

Лекция 7

ОСНОВЫ ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ

Лекция 7. Основы визуального программирования

План

1. Понятие визуального программирования.
2. Пример визуального программирования в IDE Lazarus.
3. Современные IDE для визуального программирования.

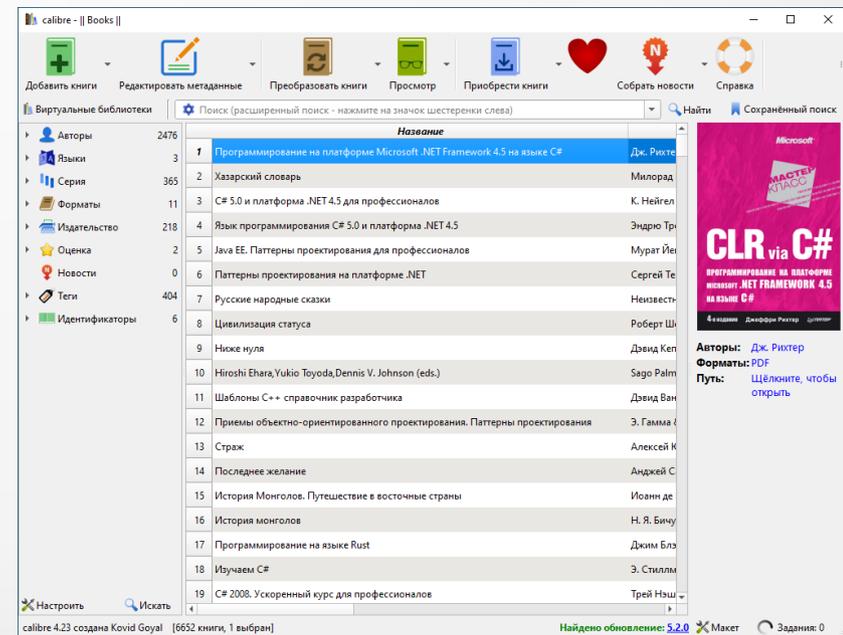
1. Понятие визуального программирования

Современные операционные системы (Windows, Linux, MacOS, Android, iOS и др.) для взаимодействия с пользователем используют графический интерфейс.

Графический интерфейс пользователя (GUI – Graphical User Interface) – система средств, предоставляющих пользователю возможность взаимодействовать с электронным устройством с помощью различных графических компонентов экрана, таких, например, как меню, кнопки, списки и т. п.



Фрагмент GUI ОС Windows



Пример программы с GUI

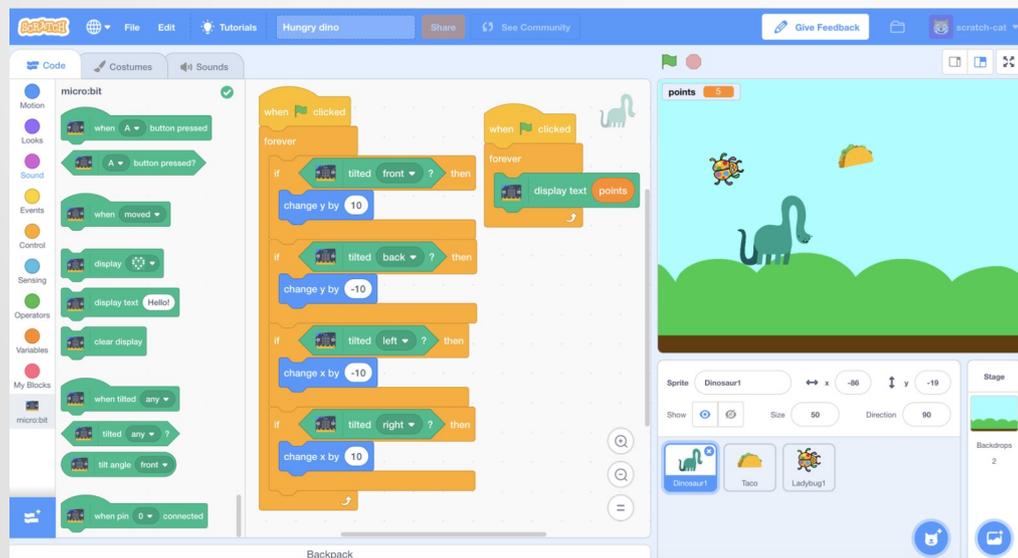
1. Понятие визуального программирования

Визуальное программирование – это технология разработки программного обеспечения, предусматривающая создание приложений с помощью наглядных (визуальных) средств.

Следует различать:

- 1) графические языки программирования;
- 2) визуальные средства разработки программ.

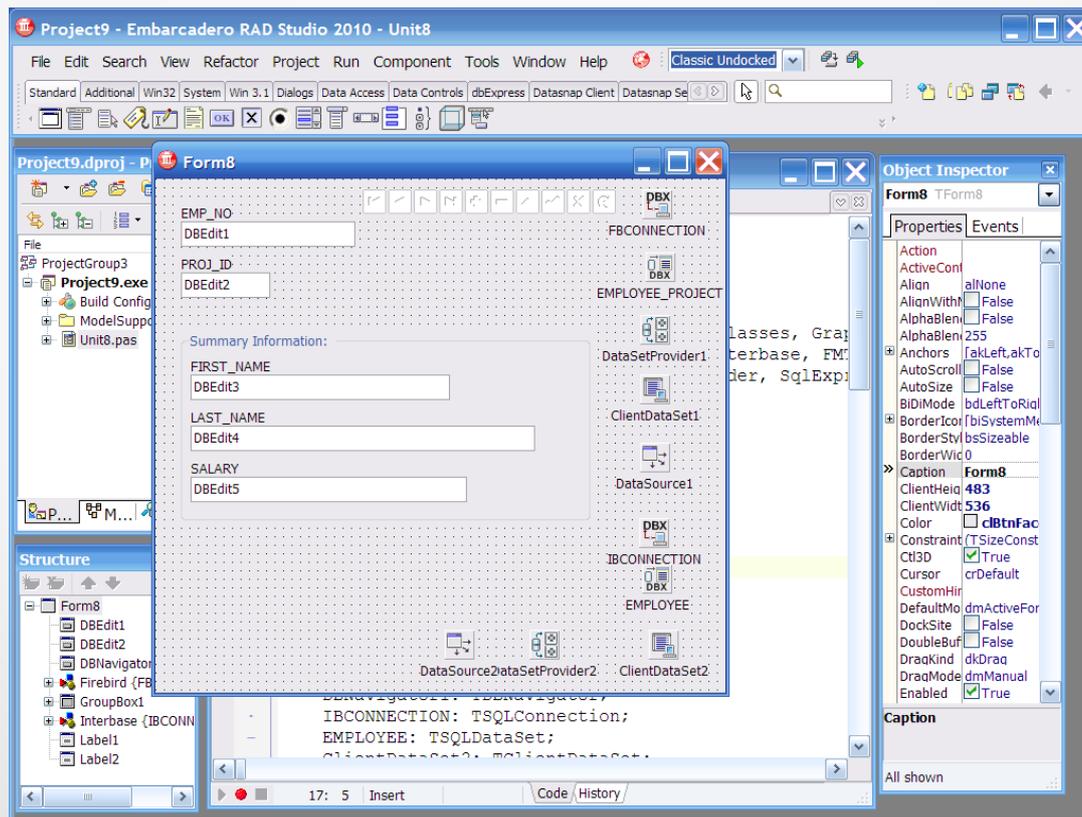
Графические языки программирования позволяют создавать программы путём манипулирования графическими объектами вместо написания исходного текста.



Одним из наиболее известных таких языков является **Scratch**, специально разработанный для обучения детей основам программирования.

1. Понятие визуального программирования

Визуальные средства разработки программ – это специальный инструментарий для проектирования GUI программ. Чаще всего в этом качестве выступают специализированные **редакторы форм**, позволяющие без написания программного кода создавать и настраивать различные элементы пользовательского графического интерфейса (C++ Builder, Delphi, Code::Blocks, Qt Creator и др.).



2. Пример визуального программирования в IDE Lazarus

Lazarus – свободная кроссплатформенная интегрированная среда разработки программного обеспечения с открытым исходным кодом, построенная на компиляторе **Free Pascal**.



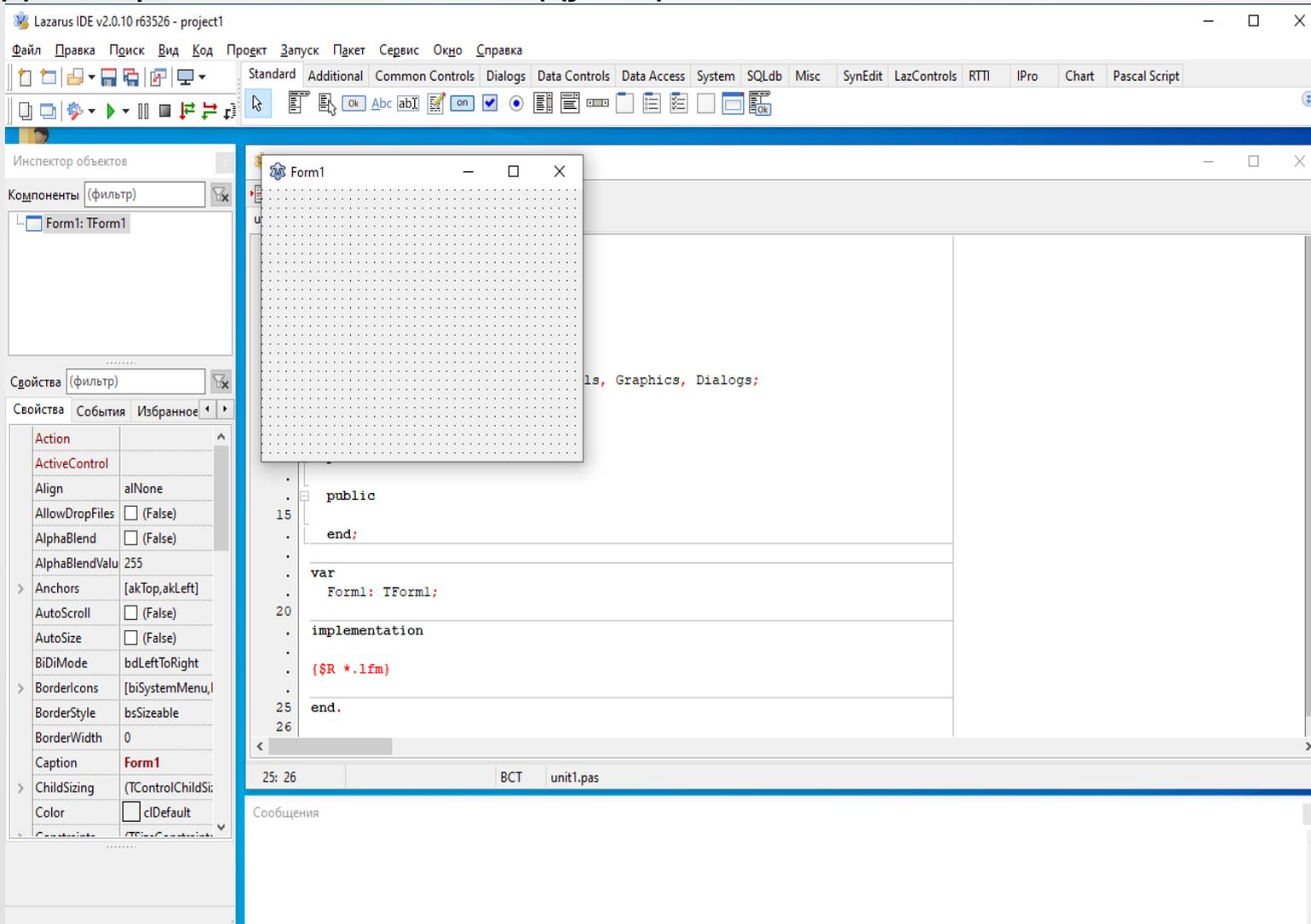
Lazarus во многом дублирует возможности проприетарной IDE **Borland Delphi** и базируется на использовании библиотеки визуальных компонентов **VCL** (Visual Component Library).



Скачать Lazarus можно на официальном сайте проекта: <https://www.lazarus-ide.org/>

2. Пример визуального программирования в IDE Lazarus

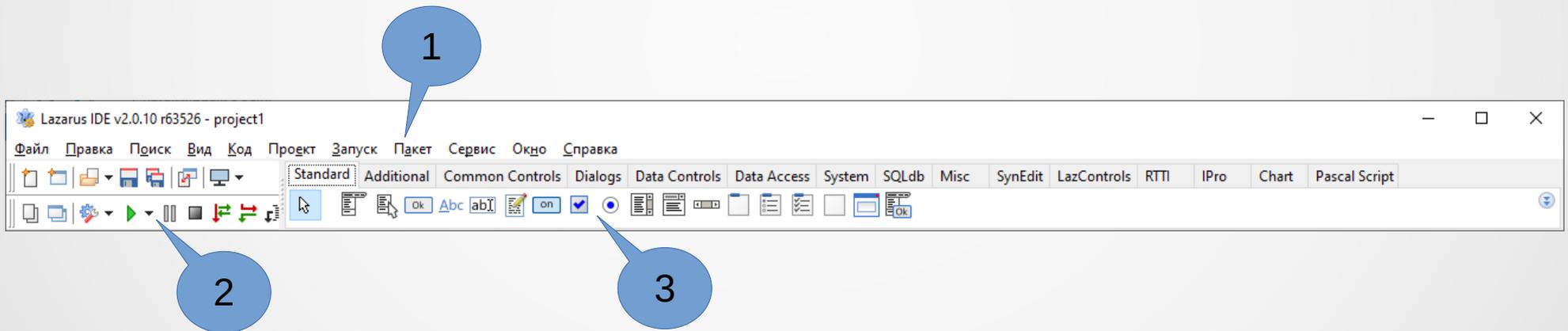
После загрузки, установки и запуска IDE Lazarus пользователь должен увидеть приблизительно следующее:



2. Пример визуального программирования в IDE Lazarus

Главное окно IDE Lazarus содержит такие элементы, как:

- 1) **главное меню программы;**
- 2) **панель инструментов,** где в виде соответствующих пиктограмм размещены наиболее часто используемые команды главного меню;
- 3) **палитру компонентов,** где размещены доступные пользователю визуальные компоненты.



2. Пример визуального программирования в IDE Lazarus

Окно «Инспектор объектов» содержит список доступных пользователю свойств и методов активного графического компонента.

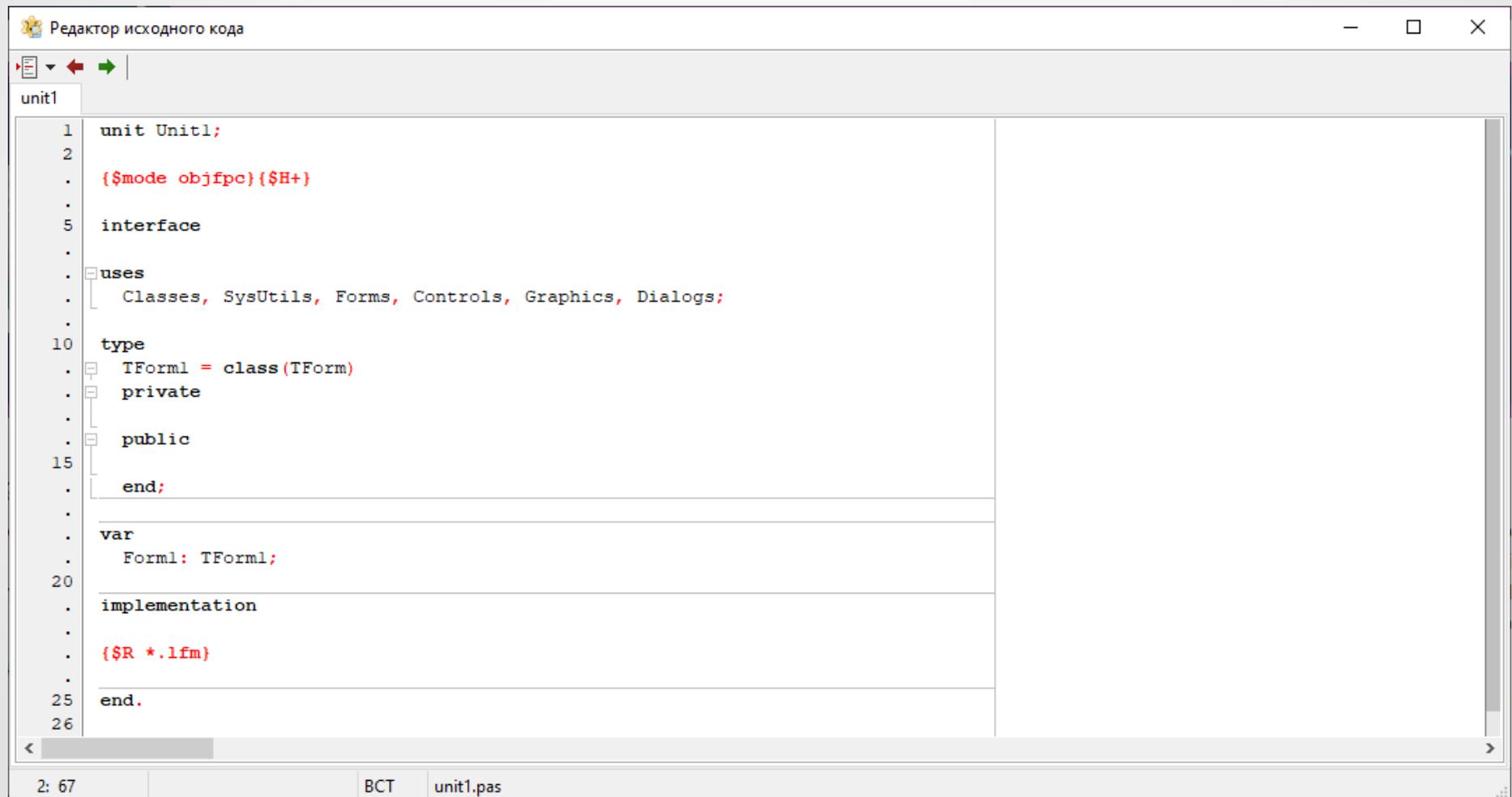
The screenshot shows the 'Инспектор объектов' (Object Inspector) window in the Lazarus IDE. It is divided into several sections:

- Компоненты (фильтр):** A list of components, with 'Form1: TForm1' selected. A callout points to this list, stating 'Текущий активный компонент' (Current active component).
- Свойства (фильтр):** A list of properties for the selected component. A callout points to the 'Caption' property, stating 'Название свойства' (Property name).
- Свойства | События | Избранное:** A tabbed interface. The 'Свойства' (Properties) tab is active. A callout points to the value of the 'Caption' property, 'Form1', stating 'Значение свойства' (Property value).
- Переключатель «Свойства/События»:** A callout points to the 'Свойства' tab, stating 'Переключатель «Свойства/События»' (Toggle 'Properties/Events').

Property Name	Value
Action	
ActiveControl	
Align	alNone
AllowDropFiles	<input type="checkbox"/> (False)
AlphaBlend	<input type="checkbox"/> (False)
AlphaBlendValue	255
> Anchors	[akTop,akLeft]
AutoScroll	<input type="checkbox"/> (False)
AutoSize	<input type="checkbox"/> (False)
BiDiMode	bdLeftToRight
> BorderIcons	[biSystemMenu,biCaption,biResize]
BorderStyle	bsSizeable
BorderWidth	0
Caption	Form1
> ChildSizing	(TControlChildSizing)csNone
Color	<input type="checkbox"/> clDefault
> Constraints	(TSizeConstraints)csNone
Cursor	crDefault

2. Пример визуального программирования в IDE Lazarus

Окно «Редактор исходного кода» содержит текстовый редактор с исходным кодом программы на языке Free Pascal.



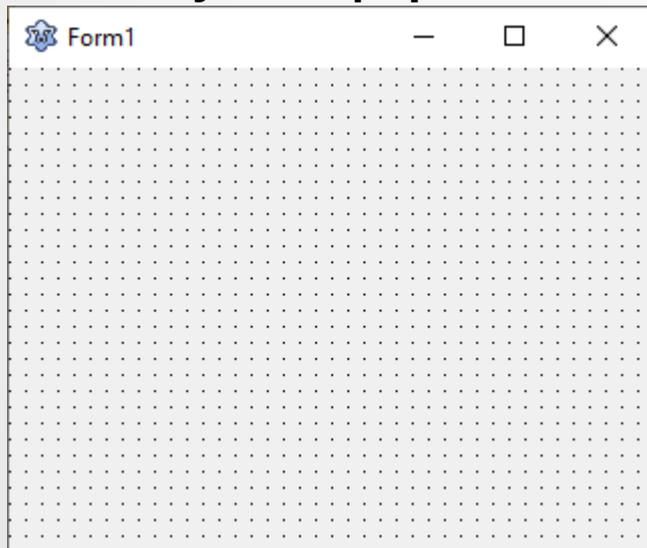
```
1  unit Unit1;
2
3  {$mode objfpc}{$H+}
4
5  interface
6
7  uses
8  [
9      Classes, SysUtils, Forms, Controls, Graphics, Dialogs;
10 ]
11
12 type
13 [
14     TForm1 = class(TForm)
15     private
16     public
17     end;
18 ]
19
20 var
21     Form1: TForm1;
22
23 implementation
24
25     {$R *.lfm}
26
27 end.
```

2: 67 BCT unit1.pas

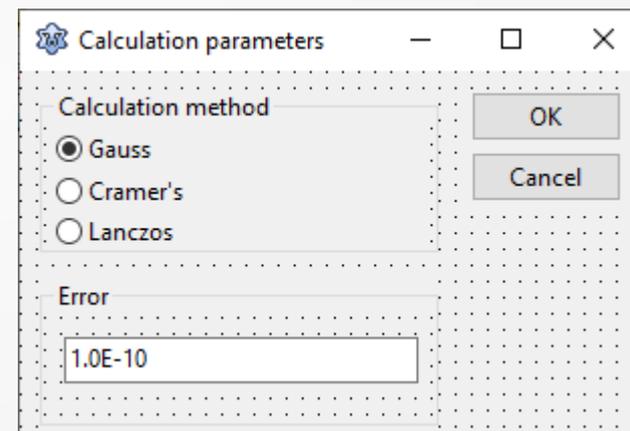
2. Пример визуального программирования в IDE Lazarus

Окно **формы** является прототипом будущего окна программы. Непосредственно на форме пользователь размещает другие компоненты графического интерфейса программы, такие как меню, кнопки, поля ввода и т. п.

Первоначальная
пустая форма

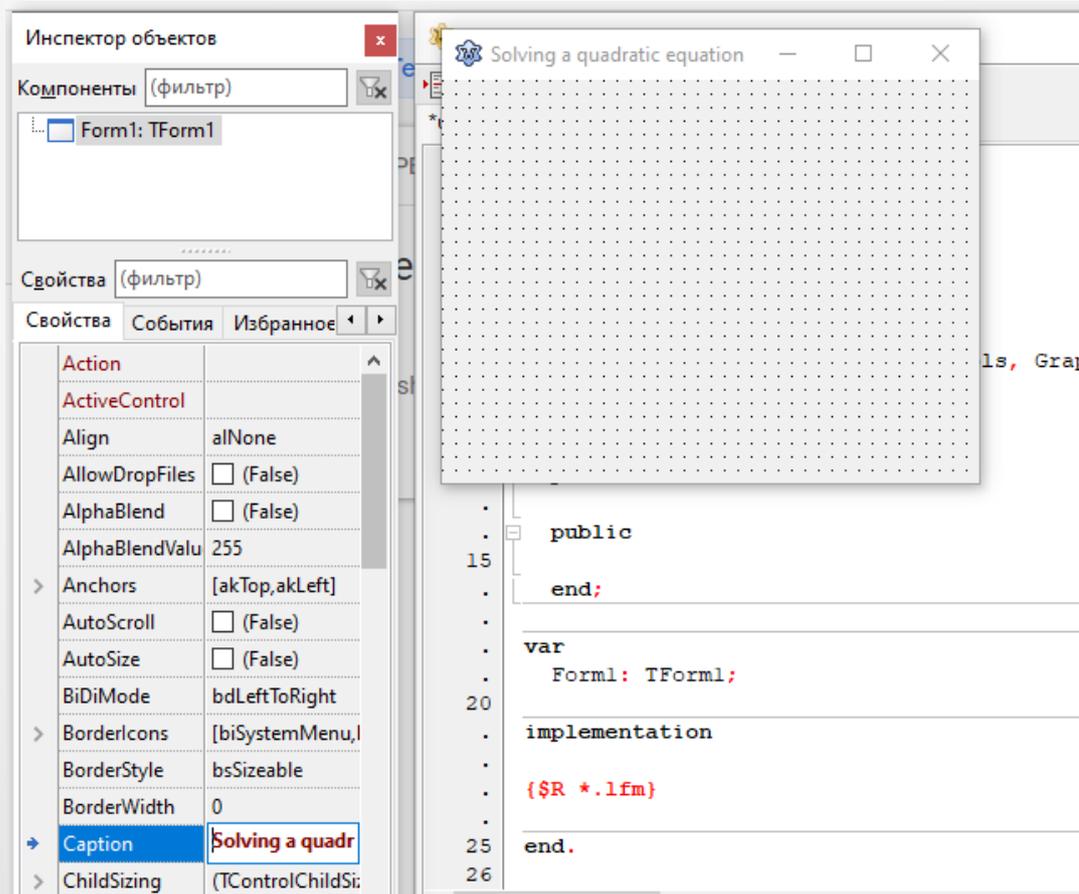


Форма с визуальными
компонентами



2. Пример визуального программирования в IDE Lazarus

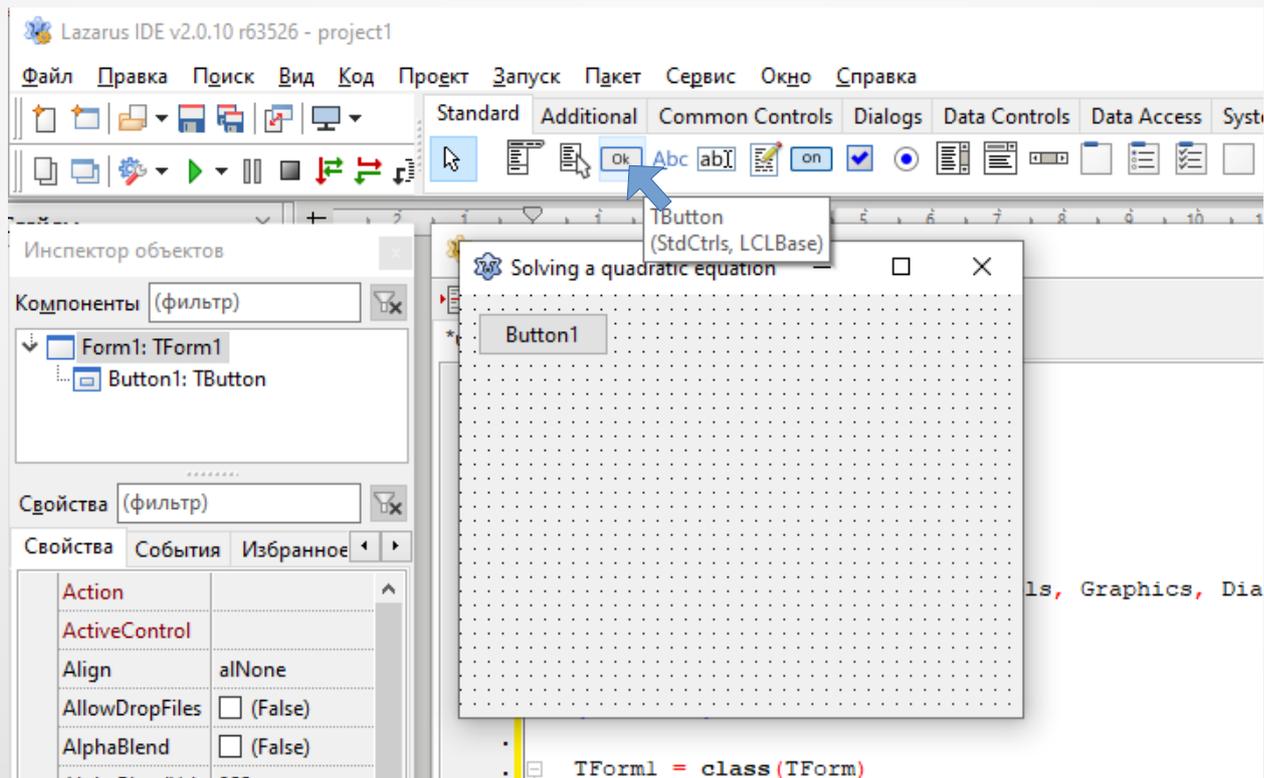
Доступные пользователю свойства и события формы отображаются в инспекторе объектов, если форма активна, т. е. пользователь кликнул по ней мышкой. Для изменения значения свойства пользователь должен ввести в соответствующее поле новое значения. Например, для изменения заголовка формы нужно изменить свойство «**Caption**» в окне инспектора объектов.



2. Пример визуального программирования в IDE Lazarus

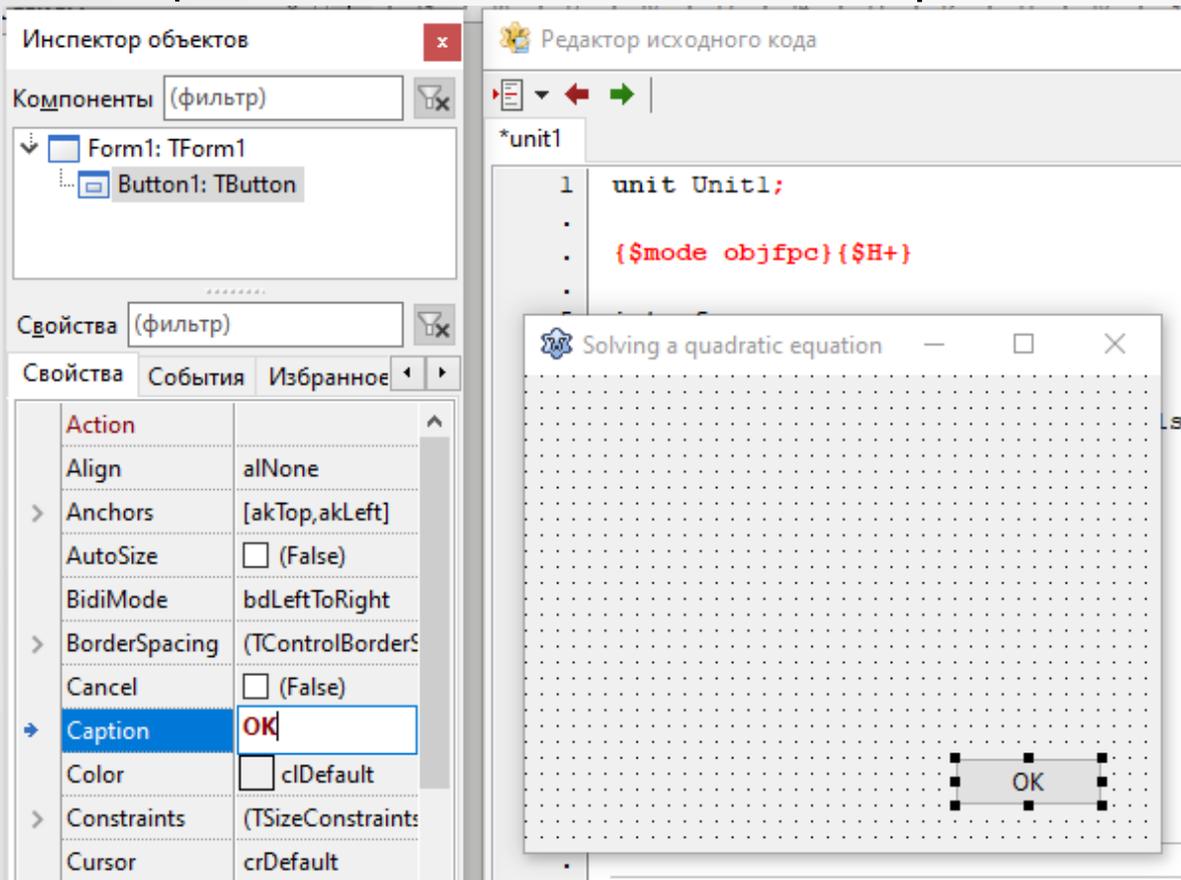
Для добавления на форму нового визуального компонента необходимо найти его в палитре компонентов главного окна Lazarus и два раза кликнуть на него мышкой.

Например, для того, чтобы добавить на форму кнопку, нужно в палитре компонентов найти элемент TButton и два раза кликнуть на него.



2. Пример визуального программирования в IDE Lazarus

После добавления компонента его следует настроить. В начале, кликнув один раз на него мышью, сделать его активным, после чего в окне инспектора объектов можно править его свойства и события. При необходимости, с помощью мыши его также можно переместить в нужную позицию на форме.

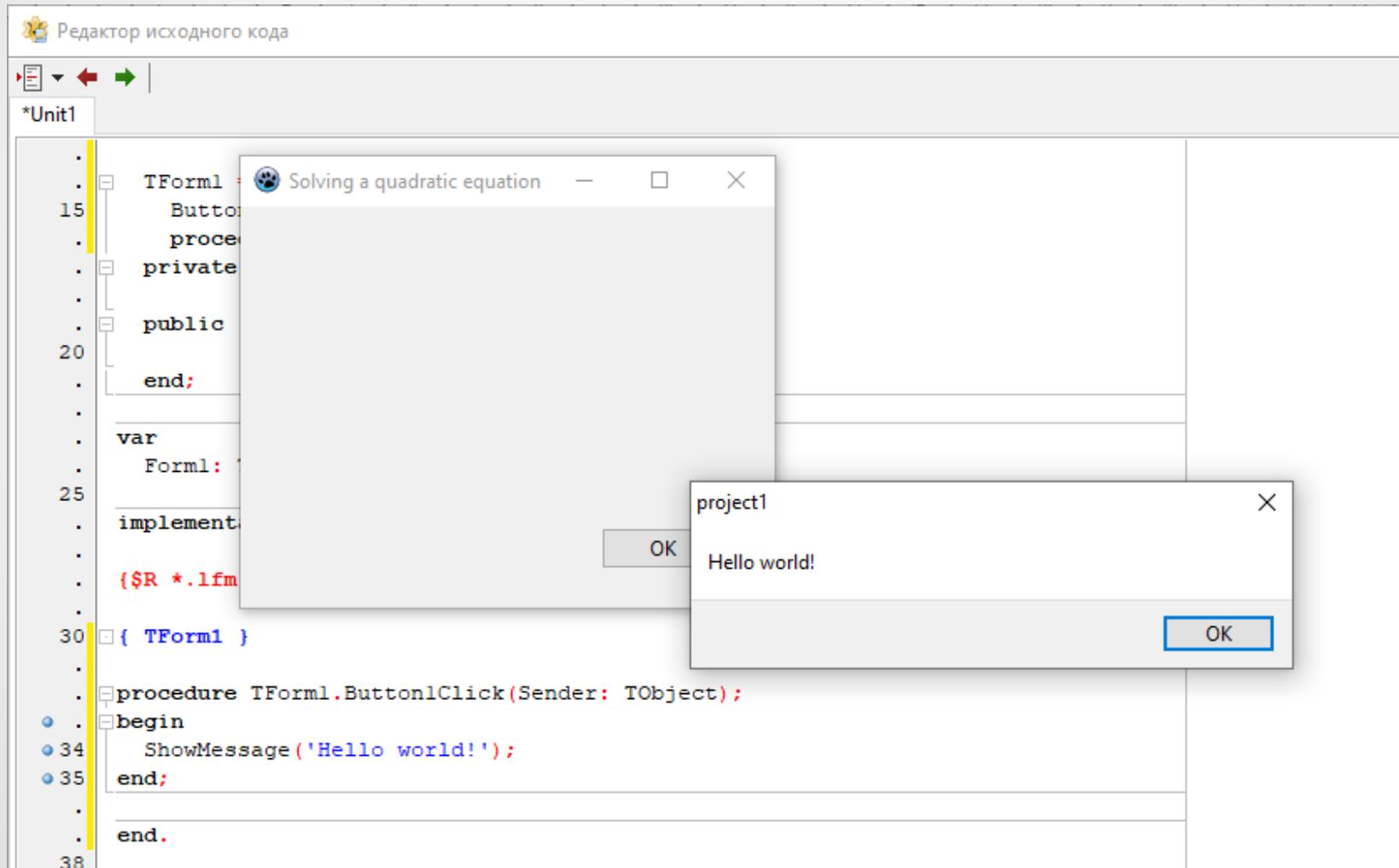


Изменяя, например, свойство «Caption», можно задать текст на кнопке, «Width» и «Height» – ее ширину и высоту и т. д.

Важно! В инспекторе объектов отображаются свойства только текущего компонента. Активная форма маркерами по периметру не выделяется.

2. Пример визуального программирования в IDE Lazarus

Например, для вывода сообщения при нажатии кнопки в обработчике соответствующего события можно написать такой код:



The screenshot shows the Lazarus IDE interface. The title bar reads "Редактор исходного кода". The code editor displays the following code for a unit named *Unit1:

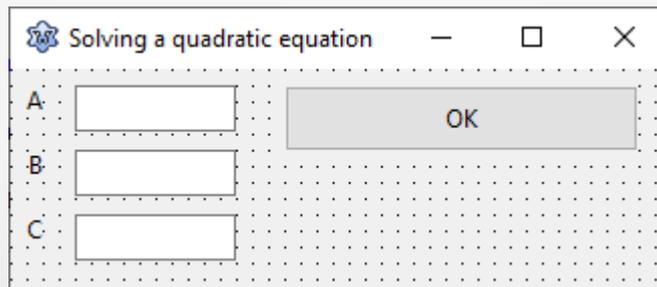
```
15 TForm1
    Button
    procedure
    private
    public
20 end;
var
    Form1:
25 implement
    {$R *.lfm
30 { TForm1 }
    procedure TForm1.Button1Click(Sender: TObject);
    begin
    34     ShowMessage('Hello world!');
    35 end;
    end.
```

Two dialog boxes are overlaid on the code editor. The top dialog box is titled "Solving a quadratic equation" and is empty. The bottom dialog box is titled "project1" and contains the text "Hello world!" with an "OK" button.

2. Пример визуального программирования в IDE Lazarus

Рассмотрим пример создание программы решения квадратного уравнения с графическим интерфейсом. Для этого.

1. Разместим на форме три компонента класса TLabel для вывода статической надписей. Их свойствам «Caption» присвоим значения «А», «В» и «С» соответственно.
2. Рядом с этими элементами разместим три компонента класса TEdit для ввода текстовой информации. Для красоты очистим значение их свойств «Text», которые содержат введенный (начальный) текст в поле ввода.
3. Добавим кнопку TButton и настроим ее свойство «Caption» таким образом, чтобы на кнопке отображалась надпись «ОК».



Пример графического интерфейса программы решения квадратного уравнения

2. Пример визуального программирования в IDE Lazarus

Кнопка «ОК» здесь предназначена для инициирования события начала решения уравнения. Таким образом, обработчик события OnClick для данной визуальной компоненты должен:

- 1) считать содержимое полей ввода Edit1, Edit2 и Edit3 (оно находится в свойстве «Text» класса TEdit) и преобразовать его к вещественным значениям;
- 2) в случае некорректно введенных данных вывести соответствующее сообщение;
- 3) в противном случае вычислить дискриминант, затем корни уравнения и вывести их на экран пользователю.

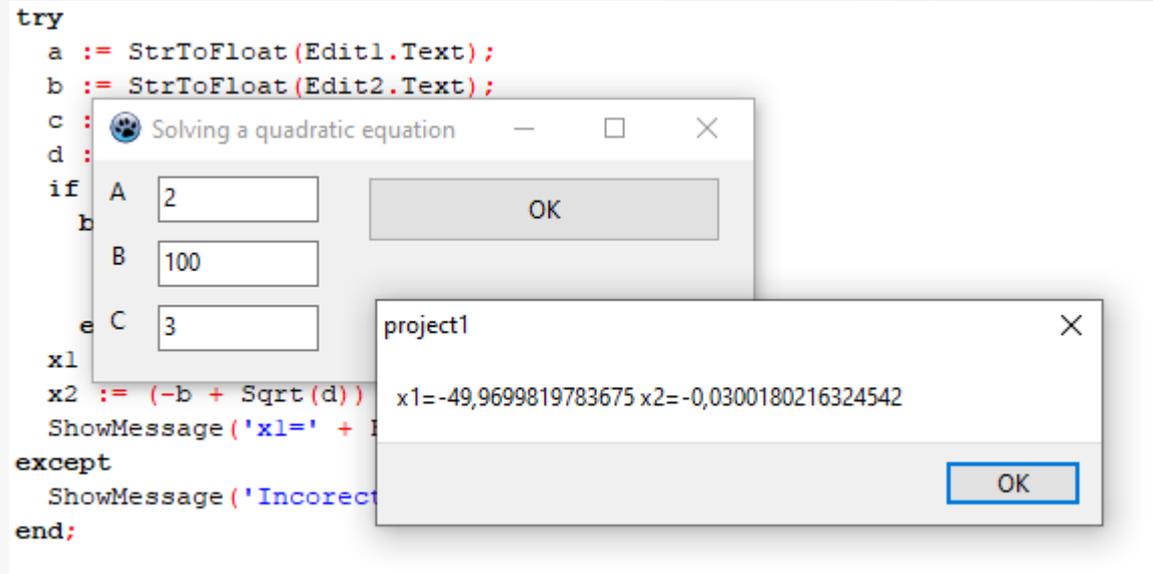
Примечание. Программы с GUI являются событийно-ориентированными. Для запуска процедуры расчета требуется получить соответствующее сообщение.

2. Пример визуального программирования в IDE Lazarus

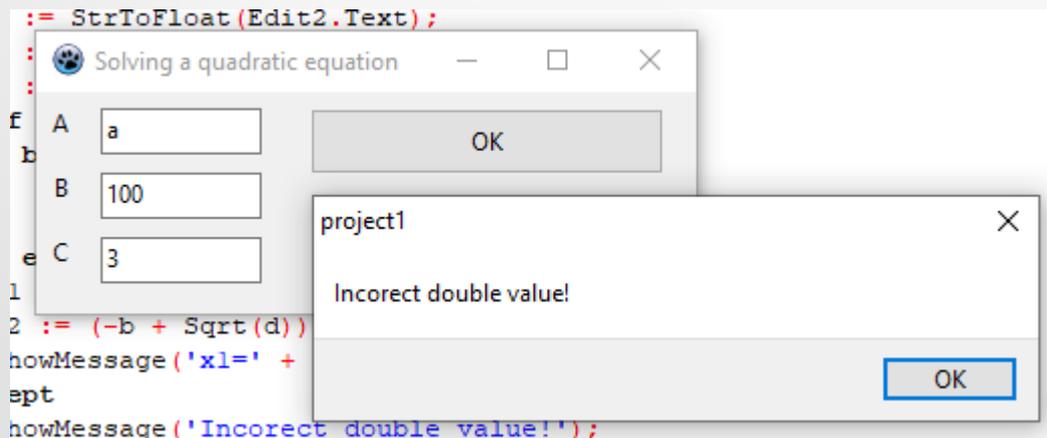
Тогда программа будет работать следующим образом:

В случае корректно введенных данных

```
try
  a := StrToFloat(Edit1.Text);
  b := StrToFloat(Edit2.Text);
  c := StrToFloat(Edit3.Text);
  d := 4 * a * c - b * b;
  if d >= 0 then
  begin
    x1 := (-b + Sqrt(d)) / (2 * a);
    x2 := (-b - Sqrt(d)) / (2 * a);
    ShowMessage('x1=' + FloatToStr(x1) + ' x2=' + FloatToStr(x2));
  end
except
  ShowMessage('Incorect double value!');
end;
```



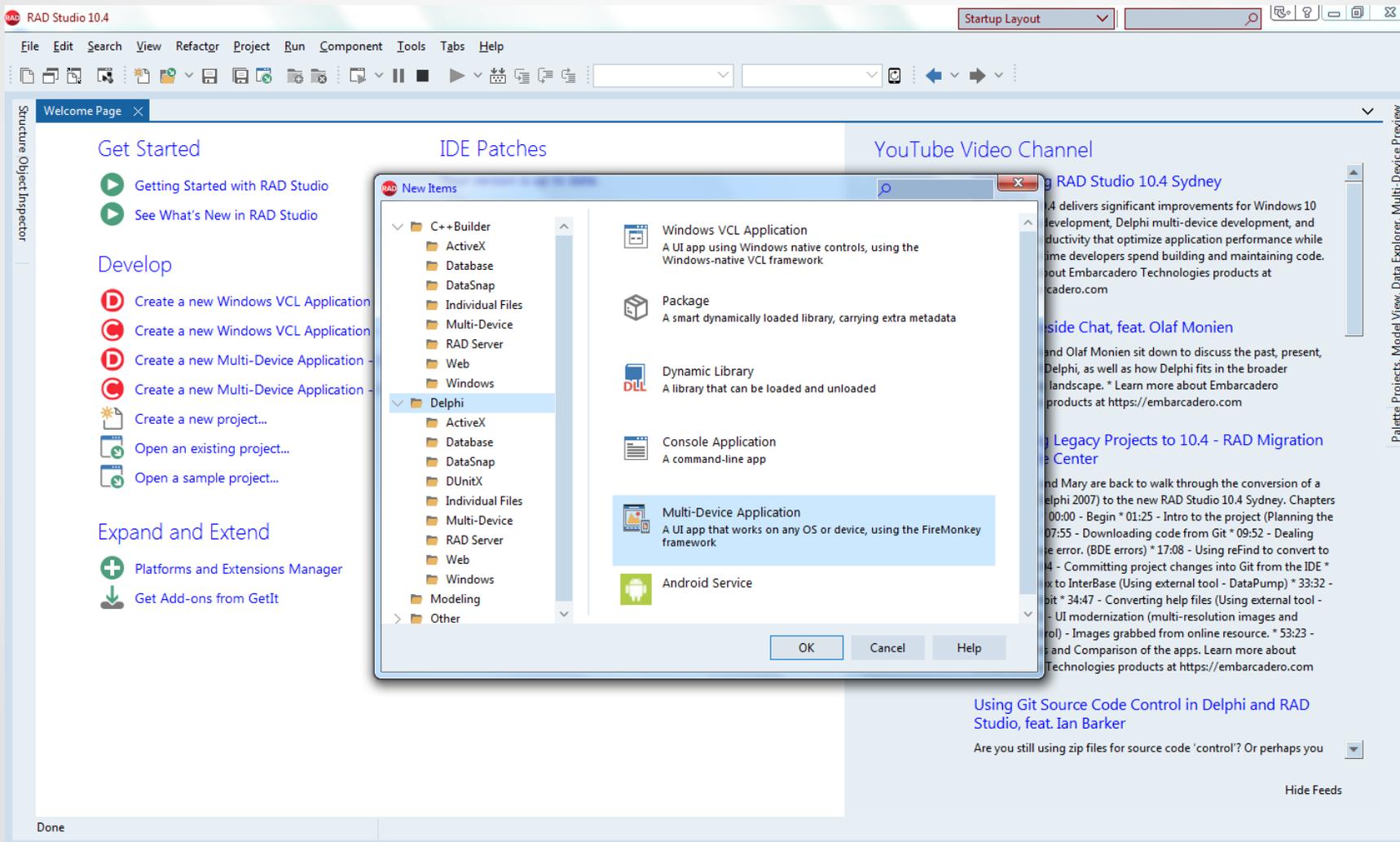
```
:= StrToFloat(Edit2.Text);
```



В случае некорректно введенных данных

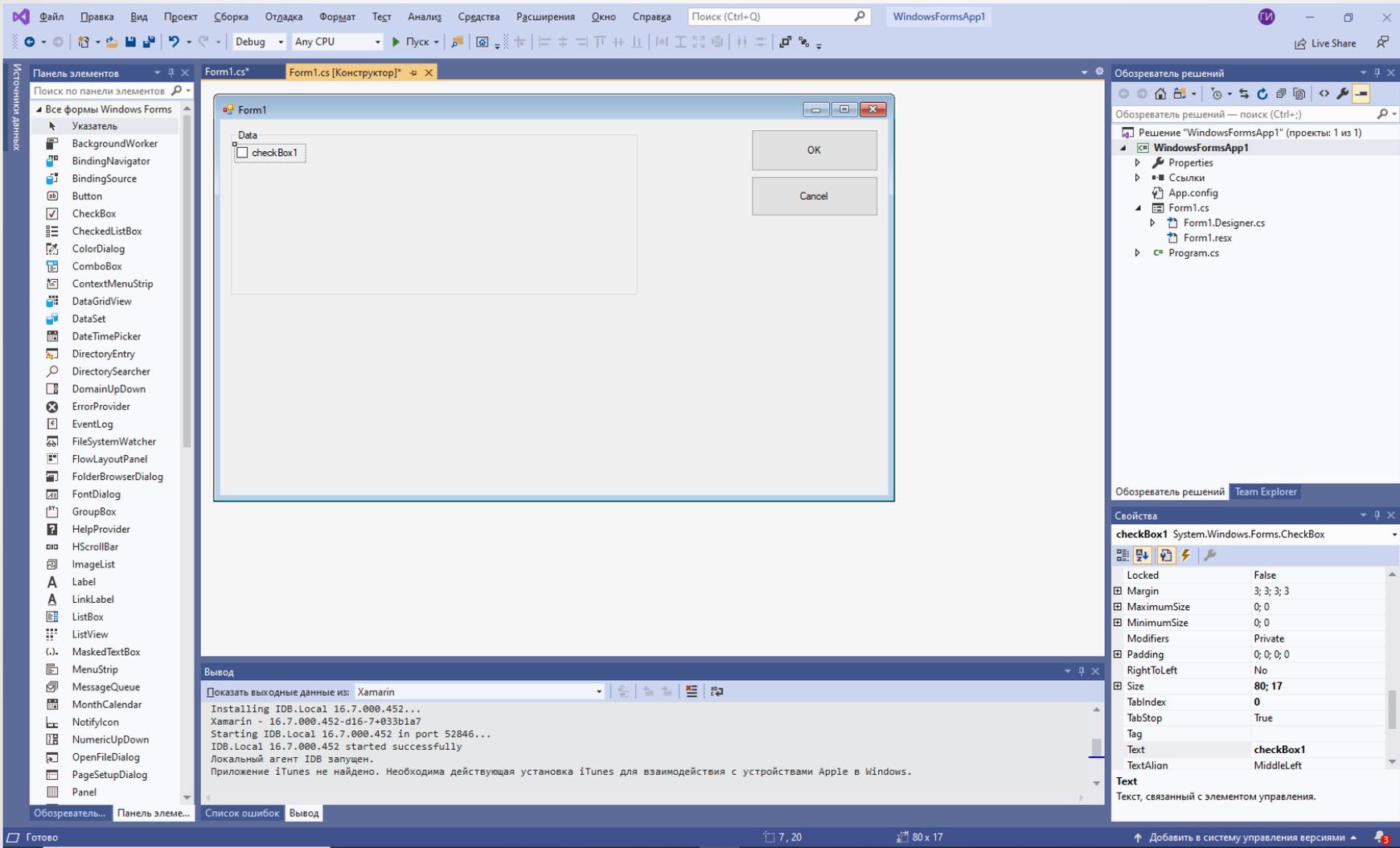
3. Современные IDE для визуального программирования

Embarcadero RAD Studio



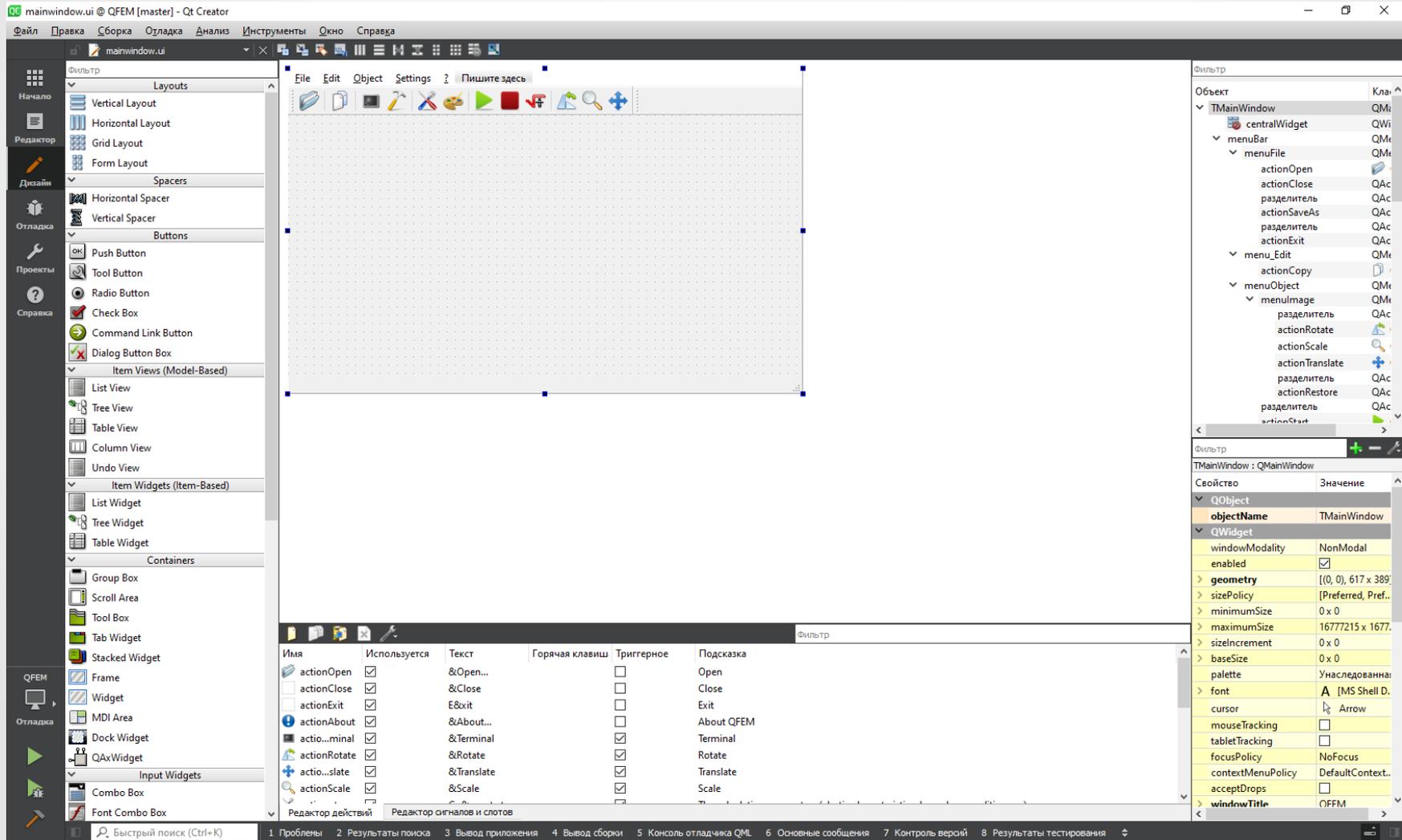
3. Современные IDE для визуального программирования

Microsoft Visual Studio



3. Современные IDE для визуального программирования

Qt Creator



3. Современные IDE для визуального программирования

Code::Blocks

