Visual Basic .NET

VB .NET: первый опыт

Не так давно вышла новая версия интегрированной среды разработки Visual Studio — Visual Studio .NET. Изменилось все: интерфейс, компоновка, структура языков программирования. Попробуем разобраться в основных нововведениях пакета и напишем простейшую программу на Visual Basic .NET.

Глоссарий

Это важно знать

- **Решение (solution)** — аналог проектной группы в VB 5/6, но с изменениями и дополнениями (термин наверняка известен тем, кто работал в программе Visual InterDev 6).
- **Проект** — часть решения, выполняющая свои конкретные задачи.
- **Класс** — объект приложения со своими свойствами, методами и событиями. Класс можно наследовать и перегружать.
- **Интерфейс** — разновидность класса.
- **Наследование** — копирование одного класса другим.
- **Перегрузка** — изменение процедур и функций наследованного класса.

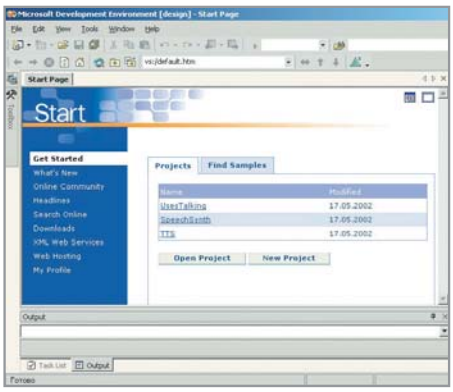
Эта статья будет полезна прежде всего программистам, переходящим с VB 5/6 на VB .NET. Однако она может быть интересна также для всех пользователей, желающих быть в курсе последних разработок в области объектно-ориентированных языков программирования. Даже если у вас нет опыта программирования на Visual Basic, вы можете вместе со мной написать небольшую программу для подсчета количества заказанных журналов Chip.

Установка и первый запуск

Установка Visual Studio .NET в полной комплектации длится три часа, если устанавливать только VB .NET — полтора часа. Новая версия занимает около 1 Гбайт на вашем жестком диске (включая базу данных Microsoft Developer's Network и примеры программ).

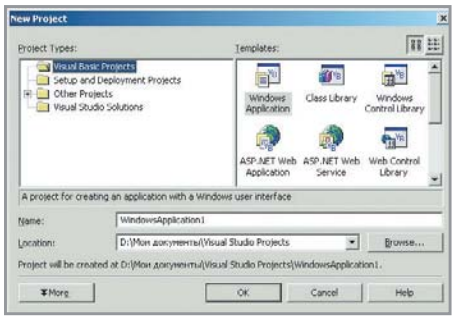
В целом процесс нареканий не вызывает, однако VS .NET не устанавливается на системы Windows 9x/Me, которые сейчас наиболее распространены в среде пользователей. В принципе, VS .NET может запуститься на машине с 64 Мбайт ОЗУ. В действительности же вам придется ждать выполнения простейшего действия в течение длительного времени. Опыт показывает, что минимальный объем памяти для комфортной работы составляет 96 Мбайт.

Теперь все программы пакета содержатся в одной оболочке, поэтому для работы со всеми приложениями VS .NET достаточно запустить программу Microsoft Visual Studio .NET, выбрав в меню «Пуск -> Программы» соответствующий пункт. Через некоторое время появится главное окно пакета программ (рис. 1). »



▲ Рис. 1. Новый интерфейс — новые возможности

▲ Рис. 2. Восстановите старые сочетания клавиш VB5/6



▲ Рис. 3. Создание нового проекта. Планов громадье

» Если вы привыкли к работе в среде Visual Basic 5/6 и не хотите переучиваться, то можете по-прежнему использовать «горячие клавиши» VB 5/6. Для этого щелкните по ссылке «My Profile» (рис. 2) и выберите необходимую конфигурацию — «Visual Basic 6».

Создаем проект

Для создания проекта переключитесь на раздел «Get Started» страницы Start Page и выберите пункт «New Project» (рис. 3).

На закладке «Visual Basic Projects» выберите тип приложения «Windows Application». Отмечу, что VS .NET по-прежнему предоставляет большое количество типов проектов — привычные SDI/MDI-приложения, консольные приложения, библиотеки классов и т. п.

Введите название «my first app in vb.net» в поле «Name» и щелкните мышкой по кнопке «OK». Вслед за этим начнется создание проекта.

После этого появится главное окно редактирования решения (рис. 4), в котором сосредоточено все управление проектом. Очень хорошее впечатление производит панель «Toolbox» (рис. 5). Вообще весь интерфейс VS .NET а-ля Office XP кажется достаточно продуманным, удобным и эргономичным.

В VS .NET можно получить оперативную справку по нескольким темам, просмотреть полную справку в окне «Contents» (рис. 11), а также вызвать предметный указатель и поиск по теме. Новинкой является окно «Dynamic Help», которое может выдать оперативную справку по теме. Это позволяет быстро освоить новые возможности Visual Studio .NET.

Создадим элементы пользовательского интерфейса и установим свойства. Дважды щелкните на значке «Form1.vb» в окне «Solution Explorer». Вслед за этим откроется форма пользовательского интерфейса.

Наведите курсор на значок «Toolbox». Когда всплывающая панель покажется на экране, щелкните кнопку «Auto Hide» — и она останется.

Создайте один элемент класса GroupBox, две кнопки, четыре флажка внутри элемента GroupBox и один элемент LinkLabel.

Придайте значение «Отличная форма» свойству «Text» самой формы.

Задайте свойства, указанные в табл. 1. В результате должен получиться примерно такой макет, какой изображен на рис. 7.

Написание кода для формы и флажков

Дважды щелкните по форме. Откроется окно редактирования программного кода (рис. 8). Заметьте, теперь весь скрытый код формы виден в окне редактора.

После строки Inherits System.Windows.Forms.Form объявим переменную JournalCount:

```
Dim JournalCount as Short=0
```

В VB .NET можно придавать начальное значение переменной при объявлении. Следует обратить внимание на то, что тип Short эквивалентен типу Integer из VB6, тип Integer — Long, и т. д.

Выберите объект «CheckBox1» в окне «Form1.vb» и введите следующий программный код:

```
If CheckBox1.Checked = True Then JournalCount = JournalCount + 1 : Exit Sub
JournalCount = JournalCount — 1
```

Выберите объект «CheckBox2» и напишите аналогичный код (изменилось только имя объекта): »

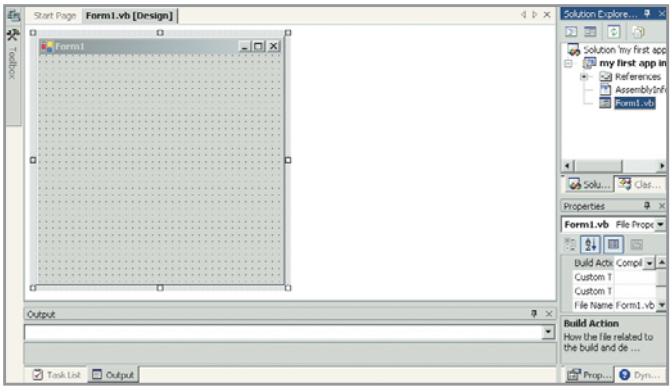


Рис. 4. Окно редактирования решения ▸

Рис. 5. Плавающая панель «Toolbox» ▸

Для сведения

Почему именно Visual Studio .NET?

Действительно, почему? Ведь существует достаточное количество интегрированных средств разработки, в том числе и RAD (Rapid Application Development) — Borland C++ Builder/Delphi/Java Builder. Скажем так, каждая интегрированная среда разработки по-своему хороша. При помощи других IDE (Integrated development environment) вы также можете писать интересные программы.

Ключом потенциального успеха Visual Studio .NET является возможность не только писать приложения на разработанном Microsoft высокоуровневом языке C# («Си-шарп»), но и транслировать свой код, написанный на Visual Basic, C++, Java или любом другом языке (к примеру, на Python) в код на C#.

В принципе, C# переносим на любую возможную платформу не в последнюю очередь благодаря архитектуре CLR (Common Language Runtime), доказательством чему являются проекты Mono (http://go-mono.net) и DotGNU (http://dotgnu.org). Однако наиболее полно C# реализован именно в Visual Studio .NET, поэтому тем разработчикам, которые планируют в ближайшем году заниматься смежными Visual Basic/C# проектами, мы рекомендуем использовать именно VS .NET.

Невозможно утверждать наверняка, что C# со временем поглотит Visual Basic или C++, поскольку главный противник Microsoft на этом поле битвы, Sun Microsystems, также не дремлет и понемногу продвигает свою технологию Sun One на основе Java 2, да и популярность скриптовых языков постоянно возрастает — в последнее время это особенно касается Ruby.

Ну а что произойдет с Visual Studio .NET и C# в дальнейшем — покажет время.

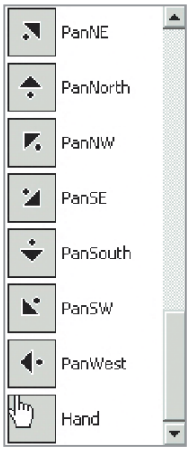
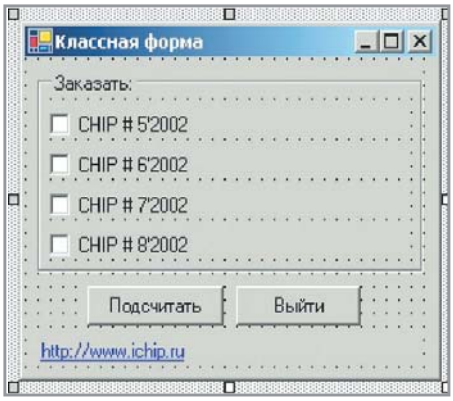


Рис. 6. Выбор курсоров с просмотром изображения. Дождались!



▲ Рис. 7. Вид формы после установки свойств

```
If CheckBox2.Checked = True Then JournalCount = JournalCount + 1 : Exit Sub
JournalCount = JournalCount — 1
```

Проделайте то же самое с флажками «CheckBox3» и «CheckBox4».

Таким образом, программный код для всех флажков един, необходимо только менять имя объекта! Это очень удобный прием, и он широко применяется на практике.

Обработка нажатия кнопок

Выберите объект «Button1» и введите код для завершения программы — «End». После этого выберите другую кнопку и введите этот код:

```
MsgBox("Вы заказали "&Trim(Str(JournalCount))&"# CHIP",MsgBoxStyle.Information _,"Заказ журналов CHIP")
```

Тестовый запуск

Нажмите F5 или выберите пункт меню «Debug -> Start». Через некоторое время в окне «Output» увидите надпись «Done» (рис. 9) и появится главное окно программы. Пометьте несколько флажков в окне и нажмите кнопку «Подсчитать». Результат этого действия показан на рис. 13.

После успешного тестирования закройте главное окно нашей программы — вы вернетесь в среду разработки.

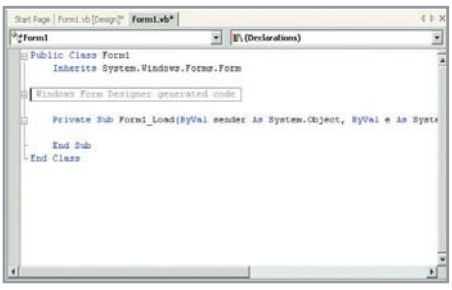
Разбираемся в коде

Для просмотра кода следует в окне «Solution Explorer» выбрать форму «Form1.vb» и щелкнуть по кнопке «View Code».

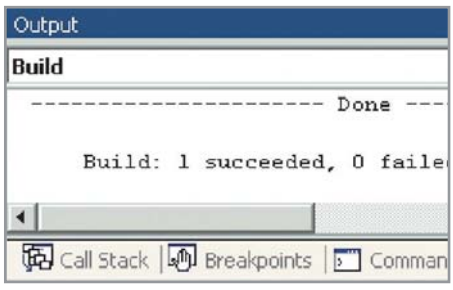
Стоит заметить, что ваш листинг может и должен отличаться от приведенного нами (в большинстве случаев это несущественные различия в служебных данных Windows Forms)!

Код нашей программы начинается со строки «Public Class Form1». Это самая важная строка — она определяет форму «Form1» как класс. Далее идет атрибут «Inherits». Он позволяет наследовать один класс или интерфейс другим классом или интерфейсом. В данном случае — класс «System Windows.Forms.Form» наследуется классом «Form1». Более подробную справку можно получить, поставив курсор перед словом «Inherits» в окне редактора (рис. 10).

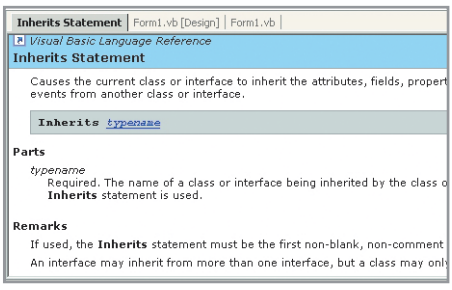
Далее идут стандартные объявления переменных и служебный код «Windows Forms», определяющий все свойства объектов формы, изменять которые не рекомендуется. После служебного кода и объ-



▲ Рис. 8. Новый редактор кода



▲ Рис. 9. Окно «Output» — информация о компиляции



▲ Рис. 10. Встроенная справка по атрибуту «Inherits»

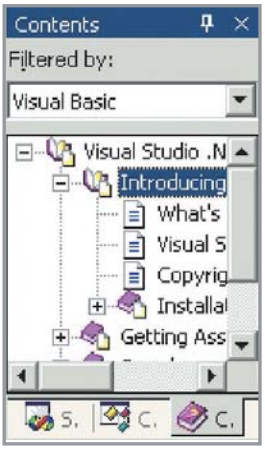


Рис. 11. Окно содержания справки

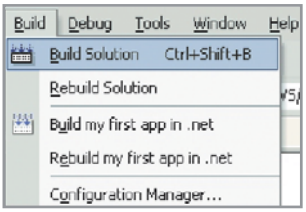


Рис. 12. Построить решение!

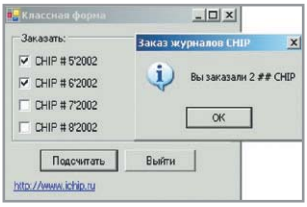


Рис. 13. Сколько мы заказали?

Рис. 14. Мастер преобразования проектов

» явлений переменных идут процедуры обработки объектов. Завершающая строка — «End Class». После нее ни одна строчка кода не должна исполняться в этом классе (в данном случае — в форме).

Постройка решения

Для постройки решения выберите «Build -> Build Solution» (рис. 12). После постройки в папке «\Мои документы\Visual Studio Projects\my first app in vb.net\bin\» появится файл «my first app in vb.net.exe».

Проблемы преобразования проектов

VS .NET позволяет преобразовать проект VB 5/6 в решение VS .NET. Для этих целей существует мастер «Upgrade Wizard» (рис. 14). Для того чтобы его вызвать, необходимо открыть любой проект в формате VB 5/6 (VBP). Однако не стоит считать, что старый проект преобразуется в новый формат безупречно. Ведь многие архаизмы VB 5/6 (вроде графических методов «Line», «Circle», «Scale», переходов «GoSub») были удалены, а некоторые старые классы — модифицированы.

Приведу пример: объект «Clipboard» теперь не содержит методов «GetData» и «SetData», однако присутствуют «SetDataObject» и «GetDataObject». Конечно, с использованием объектов данных гибкость обработки Буфера обмена возрастает, но все же такой синтаксис непривычен для многих программистов.

Сравните:

```
' Код на VB 6
Text1.SelText = Clipboard.GetText

' Код на VB.NET
Dim datobj As New
System.Windows.Forms.DataObject()
datobj =
System.Windows.Forms.Clipboard.GetData-
Object()
Text1.SelText = datobj.GetData(System.Win-
dows.Forms.DataFormats.Text)
```

Новые понятия+старый код=?

Как вы уже видели, код на VB 6 и код на VB .NET серьезно различаются. Причем не столько тем, что появились новые определения. Просто вся структура языка Visual Basic была кардинально переопределена.

Самые главные изменения и дополнения коснулись работы с классами: теперь, в частности, их можно наследовать и перегружать.

Наследование и перегрузка — вот чего не хватало многим программистам на VB 6 для полноценной работы с использованием объектно-ориентированного программирования!

То есть, если при работе с шестой версией языка мы не могли быстро и удобно создавать одинаковые классы, то теперь при помощи наследования с этим справится даже ребенок. Кроме того, если наследованные классы должны незначительно различаться, не надо отказываться от наследования: с помощью перегрузки вы можете изменить параметры вызываемой процедуры и функции без изменения наследуемого класса!

Очень значительно изменилась и сама иерархия программного кода: теперь самым главным понятием является так называемое пространство имен (namespace). Далее идут интерфейсы, затем классы, после них объекты и т. д.

Но это далеко не полный список нововведений VB .NET.

Выводы

Переходить на Visual Basic .NET с Visual Basic 6 или нет — ваше личное дело. Однако хотя бы ознакомиться с этим продуктом стоит. Тем более что после 2005 года Microsoft поддерживать программистов VB 6 уже не будет... ■ ■ ■ Николай Амеличев

Элемент	Свойство	Значение
GroupBox1	Text	Заказать
Button1	Text	Выйти
Button2	Text	Подсчитать
CheckBox1	Text	CHIP #5'2002
CheckBox2	Text	CHIP #6'2002
CheckBox3	Text	CHIP #7'2002
CheckBox4	Text	CHIP #8'2002
LinkLabel1	Text	http://www.ichip.ru
LinkLabel1	Cursor	Hand (рис. 6)

Табл. 1. Введите указанные свойства для объектов на созданной нами форме

Microsoft Visual Studio .NET

Разработчик ▸ Microsoft

Сайт разработчика ▸ www.microsoft.com/rus/msdn/vs

Операционная система ▸ Windows

Размер ▸ 1,2 Гбайт при стандартной установке; 2,6 Гбайт при полной

Условия распространения ▸ shareware

Цена ▸ варьируется в зависимости от варианта комплектации