C++Builder и объясните их назначение.
- 9) Какие компоненты ИСР C++Builder можно использовать для ввода и вывода текстовой информации?
- 10) В чем отличие между компонентами CheckBox и RadioButton с точки зрения выбора возможных вариантов?
- 11) Какие компоненты ИСР C++Builder можно использовать для выбора элементов из списка?
- 12) Объясните принцип метода Navigate компонента CppWebBrowser. Каким образом следует задать параметры метода Navigate?

Список литературы

1. Иванова Г. С. Технология программирования [Текст]: учебник / Г. С. Иванова.- М.: КНОРУС, 2011. – 336 с.
2. Павловская Т. А. С/С++. Программирование на языке высокого уровня [Текст] : учебник / Т. А. Павловская. - СПб. : Питер, 2006. - 461 с.
3. Зотов И. В. Разработка программ на С++. Теория, примеры решения задач, эффективность [Текст] : учебное пособие / Т. А. Ширабакина. - Курск : Курск. гос. техн. ун-т, 2003. - 172 с.
4. Культин Н. Б. С++Builder. – 2-е изд. перераб. и доп. – СПб.: БХВ - Петербург, 2008. - 464 с.
5. Пахомов Б. И. С/С++ и Borland С++Builder для начинающих. –Спб.: БХВ –Петербург, 2007. – 640 с.
6. Торрес Р. Дж. Практическое руководство по проектированию и разработке пользовательского интерфейса / Торрес Р. Дж. – М.: Вильямс, 2002. – 400с.
7. Хорев П. Б. Технологии объектно-ориентированного программирования [Текст]: учебное пособие / П. Б. Хорев.- М.: Академия, 2004.-448 с.