

Программирование и алгоритмизация

Лекция 4

Введение в C++ Builder

Интегрированная среда разработки объединяет

- ***Редактор форм***
- ***Инспектор объектов***
- ***Палитру компонент***
- ***Администратор проекта***
- ***Полностью интегрированные Редактор
кода и Отладчик - инструменты
быстрой разработки программных
приложений, обеспечивающие полный
контроль над кодом и ресурсами.***

- ***Механизмы двунаправленной разработки (two-way-tools) устраняют барьеры между программистом и его кодом.***
- ***Технология двунаправленной разработки обеспечивает контроль за вашим кодом посредством гибкого, интегрированного и синхронизированного взаимодействия между инструментами визуального проектирования и Редактором кода.***

Конструирование по способу "drag-and-drop"

- **позволяет создавать приложение простым перетаскиванием захваченных мышью визуальных компонент из Палитры на форму приложения.**
- **Инспектор объектов предоставляет возможность оперировать со свойствами и событиями компонент, автоматически создавая заготовки функций обработки событий, которые наполняются кодом и редактируются в процессе разработки.**

- **Свойства, методы и события - это именно те элементы языка, которые обеспечивают быструю разработку приложений в рамках объектно-ориентированного программирования.**
- **Свойства позволяют легко устанавливать разнообразные характеристики объектов.**
- **Методы производят определенные, иногда довольно сложные, операции над объектом.**
- **События связывают воздействия пользователя на объекты с кодами реакции на эти воздействия. События могут возникать при таких специфических изменениях состояния объектов как обновление данных в интерфейсных элементах доступа к базам данных.**
- **Работая совместно, свойства, методы и события образуют среду RAD (Rapid Application Development) быстрого и интуитивного программирования надежных приложений для Windows.**

- ***Визуальное наследование форм воплощает важнейший аспект объектно-ориентированного программирования в удобном для пользования инструменте визуального проектирования.***
- ***Характеристики новой формы приложения могут быть унаследованы от любой другой существующей формы, что обеспечивает централизованную репродукцию изменений пользовательского интерфейса, облегчает контроль за кодом и уменьшает временные затраты на введение новых качественных атрибутов.***

- **Испытание прототипа позволяет без труда переходить от прототипа приложения к полностью функциональному, профессионально оформленному программному продукту, действуя в пределах интегрированной среды.**
- **Чтобы удостовериться, что ваша программа производит ожидаемые результаты, раньше приходилось многократно проходить по циклу редактирование => компиляция => сборка, непроизводительно расходуя время. C++Builder объединяет три этапа разработки в единый производственный процесс. В результате удается строить приложения, базирующиеся на текущих требованиях заказчика, вместе с тем гибкие настолько, чтобы быстро адаптировать их к новым запросам пользователей.**

Библиотека Визуальных Компонент

- **Библиотека Визуальных Компонент VCL приобрела статус нового промышленного стандарта и в настоящее время применяется более чем полумиллионом пользователей, существенно ускоряя разработку надежных приложений любой степени сложности. VCL содержит более 100 повторно используемых компонент, которые реализуют все элементы пользовательского интерфейса операционной системы Windows.**
- **Кроме того, VCL предоставляют в распоряжение программистов такие оригинальные объекты, как записные книжки с закладками, табличные сетки для отображения содержимого баз данных и даже органы управления устройствами мультимедиа.**
- **Находясь в среде объектно-ориентированного Программирования C++Builder, компоненты можно использовать непосредственно, менять их свойства, облик и поведение или породить производные элементы, обладающие нужными отличительными характеристиками.**

- ***Хранилище объектов является инструментом новой методики хранения и повторного использования модулей данных, объектов, форм и программной бизнес-логики.***
- ***Поскольку построение нового приложения на существующем фундаменте значительно экономит временные затраты, хранилище объектов предоставляет для повторного использования готовые структуры: формы и законченные программные модули.***
- ***Создавая прототип нового приложения, вы можете наследовать, ссылаться или просто копировать существующую структуру - точно так же архитектор приступает к проектированию нового здания.***

File Edit Search View Project Run Component Database Tools Window Help <None>


Standard Additional Win32 System Data Access Data Controls dbExpress DataSnap BDE ADO InterBase WebServices Internet...

Standard toolbar icons: File, Edit, View, Run, Component, Database, Tools, Window, Help, Standard, Additional, Win32, System, Data Access, Data Controls, dbExpress, DataSnap, BDE, ADO, InterBase, WebServices, Internet...

Object TreeView

- Form1

Form1



Object Inspector

1 TForm1

Properties | Events

ActiveControl	
Align	alNone
AlphaBlend	false
AlphaBlendValue	255
Anchor	[akLeft,akTop]
AutoScroll	true
AutoSize	false
BorderStyle	bdLeftToRight
BorderStyle	[biSystemMenu, bsSizeable]
BorderStyle	0
Caption	Form1
ClientHeight	446
ClientWidth	688
Color	clBtnFace
Constraints	[TSizeConstraint]
Owner	

1: 1 Modified Insert \Unit1.cpp\Unit1.h\Diagram\

today



Палитра компонент.

Состоит из страниц:

- Standard
- Additional
- Win32
- System
- Data Access
- Data Controls
- dbExpress
- DataSnap
- BDE
- ADO
- InterBase
- WebServices
- InternetExpress
- Internet
- WebSnap
- FastNet
- Decision Cube
- QReport
- Dialogs
- Win 3.1
- Samples
- ActiveX
- Com+
- InterBase Admin
- Servers
-

Палитра компонент. Standard

TМашМени	Создает панель команд главного меню для формы.
TPopUpMerm	Создает "выскакивающее" меню для формы или для другой компоненты.
TLabel	Отображает на форме текст названия, который нельзя редактировать.
TEdit	Отображает область редактируемого ввода одиночной строки информации на форме.
TMemo	Отображает область редактируемого ввода множественных строк информации на форме.
TButton	Создает кнопку с надписью.
TCheckBox	Создает элемент управления с двумя состояниями.
TRadioButton	Создает элемент управления с двумя состояниями.
TListBox	Отображает область списка текстовых строк.
TComboBox	Создает комбинацию области редактирования и выпадающего списка текстовых строк.
TScrollBar	Создает линейку прокрутки для просмотра содержимого окна, формы, списка или диапазона значений.
TGroupBox	Создает контейнер, объединяющий на форме логически связанную группу некоторых компонент.
TRadioGroup	Создает контейнер, объединяющий на форме группу логически взаимоисключающих радио-кнопок.
TPanel	Создает панель инструментов или строк состояния.

Палитра компонент. Win32

TTabControl	Отображает набор полей, имеющих вид частично перекрывающихся друг друга картотечных вкладок.
TPageControl	Отображает набор полей, имеющих вид частично перекрывающихся друг друга картотечных вкладок, для организации многостраничного диалога.
TTreeView	Отображает древовидный перечень элементов - заголовков документов, записей в указателе, файлов или каталогов на диске.
TListView	Отображает древовидный перечень элементов в различных видах - по столбцам с заголовками, вертикально, горизонтально, с пиктограммами.
TImageList	Создает контейнер для группы изображений.
THeaderControl	Создает контейнер для заголовков столбцов.
TRichEdit	Отображает область редактируемого ввода множественных строк информации в формате RTF.
TStatusBar	Создает строку панелей состояния для отображения статусной информации.
TTrackBar	Создает шкалу с метками и регулятором текущего положения.
TProgressBar	Создает индикатор процесса выполнения некоторой процедуры в приложении.
TUpDown	Создает спаренные кнопки со стрелками "вверх" и "вниз". Нажатие этих кнопок вызывает увеличение или уменьшение значения свойства Position .
THotKey	Используется для установки клавиш быстрого вызова во время выполнения программы.

Палитра компонент. Additional

TBitBtn	Создает кнопку с изображением битового образа.
TSpeedButton	Создает графическую кнопку быстрого вызова.
TMaskEdit	Создает область редактируемого ввода данных специфического формата.
TStringGrid	Создает сетку для отображения строк по строкам или столбцам.
TDrawGrid	Создает сетку для отображения графических данных по строкам или столбцам.
TImage	Создает на форме контейнер для отображения битового образа, пиктограммы или метафайла.
TShape	Рисует простые геометрические фигуры.
TBevel	Создает линии и рамки с объемным видом.
TScrollBar	Создает контейнер переменного размера с линейками прокрутки, если это необходимо

Компоненты

- Класс не имеет физической сущности, его ближайшей аналогией является объявление структуры. Память выделяется только тогда, когда класс используется для создания объекта. Этот процесс также называется созданием *экземпляра класса (class instance)*.
- C++Builder вводит понятие *компонент (components)* - специальных классов, свойства которых представляют атрибуты объектов, а их методы реализуют операции над соответствующими экземплярами компонентных классов. Понятие *метод* обычно используется в контексте компонентных классов и внешне не отличается от термина *функция-член* обычного класса. C++Builder позволяет манипулировать видом и функциональным поведением компонент не только с помощью методов (как это делают функции-члены обычных классов), но и посредством свойств и событий, присущих только классам компонент. Работая в среде C++Builder, вы наверняка заметите, что манипулировать с компонентным объектом можно как на стадии проектирования приложения, так и во время его выполнения.

Компоненты (2)

- *Свойства (properties) компонент представляют собой расширение понятия членов данных и хотя не хранят данные как таковые, однако обеспечивают доступ к членам данных объекта.*
- *C++Builder использует ключевое слово **_property** для объявления свойств. При помощи событий (events) компонента сообщает пользователю о том, что на нее оказано некоторое предопределенное воздействие. Основная сфера применения методов в программах, разрабатываемых в среде C++Builder - это обработчики событий (event handlers), которые реализуют реакцию программы на возникновение определенных событий.*
- *Легко заметить некоторое сходство событий и сообщений операционной системы Windows. Типичные простые события —нажатие кнопки или клавиши на клавиатуре. Компоненты инкапсулируют свои свойства, методы и события.*

Компоненты (3)

- На первый взгляд компоненты ничем не отличаются от других объектных классов языка C++, за исключением ряда особенностей, среди которых отметим следующие:
- Большинство компонент представляют собой элементы управления интерфейсом с пользователем, причем некоторые обладают весьма сложным поведением.
- Все компоненты являются прямыми или косвенными потомками одного общего класса-прародителя (TComponent).
- Компоненты обычно используются непосредственно, путем манипуляции с их свойствами; они сами не могут служить базовыми классами для построения новых подклассов.
- Компоненты размещаются только в динамической памяти *кучи (heap)* с помощью оператора **new**, а не на стеке, как объекты обычных классов.
- Свойства компонент включают в себе RTTI - идентификацию динамических типов.
- Компоненты можно добавлять к Палитре компонент и далее манипулировать с ними посредством Редактора форм интегрированной среды визуальной разработки C++Builder.