

Введение в Visual Basic .NET 2005 Express Edition

Средство для разработки Windows-приложений VB 6 уже давно снискало популярность миллионов разработчиков во всем мире. С выходом Visual Basic .NET 2005 Express Edition данный продукт стал еще мощнее и доступнее

Сергей Байдачный, Дмитрий Маленко

Среда для создания и запуска приложений .NET Framework стала еще доступней для начинающих разработчиков с выходом Visual Basic .NET 2005 Express Edition. Этот продукт позволит разрабатывать как консольные приложения и библиотеки, так и Windows-приложения с графическим интерфейсом. Причем для построения последних используется уже полюбившийся редактор форм, который был ранее доступен лишь в полной версии Visual Studio .NET.

Рассматривая Visual Basic 2005 Express, который является компонентом Visual Studio 2005 Express, следует остановиться на двух моментах: доступность продукта и возможности среды. Сначала поговорим о доступности. Visual Basic 2005 Express, как и любой другой продукт новой серии Express, доступен для загрузки с сайта Microsoft совершенно бесплатно. Чтобы установить продукт на свою машину, воспользуйтесь либо ссылкой: <http://lab.msdn.microsoft.com/express/vbasic/default.aspx>, либо дистрибутивом продукта, размещенным на CHIP-CD 7/2005.

После установки продукт можно ис-

пользовать как для обучения, так и для создания небольших коммерческих приложений. Такой подход со стороны Microsoft позволит значительно сэкономить деньги небольшим организациям, разрабатывающим desktop-приложения, а также учебным заведениям, внедряющим технологии .NET в систему обучения.

Каковы возможности Visual Basic 2005 Express по сравнению с его старшим братом Visual Studio .NET 2005? Ответить на этот вопрос кратко очень непросто, так как возможности этого продукта весьма значительны. Поэтому, прежде чем перейти к их описанию, перечислим сначала ограничения Visual Basic 2005 Express:

- ▶ продукт позволяет создавать только Windows-приложения или библиотеки классов (для создания ASP.NET-приложений можно использовать Visual Web Developer Express Edition)
- ▶ отсутствует поддержка проектов для мобильных устройств
- ▶ отсутствует возможность удаленного доступа к данным
- ▶ отсутствует поддержка серверов
- ▶ нет возможности автоматизировать весь процесс разработки, напри-

мер тестирование, контроль версий и анализ кода (такие возможности присутствуют в Visual Studio .NET 2005 Team System).

Как следует из приведенного списка, Visual Basic 2005 Express не имеет только тех возможностей, которые с большой долей вероятности используются для разработки коммерческих проектов с большим бюджетом в крупных компаниях.

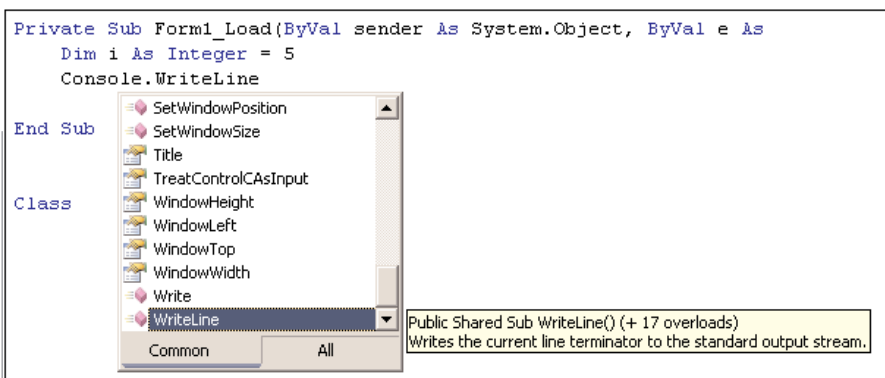
Новые возможности Visual Basic 2005 Express

Приступим к описанию возможностей Visual Basic 2005 Express. Обсудим такие вопросы, как новые возможности языка и редактора, а также переход с более ранних версий этого языка программирования.

Новая версия Visual Basic содержит большое количество изменений и улучшений, которые помогут программистам создавать более качественное программное обеспечение в короткие сроки. Основные улучшения в языке включают поддержку обобщенных классов, новое пространство имен **My**, новые операторы, а также возможность перегрузки операторов для классов, определяемых программистом. Улучшения коснулись также и компилятора. Так, например, была добавлена поддержка XML-комментариев и предупреждений компилятора.

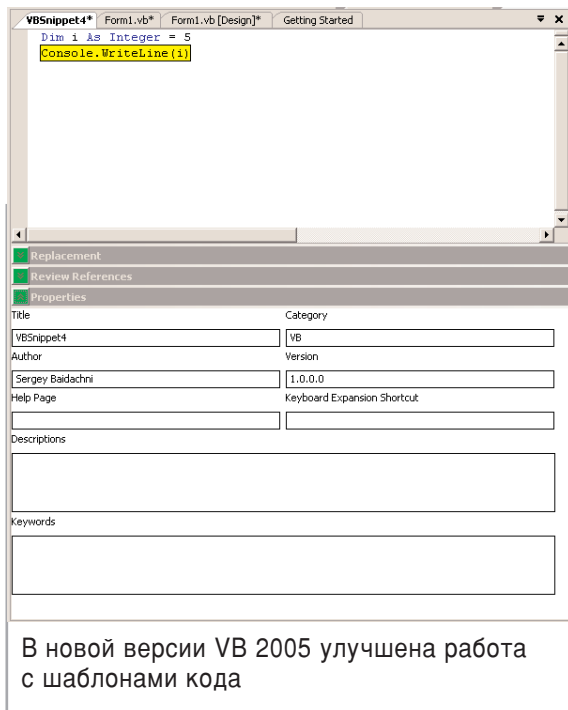
Visual Basic 2005, как и предыдущая версия языка, является полноценным объектно-ориентированным языком, который обеспечивает широкую поддержку всех средств объектно-ориентированного программирования: инкапсуляцию, наследование, полиморфизм.

Первым значительным улучшением в языке является возможность зада-



Использование функции IntelliSense в Visual Basic 2005 позволяет значительно ускорить процесс кодирования

Visual Basic .NET 2005



В новой версии VB 2005 улучшена работа с шаблонами кода

вать различные модификаторы доступа для методов Get и Set. Это позволяет создавать свойства, которые для внешних пользователей класса будут доступны только для чтения, в то время как внутри класса (или сборки) значение этого свойства можно будет еще и устанавливать:

```
Private _propertyValue As String
Public Property MyProp() As String
Get
Return _propertyValue
End Get
Friend Set(ByVal value As String)
If value.Trim.Length > 0 Then
_propertyValue = value.Trim
Else
value = «{not set}»
End If
End Set
End Property
```

Другим значительным расширением возможностей объектно-ориентированного программирования стала поддержка обобщенных типов (Generics). Так, все программисты имели дело с коллекциями, тем более что .NET Framework содержит множество разнообразных типов коллекций. В предыдущей версии платформы при описании класса коллекции приходилось работать с переменными типа Object, так как именно это позволяло хранить в коллекции типы, производные от Object (следовательно, произвольные). Однако негативным моментом было то, что компиля-

тор не контролировал правильность типа при помещении и извлечении объектов из коллекции. Кроме того, при помещении в нее элементов типов-значений использовался такой механизм, как упаковка/распаковка (boxing/unboxing), что не лучшим образом сказывалось на производительности приложения. Теперь у программиста есть возможности определить класс, который будет иметь тип в качестве параметра (например, класс коллекции будет получать как параметр тип элементов).

Практически все классы коллекций, которые были доступны в пространстве имен System.Collections, в новой версии получили свои аналоги, определенные с использованием механизма обобщенных классов. Эти новые классы были помещены в пространство имен System.Collections.Generic. Так, например, выглядит использование обобщенного класса List для создания списка строк, где тип String передается в момент создания объекта:

```
Dim strings As New System.
Collections.Generic.List(Of String)
```

Еще одним новшеством является поддержка перегрузки операторов для определяемых пользователем классов. Это позволит в некоторых случаях сделать использование пользовательских классов более прямоли-

нейным. Традиционный пример для демонстрации такой возможности — определение класса для работы с комплексными числами. Гораздо удобнее использовать оператор «+» вместо вызова метода Add при сложении комплексных чисел. Определения перегруженной версии оператора «+» для класса Complex может выглядеть так:

```
Shared Operator +(ByVal lhs As Complex,
ByVal rhs As Complex) As Complex
Return New Complex(lhs.Real + rhs.Real,
lhs.Imag + rhs.Imag)
End Operator
```

Среди новых операторов и ключевых слов Visual Basic 2005 — Using, Continue, TryCast, IsNot.

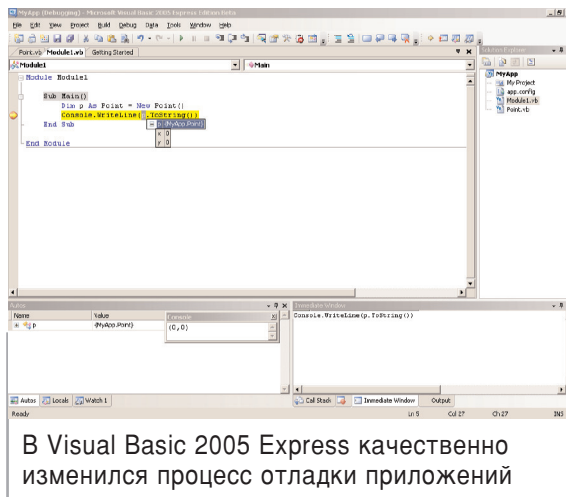
Ключевое слово Using позволяет явно управлять временем жизни объекта. В начале блока Using происходит создание объекта, а в конце блока гарантированно происходит его уничтожение (вызывается метод Dispose для объекта), даже если при выполнении самого блока происходит исключение. Использование этого оператора позволяет создавать более лаконичный и выразительный код:

```
Using br As System.Drawing.SolidBrush = _
New SolidBrush(System.Drawing.Color.Blue)
g.FillRectangle(br, New Rectangle
(30, 50, 230, 200))
End Using
```

Также была добавлена поддержка оператора Continue, который прерывает выполнение текущей итерации цикла и осуществляет переход на следующую итерацию. Мы не рекомендуем использовать этот оператор, так как он делает код более сложным и демонстрирует плохой стиль программирования.

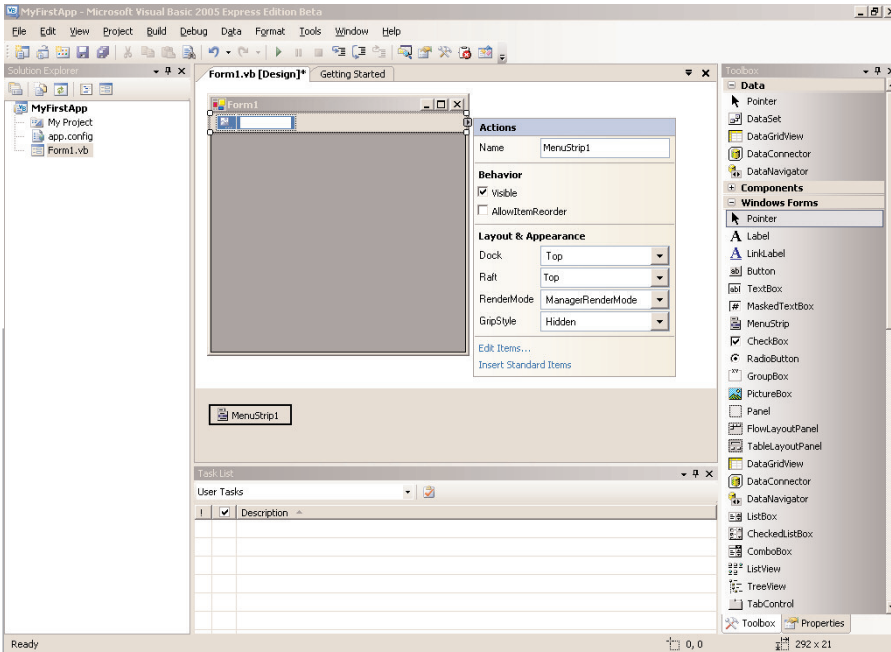
Оператор IsNot предназначен для сокращения записи при совместном использовании операторов Is и Not, т. е. вместо Not obj Is Nothing теперь можно просто написать obj IsNot Nothing.

Оператор TryCast позволяет выполнить попытку преобразования объекта к заданному типу. При этом в отличие от оператора CType он не выбрасывает исключение, если преобразование невозможно, а возвращает Nothing:



В Visual Basic 2005 Express качественно изменился процесс отладки приложений

Visual Basic .NET 2005



Для построения Windows-приложений используется редактор форм, который был ранее доступен лишь в Visual Studio .NET

```
Dim box As TextBox =
TryCast(Me.Controls(3), TextBox)
If box IsNot Nothing Then
' Do something with box
End If
```

Еще одно значительное нововведение в Visual Basic 2005 — новое пространство имен *My*, которое предоставляет быстрый доступ к наиболее часто используемым возможностям библиотеки. Для программиста больше нет необходимости запоминать множество названий классов или пространств имен — доступ ко всем необходимым функциям и свойствам можно получить через пространство имен *My*. К примеру, доступ к информации о пользователе, работающем с приложением, о самом приложении, среде, в которой оно выполняется, можно получить через такие объекты, как *My.User*, *My.Application* и *My.Computer*, соответственно. Теперь нет необходимости пробираться через дебри библиотеки, чтобы узнать, например, название операционной системы, на которой запущено приложение, достаточно воспользоваться пространством имен *My*: *My.Computer.Info.OSFullName*. При этом надо помнить, что пространство имен *My* — это только сокращения для той функциональности, которая уже есть в библиотеке .NET Framework.

Еще одно удобство использования *My* проявляется при работе с формами, ресурсами и настройками приложе-

ния. Так доступ к любой форме приложения можно получить посредством коллекции *My.Forms*. Объекты *My.Resources* и *My.Settings* предоставляют типизированный доступ к ресурсам и настройкам приложения, которые можно создать непосредственно в оболочке Visual Basic 2005.

Рассмотрим, какие основные возможности характерны для среды Visual Basic 2005 Express. Начнем с редактора кода, поскольку именно с ним разработчику приходится проводить максимум времени.

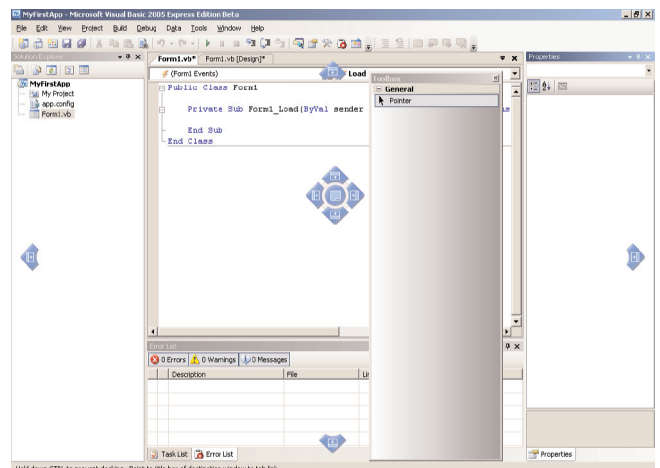
Использование IntelliSense

Наиболее интересной особенностью редактора кода является система IntelliSense. Благодаря тому, что сборки хранят свое описание в метаданных, оболочка имеет возможность получить информацию о классах, методах, свойствах и данных в используемой сборке. Именно это и позволяет работать IntelliSense. Система вызывается автоматически, когда вы набираете точку после имени прост-

ранства имен или класса или открываете скобку, чтобы записать параметры. В первом случае система отображает список доступных компонент, а во втором — описание одного из перегруженных методов, с возможностью выбора нужного. Аналогичным образом система активируется и при редактировании параметров, или названия метода, класса или пространства имен. Использование IntelliSense позволяет значительно ускорить процесс кодирования, так как теперь нет необходимости набирать длинные имена, а также вспоминать точное название метода и список его параметров. Кроме того, это снижает количество ошибок.

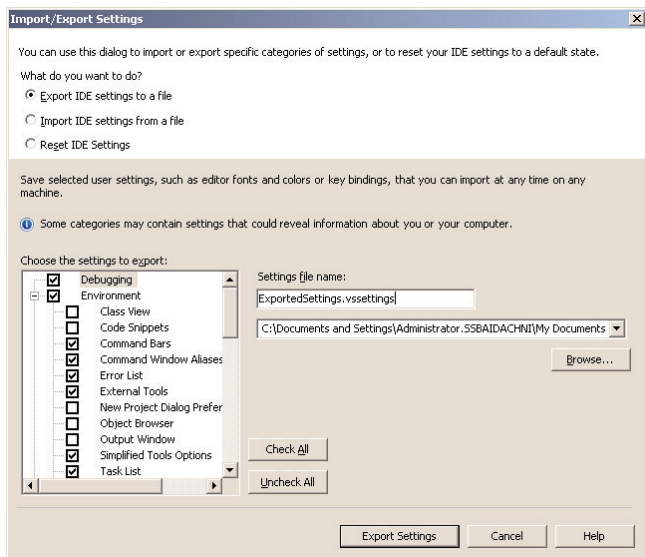
Систему IntelliSense можно активировать принудительно, используя пункт *Edit | IntelliSense*. Это может понадобиться тогда, когда код уже написан, но программист собирается его модифицировать, при этом ему требуется проанализировать, например, возможный список параметров.

В редакторе Visual Basic 2005 начали зарождаться и возможности рефакторинга. В отличие от редактора C эти возможности пока еще слабые, но наверняка будут расширяться. Сейчас для рефакторинга доступен лишь один пункт меню — *Rename*. Этот пункт становится доступен из контекстного меню, когда программист выделяет имя метода при вызове или объявлении имени переменной. Благодаря этой возможности вы можете переименовать метод или переменную, как в месте объявления, так и во всех местах их использования.



Visual Basic 2005 Express обладает улучшенной системой стыковки окон

Visual Basic .NET 2005



Все настройки можно сохранить в один файл и загрузить их на другой машине

Сокращение рутинных операций

Редактор кода также позволяет устанавливать закладки. Благодаря этому программист может перейти в нужное место кода, просто перебрав все имеющиеся закладки. Такая возможность представлена в пункте **Edit / Bookmarks** или на панели инструментов. К сожалению, Visual Basic 2005 Express не имеет окна, отображающего все имеющиеся закладки. Подобное окно присутствует в остальных редакциях Visual Studio и позволяет перейти в нужное место, просто щелкнув на нужной закладке, кроме того, это окно позволяет группировать и перемещать закладки в разные папки.

Кроме перечисленных возможностей, редактор кода позволяет вставлять наиболее общие фрагменты кода, связанные с рутинными операциями. Это возможно с помощью пункта контекстного меню **Insert Snippets...** В появившемся окне программист должен выбрать категорию, в которую входит код, после чего указать необходимый шаблон. При этом вставленный код будет содержать специально выделенные области, подлежащие замене. Данная возможность была бы не полной, если бы программист не мог создать свои шаблоны. Это можно сделать, выделив часть кода и выбрав пункт контекстного меню **Create Snippet...** После этого на экране появится специальный редактор шаблонов. Тут программист может создавать об-

ласти, подлежащие замене, задавать ссылки, которые должны быть добавлены в проект с кодом, а также свойства шаблона, включая автора и категорию.

Произошло много изменений и в редакторе форм. Теперь здесь есть smart-теги, которые поддерживаются практически всеми новыми элементами управления. Благодаря этой возможности программист может

получить быстрый доступ к наиболее важным свойствам всего элемента или его части.

Немало и новых элементов управления. Тут и элементы, позволяющие задавать стиль компоновки для компонент на форме, и новые панели инструментов, меню, строк состояния, а также большое количество элементов по работе с данными. Все эти элементы позволяют создавать профессиональный интерфейс. На сайте компании Microsoft доступно множество примеров, демонстрирующих возможности новых элементов.

Функции отладки

Следующая возможность, которая необходима программистам — простой и удобный отладчик. Visual Basic 2005 поддерживает такую функцию, как **Edit and Continue**, что позволяет в режиме отладки произвести изменения в коде, например, значения некоторых переменных, и продолжить выполнение программы. Разработчики постарались, и эта возможность отладчика работает достаточно быстро, что делает для программистов незаметным процесс перекомпиляции кода.

Как и в VB 6, чтобы установить точку останова, достаточно щелкнуть на специальной панели слева от окна кода. Среда поместит специальный элемент в виде шарика в выбранное место, щелкнув на который, можно убрать или на время отключить точку останова. В Visual Basic 2005 появилось знакомое программистам на VB 6 окно **Immediate**, позволяющее выполнять различные команды в процессе отладки. Также поддерживается окно **Console**, которое позволяет видеть все сообщения, посылаемые на консоль, не покидая отладчик. И, наконец, отладчик обладает возможностью, позволяющей не только просматривать значение переменной, но и детальную информацию о содержимом объекта на любом уровне вложенности.

Интересная возможность появилась при работе с окнами. Раньше приходи-

New Filename	Original Filename	File Type	Status	Errors	Warnings	Total Issues
Global update issues:						
None						
PicForm.vb	PicForm.frm	Form	Upgraded with issues	6	0	6
Upgrade Issues for PicForm.frm:						
#	Severity	Location	Object Type	Object Name	Property	Description
1	Compile Error	ClearPictureBox_Click	PictureBox	Picture1	Cls	PictureBox.method Picture1.Cls was not upgraded.
2	Compile Error	Timer1_Timer	PictureBox	Picture1	Cls	PictureBox.method Picture1.Cls was not upgraded.
3	Compile Error	Timer1_Timer	PictureBox	Picture1	PaintPicture	PictureBox.method Picture1.PaintPicture was not upgraded.
4	Compile Error	Timer1_Timer	PictureBox	Picture1	Print	PictureBox.method Picture1.Print was not upgraded.
5	Design Error	(Layout)	PictureBox	Picture1	AutoRedraw	PictureBox.property Picture1.AutoRedraw was not upgraded.
6	Design Error	(Layout)	PictureBox	Picture1	FillStyle	PictureBox.property Picture1.FillStyle was not upgraded.
1 File(s)				Forms: 1	Upgraded: 1	Not upgraded: 0
					6	0
					6	6

Upgrade Settings
 Log File: Drawing.log
 Migrate Project To: WinExe
 Output Directory: C:\Program Files\Microsoft Visual Studio .NET 2003\VB7\VB Samples\Migration\WalkThrough\Drawing.NET
 Project Name: Drawing
 Project Path: C:\Program Files\Microsoft Visual Studio .NET 2003\VB7\VB Samples\Migration\WalkThrough\Drawing.vbp

Отчет о преобразовании проекта содержит перечень всех проблем, с которыми столкнулся Мастер преобразования

лось долго перетягивать окна, чтобы получить для себя наиболее удобный интерфейс, а наложить два окна друг на друга было совсем даже не просто. Теперь Visual Basic 2005 Express обладает улучшенной системой стыковки окон, которая позволит быстро поместить окно в нужное место экрана. Если же необходимо наложить два окна, то достаточно совместить их заголовки. Вот как выглядит эта система при попытке перетащить окно:

И последняя возможность оболочки, которую хотелось бы рассмотреть, — возможность сохранять пользовательские настройки и загружать их на другой машине. Это можно сделать с помощью пункта меню **Tools Import/Export Settings...** Дело в том, что нередко приходится менять один ПК на другой. Это может быть связано с выездами на лекции, с модернизацией оборудования или плановой переустановкой. И если раньше приходилось первым делом заниматься настройкой оболочки «под себя», то теперь в этом нет необходимости, так как можно сохранить все настройки в один файл и загрузить их на другой машине.

Проблемы миграции

Следующий и последний вопрос, который мы рассмотрим, — миграция приложений, написанных в VB 6, на платформу .NET. Эта проблема возникает очень часто, а учитывая то, что с появлением Visual Basic 2005 Express отпадает проблема покупки средства разработки, то для многих мелких компаний этот вопрос станет очень актуальным. Ведь на сегодняшний день все еще существует огромное количество приложений, написанных с использованием Visual Basic 6, эти приложения были написаны годы назад и все еще продолжают служить верой и правдой. Но уже в марте 2005 года закончился срок поддержки VB 6, и, хотя этот продукт все еще будет находиться в состоянии «расширенной поддержки», это лишний повод разработчикам подумать о переходе на платформу .NET, тем более что Microsoft прикладывает много усилий для того, чтобы сделать процесс миграции максимально простым.

Основным инструментом для переноса приложений Visual Basic 6 на платформу .NET является Visual Basic

Upgrade Wizard, который встроен во все редакции Visual Studio 2005, в том числе и в Visual Basic 2005 Express. Этот мастер активизируется, когда вы пытаетесь открыть проект VB 6 (*.vbp), и выполняет преобразование вашего проекта в решение на Visual Basic 2005. При этом поддерживается 2 типа проектов: .EXE и библиотеки .DLL. Следует, однако, отметить, что в текущей бета-версии мастер иногда работает недостаточно стабильно. Например, он может отказаться выполнять преобразование, если на компьютере установлены форматы величин, отличные от английских, но к финальному релизу проблему должны устранить. После завершения преобразования создается проект на Visual Basic .NET 2005, в который помещаются все формы и файлы из старого проекта. Дополнительно в проект будет входить HTML-файл, содержащий отчет о преобразовании — все проблемы, с которыми столкнулся мастер, будут отражены в нем. Используя сформированный отчет, программист может перейти в указанное место кода и устранить ошибку.

В оболочку Visual Basic Express встроен еще один инструмент, призванный помочь в миграции с VB 6. Это своеобразный переводчик, который доступен через пункт меню **Tools|Upgrade Visual Basic 6 Code**. Выбрав этот пункт меню, вы вызываете мастер, куда можно поместить старый код, а также можно указать ссылки на COM-объекты, используемые в коде VB 6. Благодаря последней возможности, можно корректно обновить код, работающий с нестандартными COM-компонентами. Код, сформированный мастером в процессе преобразования, вставляется в основной код приложения в том месте, где стоял курсор перед вызовом мастера. При этом возможные проблемы с преобразованием будут отмечены соответствующими комментариями.

Оба инструмента используют один и тот же компонент для преобразования, так что если с вашим кодом не справился Upgrade Wizard, то вряд ли чем-то поможет второй мастер.

Для обеспечения более простого переноса старых приложений на новую платформу Microsoft предоставляет специальную библиотеку Visual Basic

Compatibility Library, которая предназначена для использования инструментами автоматического обновления кода. И хотя она предоставляет функции VB 6, которые реализованы средствами .NET, использовать ее при реализации новых проектов не рекомендуется, т. к. поддержка этой библиотеки в будущем не гарантируется. Лучше применять новые средства платформы .NET.

Обновление платформы

Кроме автоматических утилит миграции приложений Microsoft приложила много дополнительных усилий для облегчения перехода на новую платформу. Хотя в Visual Basic 2005 встроена вторая по счету версия языка для платформы .NET, но даже в ней есть новшества, которые предназначены для облегчения разработчикам миграции на новую платформу с VB 6. Например, в версии 2005 доступ к элементам управления можно осуществлять через коллекцию Controls, как и в VB 6:

```
Form1.Controls(«btnOK»).Caption = «&OK»
```

Кроме того, добавлена поддержка экземпляров форм, создаваемых по умолчанию. Если в Visual Basic 2003, перед тем как отобразить форму на экране, приходилось создавать экземпляр класса формы, то в версии 2005 в этом уже нет необходимости — система сама создает и инициализирует экземпляры форм по умолчанию. Доступ к любой форме в проекте можно получить с помощью коллекции `My.Forms`, которая во многом аналогична глобальной коллекции `Forms` из VB 6.

Несмотря на то что Microsoft позиционирует Visual Basic .NET как абсолютно новый язык, авторы статьи считают, что проблем с переходом со старой версии практически не возникает. Так, за последние два года нам пришлось работать более чем с сотней специалистов, использующих VB 6, в число которых входили как преподаватели программирования, так и действующие программисты. Почти для всех переход оказался безболезненным. Знания, полученные ранее, позволили нам быстро приступить к работе, изучая при этом новые возможности, которые предоставляет платформа .NET.