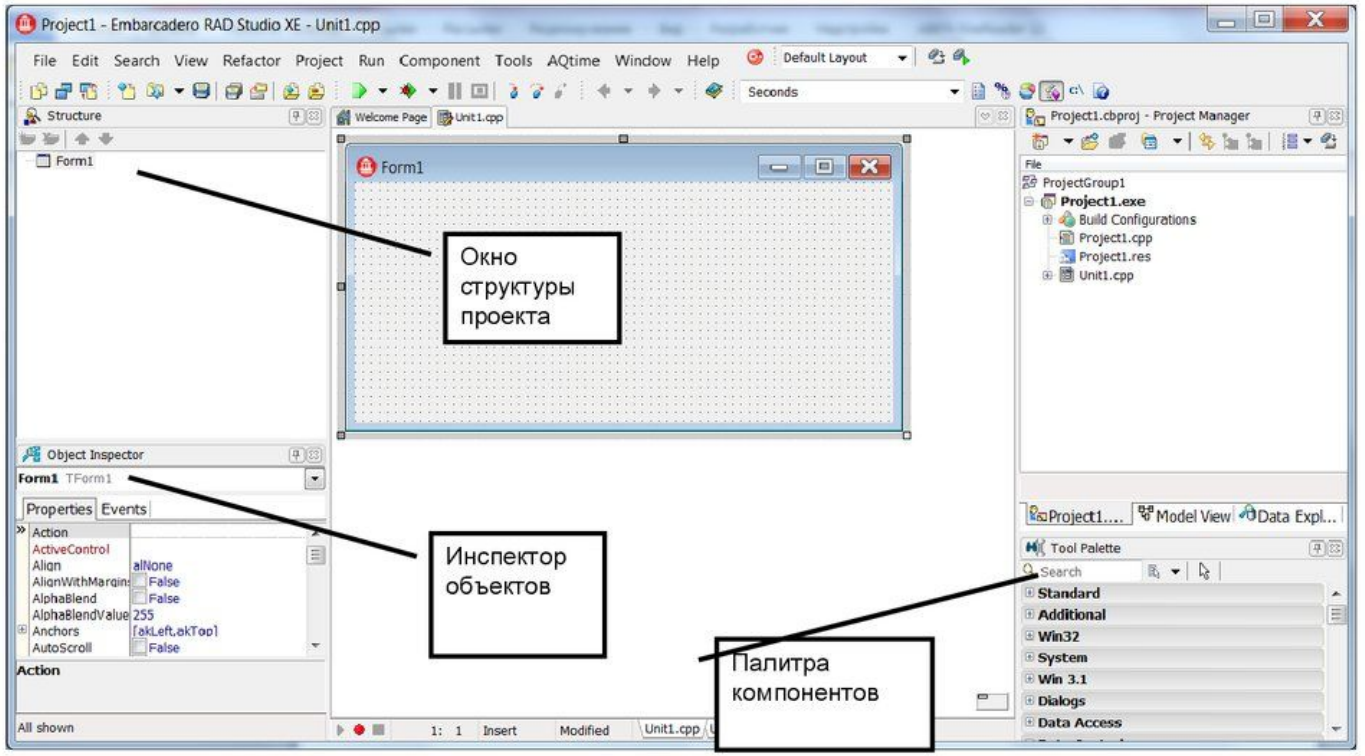
Работа с графикой в C++ Bulder

Embarcadero RAD Studio

Embarcadero RAD Studio



Библиотека классов системы

Сегодня в C++ Builder библиотекой классов является библиотека визуальных компонентов VCL – Visual Component Library) – библиотека визуальных компонентов. В ее основе лежит концепция свойств, методов и событий.

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

Инспектор объектов

С помощью инспектора объектов можно задавать начальные значения свойств объектов и их реакцию на стандартные события.

Окно инспектора объектов содержит список компонентов текущей формы, а также две закладки: свойства (Propeties) и события (Events).

Инспектор объектов

Свойства являются атрибутами компонента, определяющими его внешний вид и поведение. Каждый компонент имеет свой собственный набор обработчиков событий.

Label1->Caption = “First”;

В C++ Builder следует писать функции, называемые обработчиками событий, и связывать события с этими функциями. Создавая, обработчик того или и ого события, вы поручаете программе выполнить написанную функцию, если это событие произойдет.

Инспектор объектов

Метод является функцией, которая связана с компонентом, и которая объявляется как часть объекта. Создавая обработчики событий, можно вызывать методы, используя следующую нотацию:
->, например:

```
Label1->Show();
```

ЗАМЕЧАНИЕ: при создании формы связанные с ней модуль и заголовочный файл с расширением *.h генерируются обязательно, тогда как при создании нового модуля он не обязан быть связан с формой (например, если в нем содержатся процедуры расчетов).

Имена формы и модуля можно изменить, причем желательно сделать это сразу после создания, пока на них не появилось много ссылок в других формах и модулях.

Компонент форма - TForm

Как и любой другой визуальный компонент, форма имеет свойства, методы и события, общие для всех визуальных компонентов.

По умолчанию проект первоначально содержит файлы для одной формы и исходного кода одного модуля. Однако большинство проектов содержат несколько форм и модулей.

Компонент форма - TForm

- Caption — сюда помещается название формы. По умолчанию первая форма проекта получает имя Form1, вторая — Form2 и т. д.
- ActiveControl — это свойство определяет, какой компонент, помещенный на форму, в данный момент является активным (или, как еще говорят, имеет фокус).
- AutoScroll — это свойство определяет, будут ли автоматически появляться полосы прокрутки формы, если на ней не будут помещаться компоненты (т. е., чтобы их увидеть, надо будет форму "прокрутить"). Если значение этого свойства равно true, то полосы прокрутки будут появляться, а если false — нет.
- AutoSize — если значение свойства равно true, то форма автоматически будет принимать прежние размеры, как бы вы ее не растягивали. При свойстве, равном false, форма "позволяет" себя растягивать.
- BorderStyle — свойство задает появление и поведение границ формы: можно ли мышью менять размеры формы, когда приложение находится в режиме исполнения

Компонент форма - TForm

- OnActivate — возникает, когда форма активизируется;
- OnClick — возникает при щелчке мышью на форме;
- OnClose — возникает, когда форма закрывается;
- OnCreate — возникает, когда форма создается;
- OnDeactivate — возникает, когда форма перестает быть активной;
- OnKeyDown — возникает, когда пользователь нажимает некоторую клавишу на клавиатуре. Это событие может возникать в ответ на нажатие любых клавиш, включая функциональные (<F1>—<F12>) и комбинации с <Shift>, <Alt> и <Ctrl>;
- OnKeyPress — возникает, когда пользователь нажимает некоторую (только одну) клавишу на клавиатуре, кроме функциональных;
- OnKeyUp — возникает, когда пользователь отпускает клавишу клавиатуры, которая до этого была нажата. Это событие возникает и тогда, когда до этого были нажаты комбинации клавиш с <Shift>, <Alt> и <Ctrl>, а также функциональные клавиши;

Компонент форма - TForm

- OnMouseDown — возникает при нажатии любой кнопки мыши, а также при комбинации клавиш <Shift>, <Ctrl> и <Alt> и кнопок мыши. В обработчике этого события X и Y — это координаты в пикселах указателя мыши;
- OnMouseMove — возникает, когда пользователь двигает указатель мыши, пока он находится над объектом (в нашем случае — над формой). В обработчике этого события X и Y — это координаты в пикселах указателя мыши;
- OnMouseUp — возникает, когда пользователь отпускает кнопку мыши, которая была нажата на компоненте. В обработчике этого события X и Y — это координаты в пикселах указателя мыши;
- OnPaint — возникает, когда начинается прорисовка формы;
- OnShow — возникает, когда форма появляется на экране.

Построение графиков

Обычно результаты расчетов представляются в виде графиков и диаграмм.

Способы построения графиков и диаграмм:

1. Использование компонента TChart
2. Использование свойства Canvas формы
3. Использование свойства Canvas PaintBox

Построение графиков

TCanvas (Канва) - это класс, предназначенный для вывода и хранения графических объектов в C++ Builder.

Канва входит в состав большинства визуальных компонентов, кроме стандартных оконных контролеров (TButton, TMemo, TPanel и т.п.).
При помощи методов этого класса можно рисовать как и стандартные примитивы (линии, эллипсы, прямоугольники), так и графические объекты типа Graphics::TBitmap.

Построение графиков

Методы класса TCanvas-, используемые для создания графика:

MoveTo(x,y) – перейти к точке холста с координатами (x,y) (в пикселах);

LineTo(x,y) – нарисовать линию из предыдущей точки в точку с координатами (x,y).

Функции Канвы

Метод (Функция)	Действие
MoveTo	Определяет текущую позицию пера
LineTo	Рисует прямую до заданной точки
Rectangle	Рисует прямоугольник
Ellipse	Рисует эллипс
Arc	Рисует дугу
Polyline	Рисует ломаную линию
PolyBezier	Рисует кривую Безье
Chord	Рисует сектор
DrawFocusRect	Рисует прямоугольник
FrameRect	Выводит рамку вокруг прямоугольника
Pie	Выводит сектор круга
TextOut	Выводит текстовую строку
TextHeight	Задает высоту текстовой строки
TextWidth	Задает ширину для вывода текстовой строки
TextRect	Вывод текста внутри прямоугольника
FillRect	Заливка указанного прямоугольника цветом и текстурой текущей кисти
FloodFill	Заливка области канвы (произвольной формы) заданным цветом

Цвета Pen

Form1 –> Canvas->Pen->Color = clRed;

Таблица констант:

Константа	Значение цвета
clBlack	Черный
clMaroon	Темно-бордовый
clGreen	Зеленый
clOlive	Оливково-зеленый
clNavy	Темно-синий
clPurple	Пурпурный
clTeal	Морской воды
clGray	Серый
clSilver	Серебряный
clRed	Красный
clLime	Лимонно-зеленый
clBlue	Синий
clYellow	Желтый
clFuchsia	Сиреневый
clAqua	Голубой
clWhite	Белый

Стиль Pen

Form1 –> Canvas->Pen->Style = psSolid;

Значение	Описание
psSolid	Сплошная линия
psDash	Штриховая линия
psDot	Пунктирная линия
psDashDot	Штрих-пунктирная линия
psDashDotDot	Линия, чередующая штрих и два пунктира
psClear	Отсутствие линии
psInsideFrame	Сплошная линия, но при Width>1 допускающая цвета, отличные от палитры Windows

Построение графиков

Система C++Builder имеет мощный пакет стандартных программ вывода на экран и редактирования графической информации, который реализуется с помощью визуально отображаемого на форме компонента TChart (вкладка TeeCartStd)

Построение графиков

* Компонент TChart осуществляет всю работу по отображению графиков, переданных в объект Seriesk: строит и размечает оси, рисует координатную сетку, подписывает название осей и самого графика, отображает переданную таблицу в виде всевозможных графиков или диаграмм.
* При необходимости, с помощью встроенного редактора EditingChart компоненту TChart передаются данные о толщине, стиле и цвете линий, параметрах шрифта подписей, шагах разметки координатной сетки и другие настройки.

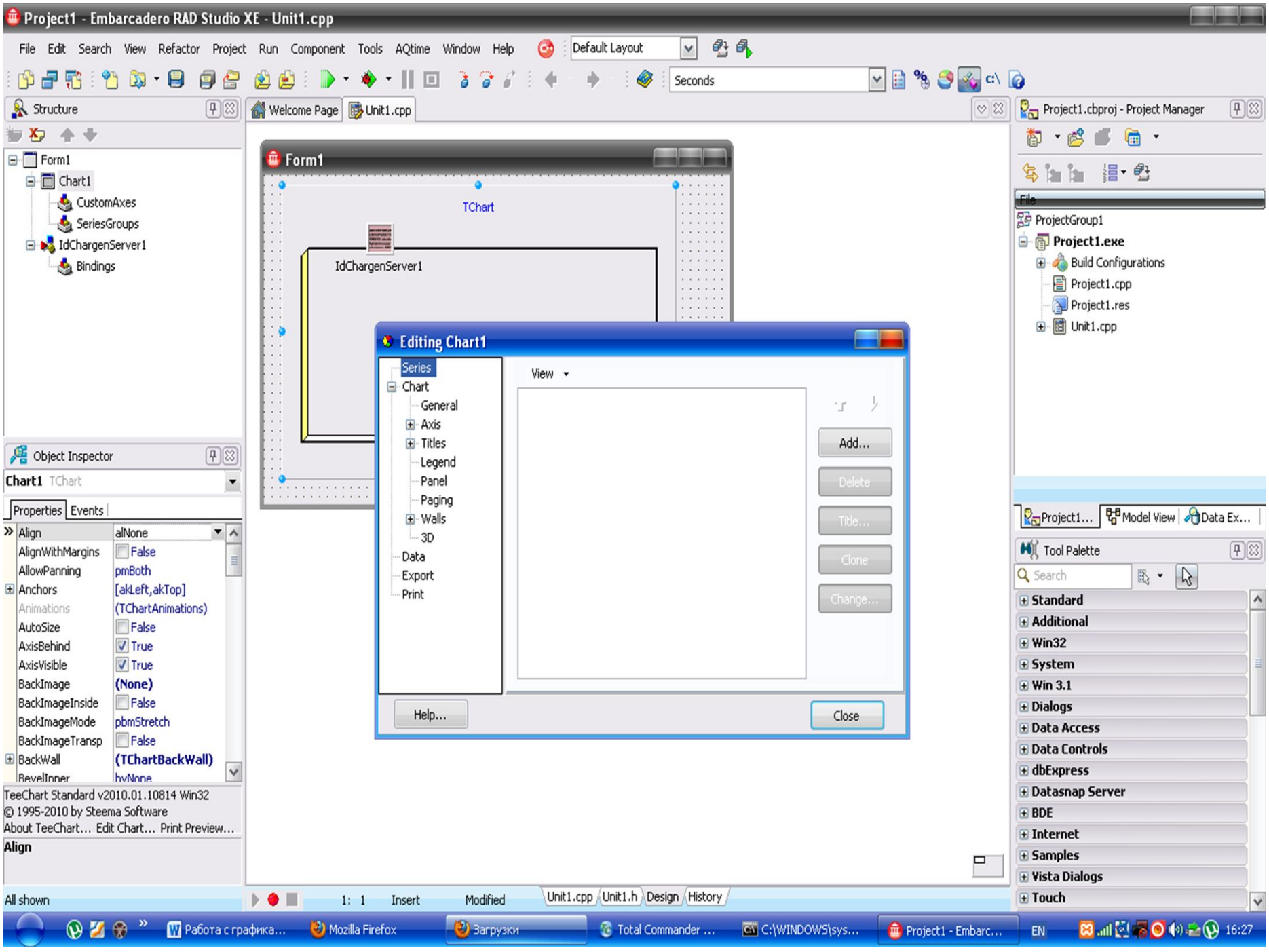
Построение графиков

Вызов редактора EditingChart:

двойной щелчок на объекте Chart1

контекстное меню на объекте Chart1 и выбрать Edit Chart…

ЗАМЕЧАНИЕ: график можно настраивать и изменять параметры и в ходе выполнения программы, то есть его настройка не ограничивается одним окошком.


Project1 - Embarcadero RAD Studio XE - Unit1.cpp
File Edit Search View Refactor Project Run Component Tools AQtime Window Help
Default Layout
Seconds
Structure
Form1
Chart1
CustomAxes
SeriesGroups
IdChargenServer1
Bindings
Welcome Page
Unit1.cpp
Form1
TChart
IdChargenServer1
Editing Chart1
Series
Chart
General
Axis
Titles
Legend
Panel
Paging
Walls
3D
Data
Export
Print
View
Add...
Delete
Title...
Clone
Change...
Help...
Close
Project1.cbproj - Project Manager
File
ProjectGroup1
Project1.exe
Build Configurations
Project1.cpp
Project1.res
Unit1.cpp
Object Inspector
Chart1 TChart
Properties
Events
Align alNone
AlignWithMargins False
AllowPanning pmBoth
Anchors [akLeft,akTop]
Animations (TChartAnimations)
AutoSize False
AxisBehind True
AxisVisible True
BackImage (None)
BackImageInside False
BackImageMode pbmStretch
BackImageTransp False
BackWall (TChartBackWall)
BevelInner bvNone
TeeChart Standard v2010.01.10814 Win32
© 1995-2010 by Steema Software
About TeeChart... Edit Chart... Print Preview...
Align
Project1...
Model View
Data Ex...
Tool Palette
Search
Standard
Additional
Win32
System
Win 3.1
Dialogs
Data Access
Data Controls
dbExpress
Datasnap Server
BDE
Internet
Samples
Vista Dialogs
Touch
All shown
1: 1
Insert
Modified
Unit1.cpp
Unit1.h
Design
History
Работа с графика...
Mozilla Firefox
Загрузки
Total Commander ...
C:\WINDOWS\sys...
Project1 - Embarc...
EN
16:27

Построение графиков

Для добавления координат точек из таблицы в двумерный массив объекта Seriesk используется функция

Series1->AddXY(AXValue;AYValue;AXLabel;AColor) ,

где AXValue, AYValue – координаты точки по осям X и Y;

AXLabel – текстовая надпись добавленной точки;

AColor задает цвет линий (если равен clTeeColor, то принимается цвет, определенный при проектировании формы).

Например: Series1->Add(y,x,clRed);

Задание

1) 2 графика на Form1 -> Canvas с основными и дополнительными осями.

2) 3 графика в Tchart с элементами управления

3) Анимацию в PaintBox

Варианты

Вариант	4.1	4.2	4.3
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	7	7	7
8	8	8	8
9	9	9	9
10	10	10	10
11	11	1	11
12	12	2	1
13	13	3	2
14	14	4	3
15	15	5	4
16	16	6	5

Варианты

Вариант	4.1	4.2	4.3
17	16	7	6
18	15	8	7
19	14	9	8
20	13	10	9
21	12	1	10
22	11	2	11
23	10	3	1
24	9	4	2
25	8	5	3
26	7	6	4
27	6	7	5
28	5	8	6
29	4	9	7
30	3	10	8
31	2	1	9
32	1	2	10