**4.6.1 Компонент Chart**

Теперь рассмотрим компонент **Chart**. Этот компонент позволяет строить различные диаграммы и графики, которые выглядят очень эффектно (рис. 4.8). Компонент **Chart** имеет множество свойств, методов, событий, так что если все их рассматривать, то этому пришлось бы посвятить целую главу. Поэтому ограничимся рассмотрением только основных характеристик **Chart**. А с остальными вы можете ознакомиться во встроенной справке Delphi или просто опробовать их, экспериментируя с диаграммами.

|  |  |  |
| --- | --- | --- |
| **Рис. 4.8** Пример приложения с диаграммами: начальное состояние (а) и состояние при изменении типа диаграммы и увеличении фрагмента графика (б) | а) | http://www.beluch.ru/progr/100comp/gif/4_8a.gif |
|  | б) | http://www.beluch.ru/progr/100comp/gif/4_8b.gif |

Компонент **Chart** является контейнером объектов **Series** типа **TChartSeries** — серий данных, характеризующихся различными стилями отображения. Каждый компонент может включать несколько серий. Если вы хотите отображать график, то каждая серия будет соответствовать одной кривой на графике. Если вы хотите отображать диаграммы, то для некоторых видов диаграмм можно наложить друг на друга несколько различных серий, для других (например, для круговых диаграмм) это, вероятно, будет выглядеть некрасиво. Однако, и в этом случае вы можете задать для одного компонента **Chart** несколько серий одинаковых данных с разным типом диаграммы. Тогда, делая в каждый момент времени активной одну из них, вы можете предоставить пользователю выбор типа диаграммы, отображающей интересующие его данные.

Разместите один или два (если захотите воспроизвести рис. 4.8) компонента **Chart** на форме и посмотрите открывшиеся в Инспекторе Объектов свойства. Приведем пояснения некоторых из них.

|  |  |
| --- | --- |
| **AllowPanning** | Определяет возможность пользователя прокручивать наблюдаемую часть графика во время выполнения, нажимая правую кнопку мыши. Возможные значения: **pmNone** — прокрутка запрещена, **pmHorizontal**, **pmVertical** или **pmBoth** — разрешена соответственно прокрутка только в горизонтальном направлении, только в вертикальном или в обоих направлениях. |
| **AllowZoom** | Позволяет пользователю изменять во время выполнения масштаб изображения, вырезая фрагменты диаграммы или графика курсором мыши (на рис. 4.8 б внизу показан момент просмотра фрагмента графика, целиком представленного на рис. 4.8 а). |
| **Title** | Определяет заголовок диаграммы. |
| **Foot** | Определяет подпись под диаграммой. По умолчанию отсутствует. Текст подписи определяется подсвойством **Text**. |
| **Frame** | Определяет рамку вокруг диаграммы. |
| **Legend** | Легенда диаграммы — список обозначений. |
| **MarginLeft , MarginRight , MarginTop, MarginBottom** | Значения левого, правого, верхнего и нижнего полей. |
| **BottomAxis, LeftAxis, RightAxis** | Эти свойства определяют характеристики соответственно нижней, левой и правой осей. Задание этих свойств имеет смысл для графиков и некоторых типов диаграмм. |
| **LeftWall, BottomWall, BackWall** | Эти свойства определяют характеристики соответственно левой, нижней и задней граней области трехмерного отображения графика (см. рис. 4.8 а, нижний график). |
| **SeriesList** | Список серий данных, отображаемых в компоненте. |
| **View3d** | Разрешает или запрещает трехмерное отображение диаграммы. |
| **View3DOptions** | Характеристики трехмерного отображения. |
| **Chart3DPercent** | Масштаб трехмерности (для рис. 4.8 это толщина диаграммы и ширина лент графика). |

Рядом со многими из перечисленных свойств в Инспекторе Объектов расположены кнопки с многоточием, которые позволяют вызвать ту или иную страницу Редактора Диаграмм — многостраничного окна, позволяющего установить все свойства диаграмм. Вызов Редактора Диаграмм возможен также двойным щелчком на компоненте **Chart** или щелчком на нем правой кнопкой мыши и выбором команды Edit Chart во всплывшем меню.

Если вы хотите попробовать воспроизвести приложение, показанное на рис. 4.8, сделайте двойной щелчок на верхнем компоненте **Chart**. Вы попадете в окно Редактора Диаграмм (рис. 4.9) на страницу Chart, которая имеет несколько закладок. Прежде всего вас будет интересовать на ней закладка Series. Щелкните на кнопке Add — добавить серию. Вы попадете в окно (рис. 4.10), в котором вы можете выбрать тип диаграммы или графика. В данном случае выберите Pie — круговую диаграмму. Воспользовавшись закладкой Titles вы можете задать заголовок диаграммы, закладка Legend позволяет задать параметры отображения легенды диаграммы (списка обозначений) или вообще убрать ее с экрана, закладка Panel определяет вид панели, на которой отображается диаграмма, закладка 3D дает вам возможность изменить внешний вид вашей диаграммы: наклон, сдвиг, толщину и т.д.

Когда вы работаете с Редактором Диаграмм и выбрали тип диаграммы, в компонентах **Chart** на вашей форме отображается ее вид с занесенными в нее условными данными (см. рис. 4.11).

|  |  |
| --- | --- |
| **Рис. 4.9** Редактор Диаграмм, страницаChart, закладкаSeries | http://www.beluch.ru/progr/100comp/gif/4_9.gif |

|  |  |
| --- | --- |
| **Рис. 4.10** Выбор типа диаграммы в Редакторе Диаграмм | http://www.beluch.ru/progr/100comp/gif/4_10.gif |

Поэтому вы сразу можете наблюдать результат применения различных опций к вашему приложению, что очень удобно.

Страница Series, также имеющая ряд закладок, дает вам возможность выбрать дополнительные характеристики отображения серии. В частности, для круговой диаграммы на закладке Format полезно включить опцию Circled Pie, которая обеспечит при любом размере компонента **Chart** отображение диаграммы в виде круга. На закладке Marks кнопки группы Style определяют, что будет написано на ярлычках, относящихся к отдельным сегментам диаграммы: Value — значение, Percent — проценты, Label — названия данных и т.д. В примере рис. 4.8 включена кнопка Percent, a на закладке General установлен шаблон процентов, обеспечивающий отображение только целых значений.

|  |  |
| --- | --- |
| **Рис. 4.11** Форма приложения рис. 4.8 с занесенными в нее условными данными | http://www.beluch.ru/progr/100comp/gif/4_11.gif |

Вы можете, если хотите, добавить на этот компонент **Chart** еще одну тождественную серию, нажав на закладке Series страницы Chartкнопку Clone, а затем для этой новой серии нажать кнопку Change (изменить) и выбрать другой тип диаграммы, например, Bar. Конечно, два разных типа диаграммы на одном рисунке будут выглядеть плохо. Но вы можете выключить индикатор этой новой серии на закладкеSeries, а потом предоставить пользователю выбрать тот или иной вид отображения диаграммы (ниже будет показано, как это делается).

Выйдите из Редактора Диаграмм, выделите в вашем приложении нижний компонент **Chart** и повторите для него задание свойств с помощью Редактора Диаграмм. В данном случае вам надо будет задать две серии, если хотите отображать на графике две кривые, и выбрать тип диаграммы Line. Поскольку речь идет о графиках, вы можете воспользоваться закладками Axis и Walls для задания координатных характеристик осей и трехмерных граней графика.

На этом проектирование внешнего вида приложения завершается. Осталось написать код, задающий данные, которые вы хотите отображать. Для тестового приложения давайте зададим в круговой диаграмме просто некоторые константные данные, а в графиках — функции синус и косинус.

Для задания отображаемых значений надо использовать методы серий **Series**. Остановимся только на трех основных методах.

Метод **Clear** очищает серию от занесенных ранее данных.

Метод **Add**:

Add(Const AValue: Double; Const ALabel: String;

AColor: TColor)

позволяет добавить в диаграмму новую точку. Параметр **AValue** соответствует добавляемому значению, параметр **ALabel** — название, которое будет отображаться на диаграмме и в легенде, **AColor** — цвет. Параметр **ALabel** — не обязательный, его можно задать пустым: ''.

Метод **AddXY**:

AddXY(Const AXValue, AYValue: Double;

Const ALabel: String; AColor: TColor)

позволяет добавить новую точку в график функции. Параметры **AXValue** и **AYValue** соответствуют аргументу и функции. Параметры **ALabel**и **AColor** те же, что и в методе **Add**.

Таким образом, процедура, обеспечивающая загрузку данных в нашем примере, может иметь вид:

**const**

A1=155;

A2=251;

A3=203;

A4=404;

**var**

i: word;

**begin**

With Series1 **do**

**begin**

Clear;

Add(A1, 'Цех 1', clYellow);

Add(A2, 'Цех 2', clBlue);

Add(A3, 'Цех 3', clRed);

Add(A4, 'Цех 4', clPurple);

**end**;

Series2.Clear;

Series3.Clear;

**for** i:=0 **to** 100 **do**

**begin**

Series2.AddXY(0.02\*Pi\*i, sin(0.02\*Pi\*i), '', clRed);

Series3.AddXY(0.02\*Pi\*i, cos(0.02\*Pi\*i), '', clBlue);

**end**;

**end**;

Эту процедуру можно включить в обработку щелчка какой-нибудь кнопки, в команду меню или просто в событие **OnCreate** формы. Операторы **Clear** нужны, если в процессе работы приложения вы собираетесь обновлять данные. Без этих операторов повторное выполнение методов **Add** и **AddXY** только добавит новые точки, не удалив прежние.

Если вы предусмотрели, например, для данных, отображаемых в диаграмме, две серии **Series1** и **Series4** разных видов — **Pie** и **Bar**, то можете ввести процедуру, изменяющую по требованию пользователя тип диаграммы. Эту процедуру можно ввести в событие **OnClick** какой-нибудь кнопки, в команду меню или, например, просто в обработку щелчка на компоненте **Chart**. Для того, чтобы загрузить данные в **Series4**и сделать эту диаграмму в первый момент невидимой, можно вставить в конце приведенной ранее процедуры операторы

Series4.Assign(Series1);

Series4.Active:=false;

Первый из этих операторов переписывает данные, помещенные в **Series1**, в серию **Series4**. А второй оператор делает невидимой серию**Series4**. Смена типа диаграммы осуществляет процедура

Series1.Active := **not** Series1.Active;

Series4.Active := **not** Series4.Active;

На рис. 4.8 б вы можете видеть результат переключения пользователя на другой вид диаграммы.