

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

---

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(национальный исследовательский университет)  
Кафедра 410

---

**А.В. Бруханский**

**Разработка приложения «Калькулятор-редактор»**

**в среде C++Builder**

*Учебное пособие к лабораторной работе*

Москва  
МАИ  
2019

*Цели выполнения лабораторной работы:*

- знакомство с интегрированной средой быстрой разработки приложений Embarcadero RAD Studio C++Builder;
- знакомство с библиотеками визуальных компонентов и функций;
- освоение приемов отладки приложений.

Выбираем тип создаваемого проекта: « VCL Forms Application – C++Builder».

Создаем форму Form1 (рисунок 1). Все визуальные компоненты кроме диалогов OpenFileDialog и SaveDialog выбираются из палитры инструментов (Tool Palette) **Standard**, компоненты диалогов – из палитры **Dialogs**.

Окна ввода чисел-операндов и результата вычислений калькулятора, а также пути к файлу – компоненты Edit, панель с радиокнопками – RadioGroup, окно ввода многострочного текста – компонент Мемо.

### Расположение компонентов в окне VCL-Form приложения «Калькулятор-редактор»

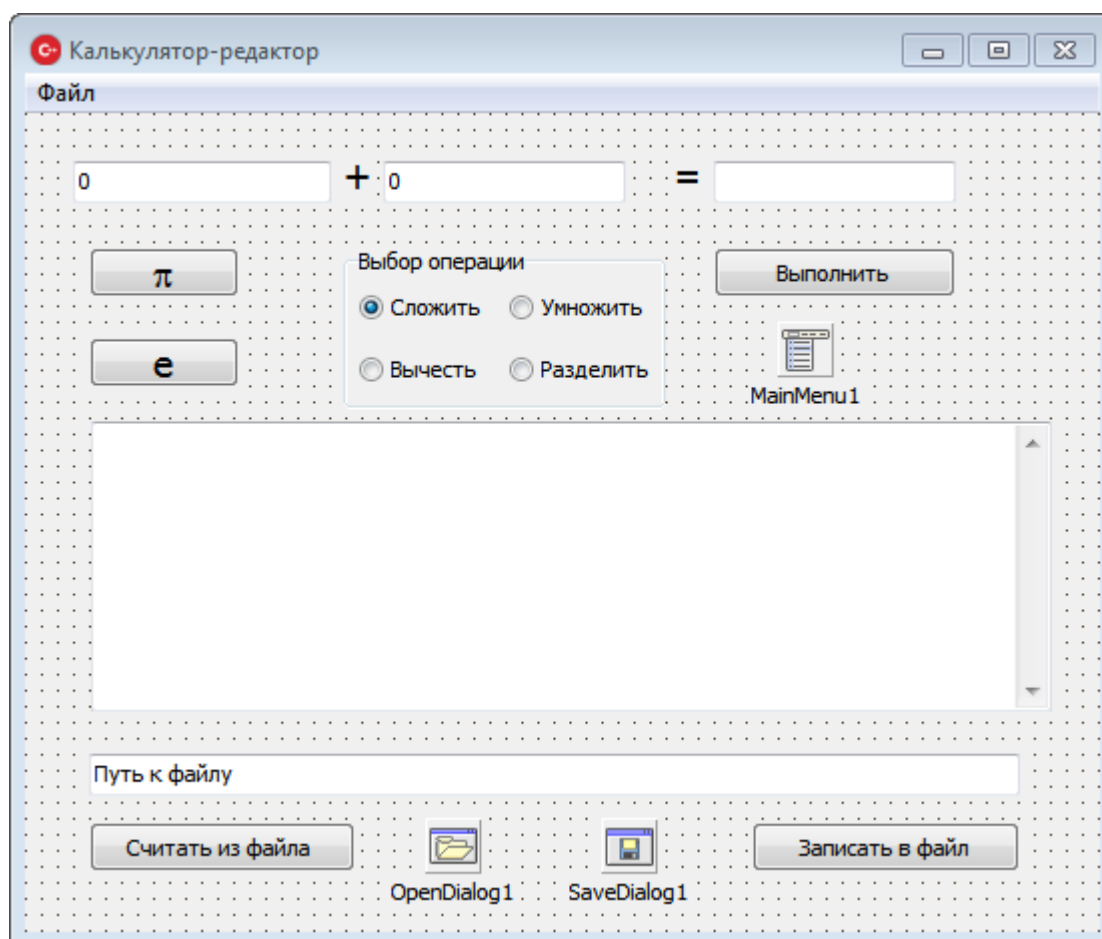


Рисунок 1. Расположение компонентов в окне VCL-Form приложения  
«Калькулятор-редактор»

Для задания количества кнопок на панели RadioGroup «Выбор операции» раскрываем свойство Items компонента RadioGroup1 и добавляем 4 строки:

Сложить  
Вычесть  
Умножить  
Разделить

В свойстве Columns этого компонента записываем цифру 2, а свойство ItemIndex устанавливаем в 0, что означает выбор операции «Сложить» по умолчанию.

Надпись на кнопке ввода константы  $\pi$  выполняется шрифтом Symbol (панель свойств «Object Inspector» → Properties → Font → Name). Размер шрифта (Size) – 14.

Редактор пунктов главного меню, рисунок 2, открывается двойным щелчком по компоненту MainMenu, предварительно помещенному в форму.

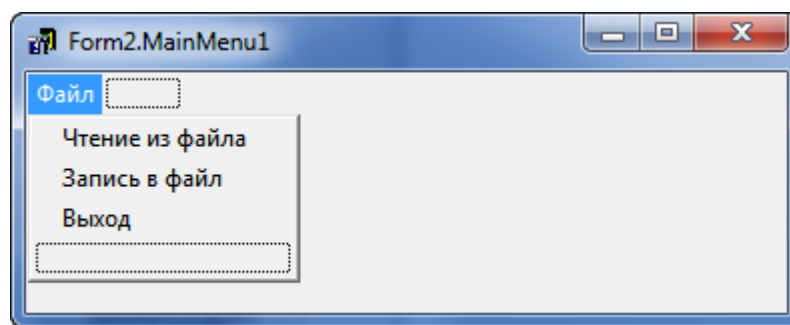


Рисунок 2. Окно редактора главного меню

Далее приводится исходный код модуля Unit1.cpp. Переключение между окнами с формой и текстом программы (функциями обратного вызова) осуществляется функциональной клавишей «F12».

Каждую очередную функцию обратного вызова вводим в заготовку, создаваемую средой автоматически при двойном щелчке на соответствующий визуальный компонент. Нельзя набирать текст программы подряд в надежде связать его с нажатием кнопок впоследствии. После ввода текста каждой функции проверяйте с каким событием (Events) соответствующего визуального компонента инспектор объектов («Object Inspector») связал выполнение этой функции.

Объявление глобальных переменных и операторы #include, предшествующие тексту функций вводятся непосредственно в начало текста программы без связи с какими-либо визуальными компонентами.

Запуск программы на трансляцию и выполнение осуществляется из пункта меню «Run» окна RAD Studio или нажатием на зеленый треугольник в строке инструментов.

После двойного щелчка по изображению кнопки или другого компонента формы выбирайте текст нужной функции из приведенного далее текста и вписывайте его в заготовку, созданную средой программирования C++Builder.

Дублировать запись функций ввода текста из файла и записи текста в файл с помощью кнопок главного меню «Чтение из файла», «Запись в файл» не

нужно. Достаточно связать выбор этих кнопок с выполнением функций, связанных с кнопками формы «Считать из файла», «Записать в файл», как это сделано в приведенном ниже тексте программы.

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
#include <graphics.hpp>  
#include "Unit1.h"  
#include <math.h>  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
float a, b, c;  
int indinput=1;  
AnsiString fname;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
void __fastcall TForm1::Button1Click(TObject *Sender)  
{  
    a = StrToFloat(Edit1->Text);  
    b = StrToFloat(Edit2->Text);  
    switch (RadioGroup1->ItemIndex) {  
        case 1:  
            c = a - b;  
            Label1->Caption="-";  
            break;  
        case 2:  
            c = a * b;  
            Label1->Caption="*";  
            break;  
        case 3:  
            c = a / b;  
            Label1->Caption=":";  
            break;  
        default:  
            c = a + b;  
            Label1->Caption="+";  
            break;  
    }  
    Edit3->Text = FloatToStrF(c, ffGeneral, 7, 2);  
}
```

```

· //-----
- void __fastcall TForm1::Edit1Focus(TObject *Sender)
· {
·     indinput=1;
· }
· //-----
50 void __fastcall TForm1::Edit2Focus(TObject *Sender)
· {
·     indinput=2;
· }
· //-----
- void __fastcall TForm1::Button2Click(TObject *Sender)
· {
·     if(indinput==1)
·         Edit1->Text = M_PI;
·     else if (indinput==2)
60     Edit2->Text = M_PI;
· }
· //-----
· void __fastcall TForm1::Button3Click(TObject *Sender)
· {
- if(indinput==1)
· Edit1->Text=exp(1.0);
· else if (indinput==2) {
·     Edit2->Text=exp(1.0);
·     }
70 }
· //-----
· void __fastcall TForm1::Button4Click(TObject *Sender)
· {
·     if(OpenDialog1->Execute()) {
-     fname=OpenDialog1->FileName;
·     Edit4->Text=fname;
·     Mem1->Lines->LoadFromFile(fname);
·     }
· }
80 //-----
· void __fastcall TForm1::Button5Click(TObject *Sender)
· {
·     SaveDialog1->FileName=Edit4->Text;
·     if(SaveDialog1->Execute()) {
-     fname=SaveDialog1->FileName;
·     Edit4->Text=fname;
·     Mem1->Lines->SaveToFile(fname);
·     }
· }

```

```

90 //-----
· void __fastcall TForm1::N2Click(TObject *Sender)
· {
·   Form1->Close();
· }
- //-----
· void __fastcall TForm1::N3Click(TObject *Sender)
· {
·   TForm1::Button4Click(Sender);
· }
100 //-----
· void __fastcall TForm1::N4Click(TObject *Sender)
· {
·   TForm1::Button5Click(Sender);
· }
- //-----

```

После успешного запуска программы протестируйте работу калькулятора и редактора многострочных текстов (рисунок 3)

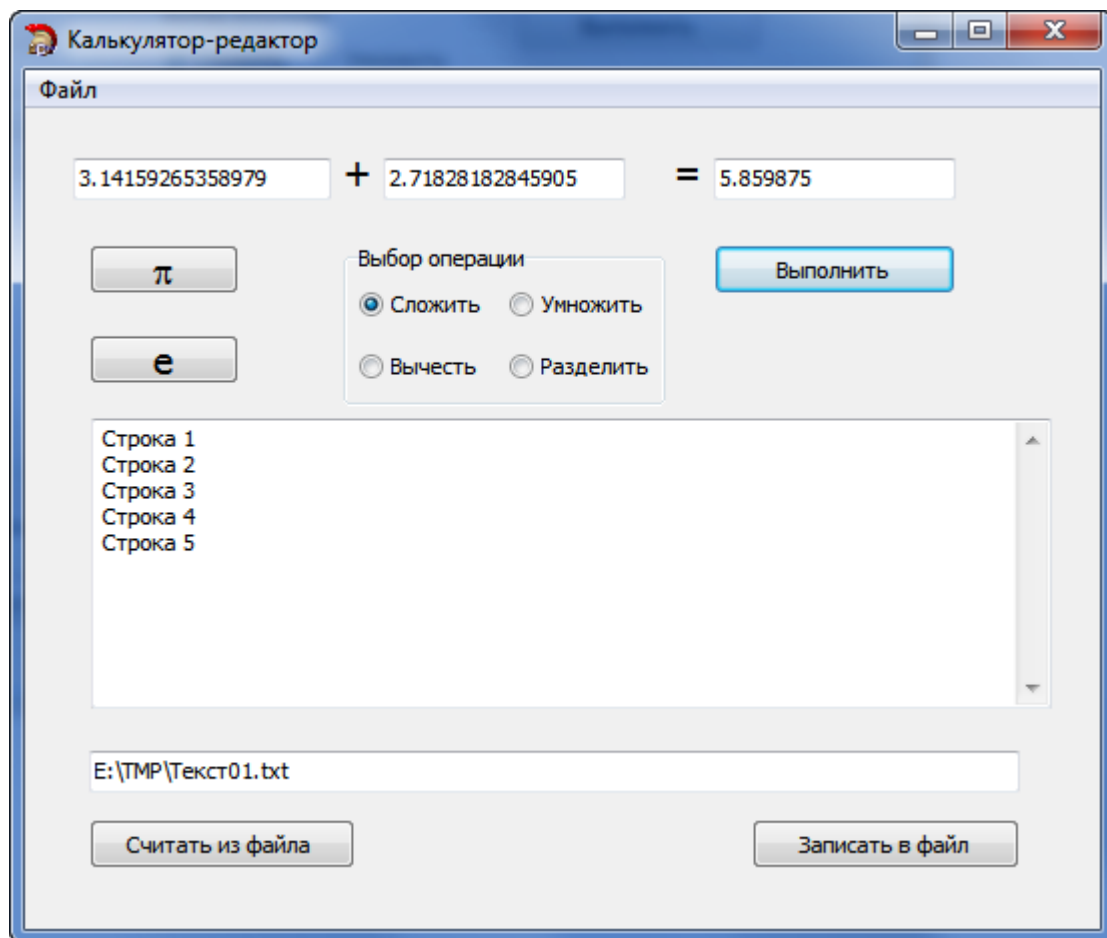


Рисунок 3. Окно выполняющегося приложения «Калькулятор-редактор»

При помещении курсора в одно из окон ввода операндов и последующего щелчка на кнопке ввода константы  $\pi$  или  $e$  числовое значение константы должно появиться в выбранном окне Edit.

Текстовый файл, из которого будет считываться текст в окно компонента Метод, создайте предварительно в папке проекта. Лучше всего поместить его в папку, в которой находится исполняемый файл программы, то есть файл с расширением «.exe». Это папки Debug или Release, расположенные внутри папки проекта.