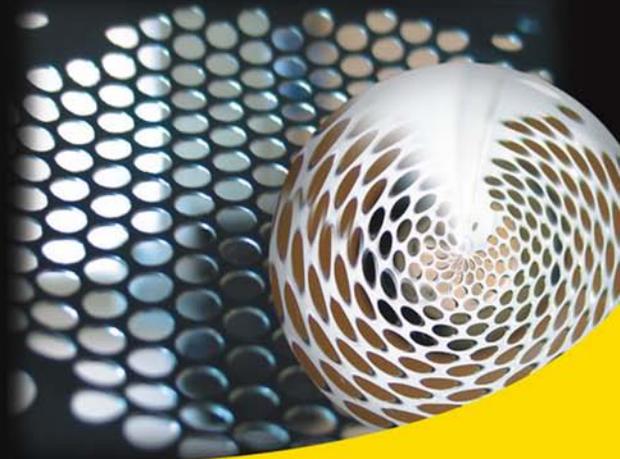


Анатолий Хомоненко

Delphi .NET



Техника работы в среде Delphi 2006

Приемы создания приложений

Работа с базами данных

Примеры программ

Анатолий Хомоненко

**Самоучитель
Delphi .NET**

Санкт-Петербург

«БХВ-Петербург»

2006

УДК 681.3.068+800.92Delphi2006
ББК 32.973.26-018.1
Х76

Хомоненко А. Д.

Х76 Самоучитель Delphi .NET. — СПб.: БХВ-Петербург, 2006. —
464 с.: ил.

ISBN 5-94157-899-7

Рассмотрена разработка приложений в Delphi 2006 для Microsoft .NET Framework. Описаны визуальные компоненты, используемые для создания интерфейса приложений, Windows Forms и VCL.NET; компоненты и техника работы с текстовой информацией, кнопками и переключателями, а также формами и элементами меню. Уделено внимание использованию графики и мультимедиа, работе с файлами и каталогами. Приведены основные понятия, связанные с проектированием баз данных, рассмотрены элементы реляционных баз данных и техника их использования. Показаны особенности технологий ADO.NET и BDP.NET. Описано создание приложений для работы с базами данных по технологиям BDE.NET, dbGo.NET и dbExpress.NET. Показаны приемы работы с данными и подготовка отчетов с помощью генератора Rave Reports. Приводятся многочисленные примеры.

Для начинающих программистов

УДК 681.3.068+800.92Delphi2006
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>Екатерина Капальгина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 27.07.06.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 37,41.

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 ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-899-7

© Хомоненко А. Д., 2006

© Оформление, издательство "БХВ-Петербург", 2006

Оглавление

Предисловие	11
ЧАСТЬ I. ОБЩАЯ ХАРАКТЕРИСТИКА DELPHI .NET 2006.....	15
Глава 1. Состав и назначение Delphi .NET 2006.....	17
Характеристика платформы Microsoft .NET.....	17
Типы приложений Delphi .NET 2006.....	18
Глава 2. Язык программирования Delphi .NET	21
Основные понятия.....	21
Алфавит.....	21
Словарь языка.....	22
Структура программы.....	24
Комментарии.....	27
Типы данных.....	27
Операторы.....	28
Директивы компилятора.....	28
Простые типы данных.....	29
Целочисленные типы.....	30
Литерные типы.....	30
Логические типы.....	31
Перечислимые типы.....	31
Интервальные типы.....	32
Вещественные типы.....	32
Структурные типы данных.....	33
Строки.....	34
Массивы.....	35
Множества.....	37
Записи.....	39
Файлы.....	40

Другие типы данных	41
Указатели	41
Небезопасный код	42
Процедурные типы	43
Вариантные типы	43
Выражения	45
Арифметические выражения	46
Логические выражения	49
Строковые выражения	50
Простые операторы	52
Оператор присваивания	52
Оператор перехода	53
Пустой оператор	54
Оператор вызова процедуры	54
Структурированные операторы	54
Составной оператор	54
Условный оператор	55
Оператор выбора	55
Операторы цикла	56
Оператор цикла с параметром	57
Оператор цикла с постусловием	58
Оператор цикла с предусловием	59
Оператор цикла перебора членов контейнера	59
Оператор доступа	60
Подпрограммы	61
Процедуры	62
Функции	64
Рекурсивные подпрограммы	65
Параметры и аргументы	66
Модули	67
Особенности объектно-ориентированного программирования	69
Основные концепции ООП	69
Классы и объекты	70
Поля	72
Свойства	72
Методы	73
Сообщения и события	75
Динамическая информация о типе	78
Библиотека визуальных компонентов	80
Глава 3. Интегрированная среда Delphi .NET 2006	83
Средства интегрированной среды	83
Главное окно	84
Страница приглашения	86
Палитра инструментов	86

Конструктор формы	88
Редактор кода.....	89
Инспектор объектов	90
Менеджер проектов.....	92
Хранилище объектов.....	93
Проводник данных	94
Список для выполнения	94
Окно структуры проекта	94
Параметры инструментальных средств	95
Характеристика проекта	96
Состав проекта.....	96
Файл проекта.....	96
Файл описания формы приложения VCL.NET	99
Файлы модуля формы	101
Файлы модулей.....	103
Файл ресурсов.....	103
Параметры проекта.....	103
Компиляция и выполнение проекта.....	104
Разработка приложения	105
Простейшее приложение	106
Создание интерфейса приложения.....	107
Определение функциональности приложения	113
Встроенный отладчик.....	115
Рефакторинг.....	116
Справочная система	116

ЧАСТЬ II. ПРИЛОЖЕНИЯ WINDOWS FORMS 119

Глава 4. Характеристика управляющих компонентов 121

Страницы с компонентами	121
Общие свойства визуальных компонентов	124
Общие методы визуальных компонентов.....	125
Общие события визуальных компонентов	127
Обработка событий клавиатуры.....	129
Обработка событий мыши.....	130
Перетаскивание методом Drag and Drop	131

Глава 5. Работа с текстом 136

Отображение текста	136
Текст с гиперссылкой.....	137
Простейший текстовый редактор.....	138
Редактор текста с форматированием	141
Обмен данными с текстовыми процессорами.....	141
Общие свойства редакторов.....	148

Глава 6. Кнопки, списки, переключатели	149
Работа с кнопками	149
Списки	152
Простой список	152
Комбинированный список	154
Общие свойства списков	157
Переключатели	159
Флажок — независимый переключатель	159
Зависимый переключатель	162
Комбинация списка и флажка	163
Глава 7. Форма, контейнеры, приложение	165
Форма	165
Характеристика формы	165
Свойства формы	167
Связи между формами	169
Формы приложения MDI	170
Методы и события формы	170
Контейнеры	173
Группа	175
Панель	175
Приложение	176
Глава 8. Диалоги, панель инструментов, меню	178
Диалоги	178
Диалоги открытия и закрытия файлов	179
Диалоги печати файлов документов	181
Диалог настройки шрифта	183
Диалог выбора цвета	184
Панель инструментов	185
Меню	188
Главное меню	188
Контекстное меню	194
ЧАСТЬ III. ПРИЛОЖЕНИЯ VCL.NET	195
Глава 9. Визуальные компоненты	197
Страницы с визуальными компонентами	197
Базовый класс <i>TControl</i>	201
Свойства	202
События	211
Методы	215

Глава 10. Исключения.....	216
Виды ошибок.....	216
Классы исключений.....	219
Обработка исключений.....	222
Глобальная обработка.....	222
Локальная обработка.....	223
Глава 11. Развитые элементы интерфейса.....	230
Диапазоны значений.....	230
Реверсивные счетчики.....	235
Компонент <i>UpDown</i>	235
Компонент <i>SpinButton</i>	238
Компонент <i>SpinEdit</i>	238
Строка состояния.....	239
Элементы с вкладками.....	242
Одностраничный блокнот.....	242
Многостраничный блокнот.....	246
Глава 12. Работа с графикой.....	247
Возможности рисования при выполнении программы.....	248
Поверхность рисования.....	250
Графические компоненты.....	260
Компонент <i>Shape</i>	261
Компонент <i>Bevel</i>	261
Компонент <i>Image</i>	262
Компонент <i>PaintBox</i>	265
Компонент <i>ImageList</i>	265
Глава 13. Работа с мультимедиа.....	270
Воспроизведение видеоклипов.....	271
Управление мультимедийными устройствами.....	274
ЧАСТЬ IV. РАБОТА С БАЗАМИ ДАННЫХ .NET.....	285
Глава 14. Введение в базы данных.....	287
Основные понятия.....	287
Банки данных.....	287
Модели данных.....	288
Базы данных и приложения.....	289
Варианты архитектуры.....	290
Реляционные базы данных.....	291
Таблицы баз данных.....	291
Ключи и индексы.....	293
Способы доступа к данным.....	296
Связь между таблицами.....	297

Механизм транзакций	300
Бизнес-правила	300
Форматы таблиц	301
Инструменты.....	304
Создание информационной системы	305
Создание таблиц базы данных.....	305
Описание полей	308
Задание индексов.....	309
Ограничения на значения полей.....	311
Ссылочная целостность	312
Пароль и языковой драйвер.....	312
Изменение структуры таблицы	313
Глава 15. Технология ADO.NET	314
Общая характеристика	314
Схема доступа к данным.....	316
ADO.NET в Delphi 2006.....	316
Глава 16. Использование провайдеров BDP.NET	319
Провайдеры данных для .NET.....	319
Провайдеры данных BDP.NET.....	320
Приложение Windows Forms с ADO.NET и BDP.NET.....	322
Глава 17. Подготовка отчетов Rave Reports	327
Характеристика генератора отчетов	327
Визуальное конструирование отчетов	329
Интерфейс визуального конструктора.....	329
Состав проекта отчетов.....	330
Редактор событий	333
Компоненты многостраничной панели инструментов	334
Компоненты отображения данных.....	336
Компоненты управления отчетом	339
Компонент-проект отчета	339
Компонент управления отчетом.....	340
Компоненты установления соединения.....	340
Схема управления отчетом и подсоединения данных.....	341
Примеры создания и просмотра отчетов.....	343
Предварительный просмотр отчета	344
Простой отчет приложения базы данных.....	346
ЧАСТЬ V. РАБОТА С БАЗАМИ ДАННЫХ VCL.NET	349
Глава 18. Компоненты приложения для работы с базами данных	351
Характеристика приложения для работы с базами данных.....	351
Источник данных.....	353

Визуальные компоненты.....	354
Отображение и редактирование значения строкового поля.....	356
Отображение и редактирование значения логического поля.....	357
Представление записей с помощью сетки.....	358
Столбцы сетки.....	360
Использование навигатора.....	365
Глава 19. Технология BDE.NET.....	367
Наборы данных.....	367
Общая характеристика.....	367
Состояния и режимы.....	370
Особенности компонента <i>Table</i>	374
Особенности компонента <i>Query</i>	377
Объекты-поля.....	381
Редактор полей.....	383
Операции с полями.....	385
Операции с данными.....	388
Сортировка записей.....	388
Навигация по набору данных.....	390
Фильтрация записей.....	393
Поиск записей.....	397
Модификация набора данных.....	400
Редактирование записей.....	401
Вставка и добавление записей.....	405
Удаление записей.....	407
Пример формы приложения.....	409
Связывание таблиц.....	416
Глава 20. Технология dgGo.NET.....	420
Общая характеристика.....	420
Установление соединения.....	422
Управление соединением и транзакциями.....	426
Компоненты доступа к данным.....	428
Доступ к таблицам.....	430
Выполнение запросов.....	430
Вызов хранимых процедур.....	431
Компонент <i>ADODataset</i>	431
Команды.....	432
Пример приложения.....	433
Глава 21. Технология dbExpress.NET.....	437
Общая характеристика.....	437
Установление соединения с сервером.....	438
Компоненты доступа к данным.....	442

Универсальный доступ к данным.....	443
Просмотр таблиц	448
Выполнение SQL-запроса	449
Выполнение хранимых процедур.....	450
Компонент редактирования набора данных.....	450
Отладка соединения с сервером.....	453
Предметный указатель.....	455

Предисловие

В книге рассматривается Delphi .NET 2006, обозначающая систему программирования Delphi for the Microsoft .NET Framework, которая входит в состав продукта Borland Developer Studio 2006. Названный продукт представляет собой интегрированную среду разработки и, кроме того, включает следующие системы программирования: C# Builder, C++ Builder и Delphi for Microsoft Win32.

Delphi .NET 2006 обеспечивает объединение возможностей инструментальных средств системы Delphi версий до 7 включительно с новыми средствами и решениями, которые предоставляет платформа Microsoft .NET Framework.

С помощью системы программирования Delphi .NET 2006 можно разрабатывать приложения для платформы .NET, используя компоненты двух библиотек: Visual Component Library (VCL) для .NET — библиотеки визуальных компонентов фирмы Borland для платформы .NET; Framework Class Library (FCL) — библиотеки классов .NET Framework — разработки фирмы Microsoft. Названные библиотеки обладают во многом похожими наборами компонентов и сопоставимыми возможностями.

Книга посвящена основам работы с Delphi .NET 2006 и освоению приемов программирования с использованием визуальных средств обеих библиотек. Чтобы избежать повторов, применение каждой библиотеки дается на не перекрывающихся наборах компонентов. Рассматриваются основные средства системы, технология создания приложений для решения общих задач (от простейших приложений до не очень сложных приложений для работы с базами данных), с которыми приходится сталкиваться на начальном этапе освоения системы программирования. В книге рассматривается большое число примеров, демонстрирующих основные возможности Delphi .NET 2006.

Книга включает пять частей.

Часть I. Общая характеристика Delphi .NET 2006. Содержит краткое описание платформы Microsoft .NET и характеристику типов приложений Delphi .NET 2006. Описывается язык программирования Delphi .NET: типы данных, ос-

новые конструкции языка, важнейшие приемы программирования, понятия объектно-ориентированного программирования, а также ограничения и отличия языка от предыдущих версий. Рассматривается интегрированная среда Delphi .NET 2006, состав и характеристика элементов проекта приложения, компиляция и выполнение проекта, основы технологии разработки приложений.

Часть II. Приложения Windows Forms. Рассматриваются важнейшие визуальные компоненты, используемые для создания интерфейса приложений Windows Forms, разрабатываемых с помощью библиотеки Framework Class Library — библиотеки классов .NET Framework. При этом дается состав страниц **Windows Forms, Components** и **Dialogs** Палитры инструментов, содержащих визуальные компоненты. Все компоненты, имеющие визуальное представление, наследуются от класса `Control`. Коротко описываются общие наиболее важные, на наш взгляд, свойства, методы и события визуальных компонентов, происходящие от класса `Control`. Рассматриваются вопросы обработки событий клавиатуры и мыши, перетаскивание методом `Drag and Drop`.

Рассматриваются компоненты и техника работы с информацией (текстом) по ее отображению, вводу и редактированию. Описывается используемый для отображения надписей компонент `Label`, называемый меткой; компонент `LinkLabel`, задающий надпись с гиперссылкой; простейший текстовый редактор — компонент `TextBox`; редактор текста с форматированием — `RichTextBox`; обмен данными с текстовыми процессорами, а также общие свойства текстовых редакторов.

Описываются компоненты и техника работы с кнопками, списками и переключателями. В частности, рассматриваются простой и комбинированный списки и общие свойства списков, зависимый переключатель и независимый переключатель, комбинация списка и флажка. При этом освещается техника объединения, или группирования, различных элементов управления, которая может понадобиться, например, при работе с переключателями на форме или при создании панели инструментов.

Рассматривается общая характеристика форм, контейнеров и приложений. При этом характеризуются общие свойства, методы и события формы, а также важные свойства и методы контейнеров и приложения. Описываются компоненты и техника создания форм, являющихся центральной частью практически любого приложения и представляющих собой видимые окна Windows. Приводятся характеристики формы, приемы организации взаимодействия форм.

Освещаются *контейнеры*, используемые для объединения, или группирования, различных элементов управления. При этом рассматриваются так называемые *группы*, которые служат в основном для визуального выделения функционально связанных управляющих элементов, и *панели*, применяемые для группирования элементов управления, а также для создания панелей инструментов и строк состояния.

Дается общая характеристика стандартных диалогов, панели инструментов и меню. При этом указываются основные свойства, методы и события соответствующей

щих компонентов и описывается техника их создания. Эти компоненты играют важную роль в организации пользовательского интерфейса приложения.

Часть III. Приложения VCL.NET. Дается общая характеристика визуальных компонентов библиотеки VCL.NET, приводится состав страниц **Standard**, **Additional** и **Win32** Палитры инструментов. Описывается класс `TControl`, который является базовым для большинства визуальных компонентов и включает в себя общие для визуальных компонентов свойства, события и методы.

Описываются важные для отладки приложений вопросы обработки исключительных ситуаций, связанных с ошибками при выполнении программ. При этом характеризуются виды ошибок и классы исключений, рассматриваются варианты глобальной и локальной обработки исключений.

При изучении визуальных компонентов для приложений Windows Forms рассмотрены наиболее простые элементы управления. Для приложений VCL.NET описываются более сложные элементы пользовательского интерфейса: полоса прокрутки, ползунок, счетчик, строка состояния, таблица и блокноты, многостраничный и одностраничный.

Освещаются вопросы работы с графикой: рисование в процессе выполнения программ, используемые важнейшие графические компоненты `Shape`, `Bevel`, `Image`, `PaintBox` и `ImageList`. Описываются вопросы воспроизведения видеоклипов и управления мультимедийными устройствами.

Часть IV. Работа с базами данных .NET. Рассматриваются основные понятия баз данных; характеризуются элементы реляционных баз данных и техника их использования (таблицы, ключи и индексы, способы доступа к данным, связь между таблицами, механизм транзакций и др.). Описывается технология создания таблиц базы данных.

Дается характеристика технологии ADO.NET, рассматривается схема доступа к данным и описываются компоненты, используемые при реализации этой технологии в Delphi .NET.

Рассматриваются провайдеры данных BDP.NET, которые при использовании технологии ADO.NET служат для соединения с базой данных, выполнения команд и получения результатов. Приводится пример приложения Windows Forms для работы с базами данных с применением технологии ADO.NET и BDP.NET.

Рассматривается подготовка отчетов (печатных документов, содержащих данные, аналогичные получаемым в результате выполнения запроса к базе данных) с помощью генератора Rave Reports. При этом описываются компоненты, предназначенные для создания отчетов; процедура печати отчета; технология подготовки простого отчета. Приводятся примеры создания отчетов.

Часть V. Работа с базами данных VCL.NET. Отмечается, что одно- и двухуровневые приложения VCL.NET могут осуществлять доступ к локальным и удаленным БД с использованием следующих технологий: BDE.NET, dbGo.NET

на основе ADO, dbExpress.NET и InterBase.NET. Описывается состав компонентов приложения для работы с базами данных VCL.NET. Рассматривается компонент-источник данных, используемый для подключения наборов данных, и визуальные компоненты, которые служат для навигации по набору данных, отображения и редактирования записей. Эти два типа компонентов применяются при создании приложений для работы с базами данных VCL.NET с использованием любой из названных технологий доступа.

Рассматриваются основные наборы данных (компоненты `Table` и `Query`), используемые для доступа к данным с помощью механизма BDE.NET. Описываются технологии выполнения операций с данными: сортировка, навигация, фильтрация, поиск записей и модификация набора данных. Приводится пример приложения для работы с базами данных по технологии BDE.NET. Поясняется работа со связанными таблицами.

Описывается технология dgGo.NET, которая в среде Delphi .NET представляет собой новое название технологии ADO, введенное с той целью, чтобы отличить технологии ADO и ADO.NET. Рассматривается установление соединения, управление соединениями и транзакциями, компоненты доступа к данным: доступ к таблицам, выполнение запросов, вызов хранимых процедур. Приводится пример приложения для работы с базами данных по технологии dgGo.NET.

Дается характеристика технологии dbExpress.NET, в основе которой лежит использование множества легковесных драйверов, компонентов, объединяющих соединения, транзакции, запросы и наборы данных, а также интерфейсов, реализующих универсальный доступ к соответствующим функциям. Рассматривается установление соединения с сервером, компоненты доступа к данным: универсальный доступ к данным, просмотр таблиц, выполнение SQL-запросов и хранимых процедур, а также отладка соединения с сервером.

Книга ориентирована на начинающих программистов, может быть полезной для специалистов-разработчиков приложений в качестве справочного издания.



ЧАСТЬ I

Общая характеристика Delphi .NET 2006

Глава 1. Состав и назначение Delphi .NET 2006

Глава 2. Язык программирования Delphi .NET

Глава 3. Интегрированная среда Delphi .NET 2006

ГЛАВА 1



Состав и назначение Delphi .NET 2006

Термином Delphi .NET 2006 называют систему программирования Delphi for the Microsoft .NET Framework, которая входит в состав продукта Borland Developer Studio 2006. Названный продукт представляет собой интегрированную среду разработки и, кроме того, включает следующие системы программирования: C# Builder, C++ Builder и Delphi for Microsoft Win32. Соответствующие типы приложений могут разрабатываться либо в рамках интегрированной среды Borland Developer Studio 2006, либо с помощью перечисленных систем программирования, вызываемых с помощью отдельных модулей.

С помощью Borland Developer Studio 2006 и языка программирования Delphi можно разрабатывать приложения для двух платформ: Win32 и .NET. Мы будем вести речь о технологиях разработки приложений только для платформы .NET.

Характеристика платформы Microsoft .NET

Основу Microsoft .NET составляет .NET Framework, которая представляет собой платформу для разработки и исполнения приложений. В ее состав входят следующие основные компоненты:

- CLR (Common Language Runtime) — общезыковая исполняемая среда;
- FCL (Framework Class Library) — библиотека классов .NET Framework.

CLR управляет выполнением кода приложений и обеспечивает службы (services), упрощающие процесс разработки приложений. В качестве примера можно назвать службы управления памятью и многоязыковой поддержки.

Библиотека классов FCL содержит множество объектно-ориентированных компонентов, служащих для разработки приложений .NET, позволяющих использовать все достоинства применения служб среды CLR.

Среда Microsoft .NET Framework обеспечивает *многоязыковую совместимость* приложений. Благодаря общеязыковой исполняемой среде приложение .NET может быть написано на любом .NET-совместимом языке программирования высокого уровня, к примеру, Visual Basic .NET и Visual C# .NET. Естественно, что теперь здесь может использоваться и язык Delphi.

При создании с помощью компилятора для платформы .NET модуля (двоичного файла) с расширением dll или exe его содержимое представляет собой *сборку* на промежуточном языке Microsoft Intermediate Language (MSIL). Кроме команд на языке MSIL, двоичный модуль .NET содержит метаданные, которые предназначены для описания всех типов в этом модуле (классов, структур, перечислений и т. д.).

В сборку может входить произвольное количество типов. Для указания определенного типа требуется обеспечить уникальность имен по отношению к другим типам в сборке. Для решения этой проблемы и предназначены пространства имен, выполняющие группирование стандартных типов и дающие возможность их использования при разработке собственных типов.

Пространство имен представляет собой логическую структуру, предназначенную для организации используемых в приложении имен с целью предотвращения конфликтов между именами из разныхборок. Например, пусть в приложении .NET есть обращения к двум разным внешним сборкам. Если при этом в каждой из них имеется тип с одним и тем же именем, то по своим функциональным возможностям эти типы различны, и смешивать их недопустимо. Следовательно, при программировании нужно указать, к какому типу и из какой сборки выполняется обращение. Для этого достаточно к имени типа добавить наименование нужного пространства имен. Например, строка `System.Int32` указывает на тип `Int32`, который принадлежит пространству имен `System`.

Типы приложений Delphi .NET 2006

С помощью системы программирования Delphi .NET 2006 можно разрабатывать приложения для платформы .NET, используя компоненты двух библиотек:

- Visual Component Library (VCL) для .NET — библиотеки визуальных компонентов фирмы Borland для платформы .NET;
- Framework Class Library (FCL) — библиотеки классов .NET Framework — "родной" для платформы .NET разработки фирмы Microsoft.

Обе указанные библиотеки обладают во многом похожими наборами компонентов и сопоставимыми возможностями. Есть заметные различия в возможностях применения названных библиотек в создании приложений для работы с базами данных и Web.

Библиотека VCL для .NET представляет собой платформу программирования для разработки приложений в среде Delphi .NET 2006 с помощью VCL. С помощью

Delphi 2006 и VCL для .NET можно разрабатывать новые приложения, а также переносить имеющиеся приложения Win32 на платформу .NET, причем с сохранением большей части кода.

Библиотека FCL.NET используется для разработки приложений для платформы .NET, называемых приложениями Windows Forms, Web-приложений с ASP.NET и Web-служб с ASP.NET.

Таким образом, в среде Delphi .NET 2006 поддерживается разработка приложений на платформе .NET с помощью языка Delphi, а также с использованием элементов управления VCL для .NET и элементов управления Windows Forms. Схематично это показано на рис. 1.1. Как видим из приведенной схемы, в Delphi 2006 поддерживается создание приложений для работы с базами данных для всех известных нам по предыдущим версиям системы технологий доступа, но применительно к платформе .NET (ADO.NET, dbExpress.NET, BDE.NET, IBX.NET, DataSnap.NET). В случае использования ADO.NET применяются провайдеры данных фирмы Borland — BDP.NET.

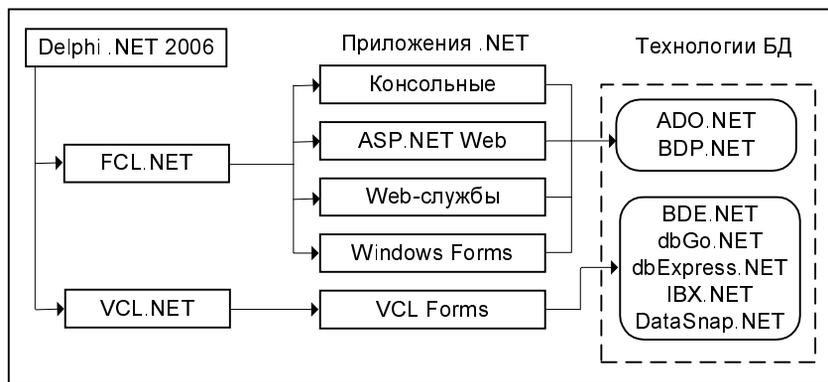


Рис. 1.1. Варианты приложений Delphi .NET 2006

Библиотека VCL для .NET представляет собой большое подмножество наиболее общих классов в составе библиотеки VCL для Win32.

Платформа .NET разработана так, чтобы обеспечить согласование с любым языком, входящим в ее состав. Поэтому код Delphi, оперирующий с классами и функциями VCL для Win32, во многих случаях может быть перекомпилирован на платформу .NET с минимальными изменениями.

Классы библиотеки VCL для .NET принадлежат пространству имен `Borland.VCL`. При этом классы, связанные с базами данных, размещены в пространстве имен `Borland.VCL.DB`. Библиотека классов времени выполнения (Runtime Library Classes) находится в пространстве имен `Borland.VCL.RTL`. Файлы модулей (unit) были собраны в соответствующие пространства имен `Borland.VCL`. Некоторые

модули были перемещены, тем не менее, пространства имен позволяют облегчить поиск объектов, требуемых для достижения нужной функциональности приложения. Файлы с исходным кодом для всех объектов Delphi 2006 размещены в подкаталоге C:\Program Files\Borland\BDS\4.0\source.

Замечание

В книге мы рассматриваем разработку приложений только для платформы .NET. Поэтому, говоря о разработке приложений с помощью библиотеки VCL.NET, для краткости мы будем считать синонимами фразы "Приложение VCL", "Приложение VCL.NET" и "Приложение VCL Forms".

ГЛАВА 2



Язык программирования Delphi .NET

Основу языка программирования Delphi составляет язык Object Pascal. При этом система программирования Delphi накладывает ряд ограничений. Язык программирования Delphi .NET подразумевает использование платформы .NET, а это приводит к ряду дополнительных ограничений и отличий. В данной главе мы рассмотрим основные средства и приемы программирования на языке Delphi .NET, оговаривая специально эти ограничения и отличия.

Основные понятия

Алфавит

Алфавит языка Delphi включает в себя следующие символы:

- 53 буквы — прописные (A .. Z) и строчные (a .. z) буквы латинского алфавита и знак подчеркивания (_);
- 10 цифр — 0 .. 9;
- 23 специальных символа — + - * / . , ; = > < ' () { } [] # \$ ^ @ пробел.

Комбинации специальных символов образуют следующие *составные* символы:

- := — присваивание;
- <> — не равно;
- .. — диапазон значений;
- <= — меньше или равно;
- >= — больше или равно;
- (* и *) — альтернатива фигурным скобкам { и };
- (. и .) — альтернатива квадратным скобкам [и].

Словарь языка

Неделимые последовательности знаков алфавита образуют *слова*, отделяемые друг от друга разделителями и несущие определенный смысл в программе. *Разделителями* могут служить пробел, символ конца строки, комментарий, другие специальные символы и их комбинации.

Слова подразделяются на:

- ключевые слова;
- директивы;
- стандартные идентификаторы;
- идентификаторы пользователя.

Ключевые (зарезервированные) слова являются составной частью языка, имеют фиксированное написание и однозначно определенный смысл, изменить который программист не может. Полный перечень ключевых слов включает следующие слова:

and	else	inherited	packed	then
array	end	initialization	procedure	threadvar
as	except	inline	program	to
asm	exports	interface	property	try
begin	file	is	raise	type
case	final	label	record	unit
class	finalization	library	repeat	unsafe
const	finally	mod	resourcestring	until
constructor	for	nil	sealed	uses
destructor	function	not	set	var
dispinterface	goto	object	shl	while
div	if	of	shr	with
do	implementation	or	static	xor
downto	in	out	string	

В редакторе кода ключевые слова выделяются полужирным шрифтом.

Слова *above*, *private*, *protected*, *public*, *published* и *automated* являются зарезервированными словами внутри объявления класса, а в остальных случаях воспринимаются как директивы. Слова *at* и *on* имеют специальное значение и должны обрабатываться как зарезервированные слова.

Директивы имеют контекстную зависимость значения от своего местоположения в исходном коде программы. К числу директив относятся следующие слова:

absolute	dynamic	local	platform	requires
abstract	export	message	private	resident
assembler	external	name	protected	safecall
automated	far	near	public	stdcall
cdecl	forward	nodefault	published	stored
contains	implements	overload	read	varargs
default	index	override	readonly	virtual
deprecated	inline	package	register	write
dispid	library	pascal	reintroduce	writeonly

Стандартные идентификаторы служат для обозначения следующих заранее определенных разработчиками конструкций языка:

- типов данных;
- констант;
- процедур и функций.

В отличие от ключевых слов, директиву и любой из стандартных идентификаторов можно переопределить, но поскольку это может привести к ошибкам, то их лучше использовать без изменений. Примерами стандартных идентификаторов являются слова `Sin`, `Pi`, `Real`.

Идентификаторы задаются программистом для обозначения имен меток, констант, переменных, процедур, функций и типов данных. Они должны удовлетворять следующим требованиям:

- состояются из букв и цифр;
- идентификатор начинается с буквы, исключением являются метки, которыми могут быть целые числа без знака в диапазоне 0 .. 9999;
- в идентификаторе допускаются строчные и прописные буквы, компилятор интерпретирует их одинаково; использование в идентификаторах специальных символов не допускается, поэтому для наглядности отдельные составляющие идентификатора полезно выделять прописными буквами, например, `NumberLines` или `btnOpen`;
- между двумя идентификаторами в программе должен быть, по крайней мере, один разделитель.

Структура программы

Исходный текст программы представляется в виде последовательности строк, при этом текст строки может начинаться с любой позиции. Структурно программа состоит из заголовка и блока.

Заголовок находится в начале программы и имеет вид:

```
Program <Имя программы>;
```

Блок состоит из двух частей: описательной и исполнительской. В *описательной части* содержится описание элементов программы, а в *исполнительской* указываются действия с различными элементами программы, позволяющие получить требуемый результат.

В общем случае описательная часть состоит из следующих разделов:

- подключения модулей;
- объявления меток;
- объявления констант;
- описания типов данных;
- объявления переменных;
- описания процедур и функций.

В конце каждого из указанных разделов ставится точка с запятой.

Замечание

Подчеркнем различие между терминами *объявление* и *описание*. Объявление некоторого объекта в программе предполагает выделение основной памяти для его размещения. Описание некоторой конструкции в программе, в отличие от объявления, выделения памяти не требует.

Структуру программы в общем случае можно представить следующим образом:

```
Program <Имя программы>;  
Uses <Список модулей>;  
Label <Список меток>;  
Const <Список констант>;  
Type <Описание типов>;  
Var <Объявление переменных>;  
<Описание процедур>;  
<Описание функций>;  
Begin  
<операторы>;  
End.
```

В программе любой из разделов описания и объявления может отсутствовать. Разделы описаний и объявлений, кроме раздела подключения модулей, который

располагается сразу после заголовка программы, могут встречаться в программе произвольное число раз и следовать в произвольном порядке. При этом все описания и объявления элементов программы должны быть сделаны до того, как они будут использованы. Рассмотрим подробнее отдельные разделы программы.

Раздел *подключения модулей* включает зарезервированное слово `Uses` и список имен подключаемых стандартных и пользовательских библиотечных модулей. Формат этого раздела:

```
Uses <Имя1>, <Имя2>, ..., <ИмяN>;
```

Например,

```
Uses Crt, Dos, MyLib;
```

Раздел *объявления меток* включает зарезервированное слово `Label` и список имен меток. Формат данного раздела:

```
Label <имя1>, <имя2>, ..., <имяN>;
```

Например,

```
Label met1, met2, 11, 67;
```

Раздел *объявления констант* начинается ключевым словом `Const`, за которым следует ряд операторов, присваивающих константам значения. Имя константы отделено от выражения знаком равенства, в конце оператора ставится точка с запятой. Формат этого раздела:

```
Const <идентификатор1> = <Выражение>;  
    ...  
    <идентификаторN> = <Выражение>;
```

Например,

```
Const st1 = 'Hello'; ch = '3'; pi1 = 3.1415;
```

Тип константы распознается автоматически на основании типа выражения.

В Delphi имеется большое количество констант, которые можно использовать без их предварительного объявления, например, `Nil`, `True` и `MaxInt`.

В разделе *описания типов* описываются типы данных пользователя. Этот раздел не является обязательным, и типы могут быть описаны неявно в разделе объявления переменных. Раздел описания типов начинается ключевым словом `Type`, за которым располагаются имена типов и их описания, разделенные знаком равенства. В конце описания ставится точка с запятой. Формат раздела:

```
Type <Имя типа1> = <Описание типа>;  
    ...  
    <Имя типаN> = <Описание типа>;
```

Например,

```
Type char2 = ('a'..'z');  
massiv = array[1..50] of real;  
week = 1..7;
```

В Delphi имеется много *стандартных* типов, не требующих предварительного описания: Real, Integer, Char, Boolean и др.

Каждая переменная программы должна быть объявлена. Объявление обязательно предшествует использованию переменной. Раздел *объявления переменных* начинается с ключевого слова Var, после которого через запятые перечисляются имена переменных и через двоеточие указывается их тип. Формат раздела:

```
Var <идентификаторы> : <тип>;  
    ...  
    <идентификаторы> : <тип>;
```

Например,

```
Var a, u7: real;  
    simv: char;  
    il, j2: integer;
```

При компиляции операторов объявления переменных под них выделяется соответствующий их типу участок памяти для размещения без присваивания начальных значений.

Подпрограммой называют логически законченную и специальным образом оформленную часть программы, которая может вызываться для выполнения из других точек программы неограниченное число раз. В языке Delphi подпрограммы разделяют на два вида: *процедуры* и *функции*. Каждая подпрограмма представляет собой блок и должна быть определена в разделе *описания процедур и функций*. Описание процедур и функций рассматривается далее.

Раздел *операторов* начинается с ключевого слова Begin, после которого следуют операторы языка, разделенные точкой с запятой. Завершает этот раздел ключевое слово End, после которого указывается точка. Формат раздела:

```
Begin  
<оператор1>;  
    ...  
<операторN>;  
End.
```

Здесь могут использоваться любые операторы языка, например, оператор присваивания или условный оператор.

Комментарии

Комментарий представляет собой пояснительный текст, который можно записывать в любом месте программы, где разрешен пробел. Текст комментария ограничен символами (* и *) или их эквивалентами { и } и может содержать любые символы языка, в том числе русские буквы. Комментарий, ограниченный указанными символами, может занимать несколько строк. Однострочный комментарий в начале строки содержит двойной слеш //.

Примеры комментариев:

```
(* Однострочный комментарий*)  
// Второй однострочный комментарий  
(* Начало многострочного комментария  
   Окончание многострочного комментария *)
```

Допускается вложение комментариев, ограниченных различными символами, например,

```
(* { Вложенные комментарии } *)
```

При компиляции комментарий игнорируется и не оказывает влияния на выполнение программы.

Типы данных

Обрабатываемые в программе данные подразделяются на переменные, константы и литералы. *Константы* представляют собой данные, значения которых установлены в разделе объявления констант и не изменяются в процессе выполнения программы. *Переменные* объявляются в разделе объявления переменных, однако в отличие от констант получают свои значения уже в процессе выполнения программы, причем допускаются изменения этих значений. К константам и переменным можно обращаться по именам. *Литерал* не имеет идентификатора и представляется в тексте программы непосредственно значением, поэтому их также называют просто *значениями*.

Каждый элемент данных принадлежит к определенному типу, при этом тип переменной указывается при ее описании, а тип констант и литералов распознается компилятором автоматически в зависимости от указанного значения.

Тип определяет множество значений, которые могут принимать элементы данных, и совокупность допустимых над ними операций. Например, значения 34 и 67 относятся к целому типу, их, соответственно, можно умножать, складывать, делить и выполнять другие арифметические операции, а значения 'abcd' и 'sdhf123' относятся к строковому типу, и их можно сцеплять (складывать), но нельзя делить или вычитать.

Типы данных можно разделить на следующие группы:

- простые;
- структурные;
- указатели;
- процедурные;
- вариантные.

В свою очередь, простые и структурные типы — это тоже группы, включающие в свой состав другие типы, например, целочисленные или массивы. Приводимое деление на типы в некоторой мере условно — иногда указатели причисляют к простым типам, а строки, которые относятся к структурным типам, выделяют в отдельный тип.

Большое значение имеет понятие *совместимости типов*, которое означает, что типы равны друг другу или один из них может быть автоматически преобразован к другому. Совместимыми, например, являются вещественный и целочисленный тип, т. к. целое число автоматически преобразовывается в вещественное (но не наоборот).

Операторы

Операторы представляют собой законченные предложения языка, которые выполняют некоторые действия над данными. Операторы Delphi можно разделить на две группы:

- простые;
- структурированные.

Например, к простым операторам относится оператор присваивания, а к структурированным — оператор разветвления и оператор цикла.

Операторы разделяются между собой точкой с запятой. Наличие между операторами нескольких точек с запятой не является ошибкой, т. к. они обозначают пустые операторы. Лишняя точка с запятой в разделе описаний и объявлений будет уже синтаксической ошибкой.

Точка с запятой может не ставиться после слова `begin` и перед словом `end`, т. к. они рассматриваются как операторные скобки, а не как операторы. В условных операторах и операторах выбора точка с запятой не ставится после слова `then` и перед словом `else`. Отметим, что в операторе цикла с параметром наличие точки с запятой сразу после слова `do` синтаксической ошибкой не является, но в этом случае тело цикла будет содержать только пустой оператор.

Директивы компилятора

Текст программы может содержать специальные команды, называемые *директивами компилятора*, которые служат для управления режимами компиляции. Ди-

рективы компилятора заключаются в фигурные скобки, а в их начале ставится символ $\$$. С помощью директив компилятора можно, например, задать способы интерпретации строковых типов, размер стека или подключить файл ресурса.

Программист обычно управляет режимами компиляции через окно параметров проекта, включая или выключая соответствующие переключатели на страницах **Compiler**, **Compiler Messages**, **Linker** и **Directories/Conditionals**. При этом установленные значения параметров сохраняются в файле параметров проекта (DOF).

Простые типы данных

Простые типы не содержат в себе других типов, и данные этих типов могут одновременно содержать только одно значение. К простым типам относятся следующие:

- целочисленные;
- литерные (символьные);
- логические (булевские);
- вещественные.

Все типы, кроме вещественного, являются *порядковыми*, т. е. значения каждого из этих типов образуют упорядоченную конечную последовательность. Номера соседних значений в ней отличаются на единицу.

Для значений и имен порядковых типов определены следующие функции:

- $Low(T)$ — минимальное значение типа T ;
- $High(T)$ — максимальное значение типа T ;
- $Ord(X)$ — порядковый номер значения выражения X ;
- $Pred(X)$ — значение, предшествующее значению выражения X ;
- $Succ(X)$ — значение, следующее после значения выражения X .

Кроме того, к ним применимы следующие процедуры:

- $Dec(X)$ — уменьшение значения переменной X на единицу;
- $Inc(X)$ — увеличение значения переменной X на единицу.

Для порядковых типов программист может создавать *перечислимые* и *интервальные* типы. Эти типы также называют *пользовательскими*, или определяемыми пользователем. Их применение улучшает наглядность программы и облегчает поиск ошибок.

Некоторые простые типы делятся на фундаментальные (физические) и общие. *Фундаментальные типы* закладываются в язык и не зависят от особенностей конкретного компьютера. *Общие типы* соответствуют одному из конкретных фундаментальных типов, и их использование считается более предпочтительным, т. к. при этом компилятор создает более эффективный код.

Целочисленные типы

Целочисленные типы предназначены для целых чисел и могут быть фундаментальными и общими. Фундаментальные целочисленные типы Delphi приводятся в табл. 2.1.

Таблица 2.1. Фундаментальные целочисленные типы

Обозначение	Диапазон	Представление в памяти
Shortint	-128 .. 127	1 байт, со знаком
Smallint	-32 768 .. 32 767	2 байта, со знаком
Longint	-2 147 483 648 .. 2 147 483 647	4 байта, со знаком
Int64	-263.. 262	8 байтов, со знаком
Byte	0 .. 255	1 байт, без знака
Word	0 .. 65 535	2 байта, без знака
Longword	0 .. 4 294 967 295	4 байта, без знака

Кроме физических, определены также два общих типа (табл. 2.2).

Таблица 2.2. Общие целочисленные типы

Обозначение	Диапазон	Представление в памяти
Integer	-2 147 483 648 .. 2 147 483 647	4 байта, со знаком
Cardinal	0 .. 4 294 967 295	4 байта, без знака

Для записи целых чисел можно использовать цифры и знаки "+" и "-". Если знак числа отсутствует, то число считается положительным. Число может быть представлено в десятичной и шестнадцатеричной системе счисления. Если число записано в шестнадцатеричной системе, то перед ним ставится знак \$ (без пробела), а допустимый диапазон значений будет \$00000000 .. \$FFFFFFF.

Литерные типы

Значениями *литерного типа* являются элементы из набора литер, т. е. отдельные символы. Для символов также имеются фундаментальные и общие типы. Фундаментальные литерные типы представлены типами AnsiChar и WideChar.

Символ типа AnsiChar занимает один байт, а для кодирования символов используется код ANSI Американского национального института стандартов (American National Standards Institute). Символ типа WideChar занимает два байта, а для кодирования символов используется международный набор символов Unicode. На-