

Евгений Марков
Владимир Никифоров



Delphi 2005

для .NET

- Архитектура Microsoft .NET
- Языки программирования Delphi и C#
- Windows Forms
- VCL.NET
- ADO.NET, DBP.NET
- ECO

Наиболее
полное
руководство

В ПОДЛИННИКЕ®

Евгений Марков
Владимир Никифоров

Delphi 2005 **для .NET**

Санкт-Петербург

«БХВ-Петербург»

2005

УДК 681.3.068+800.92Delphi2005
ББК 32.973.26-018.1
М27

Марков Е. П., Никифоров В. В.

М27 Delphi 2005 для .NET. — СПб.: БХВ-Петербург, 2005. — 896 с.: ил.
ISBN 5-94157-701-X

Рассмотрены практические аспекты программирования в Borland Delphi 2005 для .NET. Описаны вопросы реализации .NET в Delphi, а также синтаксис и объектные модели двух языков программирования — Delphi и C#. Показаны особенности разработки приложений для двух основных типов приложений — Windows Forms и VCL.NET. Большое внимание уделено созданию приложения для работы с базами данных. Рассмотрены технологии ADO.NET и BDP.NET, а также создание приложений VCL.NET, поддерживающих известные технологии доступа к данным — Borland DataBase Engine .NET, dbExpress .NET, InterBase Express .NET, dbGo и др. Описаны приемы создания приложений на основе технологии ECO, использующей перспективную архитектуру разработки приложений MDA. Излагаемый материал сопровождается примерами.

Для подготовленных программистов

УДК 681.3.068+800.92Delphi2005
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Леонид Кочин</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн обложки	<i>Игоря Цырульникова</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 28.07.05.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 72,24.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.02.953 Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ОАО "Техническая книга"

190005, Санкт-Петербург, Измайловский пр., 29

ISBN 5-94157-701-X

© Марков Е. П., Никифоров В. В., 2005
© Оформление, издательство "БХВ-Петербург", 2005

Оглавление

ЧАСТЬ I. ОСНОВЫ.....	19
Глава 1. Что и как можно разрабатывать в Delphi 2005.....	21
Многоязычная среда разработки.....	21
Язык программирования Delphi.....	22
Язык программирования C#.....	24
Язык программирования Visual Basic.....	24
Программные платформы.....	25
Компоненты .NET и VCL	26
Какие приложения можно создавать в Delphi.....	27
Перенос существующих приложений на платформу .NET.....	29
Резюме.....	30
Глава 2. Введение в архитектуру Microsoft .NET	31
Три вопроса о .NET.....	33
Что это такое?	34
Зачем это нужно?.....	35
Как это работает?.....	37
Сборки, метаданные и промежуточный код	39
Язык MSIL.....	40
Сборка (Assembly)	41
Метаданные.....	43
Особенности разработки приложений .NET	44
Пространства имен.....	45
Общая система типов (Common Type System)	46
Встроенные и определенные пользователем типы данных.....	46
Перечисления	46
Классы.....	46
Интерфейсы	47
Делегаты	47
Указатели.....	47
Массивы	47
Правила межязыкового взаимодействия Common Language Specification	47

.NET Framework	48
Common Language Runtime	48
Компиляторы	51
Безопасность приложения	51
Библиотека базовых классов .NET	52
.NET Framework SDK	52
Высокоуровневые службы	53
ASP.NET	53
ADO.NET	54
Windows Forms	54
Резюме	54
 Глава 3. Язык программирования Delphi	57
Объектно-ориентированное программирование	57
Классы и объекты	57
Поля, свойства и методы	60
События	61
Инкапсуляция	62
Наследование	62
Полиморфизм	63
Методы	65
<i>Abstract</i>	65
<i>Sealed</i>	65
<i>Static</i>	66
<i>Virtual</i> и <i>Dynamic</i>	66
<i>Override</i>	67
Перегрузка методов	69
Области видимости свойств и методов	70
Пространство имен	71
Синтаксис языка Delphi	72
Типы данных	72
Функции преобразования типов	74
Операторы	76
Структурные типы	76
Циклы	78
Обработка исключительных ситуаций	79
Резюме	80
 Глава 4. Язык программирования C#	81
Типы данных	83
Пространство имен	86
Классы	88
<i>Static</i>	89
<i>Virtual</i> и <i>Override</i>	89

<i>Abstract</i>	91
<i>Sealed</i>	92
Конструкторы	93
Синтаксис языка C#	95
Операторы	95
Константы	96
Строки	96
Формат вывода и форматирование строк	96
Массивы	97
Циклы	99
Условные предложения	101
Обработка исключительных ситуаций	103
Ввод/вывод	105
Резюме	108
Глава 5. Реализация .NET в Delphi	109
Общая система типов (Common Type System)	110
Типы данных	110
Классы	111
Интерфейсы	111
Делегаты	112
Правила межязыкового взаимодействия (Common Language Specification)	112
Пространства имен	112
Компиляция в промежуточный язык, сборки, метаданные	116
Управление памятью и сборка мусора	119
Реализация высокоуровневых служб .NET в Delphi	120
Windows Forms	120
ADO.NET	121
ASP.NET	121
Резюме	121
Глава 6. Инструментарий разработчика	123
Интегрированная среда разработки приложений	123
Окно приветствия <i>Welcome Page</i>	124
Палитра инструментов	125
Редактор кода	126
Режим <i>Sync Edit</i>	127
Рефакторинг	128
Контекстная помощь	129
Ошибки	130
Список точек останова	130
Резюме	131

ЧАСТЬ II. ПРИЛОЖЕНИЯ WINDOWS FORMS 133

Глава 7. Приложение и проект 135

Главный модуль проекта.....	135
Файл формы.....	137
Классы <i>Control</i> , <i>UserControl</i> , <i>Form</i>	140
Классы элементов управления (<i>Controls</i>).....	140
Классы компонентов (<i>Components</i>).....	141
Классы диалоговых окон (<i>Common Dialog Boxes</i>).....	141
Описание экземпляра класса.....	141
Резюме.....	146

Глава 8. Элементы управления..... 147

Компонент <i>Label</i>	147
Компонент <i>LinkLabel</i>	149
Компонент <i>TextBox</i>	150
Компонент <i>Button</i>	152
Компонент <i>Panel</i>	154
Компонент <i>CheckBox</i>	155
Компонент <i>RadioButton</i>	158
Компонент <i>ListBox</i>	160
Компонент <i>ComboBox</i>	163
Компонент <i>CheckedListBox</i>	165
Компонент <i>PictureBox</i>	168
Компонент <i>ImageList</i>	170
Компоненты <i>HScrollBar</i> и <i>VScrollBar</i>	171
Компонент <i>NumericUpDown</i>	172
Компонент <i>DomainUpDown</i>	173
Компонент <i>DateTimePicker</i>	175
Компонент <i>MonthCalendar</i>	177
Компонент <i>Timer</i>	178
Резюме.....	180

Глава 9. Стандартные программные механизмы..... 181

Интерфейс переноса Drag and Drop.....	181
Усовершенствованное масштабирование.....	188
Управление мышью.....	189
Резюме.....	190

Глава 10. Меню и панель инструментов..... 191

Компонент <i>MainMenu</i>	191
Компонент <i>ContextMenu</i>	194
Компонент <i>ToolBar</i>	197
Резюме.....	201

Глава 11. Диалоги.....	203
Стандартные компоненты диалога	203
Компонент <i>OpenFileDialog</i>	204
Компонент <i>SaveFileDialog</i>	207
Компоненты <i>PrintDialog</i> , <i>PrintDocument</i> , <i>PageSetupDialog</i> и <i>PrintPreviewDialog</i>	209
Компонент <i>FontDialog</i>	213
Компонент <i>ColorDialog</i>	214
Резюме.....	215
Глава 12. Состояние приложения	217
Компонент <i>StatusBar</i>	217
Компонент <i>ProgressBar</i>	220
Компонент <i>TrackBar</i>	222
Компонент <i>ToolTip</i>	224
Компонент <i>NotifyIcon</i>	224
Компонент <i>HelpProvider</i>	226
Компонент <i>ErrorProvider</i>	228
Резюме.....	230
Глава 13. Ввод данных	231
Ввод и обработка текста	231
Класс <i>Font</i>	231
Компонент <i>TextBox</i>	232
Компоненты <i>ComboBox</i> и <i>ListBox</i>	234
Компонент <i>DomainUpDown</i>	235
Компонент <i>RichTextBox</i>	235
Ввод данных в числовых форматах	238
Ввод даты и времени.....	239
Компонент <i>MonthCalendar</i>	240
Компонент <i>DateTimePicker</i>	242
Ввод двоичных данных	245
Резюме.....	247
Глава 14. Работа с файлами.....	249
Файл как объект файловой системы	249
Класс <i>File</i>	250
Класс <i>FileInfo</i>	255
Пути и каталоги	257
Класс <i>Directory</i>	258
Класс <i>DirectoryInfo</i>	260
Поиск файла	262
Потоки	264
Класс <i>StreamReader</i>	264

Класс <i>StreamWriter</i>	266
Класс <i>FileStream</i>	268
Асинхронный режим доступа к данным	270
Класс <i>MemoryStream</i>	272
Операции ввода/вывода	273
Создание файла и запись данных	273
Открытие файла и чтение данных	275
Резюме	276
Глава 15. Перечислители, списки, коллекции	277
Что такое коллекция	278
Как устроена коллекция	279
Интерфейс <i>ICollection</i>	280
Интерфейс <i>IList</i>	280
Интерфейс <i>IEnumerable</i>	282
Интерфейс <i>IEnumerator</i>	282
Класс <i>CollectionBase</i>	283
Коллекция строк	283
Управление коллекциями	284
Резюме	285
Глава 16. Иерархическое представление данных	287
Компонент <i>TreeView</i>	288
Класс <i>TreeNode</i>	293
Компонент <i>ListView</i>	296
Класс <i>ListViewItem</i>	302
Класс <i>ListViewSubItem</i>	304
Резюме	304
Глава 17. Использование XML	305
Что такое XML	305
Основы синтаксиса XML	308
Пролог	310
Определение	311
Тело документа. Корневой элемент	312
Объектная модель документа	313
Интерфейсы семейства <i>IDOMNode</i>	314
Свойства <i>nodeType</i> , <i>nodeName</i> и <i>nodeValue</i>	316
Свойства и методы, управляющие другими вершинами	317
Пространства имен	319
Интерфейс <i>IDOMDocument</i>	320
Пример создания приложения, использующего модель DOM	320
Реализация модели DOM в приложениях .NET	322
Класс <i>XmlNode</i>	323
Класс <i>XmlElement</i>	328

Класс <i>XMLAttribute</i>	329
Класс <i>XMLDocument</i>	329
Резюме	334

ЧАСТЬ III. ПРИЛОЖЕНИЯ VCL.NET 335

Глава 18. Приложение и проект 337

Проект как основа разработки приложения	337
Класс <i>TApplication</i>	345
Атрибуты приложения	351
Обработка сообщений	352
Реакция на действия пользователей	355
Система помощи	356
Резюме	357

Глава 19. Меню и действия 359

Редактор меню	360
Как работает меню	361
Главное меню приложения	364
Всплывающее меню	367
Действия. Компонент <i>TActionList</i>	368
События, связанные с действиями	369
Свойства, распространяемые на клиентов действия	371
Прочие свойства	372
Стандартные действия	373
Категория <i>Edit</i>	376
Категория <i>Search</i>	376
Категория <i>Help</i>	376
Категория <i>File</i>	377
Категория <i>Dialog</i>	377
Категория <i>Window</i>	377
Категория <i>Tab</i>	377
Категория <i>List</i>	377
Категория <i>Internet</i>	379
Категория <i>Format</i>	380
Категория <i>Dataset</i>	380
Категория <i>Tools</i>	380
Компонент <i>TActionManager</i>	381
Изменение и настройка внешнего вида панелей	383
Ручное редактирование коллекций панелей и действий	384
Резюме	387

Глава 20. Списки и коллекции 389

Список строк	390
Класс <i>TStrings</i>	390
Класс <i>TStringList</i>	391

Список указателей.....	399
Класс <i>TList</i>	399
Пример использования списка указателей.....	402
Коллекции	406
Класс <i>TCollection</i>	407
Класс <i>TCollectionItem</i>	408
Резюме.....	408
 Глава 21. Файлы и потоки.....	411
Потоки	411
Базовые классы <i>TStream</i> и <i>THandleStream</i>	412
Класс <i>TFileStream</i>	414
Класс <i>TMemoryStream</i>	416
Класс <i>TStringStream</i>	416
Резюме.....	417
 Глава 22. Использование графики.....	419
Графические инструменты Delphi.....	419
Класс <i>TFont</i>	419
Класс <i>TPen</i>	421
Класс <i>TBrush</i>	422
Класс <i>TCanvas</i>	422
Класс <i>TGraphic</i>	427
Класс <i>TPicture</i>	429
Класс <i>TMetafile</i>	431
Класс <i>TIcon</i>	432
Класс <i>TBitmap</i>	433
Компонент <i>TImage</i>	435
Использование диалогов для загрузки и сохранения графических файлов	436
Класс <i>TClipboard</i>	437
Класс <i>TScreen</i>	439
Резюме.....	441
 ЧАСТЬ IV. ПРИЛОЖЕНИЯ БАЗ ДАННЫХ .NET.....	443
 Глава 23. Архитектура приложений баз данных .NET	445
Как работает приложение баз данных	446
Соединение с источником данных.....	450
Адаптер данных	450
Набор данных	451
Отображение данных	452
Методика доступа к данным в приложении БД.....	452
Резюме.....	453

Глава 24. Приложения ADO.NET.....	455
Основы ADO	457
Провайдеры ADO	459
Соединение с источником данных.....	460
Пулинг соединений.....	464
Управление транзакциями	465
Обработка ошибок.....	470
Использование адаптера данных.....	472
Отбор данных и генерация набора данных	472
Выборка из одной таблицы.....	476
Вставка, изменение, удаление данных командами SQL.....	477
Схема связывания данных	480
Параметры.....	483
Набор данных	489
Таблицы данных	491
Колонка таблицы	495
Автоинкрементные колонки	498
Вычисляемые колонки.....	499
Агрегатные колонки.....	501
Фильтрация и поиск данных	502
Первичный ключ таблицы	503
Запись таблицы	504
Управление данными	510
Сортировка, поиск данных	513
Ограничения.....	515
Отношения	519
Просмотры	521
Команды SQL.....	525
Пользовательский интерфейс	530
Компоненты <i>Label</i> и <i>LinkLabel</i>	532
Компонент <i>Button</i>	532
Компонент <i>TextBox</i>	532
Компонент <i>CheckBox</i>	532
Компонент <i>RadioButton</i>	533
Компонент <i>ComboBox</i>	533
Компонент <i>ListBox</i>	533
Компонент <i>CheckedListBox</i>	534
Компоненты <i>TreeView</i> и <i>ListView</i>	534
Компонент <i>DateTimePicker</i>	534
Компоненты <i>TrackBar</i> , <i>ProgressBar</i> , <i>VScrollBar</i> и <i>HScrollBar</i>	534
Компонент <i>NumericUpDown</i>	534
Компонент <i>DomainUpDown</i>	535
Компоненты <i>GroupBox</i> и <i>StatusBar</i>	535
Компонент <i>RichTextBox</i>	535

Компонент <i>DataGrid</i>	535
Подключение данных	535
Отображение данных	537
Навигация по записям таблицы	538
Работа с ячейками таблицы	538
Сортировка данных.....	539
Отображение отношений между таблицами.....	539
Резюме.....	540
Глава 25. Приложения BDP	543
Доступ к данным	544
Механизм отображения "живых" данных	547
Компонент <i>BdpCommandBuilder</i>	548
Перенос данных между разными источниками данных.....	549
Работа с гетерогенными источниками данных	551
Обмен гетерогенными данными.....	554
Многотабличный набор данных.....	555
Работа с удаленными источниками данных	556
Удаленный сервер приложения	558
Клиентское приложение.....	560
Пример разработки распределенного приложения	562
Резюме.....	563
ЧАСТЬ V. ПРИЛОЖЕНИЯ БАЗ ДАННЫХ VCL.NET	565
Глава 26. Архитектура приложений баз данных VCL.NET	567
Набор данных	568
Абстрактный набор данных	568
Стандартные компоненты	574
Компонент таблицы	575
Компонент запроса	577
Компонент хранимой процедуры	580
Индексы в наборе данных.....	581
Механизм подключения индексов	582
Список описаний индексов	582
Описание индекса	583
Использование описаний индексов	584
Параметры запросов и хранимых процедур.....	586
Класс <i>TParams</i>	589
Класс <i>TParam</i>	590
Состояния набора данных.....	592
Поля	595
Объекты полей.....	595
Статические и динамические поля.....	598
Класс <i>TField</i>	600

Виды полей	604
Поля синхронного просмотра	605
Вычисляемые поля	607
Внутренние вычисляемые поля	608
Агрегатные поля	608
Объектные поля	609
Ограничения	610
Как работает приложение баз данных	612
Модуль данных	615
Подключение набора данных	616
Настройка компонента <i>TDataSource</i>	617
Отображение данных	619
Резюме	620
Глава 27. Процессор баз данных BDE.NET	621
Архитектура и функции BDE	622
Псевдонимы баз данных и настройка BDE	626
Соединение с источником данных	635
Компоненты доступа к данным	640
Класс <i>TBDEDataSet</i>	640
Класс <i>TDBDataSet</i>	645
Компонент <i>TTable</i>	646
Компонент <i>TQuery</i>	652
Компонент <i>TStoredProc</i>	654
Резюме	656
Глава 28. Технология dbExpress .NET	657
Доступ к данным dbExpress	658
Драйверы доступа к данным	659
Соединение с сервером баз данных	660
Управление наборами данных	665
Транзакции	668
Использование компонентов наборов данных	669
Класс <i>TCustomSQLDataSet</i>	670
Компонент <i>TSQLDataSet</i>	672
Компонент <i>TSQLTable</i>	673
Компонент <i>TSQLQuery</i>	674
Компонент <i>TSQLStoredProc</i>	675
Компонент <i>TSimpleDataSet</i>	676
Способы редактирования данных	679
Интерфейсы dbExpress	685
Интерфейс <i>ISQLDriver</i>	685
Интерфейс <i>ISQLConnection</i>	685
Интерфейс <i>ISQLCommand</i>	687
Интерфейс <i>ISQLCursor</i>	688

Отладка приложений с технологией dbExpress	689
Распространение приложений с технологией dbExpress.....	691
Резюме	692
Глава 29. Технология InterBase Express для .NET	693
Механизм доступа к данным InterBase Express	694
Компонент <i>TIBDatabase</i>	694
Компонент <i>TIBTransaction</i>	699
Компоненты доступа к данным.....	703
Область дескрипторов <i>XSQLDA</i>	705
Структура <i>XSQLVAR</i>	706
Компонент <i>TIBTable</i>	707
Компонент <i>TIBQuery</i>	708
Компонент <i>TIBStoredProc</i>	710
Компонент <i>TIBDataSet</i>	710
Компонент <i>TIBSQL</i>	712
Обработка событий	715
Информация о состоянии базы данных	717
Компонент <i>TIBDatabaseInfo</i>	717
Компонент <i>TIBSQLMonitor</i>	719
Резюме	720
Глава 30. Технология dbGo	721
Компоненты dbGo	721
Механизм соединения с хранилищем данных ADO	722
Компонент <i>TADODConnection</i>	723
Настройка соединения	723
Управление соединением.....	728
Доступ к связанным наборам данных и командам ADO.....	731
Объект ошибок ADO	734
Транзакции.....	734
Наборы данных ADO	735
Класс <i>TCustomADODDataSet</i>	736
Набор данных	736
Курсор набора данных.....	737
Локальный буфер	739
Состояние записи	740
Фильтрация.....	742
Поиск.....	743
Сортировка	743
Команда ADO	744
Групповые операции.....	745
Параметры.....	746
Класс <i>TParameters</i>	747
Класс <i>TParameter</i>	748

Компонент <i>TADODataSet</i>	749
Компонент <i>TADOTable</i>	750
Компонент <i>TADOQuery</i>	751
Компонент <i>TADOStoredProc</i>	751
Команды ADO	752
Объект ошибок ADO	754
Резюме.....	754

ЧАСТЬ VI. РАСПРЕДЕЛЕННЫЕ ПРИЛОЖЕНИЯ БАЗ ДАННЫХ VCL.NET..... 757

Глава 31. Архитектура распределенных приложений 759

Парадигма распределенных вычислений	760
Архитектура распределенных приложений	763
Уровень представления данных	767
Уровень обработки данных	768
Уровень управления данными.....	768
Уровень хранения данных	769
Расширения базовых уровней	770
Уровень бизнес-интерфейса	771
Уровень доступа к данным	771
Резюме.....	771

Глава 32. Технология DataSnap..... 773

Структура многозвенного приложения в Delphi.....	774
Трехзвенное приложение в Delphi 2005	776
Сервер приложения	777
Клиентское приложение.....	778
Механизм удаленного доступа к данным DataSnap.....	779
Компонент <i>TDCOMConnection</i>	779
Вспомогательные компоненты-брокеры соединений.....	781
Компонент <i>TSimpleObjectBroker</i>	781
Компонент <i>TLocalConnection</i>	783
Компонент <i>TSharedConnection</i>	783
Компонент <i>TConnectionBroker</i>	784
Резюме.....	785

Глава 33. Клиент многозвенного распределенного приложения..... 787

Структура клиентского приложения.....	788
Компонент <i>TClientDataSet</i>	789
Получение данных от компонента-провайдера.....	790
Кэширование и редактирование данных	792
Управление запросом на сервере	794
Использование индексов.....	795

Сохранение набора данных в файлах.....	797
Работа с данными типа BLOB	798
Представление данных в формате XML	799
Агрегаты	799
Объекты-агрегаты.....	800
Агрегатные поля	802
Группировка и использование индексов	804
Вложенные наборы данных.....	804
Дополнительные свойства полей клиентского набора данных	805
Обработка ошибок.....	806
Пример "тонкого" клиента	809
Соединение клиента с сервером приложения	814
Наборы данных клиентского приложения.....	815
Сервер приложения	816
Резюме.....	818
Глава 34. Преобразование пакетов данных в формате XML.....	819
Преобразование данных в формате XML.....	819
Схема преобразования данных XML	820
Формат пакета данных Delphi	821
Утилита XML Mapper	822
Выбор исходного файла.....	823
Создание пакета данных и документа XML и сохранение преобразованных данных	824
Связывание элементов XML и полей пакета данных.....	825
Создание трансформационного файла и преобразование данных	826
Резюме.....	827
ЧАСТЬ VII. ПРИЛОЖЕНИЯ ЕСО.....	829
Глава 35. Архитектура MDA	831
Основные понятия.....	832
Архитектура разработки приложений на основе моделей.....	833
Типы моделей	834
Уровни модели.....	834
Этапы разработки.....	835
Преобразование модели PIM в PSM.....	837
Многоплатформенные модели	838
Технологический фундамент.....	839
Что нужно знать об UML.....	839
OCL	841
Стандарты метамоделирования.....	841
XML и XMI	841
Резюме.....	842

Глава 36. Технология ЕСО	843
Что такое ЕСО	844
Проект ЕСО.....	845
Инструментарий ЕСО	850
Менеджер модели <i>Model View</i>	850
Редактор UML.....	851
Дизайнер объектного пространства	855
Общая методика разработки приложений ЕСО.....	857
Платформенно-независимая модель	858
Пакет	859
Класс	859
Атрибут	861
Оператор	862
Отношения	862
Ассоциация	863
Обобщение\вплощение	865
Платформенно-зависимая модель.....	865
Объектное пространство.....	866
Класс объектного пространства	867
Общие управляющие компоненты-дескрипторы.....	868
Компонент <i>ReferenceHandle</i>	871
Компонент <i>VariableHandle</i>	872
Компонент <i>ExpressionHandle</i>	873
Компонент <i>OclVariables</i>	874
Компонент <i>OclPSHandle</i>	874
Пользовательский интерфейс	875
Автоматическая генерация форм	876
Управление данными	877
Управление списками.....	878
Drag and Drop	879
Связывание визуальных компонентов с данными.....	879
Доступ к данным	880
Использование языка OCL	882
Пример приложения ЕСО	883
Вычислительно-независимая модель.....	883
Платформенно-независимая модель	884
Платформенно-зависимая модель.....	887
Резюме.....	889
Предметный указатель.....	891



ЧАСТЬ I

ОСНОВЫ

- Глава 1.** Что и как можно разрабатывать в Delphi 2005
- Глава 2.** Введение в архитектуру Microsoft .NET
- Глава 3.** Язык программирования Delphi
- Глава 4.** Язык программирования C#
- Глава 5.** Реализация .NET в Delphi
- Глава 6.** Инструментарий разработчика

ГЛАВА 1



Что и как можно разрабатывать в Delphi 2005

Если задуматься, какой девиз лучше всего отражает сущность среды разработки Delphi 2005, то, по нашему мнению, это будет девиз "универсальность". Конечно же, главным новшеством последних двух версий Delphi является возможность разработки приложений для операционной среды .NET корпорации Microsoft. Однако, если не полениться и выполнить команду меню **Help | About Borland Developer Studio**, то мы увидим, что официальное название программы — Borland Developer Studio for Microsoft Windows. Это означает, что здесь можно создавать приложения для любых операционных систем Windows, широко используемых в настоящее время.

И действительно, в Delphi 2005 можно программировать для Win32 и .NET, вести разработку на двух языках программирования и при этом создавать приложения многих типов (базы данных, ASP.NET, консольные приложения и т. д.). Кроме этого, здесь можно работать и с приложениями Visual Basic для .NET.

Две операционные среды, несколько языков программирования, несколько типов приложений, — сколько получится сочетаний из N по 3? Чтобы ответить на этот вопрос, написана данная глава. Здесь, прежде чем погрузиться в детали программирования в Delphi 2005, мы разберемся, что и как можно разрабатывать в Delphi.

Многоязычная среда разработки

Основа знаний и опыта любого разработчика — языки программирования. В наше время изучение языков Basic, Pascal или C — это всего лишь начальный этап, первая ступенька к освоению инструментариев и технологий программирования. Поэтому разговор о возможностях Delphi 2005 мы начнем с языков программирования.

Некоторая универсальность и известное совершенство современных высокоуровневых языков программирования является следствием долгого (порядка полувека) развития начиная с первых, мало отличавшихся от наборов машинных кодов, средств взаимодействия человека и компьютера. При этом генеалогия некоторых языков весьма и весьма прихотлива. Например, язык программирования Delphi является дальним потомком небольшого семейства языков (Eiffel, Pascal), разработанных исключительно для обучения студентов программированию. Всего же человечество придумало больше тысячи различных языков программирования.

Столь прихотливая история развития и разнообразие языков для разработчиков объясняются двумя причинами. Первая причина созидательная — это стремление создать мощное и в то же время универсальное языковое средство, способное быстро и эффективно решать подавляющее большинство задач программирования. Вторая причина деструктивная — бурное развитие вычислительной техники, появление новых технологий программирования и направлений приложения усилий программистов. Ученые создавали очередной суперсовременный и универсальнейший язык, но еще более современные архитектуры и технологии быстро вытесняли его на периферию.

Примерно такая же ситуация повторилась несколько лет назад, следствием чего и стали многоязычные возможности Delphi 2005. К началу третьего тысячелетия признанными "вождями народа" языков программирования были Visual Basic и C++, — около 90 % разработчиков использовали именно их. Язык же программирования Delphi, выступающий в одной весовой категории с C++ и ничем ему не уступающий, увы, распространен не так широко.

Но появление новой операционной среды Microsoft .NET и, как следствие, развитие новых технологий программирования, показало, что "короли" оказались не готовыми к новым веяниям времени. На сцене появился новый язык программирования — C#.

Именно стремление к универсальности, желание дать разработчику инструментарий для решения наибольшего числа задач, заставило создателей Delphi 2005 обеспечить поддержку нескольких языков программирования. Для этого потребовалось создать и включить в состав среды разработки новые компиляторы для языков C# и Visual Basic.

Итак, при работе в Delphi 2005 разработчику доступны три языка программирования: Delphi, C# и Visual Basic. У каждого из них есть свои преимущества, недостатки и предпочтительные сферы применения. Рассмотрим их.

Язык программирования Delphi

Язык программирования Delphi исторически является основным для среды разработки Delphi, он и его ближайшие предки применялись в Delphi начиная

с первой версии. Во избежание путаницы здесь следует подробнее обсудить само название языка. Официальное название основного языка программирования среды разработки Delphi 2005 — Delphi. Оно употребляется начиная с версии Delphi 6. До этого язык программирования в среде разработки Delphi назывался Object Pascal. И хотя при выходе Delphi 6 фирма Borland четко и ясно объявила об изменении названия языка, до сих пор в некоторых публикациях в периодических изданиях и даже книгах встречается старое название языка — Object Pascal.

Что же подвигло разработчиков из Borland на переименование?

Действительно, ближайшим предком (папой) современного языка Delphi является Object Pascal, а дедушкой его стал старый, добрый классический Pascal. От своих предков язык Delphi унаследовал базовые конструкции, строгую типизацию, четкий и приближенный к человеческому синтаксис.

Но название языка было изменено отнюдь не случайно.

Дело в том, что бурное развитие объектно-ориентированного программирования (ООП) и визуального компонентного программирования, которые стали одними из важных причин появления среды разработки Delphi, потребовало существенного развития и языковых средств. И на основе языка Pascal был разработан Object Pascal. Он поддерживал работу с объектами и соответствовал парадигме ООП.

Но технологии программирования продолжали развиваться. Повсеместное внедрение технологий, основанных на COM (Component Object Model, компонентная модель объектов), в очередной раз потребовало адаптации языков программирования к новым условиям. Язык Object Pascal был дополнен интерфейсами, объекты были заменены на классы, появилось множественное наследование и т. д. Основой языка стали не объекты, а классы и интерфейсы. Поэтому Object Pascal был переименован в Delphi.

Язык программирования Delphi может быть использован для разработки приложений на операционных платформах Win32 и .NET. На нем реализована библиотека визуальных компонентов Delphi (Visual Component Library, VCL). Поэтому разработчики, продолжающие работу под Win32, могут и дальше создавать приложения, используя отработанные навыки и знания.

На основе проектов на языке Delphi возможна компиляция и для платформы .NET. Это позволяет сравнительно легко перенести приложения для Win32 на новую платформу .NET, продолжая применять привычные компоненты VCL, знакомые процедуры и функции.

В то же время при помощи языка Delphi можно создавать и полноценные приложения для .NET. Для этого в состав языка включены специфические конструкции, понятия, типы данных, необходимые для работы в .NET.

А вместо библиотеки компонентов VCL имеется набор элементов управления, созданных специально для .NET и соответствующих стандартам .NET.

Подробно язык программирования Delphi рассматривается в *главе 3*.

Язык программирования C#

Язык программирования C# был разработан как средство программирования для операционной среды NET.

Авторы языка — Скотт Вилтамот (Scott Wiltamuth) и Андерс Хейлсберг (Anders Hejlsberg).

Синтаксис и возможности C# оптимизированы для работы с интерпретируемым кодом, в языке присутствуют конструкции, пригодные к использованию только при программировании для .NET. Например, язык поддерживает управление памятью в соответствии со стандартами .NET и обеспечивает сборку мусора. В синтаксическом наборе языка появились понятия сборки, пространства имен и т. д.

C# разработан на базе языков C++ и Java. Основной синтаксис, структура языка и модель типов данных свидетельствуют об этом весьма красноречиво. Но довольно существенная часть наследственных черт языков семейства C из нового языка исчезла. В C# больше нет некоторых синтаксических элементов (например, оператора "::<") и возможностей (например, опережающего объявления переменных).

В то же время несомненно и влияние на новый язык семейства языков Pascal. Поэтому исходный код на языке C# гораздо больше приближен к человеческому языку и лучше пригоден для восприятия.

В среде разработки Delphi проекты на языке C# можно создавать только для операционной среды .NET.

Подробно язык программирования C# рассматривается в *главе 4*.

Язык программирования Visual Basic

Язык программирования Visual Basic — заслуженный ветеран объектно-ориентированного и компонентного программирования, известный большинству программистов на планете Земля. Его популярность и широкое распространение объясняются простотой изучения и использования. Но его возможности сравнительно невелики.

Если представить создаваемые программистами приложения в виде пирамиды по критерию сложности, то основание пирамиды займут программы, написанные на Visual Basic. Чем дальше мы будем подниматься к вершине, тем

меньше будет встречаться Visual Basic, и тем чаще будут попадаться проекты на C++, C# и Delphi.

В сложных проектах Visual Basic также применяется, но для решения ограниченных, специализированных задач. Например, для написания частей распределенных приложений для Internet, Web-служб или тонких клиентов. Именно для таких целей в среде разработки Delphi 2005 можно создавать ограниченные по возможностям приложения на Visual Basic, но при этом нельзя применять визуальное программирование.

Программные платформы

Начиная с Delphi 8 разработчики могут создавать приложения для двух программных платформ (операционных сред): Win32 и .NET.

Традиционная платформа Win32 обеспечивает выполнение приложений, откомпилированных в двоичный машинный код, жестко привязанный к архитектуре процессора. Это обеспечивает повышенное быстродействие и эффективность программ, особенно на стадии загрузки. Ведь уже готовому машинному коду достаточно выделить область памяти, загрузить его и обеспечить общее управление операционной системой на уровне процесса.

Недостаток Win32 — жесткая зависимость от программно-аппаратной платформы и принципиальная ограниченность средств управления программным кодом. Очевидно, что однажды откомпилированный машинный код программы с точки зрения операционной системы представляет собой черный ящик со стандартизованными каналами (интерфейсами) управления и обмена данными. Управлять таким "ящиком" можно только в целом: поместить в память, выделить для работы дополнительный участок памяти, разрешить обращение к драйверам, выгрузить из памяти.

Новая платформа .NET ориентирована на 64-разрядную архитектуру процессоров, обеспечивает выполнение приложений, интерпретированных в специальный, универсальный, промежуточный язык — Intermediate Language (IL). Среда .NET выполняет первоначальную загрузку интерпретированного кода приложения, компиляцию в машинный код и управление. Такая архитектура обеспечивает независимость программ от аппаратной части компьютера и гораздо более эффективное управление, чем Win32.

К недостаткам формально следует отнести большие затраты на загрузку и управление приложением. Однако на современном этапе развития компьютерной техники этот недостаток нивелируется достаточными вычислительными мощностями. Кроме того, задержка выполнения приложения реальна только на этапе первоначальной загрузки. После компиляции в машинный код быстродействие программ .NET практически идентично Win32.

Ранее мы уже говорили о том, что в Delphi 2005 можно разрабатывать проекты на нескольких языках программирования. Основными из них являются Delphi и C#. При этом не все языки программирования одинаково универсальны. К примеру, C# специально создан для .NET, а Delphi имеет языковые подмножества для поддержки обеих платформ. Возможности Delphi 2005 по применению языков программирования для различных программных платформ представлены в табл. 1.1.

Таблица 1.1. *Применимость языков программирования к программным платформам*

	Платформа Win32	Платформа .NET
Delphi	+	+
C#	-	+

Подробно операционная среда .NET рассматривается в *главе 2*.

Компоненты .NET и VCL

Помимо языков программирования и операционных сред программист, работающий в Delphi 2005, должен учитывать и используемые здесь наборы визуальных и невидимых компонентов.

В Delphi 2005 вы можете разрабатывать приложения VCL для платформы Win32 на языке программирования Delphi (табл. 1.2).

Для программной платформы Win32 доступна библиотека компонентов VCL. Она очень разнообразна и обладает существенно большими возможностями, чем набор компонентов для .NET. И это естественно: VCL существует гораздо дольше и не так сильно связана ограничениями на совместимость, как библиотека компонентов .NET. С другой стороны, часть компонентов VCL в .NET просто не нужна, т. к. соответствующие функции реализованы в .NET на уровне платформы. Например, не менее трети компонентов VCL в Delphi 7 и Delphi 2005 (компоненты IntraWeb и другие) обеспечивают функции, полноценно представленные в .NET виде приложений ASP.NET.

В Delphi 2005 вы можете разрабатывать приложения VCL для платформы .NET на языке программирования Delphi (табл. 1.2).

Переработанная библиотека VCL для платформы .NET в Delphi 2005 получила название VCL.NET. Она позволяет задействовать весь наработанный ранее богатый инструментарий и опыт разработчиков. Благодаря VCL.NET перенос приложений VCL на платформу .NET требует минимальных усилий. Боль-

шинство стандартных компонентов VCL и компонентов для приложений баз данных есть и в VCL.NET. Однако многие применяемые ранее технологии для разработки Internet-приложений (например, WebBroker) стали вспомогательными. Их необходимо заменять компонентами IntraWeb.

В Delphi 2005 вы можете разрабатывать приложения для платформы .NET на языках программирования Delphi и C# (табл. 1.2).

Для "чистых" приложений .NET имеется собственный набор компонентов. В первую очередь, это визуальные компоненты Windows Forms, специфицированные на уровне архитектуры .NET. Другие компоненты .NET обеспечивают создание меню, диалоги, доступ к данным и т. д.

Таблица 1.2. Возможности использования наборов компонентов в приложениях Delphi 2005

	Платформа Win32		Платформа .NET	
	VCL	.NET	VCL	.NET
Delphi	+	—	+ (VCL.NET)	+
C#	—	—	—	+

Какие приложения можно создавать в Delphi

И в завершение главы мы обратимся к видам приложений, которые можно разрабатывать в Delphi 2005. Надо сказать, что разобраться в том, какие программы можно писать в Delphi 2005, не так просто. Чтобы разъяснить читателю все открывающиеся перед ним возможности и попытаться структурировать их более или менее четко, потребовалась отдельная глава! Увы, это обратная сторона медали, на лицевой стороне которой выгравирован девиз "универсальность" (см. начало этой главы). В общем-то, сложная и довольно пестрая картина возможностей, предлагаемых разработчикам в Delphi 2005, — есть объективное отражение состояния современного программирования. Сейчас мы находимся на переломе: происходит переход на новые операционные системы, новую архитектуру выполнения приложений, новые языки программирования, новые технологии и приемы работы.

В этой ситуации надо радоваться, что у нас есть универсальная среда программирования, которая обеспечивает работу и для Win32, и для .NET и переход с одной платформы на другой. А неизбежная сложность и многообразие возможностей является лишь отражением реальных процессов.

Давайте попробуем собрать полученные знания в целостную картину того, что и как можно разрабатывать в Delphi 2005. В табл. 1.3 сведены виды приложений, доступные для использования, и дана их взаимосвязь с программными платформами и наборами компонентов. А затем дано краткое описание видов приложений.

Таблица 1.3. Возможности программных платформ Win32 и .NET по созданию различных видов приложений

	Платформа Win32	Платформа .NET	
	Язык Delphi	Язык Delphi	Язык C#
Приложения VCL	+	+	—
Приложения VCL.NET	—	+	—
Приложения Windows Forms	—	+	+
Приложения ADO.NET	—	+	+
Приложения баз данных	+	—	—
Приложения ASP.NET	—	+	+
Приложения IntraWeb	+	—	—
Приложения WebBroker	+	—	—
Приложения WebSnap	+	—	—
Приложения COM	+	+	+
Приложения ECO	—	+	+
Консольные приложения	+	+	+

Под приложениями VCL, VCL.NET и Windows Forms подразумеваются лишь способы создания пользовательского интерфейса и базовой функциональности любых приложений Windows, которые зависят от выбранного набора компонентов (см. разд. "Компоненты .NET и VCL" в этой главе). Их выделение в отдельные виды в табл. 1.3 связано с тем, что эти библиотеки являются краеугольными камнями построения приложений на обеих программных платформах.

- ❑ **Приложения ADO.NET** — приложения баз данных, разрабатываются исключительно на платформе .NET и обеспечивают доступ к данным только на основе технологии ADO (ActiveX Data Objects), включенной в состав архитектуры .NET. Декларируется, что ADO является основным способом доступа к данным в .NET и именно поэтому такие приложения выделены в группу, отдельную от приложений баз данных.

- ❑ **Другие приложения баз данных** — объединяют все возможные в Delphi способы доступа к данным, за исключением ADO.NET. Например, доступ к данным через технологии dbGo, dbExpress и т. д. в Win32. Здесь необходимо отметить, что технология доступа к данным реализуется через наборы невидимых компонентов. В связи с этим нужно понимать, что доступ к данным на платформе .NET при помощи технологии ADO реализуется через ADO.NET, а такой же доступ к данным при помощи той же технологии ADO на платформе Win32 реализуется через dbGo. Просто это разные наборы компонентов, которые созданы каждый для своей программной платформы, но используют один и тот же механизм доступа к данным ADO, реализованный в операционной системе.
- ❑ **Приложения ASP.NET** — специализированный вид приложений для Internet, которые можно создавать исключительно на платформе .NET. Подробно разработка этих приложений рассматривается в *части V*.
- ❑ **Приложения IntraWeb, WebBroker и WebSnap** — объединены тем, что все они обеспечивают разработку Internet-приложений (Web-приложений и Web-служб) на программной платформе Win32. Их функциональные возможности значительно перекрываются технологией ASP.NET.
- ❑ **Приложения COM** — создаются в виде сервера или объекта COM, используют интерфейсы COM и при необходимости интерфейсы других технологий, дочерних COM. В терминологии .NET приложения COM называются приложениями с неуправляемым кодом, т. к. возможность их применения предусмотрена в архитектуре .NET, а реализованы эти приложения могут быть и на программной платформе Win32.
- ❑ **Приложения ECO (Enterprise Core Objects)** — создаются на основе возможностей архитектуры MDA (Model Driven Architecture). Такие приложения создаются с помощью модели на языке UML, на базе которой автоматически генерируется весь необходимый исходный код приложения. Подробности работы с технологиями ECO и MDA раскрываются в *части VI*.
- ❑ **Консольные приложения** — в Delphi 2005 можно разрабатывать для любой программной платформы и любого языка программирования. Эти приложения выполняются и управляются из командной строки, без использования привычного визуального пользовательского интерфейса.

Перенос существующих приложений на платформу .NET

Обсуждая возможности, предоставляемые программистам средой разработки Delphi 2005, представляется необходимым разобраться и с проблемой переноса на новую платформу уже разработанных приложений. С языком про-

граммирования все в целом понятно — Delphi не требует бросить все и начать изучение C#. Язык программирования кардинальных изменений не терпел, появились новые сущности, необходимые для работы с .NET, но это не влияет на уже существующий код для Win32.

При переносе приложений VCL с платформы Win32 на VCL.NET больших трудностей возникнуть не должно. Основные компоненты (кроме специализированных компонентов для Web-приложений и Web-служб) реализованы и в VCL.NET. Нужно будет также убрать из кода переменные, типы данных которых напрямую адресуют память, т. к. в .NET такой код считается небезопасным. Кроме того, есть еще некоторые особенности, которые подробнее рассматриваются в *главе 3*. И конечно, в секциях `uses` нужно будет изменить имена модулей на соответствующие пространства имен.

Что же касается internet-приложений, то здесь придется потрудиться. Многие компоненты, которые использовались в таких приложениях, в Win32-версиях Delphi, удалены из палитры компонентов. Для приложений VCL.NET нужно выбирать компоненты IntraWeb, а для "чистой" платформы .NET — начинать разработку заново на ASP.NET. Подробнее о переносе internet-приложений на платформу .NET рассказывается в *части V*.

При переносе приложений VCL с платформы Win32 на Windows Forms .NET неизбежны серьезные затруднения. Это касается не только компонентов VCL (в Windows Forms .NET их просто нет) и типов данных, но и собственно кода и бизнес-логики приложения. Привычные приемы работы со списками, файловой системой, памятью в .NET стали непригодными. Вы даже не сможете вывести на экран окно сообщения с сакраментальной фразой "Hello, World!", т. к. в .NET процедуры `ShowMessage` просто нет. Поэтому приложения Windows Forms для .NET лучше всего начинать писать заново.

Резюме

В этой главе кратко рассказывается о разнообразных возможностях разработки приложений в Delphi 2005. Их большое многообразие объясняется тем, что Delphi 2005 выпущена в сложный период продвижения новой архитектуры .NET, когда необходимо поддерживать разработку как традиционных приложений, так и новых, а также обеспечивать миграцию приложений с платформы на платформу.

Многоязычность среды позволяет выполнять разработки на языках программирования Delphi, C# и Visual Basic.

В Delphi 2005 поддерживаются две программные платформы (операционные среды): .NET и Win32. Это обуславливает большое разнообразие видов приложений, доступных разработчикам.

ГЛАВА 2



Введение в архитектуру Microsoft .NET

13 февраля 2002 года корпорация Microsoft официально объявила о выходе в свет новой операционной (программной) среды .NET Framework и среды разработки Visual Studio .NET. Оба этих продукта являются важнейшими составными частями платформы .NET. Вполне вероятно, что это событие обозначило начало новой эры в развитии программирования. Нет, конечно же на сегодняшний момент речь не идет о массовом сворачивании разработок для Windows и повсеместном переходе на .NET, но новая архитектура сразу же заняла серьезные позиции и пока что-то не видно, что она собирается эти позиции сдавать. Более того, представители Microsoft в официальных заявлениях и частных беседах настойчиво заявляют, что корпорация сделала основную ставку на развитие этой платформы.

В настоящее время традиционные технологии программирования для Windows и .NET развиваются параллельно. Архитектура .NET не привязана к Windows, и допускает реализацию и в других операционных системах.

Платформа .NET — это совокупность программных средств, которые реализуют новый способ разработки, распространения и выполнения приложений.

Хотя исполняемые файлы приложений .NET и имеют расширение exe и dll, их содержимое принципиально отличается от традиционных исполняемых файлов Windows. Архитектура .NET стала последним словом в технологиях создания распределенных систем, объединив преимущества Web-приложений и XML Web-служб на новой технологической платформе. Теперь любые части распределенного приложения могут исполняться на любой вычислительной платформе (рабочая станция, Pocket PC, мобильный телефон и т. д.), лишь бы они обеспечивали работу операционной среды .NET Framework — важнейшей составной части архитектуры. Для разработки приложений пригодны самые различные языки программирования и, что очень важно, это можно делать в рамках одного приложения — компилятор легко совместит фрагменты кода, написанные на разных языках.