



Самоучитель

Джо Майо

Microsoft® Visual Studio 2010



Настройка среды разработки

Вводный курс и особенности
использования языков C# и VB.Net

Разработка основных типов
приложений

Современные технологии
программирования: WPF, WCF,
ASP.NET, Silverlight и др.

Microsoft® Visual Studio® 2010

A Beginner's Guide

Joe Mayo



New York Chicago San Francisco
Lisbon London Madrid Mexico City
Milan New Delhi San Juan
Seoul Singapore Sydney Toronto

Джо Майо

Самоучитель
Microsoft®
Visual Studio
2010

Санкт-Петербург

«БХВ-Петербург»

2011

УДК 681.3.06
ББК 32.973.26-018.2
М14

Майо Дж.

М14 Самоучитель Microsoft Visual Studio 2010. — СПб.: БХВ-Петербург, 2011. — 464 с.: ил.

ISBN 978-5-9775-0609-0

Показано создание различных типов приложений в интегрированной среде разработки Microsoft Visual Studio 2010. Рассмотрены основы программирования на языках C# и VB, работа с решениями, проектами, сборками и библиотеками классов. Описаны инструменты, предназначенные для анализа и отладки кода, поиска и исправления ошибок. Рассмотрена работа с базами данных с помощью языка интегрированных запросов LINQ. Приведена информация о языках XML и XAML. Описаны основные концепции работы с системой Windows Presentation Foundation, технология Silverlight, построение Web-приложений с помощью технологии ASP.NET MVC, создание Web-сервисов с помощью Windows Communications Foundation. Рассмотрено создание собственной программы-мастера для работы над проектами, шаблонов для автоматизации генерируемых фрагментов кода и рутинных задач, добавочных модулей и др.

Для программистов

УДК 681.3.06
ББК 32.973.26-018.2

Original edition copyright © 2010 by the McGraw-Hill Companies. All rights reserved. Russian edition copyright © 2010 year by BHV – St.Petersburg. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without the prior written permission of the publisher.

Оригинальное издание выпущено McGraw-Hill Companies в 2010 году. Все права защищены. Русская редакция издания выпущена издательством БХВ-Петербург в 2010 году. Все права защищены. Никакая часть настоящей книги не может быть воспроизведена или передана в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на то нет письменного разрешения издательства

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шлишин</i>
Зав. редакцией	<i>Григорий Добин</i>
Перевод с английского и редактирование	<i>Ольги Кокоревой</i>
Компьютерная верстка	<i>Наталья Караваевой</i>
Корректор	<i>Виктория Пиотровская</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 02.08.10.

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

Тираж 2000 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.60.953.Д.005770.05.09 от 26.05.2009 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 978-0-07-166895-8 (англ.)
ISBN 978-5-9775-0609-0 (рус.)

© 2010 by The McGraw-Hill Companies
© Перевод на русский язык "БХВ-Петербург", 2010

Оглавление

Введение	1
ЧАСТЬ I. Базовая информация о Visual Studio 2010	5
Глава 1. Первое знакомство с Visual Studio 2010	7
Что позволяет делать Visual Studio 2010?	7
Автоматически генерируемый код	8
Опыт быстрого кодирования (Rapid Coding Experience).....	8
Все необходимое — всегда под рукой	9
Настраиваемость и расширяемость	9
Установка Visual Studio 2010.....	9
Навигация по среде Visual Studio 2010	17
Меню	18
Инструментальная панель (Toolbar).....	19
Рабочая область (Work Area).....	20
Инструментальный набор (Toolbox)	20
Окно Solution Explorer	20
Строка состояния.....	20
Управление окнами VS	21
Распахивание и сворачивание окон	21
Пристыковка окон	22
Плавающие окна.....	24
Окна с вкладками	24
Открытие и закрытие окон	25
Модификация настройки среды после установки	25
Экспорт выбранных параметров настройки среды.....	26
Импорт сохраненных настроек среды разработчика	29
Сброс настроек к стандартным значениям	32
Знакомство с типами проектов Visual Studio	34
Проекты Windows.....	36
Web-проекты.....	37
Проекты Office.....	37

Проекты SharePoint	38
Проекты по работе с базами данных (Database Projects).....	38
Заключение	38

Глава 2. Необходимый минимум знаний о C# и VB.NET: базовый синтаксис..... 39

Создание простейшего проекта	40
Исследование "скелета" кода будущей программы.....	42
Обзорная информация о редакторе кода VS	47
Средства обнаружения классов и их членов.....	48
Установка параметров настройки редактора.....	50
Экономия времени при помощи фрагментов (Snippets).....	51
Кодирование выражений и утверждений	53
Использование технологии Intellisense	54
Запуск программ.....	56
Простейшие типы и выражения.....	57
Тернарные операторы C# и операторы Immediate If в VB	60
Перечисления.....	61
Ветвления	62
Циклы	67
Заклучение	72

Глава 3. Изучение основ C# и VB.NET: типы и члены 73

Создание классов	73
Синтаксис класса.....	73
Наследование классов.....	75
Написание методов	77
Декларирование и использование методов.....	77
Объявление параметров и передача аргументов.....	81
Возвращение данных и использование значений, возвращаемых методами	84
Автоматически генерируемые фрагменты кода методов.....	86
Кодирование полей и свойств.....	86
Объявление и использование полей	86
Объявление и использование свойств	89
Автоматически генерируемый фрагмент кода для свойства	92
Заклучение	92

Глава 4. Необходимый минимум знаний о языках C# и VB.NET: среднеуровневый синтаксис 93

Разбираемся с делегатами и событиями	93
События.....	94
Делегаты.....	97
Завершение кода делегатов и обработчиков	99

Реализация интерфейсов	100
Создание интерфейса	101
Написание классов, реализующих интерфейсы	102
Написание кода, использующего интерфейсы	105
Автоматически генерируемый фрагмент для интерфейсов	110
Применение массивов (Arrays) и общих типов (Generics).....	110
Программирование с использованием массивов.....	111
Кодирование родовых коллекций (Generics).....	113
Заключение	114

ЧАСТЬ II. Изучаем среду разработки VS 2010 115

Глава 5. Создание и построение проектов 117

Конструирование решений и проектов.....	117
Создание новых проектов	118
Ориентируемся в окне Solution Explorer.....	120
Исследование настройки свойств	122
Компиляция приложений	135
Построение решений и проектов	135
Перестройка решений и проектов	136
Очистка решений и проектов	137
Управление зависимостями и порядком построения.....	137
Управление параметрами компиляции	139
Перемещение по проекту в режиме просмотра классов.....	142
Использование конструктора классов (Class Designer).....	143
Заключение	147

Глава 6. Отладка с помощью Visual Studio 148

Код, на примере которого в данной главе будут демонстрироваться приемы отладки.....	148
Инструменты кодирования, упрощающие разработку.....	153
Конфигурирование отладочного режима	155
Установка точек останова	161
Создание точки останова.....	162
Индивидуальная настройка точки останова	163
Управление точками останова	164
Пошаговое выполнение кода	165
Исследование состояния приложения	167
Окна <i>Locals</i> и <i>Autos</i>	168
Окно <i>Watch</i>	169
Окно <i>Immediate</i>	169
Окно <i>Call Stack</i>	170

Окно <i>QuickWatch</i>	171
Наблюдение переменных с привязкой к источнику	171
Работа с IntelliTrace	173
Решение проблем с помощью отладчика VS	174
Программа с ошибками	174
Поиск ошибки	178
Исправление первой ошибки	182
Отладка и решение проблем <code>NullReferenceException</code>	183
Заключение	188
Глава 7. Работа с данными.....	189
Работа с базами данных.....	190
Вводная информация о Server Explorer	190
Создание базы данных	191
Добавление таблиц.....	192
Установка связей между таблицами по вторичным ключам	196
Добавление хранимых процедур	200
Конфигурирование опций базы данных.....	202
Изучаем Language Integrated Query (LINQ)	203
Выполнение запросов к коллекциям объектов с помощью LINQ.....	203
Создание проекции LINQ с анонимными типами.....	206
Использование LINQ для сортировки результирующей коллекции	208
Обработка данных с помощью LINQ to SQL	208
Настройка LINQ to SQL	209
Работа с LINQ to SQL Designer	210
Введение в запросы с помощью LINQ to SQL.....	212
Выполнение запросов к множеству таблиц	214
Ввод данных с помощью LINQ to SQL	220
Обновление данных с помощью LINQ to SQL	221
Удаление данных с помощью LINQ to SQL	223
Заключение	224
ЧАСТЬ III. Разработка приложений с помощью VS 2010	225
Глава 8. Построение настольных приложений с помощью WPF.....	227
Начало работы над проектом WPF.....	228
Изучение расположения элементов (Layout)	229
Макет <i>Grid</i>	230
Макет <i>StackPanel</i>	232
Макет <i>DockPanel</i>	233
Макет <i>WrapPanel</i>	234
Макет <i>Canvas</i>	235

Использование элементов управления WPF	236
Основные окна для работы с элементами управления	236
Установка свойств.....	238
Обработка событий.....	238
Написание кода обработчиков событий	242
Работа с данными в WPF.....	244
Настройка источника данных	245
Конфигурирование поля раскрывающегося списка	251
Чтение и сохранение данных	253
Использование макета <i>DataGrid</i>	254
Заключение	257
Глава 9. Разработка Web-приложений с помощью ASP.NET MVC	259
Разбираемся с ASP.NET MVC	259
Создание нового проекта ASP.NET MVC.....	261
Создание моделей.....	263
Построение контроллеров	263
Отображение представлений.....	266
Организация файлов представления.....	267
Назначение файлов MasterPage.....	268
Управление маршрутизацией	272
Сборка приложения, предназначенного для работы с базой данных клиентов	274
Создание репозитория.....	275
Создание контроллера Customer	279
Отображение списка клиентов.....	279
Добавление нового клиента.....	285
Редактирование записей для существующих клиентов.....	290
Удаление записи о клиенте из базы данных	292
Заключение	294
Глава 10. Разработка приложений Silverlight.....	295
Запуск проекта Silverlight.....	295
Навигация в окне Silverlight Designer	300
Использование элементов управления Silverlight	301
Запуск Silverlight "вне браузера" (Out-of-Browser, OOB).....	304
Развертывание приложений Silverlight	308
Заключение	309
Глава 11. Развертывание Web-сервисов с помощью WCF	310
Создание нового проекта WCF.....	311
Указание соглашения (Contract) с WCF-интерфейсами.....	312
Изучение соглашения, сгенерированного VS.....	312

Создание собственных соглашений	315
Реализация логики с помощью классов WCF.....	318
Хостинг для сервиса WCF.....	324
Общие процедуры настройки хостинга	325
Установка IIS 7 в Windows 7	326
Создание Web-сайта под IIS 7 в Windows 7.....	328
Развертывание сервиса WCF на IIS.....	332
Взаимодействие приложений с сервисом WCF	337
Создание ссылки на Web-сервис	337
Кодирование вызовов к Web-сервису	340
Развертывание клиентского приложения, взаимодействующего с Web-сервисом	347
Создание Web-сервиса на Web-сайте	349
Заключение	349

ЧАСТЬ IV. Расширение возможностей VS 2010 **351**

Глава 12. Индивидуальная настройка среды разработки **353**

Реализация индивидуальных шаблонов	354
Создание новых шаблонов проектов.....	354
Модификация проекта	355
Экспорт шаблона проекта	356
Использование нового шаблона проекта	358
Создание шаблонов новых элементов.....	359
Создание индивидуальных автоматически генерируемых фрагментов кода	365
Создание автоматически генерируемого фрагмента кода	366
Управление библиотекой автоматических фрагментов	372
Написание макросов	374
Запись макроса	374
Сохранение макроса.....	379
Редактирование макросов.....	380
Заключение	386

Глава 13. Расширение возможностей Visual Studio 2010..... **387**

Разработка дополнительного модуля Visual Studio.....	388
Запуск программы-мастера Add-In Project Wizard.....	388
Исследование решения, созданного программой-мастером Add-In Wizard.....	393
Изучение структуры класса Connect	395
Добавление функциональных возможностей в дополнительный модуль	401
В каком направлении двигаться дальше	417
Заключение	418

ПРИЛОЖЕНИЯ	419
Приложение 1. Введение в XML	421
Редактор кода XML в VS 2010	421
Префиксы XML.....	422
Элементы XML	422
Атрибуты	423
Пространства имен.....	423
Меню XML	425
Конфигурирование опций XML Opt	425
Заключение	425
Приложение 2. Введение в XAML	426
Запуск проекта WPF	426
Элементы как классы.....	427
Атрибуты как свойства.....	428
Исполнение документа XAML	428
Элементы свойств	429
Расширения разметки	431
Заключение	433
Приложение 3. О русской версии Visual Studio 2010.....	434
Предметный указатель	437

Благодарности

Работа над любой книгой — это не только труд автора. Свой вклад в эту работу вносит весь коллектив высококвалифицированных профессионалов, каждый из которых выполнял свою часть работы, помогая автору. Я хочу поблагодарить лично всех тех членов группы подготовки этой книги, без которых ее появление было бы невозможным.

Джейн Браунлав (Jane Brownlow), ответственный редактор, оказала бесценную помощь в запуске проекта этой книги и все время помогала мне оставаться на правильном пути. Мег Морин (Megg Morin), рецензент издательства McGraw-Hill, приняла бразды правления от Джейн и помогла пройти остаток пути. Джойя Энтони (Joia Anthony), координатор проекта, следила за сдачей отдельных глав. Мэдху Бхардваж (Madhu Bhardwaj), менеджер проекта и Пэтти Мон (Patty Mon), заведующий редакцией, помогали в координации работы по редактированию и верстке. Всех этих, и многих других сотрудников McGraw-Hill, которые помогали выпуску книги, я приношу огромную благодарность за их вклад в общую работу, терпение и профессионализм.

Отдельной благодарности заслуживает Рой Огборн (Roy Ogborn), технический редактор этой книги. Я знаком с Роем уже много лет, и всегда радуюсь, когда он соглашается редактировать мои книги. Помимо того, что он очень тщательно и внимательно отлавливает все ошибки, в некоторых областях он дает и очень ценные советы касательно того, на каких технологиях следует поставить акцент, как сделать изложение более понятным для новичка или о том, какой язык выбрать для достижения той или иной цели. Огромное спасибо Рою за его выдающиеся редакторские качества и ценные советы.

Джо Майо (Joe Mayo)

Введение

Visual Studio уже в течение многих лет является ведущей интегрированной средой разработки (Integrated Development Environment, IDE) среди всех инструментальных средств разработчика, поставляемых Microsoft. Новая версия этого ПО, которой посвящается эта книга, представляет собой естественный результат эволюции, являющийся логическим продолжением всех предшествующих версий. В этой книге показано, как максимально использовать себе на благо мощный потенциал Visual Studio 2010, приобрести новые знания и навыки, а также повысить производительность труда при разработке приложений. Основное внимание в книге уделено написанию приложений для .NET (произносится как "Dot Net") — платформы Microsoft для разработки различных типов ПО, от обычных программ до Web-приложений.

Как следует из ее названия, книга представляет собой руководство для начинающих. Однако мнения о том, кто такой "новичок", могут сильно различаться, поэтому давайте сначала обсудим, кого следует считать новичком в контексте этой книги. Я, как ее автор, не считаю, что вы совсем ничего не знаете о программировании. Я предполагаю, что вы все же имеете некоторое представление о том, что такое программирование, хотя бы в общих чертах, и способны, как минимум, написать простой командный файл, макрос или командный сценарий, чтобы автоматизировать какую-нибудь рутинную задачу. Кроме того, в контексте этой книги новичками следует считать и всех тех, кто не знаком с Visual Studio, но имеет опыт разработки ПО с использованием каких-либо других технологий, например, Cobol, Dreamweaver или Java. Каким бы ни был ваш предварительный опыт и базовый уровень знаний, изложение в книге построено таким образом, чтобы идти от простого к более сложному и постепенно приобретать опыт создания приложений с помощью Visual Studio 2010.

Данная книга состоит из 13 глав, разбитых на 4 части, и содержит три приложения, в которых приведен справочный материал. Вот краткий обзор материала, представленного в этой книге:

- Часть I: *"Базовая информация о Visual Studio 2010"*. В состав этой части входят главы 1 — 4. Глава 1 содержит краткое описание VS и рассказывает о предоставляемых этой средой преимуществах и типах приложений, которые можно создавать с ее помощью. Вкратце описывается процедура установки VS, после чего даются советы и рекомендации по выбору опций установки и каталогов, в которые производится установка. Главы 2 — 4 представляют собой вводный курс по C# и VB — двум наиболее популярным у разработчиков языкам

программирования, поддерживаемым в VS. Они содержат тот минимум базовой информации, который необходим для понимания дальнейшего материала, излагаемого в данной книге. По мере ее чтения вы будете знакомиться и с остальными функциональными возможностями этих языков и учиться использовать их в своих программах. Даже если вы уже имеете опыт программирования, я рекомендую прочесть эти главы хотя бы бегло, поскольку в них приведено множество полезных советов, которые позволят вам повысить индивидуальную производительность вашего труда за счет грамотного применения особенностей VS.

- Часть II: *"Изучаем среду разработки VS 2010"*. В этой части рассматриваются некоторые распространенные задачи, которые большинство программистов выполняют ежедневно, в том числе: работа с решениями и проектами, отладка кода и работа с данными. Глава 5 описывает создание и построение проектов, а также прочие задачи, связанные с манипулированием проектами и решениями. Обратите особое внимание на указания по работе со сборками (assemblies) и библиотеками классов (class libraries), поскольку по мере того, как вы будете переходить от простых проектов к более сложным, их роль будет неуклонно возрастать. Вне зависимости от того, какой философии разработки вы придерживаетесь, вам, как и любому другому программисту, в любом случае придется искать и исправлять ошибки в коде, причем важность этого навыка со временем тоже будет только возрастать. Этому вопросу посвящена глава 6, в которой описываются инструменты VS, предназначенные для анализа кода, поиска и исправления ошибок. Наконец, еще одна важная задача, с которой сталкивается каждый программист, — это работа с данными. VS позволяет создавать базы данных, добавлять в них таблицы, устанавливать между ними взаимосвязи, и много другое. Когда база данных будет создана и готова к использованию, вы сможете написать программный код, который будет обращаться к этой базе данных. В данной книге рассматривается язык интегрированных запросов LINQ для SQL (LINQ to SQL) как одна из наиболее простых, но, в то же самое время, достаточно мощных технологий работы с базами данных, доступных разработчикам ПО.
- Часть III: *"Разработка приложений с помощью VS 2010"*. После того как вы приобретете базовые знания о поддерживаемых в VS языках программирования и в общих чертах ознакомитесь со средой разработки, можно приступить к реальному программированию с помощью VS. Платформа .NET поддерживает различные технологии, и подход данной книги заключается в изложении материала "с прицелом на будущее". Это значит, что при рассмотрении приоритет отдается новейшим технологиям, появившимся совсем недавно. Автор не ставит перед собой цели научить вас абсолютно всему, что связано с этими технологиями, потому что для описания каждой из них потребуется отдельная книга. Тем не менее, в главах из этой части показывается, как наиболее рационально пользоваться возможностями VS при разработке приложений. Здесь приводится базовая информация о том, как быстро начать разработку собственных приложений. Главы 8 и 10 содержат вводную информацию о форме языка XML

(Extensible Markup Language), называемой XML Application Markup Language (XAML). Поскольку эта книга предназначена для начинающих, основные справочные сведения о языках XML и XAML приведены в *приложениях 1 и 2*. Если вы не знакомы с этими языками, то перед чтением *глав 8 и 10* я рекомендую прочесть эти приложения. Кроме того, *главу 8* следует читать перед *главой 10*, потому что многие из концепций работы с Windows Presentation Foundation (WPF), технологией разработки обычных приложений, применимы и к Silverlight, технологии создания Web-приложений. Остальные две главы, входящие в эту часть, демонстрируют построение Web-приложений с помощью технологии ASP.NET MVC и создание Web-сервисов с помощью Windows Communications Foundation.

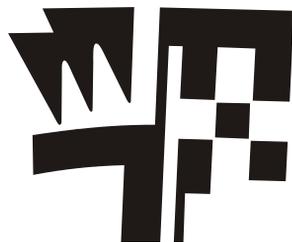
- Часть IV: "*Расширение возможностей VS 2010*". В дополнение ко всем программам-мастерам (wizards), инструментальным средствам, утилитам и средствам редактирования, которые предлагает среда VS, вы можете дополнительно расширить и нарастить ее возможности. В *главе 12* показано, как создать собственные программы-мастера для работы над проектами, как создавать собственные шаблоны для автоматически генерируемых фрагментов кода и как создавать макросы для автоматизации рутинных задач в среде VS. Если возможность написания макросов, с которой вы познакомились в VS, не кажется вам достаточно мощной, прочтите *главу 13*, в которой рассказывается о создании добавочных модулей (Add-In), программ, которые вы можете написать и установить для расширения возможностей VS. Я надеюсь, что данная книга поможет вам в овладении возможностями VS и что после этого процесс программирования будет доставлять вам удовольствие.

Джо Майо (Joe Mayo)



Ч А С Т Ь I

**Базовая информация
о Visual Studio 2010**



Глава 1

Первое знакомство с Visual Studio 2010

В этой вводной главе мы рассмотрим следующие ключевые концепции, необходимые начинающим, чтобы приступить к работе с Microsoft Visual Studio 2010:

- Чем может вам помочь Visual Studio 2010;
- Установка Visual Studio 2010 и выбор инсталляционных опций;
- Какие типы приложений можно создавать с помощью Visual Studio 2010.

Чаще всего первое знакомство с Visual Studio (VS) 2010 начнется для вас с установки этого продукта. Как и большинство других программ, среда разработки VS очень проста в установке. В данной главе будет описан процесс инсталляции, а также даны рекомендации, помогающие разобраться в предлагаемых опциях. Как только процедура установки завершится, вы сможете в первый раз запустить среду VS. После этого вам потребуется разобраться с навигацией по различным окнам среды разработчика. Эта глава дает краткое описание организации VS и показывает, как найти нужные функции и как работать с окнами. Наконец, будет рассказано, как найти типы приложений, которые можно компилировать (compile) и строить (build) с помощью VS. К этому моменту вы наверняка уже знаете, что VS позволяет создавать приложения .NET, но обсуждение мы начнем с детального описания задач, которые вы можете решать с помощью Visual Studio.

Что позволяет делать Visual Studio 2010?

Visual Studio 2010 (VS) представляет собой интегрированную среду разработки (Integrated Development Environment, IDE). IDE — это набор инструментов разработчика ПО, собранный в составе единого приложения и облегчающий труд программиста при написании приложений. Давайте вспомним, что без IDE (в данном случае — без VS) для написания программы требуется текстовый редактор, с помощью которого программист вводит весь исходный код своей будущей программы. Затем, когда исходный код написан, необходимо запустить из командной строки компилятор, чтобы создать исполняемый файл приложения. Основная проблема, связанная с использованием текстового редактора и компилятора, запускаемого из командной строки, заключается в том, что вы выполняете большое количество ручной работы и теряете при этом много времени. К счастью, с помощью VS многие из этих рутинных и трудоемких задач, связанных с повседневной рабо-

той программиста, можно автоматизировать. Последующие несколько разделов данной главы поясняют, чем может вам помочь интегрированная среда разработки и почему во главу угла в VS поставлена продуктивность труда разработчика.

Автоматически генерируемый код

В состав VS входит целый набор типовых проектов, из которых каждый разработчик может подобрать именно то, что ему в данный момент требуется. Каждый раз, когда вы создаете новый проект, VS автоматически создаст "скелет" будущего приложения, причем этот код можно немедленно скомпилировать и запустить на исполнение.

В составе каждого типового проекта имеются элементы, которые по желанию добавлять в ваш проект. Любой проект, в любом случае, содержит автоматически сгенерированный код, который представляет собой основу будущей программы. В следующей главе будет показано, как нужно действовать, чтобы создавать новые проекты, добавлять в их состав новые элементы и просматривать автоматически сгенерированный код. VS предлагает множество готовых к использованию элементов управления, включая и код, необходимый для их создания. Это экономит время разработчиков, избавляя их от необходимости каждый раз заново создавать типовой программный код для решения часто встречающихся задач. Многие из более сложных элементов управления содержат так называемые "программы-мастера" (Wizards), которые помогают настроить поведение элементов управления, автоматически генерируя код в зависимости от выбранных вами опций.

Опыт быстрого кодирования (Rapid Coding Experience)

Редактор VS оптимизирует работу программиста по кодированию. Существенная часть синтаксических элементов новой программы выделяется при помощи системы цветовых обозначений. В вашем распоряжении окажутся такие технологии, как Intellisense, известная как автодополнение. В ходе того, как вы будете вводить новый код, на экране будут появляться всплывающие подсказки. Наконец, для ускорения выполнения многих задач вам будет предоставлено большое количество клавиатурных комбинаций (keyboard shortcuts). Существует и набор средств быстрой переработки или рефакторинга (refactoring), которые позволяют быстро усовершенствовать структуру кода, не отрываясь от процесса программирования. Например, функция переименования (Rename) позволяет изменить имя идентификатора там, где оно определено, и это повлечет за собой изменение имени данного идентификатора повсюду, где он встречается в коде программы. Кроме того, VS предоставляет и множество других, не менее удобных возможностей. Например, иерархия вызовов (call hierarchy) позволяет программисту проследить всю цепочку вызовов в коде приложения, с верхних уровней до самых нижних. Еще одна возможность, автоматически генерируемые фрагменты (snippets), позволяет вводить сокращения, которые разворачиваются в шаблоны кода (code template). Наконец, списки действий (action lists) предназначены для автоматической генерации нового кода.

Все необходимое — всегда под рукой

Вам действительно необходимо научиться ориентироваться в среде VS, чтобы разобраться во всем множестве новых возможностей, призванных упростить нелегкую задачу быстрой разработки высококачественных приложений. Так, окно **Toolbox** просто "до отказа" забито различными элементами управления, окно **Server Explorer** предназначено для работы с сервисами и базами данных операционной системы, а окно **Solution Explorer** дает возможность работать с вашими проектами, тестировать утилиты, и предоставляет средства визуального проектирования. Кроме того, предоставляются и средства работы с компиляторами.

Настраиваемость и расширяемость

Многие из элементов, образующих среду VS, являются настраиваемыми. Это значит, что вы можете менять цвета отображения элементов кода, опции редактора, а также общее оформление. Набор опций настолько обширен, что и в самом деле необходимо потратить некоторое время, чтобы ознакомиться с ними и привыкнуть к тому, где и какую из них следует искать. Если стандартные настройки среды VS, которые по умолчанию предлагаются сразу же после установки, вас не устраивают, или не предлагают нужных вам опций, вы можете написать собственные макросы, чтобы автоматизировать последовательности часто выполняемых шагов. Для более сложных вариантов настройки VS предлагает специальный интерфейс прикладного программирования (Application Programming Interface, API), предназначенный для создания собственных дополнительных модулей (add-ins) и расширений (extensions). Некоторые сторонние разработчики уже предлагают приложения, способные интегрироваться в среду VS. В качестве примера можно назвать среду разработки на Delphi от компании Embarcadero¹, которая встраивается в Visual Studio. Настраиваемая и гибкая среда VS позволит вам работать так, как удобно именно вам.

По мере того как вы будете читать эту книгу, имейте в виду все эти важные концепции и внимательно читайте все рекомендации, которые позволят вам продуктивнее работать с VS. Вашим первым шагом, с которого начнется работа с VS, должна стать установка этого продукта, которая будет обсуждаться в следующем разделе.

Установка Visual Studio 2010

Предположительно, все сказанное в предыдущих разделах должно было подогреть ваше желание начать работу с VS и воспользоваться всеми возможностями новой версии данного продукта. Если до сих пор вам никогда не приходилось устанавливать VS, внимательно прочтите этот раздел, потому что именно здесь приводятся пошаговые инструкции, которые позволят вам успешно выполнить процедуру установки. Даваемые по ходу описания указания и рекомендации позволяют

¹ Дополнительную информацию см. здесь: http://en.wikipedia.org/wiki/Embarcadero_Delphi, скачать можно отсюда: <http://www.embarcadero.com/>. — Прим. перев.

вам правильно выбрать опции, нужные именно вам, и в итоге получить среду разработчика, настроенную в соответствии именно с вашими потребностями. В процессе установки VS нужно будет выполнить следующие шаги:

1. Уточнить системные требования и выяснить, удовлетворяет ли им ваша системная конфигурация.

СИСТЕМНЫЕ ТРЕБОВАНИЯ

На момент написания данной книги Microsoft рекомендует следующую аппаратную конфигурацию компьютера: процессор — 32-разрядный (x86) или 64-разрядный (x64) CPU, не менее 1 Гбайт RAM, жесткий диск со скоростью вращения 5400 RPM, 3 Гбайт свободного дискового пространства, привод DVD-ROM, Видео — DirectX с разрешением 1280×1024, тактовая частота процессора — 1.6 ГГц. Рекомендуемая операционная система: Windows Vista (все варианты, кроме Starter), Windows XP SP2 или более новый (все варианты, кроме Starter), Windows 7 (на момент написания этой книги — только редакция Ultimate), Windows Server 2003 (SP1 или R2, или более новая версия), Windows Server 2008 (SP1, R2 или более новая версия). Не забудьте посетить сайт Microsoft Developer Network (MSDN) и уточнить системные требования, потому что со временем они могут измениться.

2. Когда вы впервые вставите дистрибутивный DVD с копией VS 2010 в свой привод, на экране появится окно-заставка Microsoft Visual Studio 2010, показанное на рис. 1.1. В этом окне доступны опции **Install Microsoft Visual Studio 2010** и **Check For Service Releases**. Чтобы начать процедуру установки, выберите опцию **Install Microsoft Visual Studio 2010**.



Рис. 1.1. Окно Microsoft Visual Studio 2010 Setup

3. Следующее окно, которое вы увидите на экране, отображает приветствие Microsoft Visual Studio 2010. В примере, показанном на рис. 1.2, изображено окно

приветствия, в котором пользователь выбрал для установки версию **Ultimate**. Здесь нужно отметить, что для других вариантов установки процедура будет аналогичной, изменяться будет лишь количество и состав доступных опций, который зависит от выбранного варианта установки.

4. Если на этой странице вы установите флажок **Help Improve Setup**, то программа-инсталлятор в ходе процесса установки будет собирать информацию из журналов (logs), создаваемых в ходе установки, и отправит их в Microsoft после завершения процесса, используя действующее интернет-соединение. Чтобы помочь вам сделать информированный выбор касательно того, хотите ли вы отправлять эту информацию разработчикам продукта через Интернет, они поместили в данное окно ссылку **Privacy Statement**. Щелкните мышью по этой ссылке и внимательно прочтите информацию о том, как и с какими целями Microsoft будет использовать полученную от вас информацию. Когда вы примете окончательное решение, щелкните по кнопке **Next**. После того как загрузятся компоненты, необходимые для установки, на экране появится окно с текстом лицензионного соглашения, показанное на рис. 1.3.



Рис. 1.2. Окно приветствия программы Setup

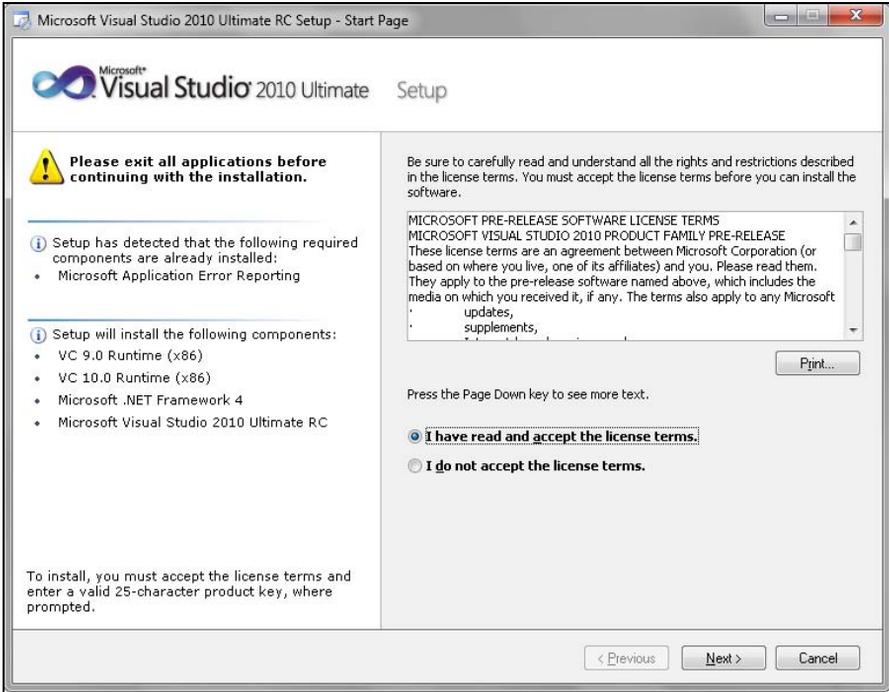


Рис. 1.3. Окно программы Setup с текстом лицензионного соглашения

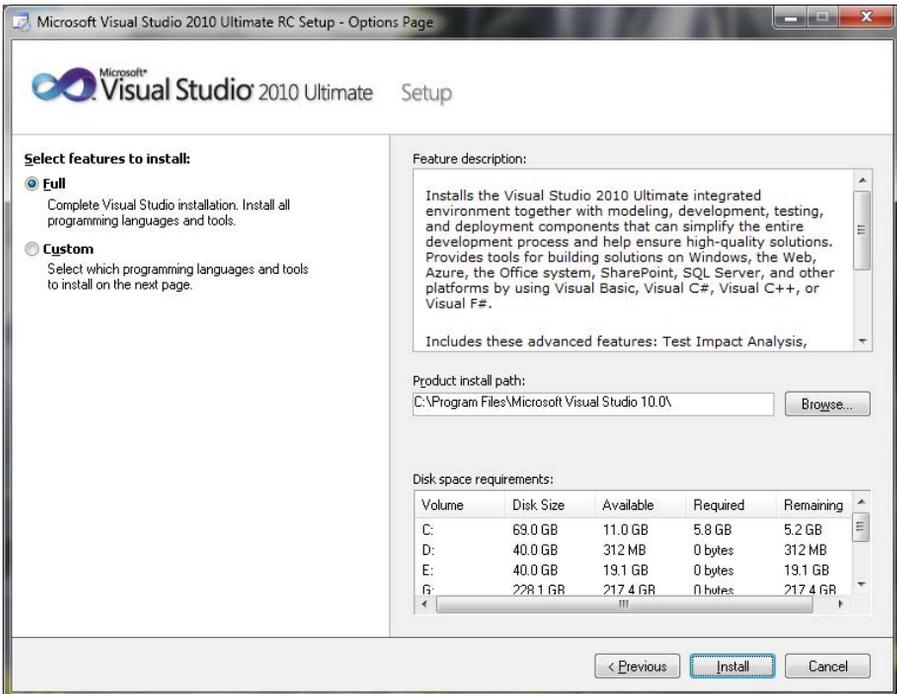


Рис. 1.4. Окно индивидуальной настройки программы Visual Studio Setup

5. В окне, показанном на рис. 1.3, вы увидите список компонентов, которые планируется установить. Кроме того, вам необходимо будет прочесть текст лицензионного соглашения с Microsoft. Текст соглашения нужно прочесть, потому что Microsoft необходимо удостовериться в том, что вы понимаете, на каких условиях вы можете пользоваться данным ПО. Условия лицензионного соглашения могут варьироваться, в зависимости от того, какой тип пакета вы приобрели, а также от конкретного региона или страны, в которой вы работаете. Прочитав условия лицензионного соглашения, вы должны будете установить флажок **I have read and accept the license terms** для продолжения установки. Далее вам потребуется ввести лицензионный ключ продукта (его можно найти на фирменной упаковке приобретенного ПО), а затем ввести свое имя. Программу-установщик автоматически введет код продукта в том случае, если вы загрузили его с сайта Microsoft Developer Network (MSDN). Нажмите кнопку **Next**, и в следующем окне вы получите возможность индивидуально настроить процесс установки продукта.
6. Окно, показанное на рис. 1.4, позволит вам выбрать между полной (Full) и выборочной (Custom) установкой. Если вы выберете опцию **Custom**, вы получите возможность индивидуально выбирать, какие компоненты VS должны быть установлены. Это — хорошая возможность отказаться от инсталляции компонентов, о которых вы точно знаете, что никогда не будете ими пользоваться.

Если вы устанавливаете VS впервые, и при этом не испытываете нехватки свободного дискового пространства, вы вполне можете пролистать список всех доступных для выбора опций и отметить их все — хотя бы для того, чтобы просто ознакомиться со всеми предлагаемыми возможностями. К процессу установки вы сможете вернуться в любое время после его завершения и вновь запустить программу Setup, чтобы внести изменения в установленную конфигурацию.

Конфигурационный экран, показанный на рис. 1.4, позволяет изменить каталог, в который будет производиться установка VS. Запомните путь к этому каталогу, потому что впоследствии именно там вы сможете найти примеры программного кода, общие сборки (common assemblies), а также множество другой информации, включая объекты, влияющие на поведение среды разработки. На данном этапе вам следует оценить объем имеющегося у вас дискового пространства и убедиться, что вам его хватит, чтобы установить все выбранные компоненты. На этом конфигурирование опций для установки будет завершено. Щелкните мышью по кнопке **Install**, и процесс установки начнется. На экране появится примерно такое окно, как на рис. 1.5, отражающее степень завершенности процесса инсталляции. Небольшие галочки в этом окне будут появляться рядом с названиями успешно установленных компонентов.

1. В ходе процесса установки Visual Studio 2010 программе VS Installer потребуется перезагрузить компьютер. Когда настанет этот момент, программа установки сообщит об этом, выведя окно, показанное на рис. 1.6. Увидев это окно, закройте все приложения, с которыми вы в данный момент работаете, предварительно убедившись в том, что все важные данные сохранены, и нажмите в этом окне кнопку **Restart Now**.

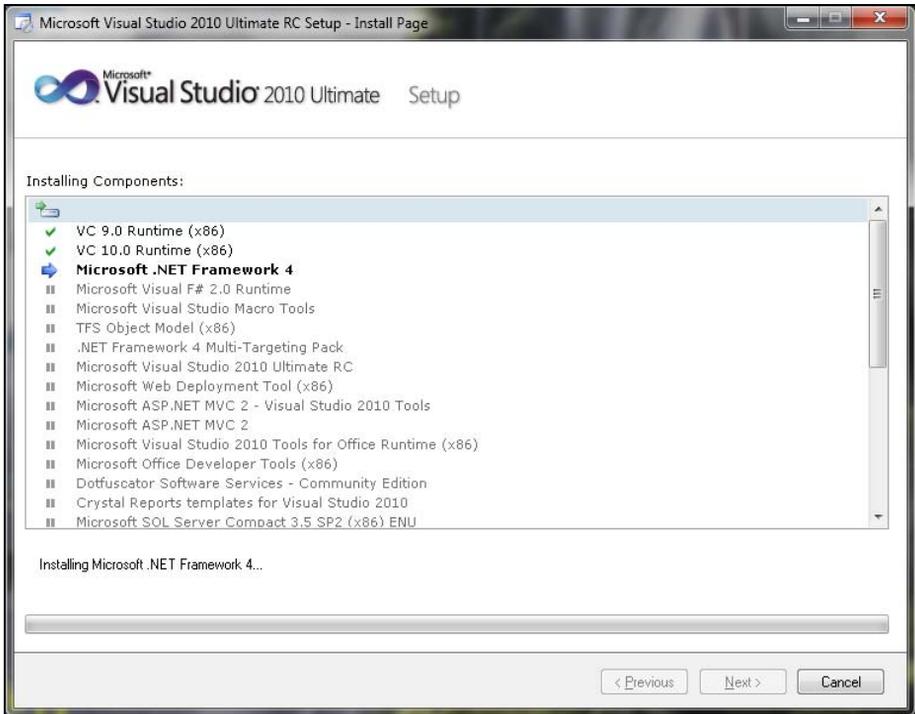


Рис. 1.5. Окно Setup Progress



Рис. 1.6. Окно с сообщением, предлагающим перезагрузить компьютер

2. После того как процедура установки завершится успехом, на экране появится окно **Success**, показанное на рис. 1.7. Если в процессе установки произойдет какая-нибудь ошибка или сбой, то в данном окне будут даны указания, помогающие понять причину неудачи и устранить проблему.

Итак, процедура установки почти завершилась. Теперь, при желании, вы можете установить и электронную документацию, нажав в окне, показанном на рис. 1.7, кнопку **Install Documentation**. На экране вновь появится окно, которое вы уже видели в начале процесса установки (рис. 1.8). В этом окне установите и опции для инсталляции сервисных релизов (Service Releases). Это следует сделать не только потому, что сервисные релизы предлагают дополнительные функции VS, но и потому, что они включают и важные обновления безопасности.

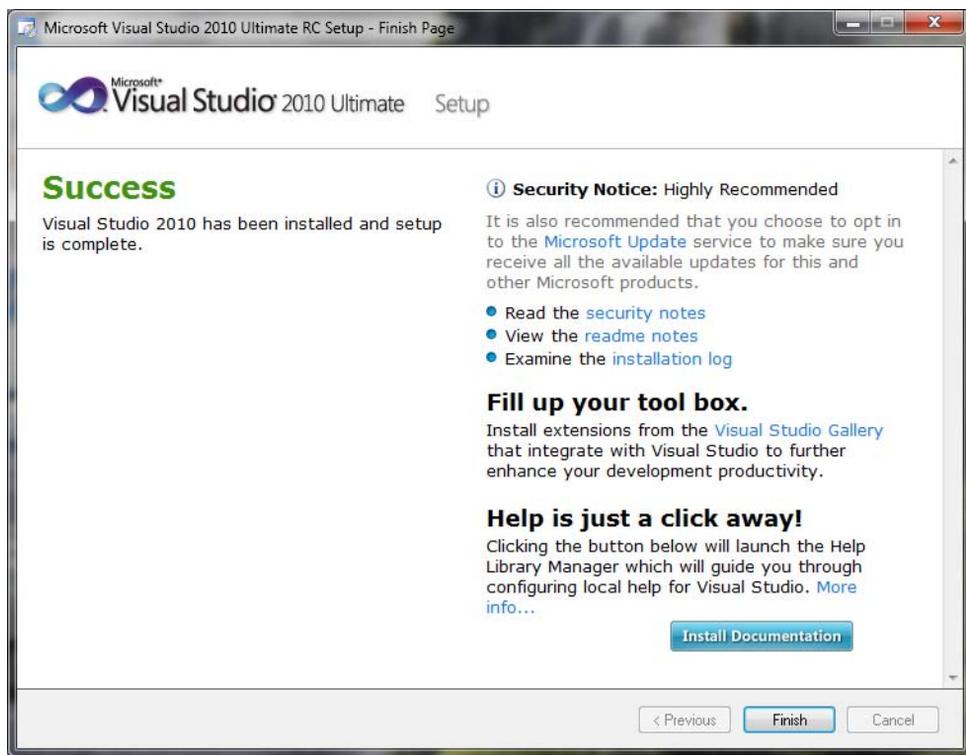


Рис. 1.7. Окно Setup Success



Рис. 1.8. Проверка наличия сервисных выпусков (service releases)

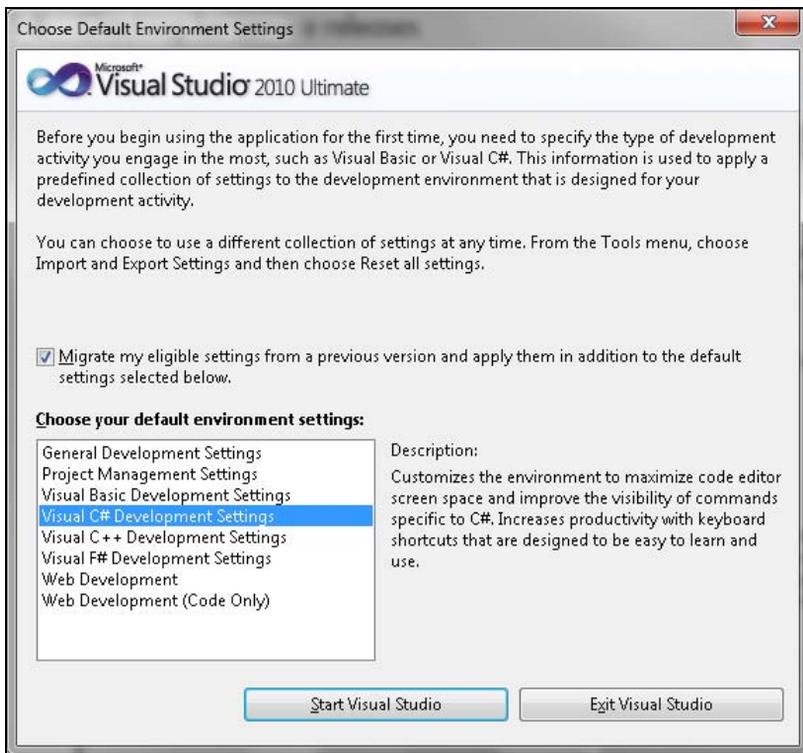


Рис. 1.9. Окно Default Environment Settings

Теперь вы готовы к тому, чтобы запустить VS в первый раз. При первом запуске вам потребуется выполнить еще один важный конфигурационный шаг — выбрать среду, которую вы хотите открывать по умолчанию, как показано на рис. 1.9.

Выбор настроек среды по умолчанию во многом зависит от того, на каком языке и в какой среде вы в основном будете писать свои программы. Впрочем, настройки среды не являются чем-то застывшим и выбираемым раз и навсегда — если впоследствии вы захотите их поменять, вы всегда сможете это сделать. О том, как изменить настройки по умолчанию, будет рассказано в следующем разделе данной главы. В этой книге речь пойдет как о Visual Basic (VB), так и C#, поэтому вам, скорее всего, будет удобнее выбрать настройки, характерные для того языка, который предпочитаете лично вы. Примеры, приведенные в этой книге, будут использовать настройки для VB или для C# — в зависимости от обсуждаемой темы. Выбор настроек определяет расположение и набор окон, отображаемых по умолчанию, а также стандартные настройки, которые по умолчанию будут использоваться средой VS IDE.

ПРИМЕЧАНИЕ

Что лучше выбрать — C# или VB? На самом деле, как C#, так и VB представляют собой первоклассные языки, поддерживающие платформу .NET. Языки как таковые ограничиваются одним лишь синтаксисом, а все дополнительные сервисы перемещены

в состав библиотеки .NET Framework Class Library, которая доступна всем поддерживаемым языкам. Между языками есть небольшое количество несущественных различий, но в реальности выбор языка зависит только от ваших личных предпочтений. На практике, знание обоих языков будет для вас преимуществом, потому что это поможет пониманию ценной информации, изложенной во множестве книг и статей, иллюстрирующих работу с платформой .NET. При этом излагаемая информация часто не зависит от используемого языка. Соответственно, совсем не рационально отказываться от чтения отличной книги или статьи о платформе .NET только потому, что приводимые там примеры написаны не на том языке, который предпочитаете лично вы.

К настоящему моменту предполагается, что вы уже успешно установили VS 2010 и настроили среду разработчика в соответствии со своим выбором и предпочтениями. Итак, откройте VS, и в следующем разделе мы приступим к рассмотрению высокоуровневых элементов интерфейса, доступных на странице **Start**.

Навигация по среде Visual Studio 2010

В этом разделе мы бегло рассмотрим высокоуровневый интерфейс VS и опишем все возможности, которые становятся вам доступными при первом запуске Visual Studio 2010. То, что вы увидите на экране, называется интегрированной средой разработчика Visual Studio (Integrated Development Environment, IDE).

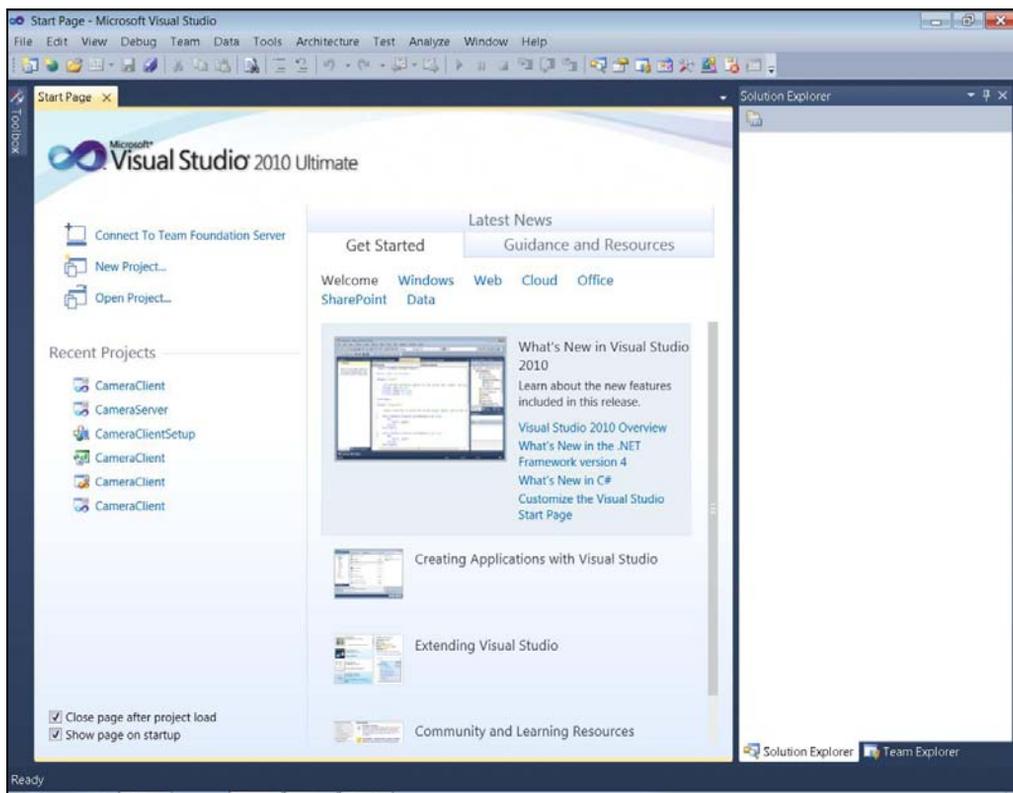


Рис. 1.10. Начальный экран Visual Studio 2010

Даже если вы — опытный программист, уже работавший с одной из предыдущих версий Visual Studio, просмотрите эту информацию хотя бы бегло, потому что благодаря ей вы быстрее научитесь ориентироваться в IDE и находить нужные вам функции. Кроме того, знание функций, доступных по умолчанию, позволит вам эффективнее различать между стандартными функциональными возможностями и контекстно-зависимыми функциями программных компонентов, над которыми вы работаете.

На рис. 1.10 показано, как выглядит среда VS сразу же после первого запуска. Это относится ко всем частям экрана, на примере которых мы и будем основываться, поясняя, как организована среда IDE. В следующем описании мы ассоциируем каждую функцию с именем, чтобы облегчить процесс поиска нужных функций, чтобы вы уже знали, где их искать, когда впоследствии будет упоминаться имя той или иной из них. В последующих нескольких разделах будут описаны все части экрана, показанного на рис. 1.10.

Меню

В верхнем левом углу экрана, показанного на рис. 1.10, вы увидите начало строки меню (menu bar), начинающейся с пунктов **File**, **Edit**, **View**, **Tools** и т. д. Строка меню представляет собой стандартный элемент интерфейса любого Windows-приложения. Помимо стандартного управления файлами, меню **File** представляет собой стандартную "начальную точку", откуда вы начинаете работать с любым проектом. Кроме того, меню **File** предоставляет возможности быстрого доступа к недавно открывавшимся файлам и проектам.

Меню **Edit** предоставляет стандартные возможности редактирования: функции вырезки в буфер (cut), копирования (copy) и вставки (paste). Кроме того, оно позволяет установить закладку (bookmark), чтобы обеспечить быстрый доступ к нужному фрагменту кода и быструю навигацию по большим файлам.

Не стоит жалеть времени на ознакомление с опциями, доступными через меню **View**, хотя бы для того, чтобы посмотреть, какие функции есть в вашем распоряжении. Но если вы только приступаете к изучению Visual Studio и являетесь начинающим программистом, то не стоит "зацикливаться" на этом сейчас — в дальнейшем мы обратим на эти функции особое внимание, обсудим предназначение каждой из них и объясним, в какой ситуации конкретное представление (view) будет особенно полезным.

Меню **View** позволяет быстро получить доступ ко всем инструментальным окнам, имеющимся в VS. Кроме того, меню **View** содержит пункт **Other Windows**, куда включены дополнительные окна приложений, которые могут оказаться удобными при написании новых программ.

Меню **Tools** предоставляет множество различных функциональных возможностей: например, с его помощью вы можете подключить отладчик, чтобы просмотреть работу ваших программ пошагово, проходя одну строку за другой, подключиться к базе данных, установить добавочные модули, макросы и выполнить множество других действий. Одна из важнейших опций меню **Tools** так и называется — **Options**, так как она предоставляет доступ к сотням и сотням настроек VS, позволяющих индивидуально настроить среду разработки.

Меню **Test** можно использовать, чтобы выполнить модульное тестирование вашей новой программы, по одному модулю одновременно. Именно в этом меню различные редакции VS предоставляют доступ к различным наборам инструментов тестирования.

Пункты меню **Analyze**, **Architecture** и **Team** предоставляют расширенные наборы инструментов, предназначенных для повышения производительности приложений, работы с их архитектурными компонентами, а также для их интеграции с Microsoft Team Foundation Server².

Меню **Windows** и **Help** имеют предназначение, стандартное для большинства приложений Windows. Так, меню **Windows** позволяет манипулировать различными окнами VS, а с помощью меню **Help** можно читать техническую документацию по VS.

РЕКОМЕНДАЦИЯ

Многие элементы меню ассоциированы с клавиатурными комбинациями, которые предоставляют более быстрый доступ к нужной функции, чем ее вызов через меню. Если вы заинтересованы в том, чтобы запомнить как можно больше клавиатурных комбинаций, то обратите внимание на то, что при открытии меню многие клавиатурные комбинации, ассоциированные с тем или иным пунктом меню, приводятся и в самом меню, справа от имени соответствующей команды. Например, чтобы открыть окно **Solution Explorer**, вы можете выбрать соответствующую команду через меню **View**, а можете нажать клавиатурную комбинацию <CTRL>+<W>, <S>.

Инструментальная панель (Toolbar)

Вернемся к рис. 1.10. Под строкой меню на этой иллюстрации находится инструментальная панель. На инструментальной панели расположены кнопки, предоставляющие быстрый доступ к наиболее часто используемым функциям. Набор кнопок инструментальной панели представляет собой подмножество команд, которые доступны через меню. Инструментальные панели контекстно-чувствительны, и набор доступных через них опций меняется в зависимости от той работы, которую вы выполняете через VS в данный момент. Любую инструментальную панель можно отобразить или скрыть. Делается это с помощью команд меню **View | Toolbars**.

Кроме того, инструментальные панели можно настраивать по вашему выбору. Для этого выполните щелчок правой кнопкой мыши по нужной панели, прокрутите появившееся контекстное меню и выберите из него опцию **Customize**. После этого на экране появится окно настройки инструментальной панели, где вы сможете добавить на нее кнопки быстрого вызова функций, которыми часто пользуетесь именно вы, но которые не были включены в набор функций предлагаемых для данной инструментальной панели по умолчанию.

² Team Foundation Server — это ПО Microsoft, предназначенное для совместной работы над проектами, доступное как в виде отдельного приложения, так и в виде серверной платформы для Visual Studio. Подробнее о Team Foundation Server см.: http://ru.wikipedia.org/wiki/Team_Foundation_Server, <http://cmcons.com/articles/microsoft/>. — Прим. перев.

Рабочая область (Work Area)

Еще раз вернемся к рис. 1.10. В центре окна, представленного на иллюстрации, находится стартовая страница (Start page). Это — та самая область, которой вы можете пользоваться для введения кода ваших программ и работы с визуальными редакторами. Стартовая страница разделена на две области: область управления проектами (project management) и информационную зону (information). Расположенная слева область управления проектами предоставляет возможность быстро начать работу над новым проектом или выбрать из списка один из проектов, с которыми вы работали недавно. Расположенная справа информационная область содержит ресурсы, помогающие новичку начать работу с VS, например, ссылки на Web-сайт Microsoft, пошаговые инструкции, помогающие разобраться с новыми возможностями, а также постоянно обновляемую вкладку, на которой можно прочесть последние новости сообщества разработчиков Microsoft.

Инструментальный набор (Toolbox)

В крайней правой части окна, показанного на рис. 1.10, располагается вертикальная вкладка, озаглавленная **Toolbox**, которая содержит контекстно-чувствительный список элементов управления (controls). Эти элементы управления можно перетаскивать мышью в текущую рабочую область, чтобы включить их в состав разрабатываемой программы.

Термин "контекстно-чувствительный" означает, что некоторые элементы списка будут отображены или скрыты, в зависимости от того, где в последний раз был выполнен щелчок мышью, или от контекста, в котором вы в данный момент работаете. Например, вы можете вводить программный код или создавать/редактировать новую Web-страницу. Если вы еще не начали ни над чем работать, то страница **Toolbox** будет пустой.

Окно Solution Explorer

Окно **Solution Explorer**, находящееся на правой границе страницы **Start** (рис. 1.10), отображает все ваши решения (solutions), проекты (projects) и элементы этих проектов. Именно здесь можно найти и требуемым образом организовать все файлы и настройки, принадлежащие проекту. На рис. 1.10 окно **Solution Explorer** пусто, потому что еще не открыто ни одного решения и не создано ни одного проекта. Если вы закроете это окно, а затем вам потребуется вновь его открыть, то это можно сделать через меню **View**, которое уже обсуждалось чуть ранее.

Строка состояния

В самой нижней части экрана, показанного на рис. 1.10, находится строка состояния (status bar), которая сообщает вам обо всем, что происходит в среде VS в настоящий момент. На рис. 1.10 в строке состояния выведено сообщение **Ready**, которое говорит о том, что среда VS готова к работе. В ходе вашей работы с VS строка состояния будет изменяться в зависимости от контекста, отображая информацию, относящуюся к задаче, выполняемой на текущий момент. Например, если