

Разработка приложений для **Windows 8** на языке **C#**



**ПРИНЦИПЫ РАЗРАБОТКИ
ПРИЛОЖЕНИЙ ДЛЯ Windows 8
В СРЕДЕ Visual Studio 2012**

**СОЗДАНИЕ ЖИВЫХ ПЛИТОК,
РАБОТА С КОНТРАКТАМИ**

**ОПРЕДЕЛЕНИЕ
МЕСТОПОЛОЖЕНИЯ, РАБОТА
С СЕНСОРАМИ И КАМЕРОЙ**

**ОСНОВНЫЕ ПРИНЦИПЫ
ДИЗАЙНА И
ПРОЕКТИРОВАНИЯ Windows
Store-ПРИЛОЖЕНИЙ**

**ХРАНЕНИЕ И ДОСТУП
К ДАННЫМ,
ИНТЕРНАЦИОНАЛИЗАЦИЯ И
РАЗМЕЩЕНИЕ ПРИЛОЖЕНИЙ
В Windows Store**

PRO

**ПРОФЕССИОНАЛЬНОЕ
ПРОГРАММИРОВАНИЕ**



*Материалы
на www.bhv.ru*

УДК 681.3.06
ББК 32.973.26-018.2
П88

Пугачев, С. В.

П88 Разработка приложений для Windows 8 на языке C# / С. В. Пугачев, А. М. Шериев, К. А. Кичинский. — СПб.: БХВ-Петербург, 2013. — 416 с.: ил. — (Профессиональное программирование)

ISBN 978-5-9775-0846-9

Рассмотрены принципы разработки Windows Store-приложений для Windows 8 на языке C# в среде Visual Studio 2012. Описаны основные возможности платформы и показаны сценарии их практического использования. Особое внимание уделяется дизайну и проектированию приложений. Описана работа с живыми плитками, контрактами, сервисом определения местоположения, сенсорами, уведомлениями и камерой. Рассказывается про хранение и доступ к данным, интернационализацию и размещение приложений в специализированном магазине приложений Windows Store.

Для программистов

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Екатерина Капальгина</i>
Редактор	<i>Леонид Кочин</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Марины Дамбиевой</i>

Подписано в печать 31.10.12.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 33,54.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.
Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-0846-9

© Пугачев С. В., Шериев А. М., Кичинский К. А., 2013
© Оформление, издательство "БХВ-Петербург", 2013

Оглавление

Почему это первая книга о Windows, которую я прочитаю от начала до конца	9
Введение	11
На кого рассчитана эта книга	11
Краткое описание глав	12
Благодарности	14
Глава 1. Платформа Windows 8.....	15
Windows Store-приложения.....	16
Программная платформа.....	21
Дизайн Windows Store-приложений.....	23
Итоги.....	25
Глава 2. Среда разработки	26
Итоги.....	30
Глава 3. Первое Windows Store-приложение	31
Создание проекта в Visual Studio	31
Структура проекта приложения.....	35
Добавляем функциональность.....	40
Отладка приложения на симуляторе.....	46
Отладка приложений на удаленных устройствах	49
Итоги.....	52
Глава 4. Страницы и навигация в приложениях.....	53
Задание начальной страницы приложения	53
Создание новой страницы.....	56
Анимация при переходе между страницами	60
Передача параметров между страницами.....	61
Сохранение состояния страниц и приложения	62
Кэширование страниц	64
Открытие ссылок из приложения.....	67
Итоги.....	68

Глава 5. Тема оформления	69
Задание темы для приложения	70
Отображение текста.....	72
Итоги.....	75
Глава 6. Панель приложения.....	77
Создание панелей приложения.....	79
Свойство <i>IsSticky</i>	83
Отображение всплывающих окон и меню.....	83
Итоги.....	86
Глава 7. RSS-клиент на основе шаблона Grid App.....	87
Итоги.....	97
Глава 8. Элементы управления <i>GridView</i>, <i>Listview</i> и <i>FlipView</i>	98
Элемент управления <i>GridView</i>	99
Задание разного размера для элементов в <i>GridView</i>	110
Установка разных шаблонов для элементов в <i>GridView</i>	114
Контекстное масштабирование (Semantic Zoom).....	115
Элемент управления <i>Listview</i>	120
Элемент управления <i>FlipView</i>	123
Итоги.....	124
Глава 9. Закрепленный режим работы и поддержка различной ориентации экрана.....	125
Поддержка закрепленного режима	126
Visual State Manager.....	130
Масштабирование изображений в зависимости от плотности пикселей.....	133
Итоги.....	135
Глава 10. Модель исполнения приложений. Многозадачность через фоновые задачи	136
Модель исполнения приложений	136
Реализация сценариев многозадачности.....	139
Фоновая загрузка/выгрузка файлов.....	139
Фоновые задачи	142
Создание фоновой задачи	146
Итоги.....	150
Глава 11. Уведомления	151
Всплывающие уведомления.....	151
Push-уведомления	155
Авторизация на WNS-сервере. Регистрация и получение ключей в Windows Store.....	156
Отправка Push-уведомлений.....	157
Итоги.....	161
Глава 12. "Живые" плитки.....	162
Плитки по умолчанию	162
Шаблоны "живых" плиток	163

Обновление плитки приложения	166
Широкие плитки	169
Шаблоны плиток с изображениями	170
Управление временем жизни плиток	171
Очередь плиток	172
Бейджи на плитках	173
Обновление плиток с помощью удаленного сервера	175
Вторичные плитки	177
Итоги	179
Глава 13. Контракт "Поиск"	180
Поддержка контракта "Поиск"	182
Поиск по мере ввода текста	186
Добавление поисковых подсказок	188
Подсказки результатов с графикой и текстом	191
Итоги	193
Глава 14. Контракт "Общий доступ"	194
Реализация поставщика данных	196
Реализация приемника данных	199
Передача и прием изображений	202
Отправка и прием нестандартных типов данных	203
Передача поставщика данных	206
Итоги	208
Глава 15. Контракт "Параметры"	209
Добавление пунктов параметров	209
Всплывающее окно с настройками	213
Итоги	215
Глава 16. Хранение и доступ к данным	216
Изолированное хранилище	216
Хранение настроек	218
Хранение файлов	219
Прямой доступ к файлам в изолированном хранилище	221
Работа с СУБД SQLite	221
Итоги	228
Глава 17. Файловые контракты и расширения	229
Расширение <i>FileOpenPicker</i>	229
Расширение <i>FileSavePicker</i>	232
Расширение <i>FolderPicker</i>	233
Разрешение на доступ к папкам с помощью манифеста приложения	233
Контракт <i>File Open Picker</i>	235
Контракт <i>File Save Picker</i>	239
Расширение <i>StorageApplicationPermissions</i> для кэширования доступа к файлам	239
Ассоциация с расширением файлов и протоколом	240
Итоги	244

Глава 18. Работа с камерой	245
Использование <i>CameraCaptureUI</i>	245
Настройка параметров для съемки фотографий	249
<i>PhotoSettings.AllowCropping</i>	249
<i>PhotoSettings.CroppedAspectRatio</i>	249
<i>PhotoSettings.CroppedSizeInPixels</i>	249
<i>PhotoSettings.Format</i>	250
<i>PhotoSettings.MaxResolution</i>	250
Настройка параметров для съемки видео	251
<i>VideoSettings.AllowTrimming</i>	251
<i>VideoSettings.Format</i>	251
<i>VideoSettings.MaxDurationInSeconds</i>	251
<i>VideoSettings.MaxResolution</i>	252
Использование расширения <i>MediaCapture</i> для прямой работы с видео/аудио	252
Дополнительные настройки расширения <i>MediaCapture</i>	256
Итоги.....	258
Глава 19. Работа с картами и определение местоположения	259
Сервис определения местоположения	259
Определение местоположения	260
Определение изменения местоположения.....	264
Работа с Bing Maps SDK	267
Итоги.....	274
Глава 20. Работа с сенсорами.....	275
Датчик света.....	275
Акселерометр.....	278
Гироскоп.....	281
Инклинометр.....	282
Компас	286
Простой сенсор ориентации	287
Итоги.....	288
Глава 21. Интернационализация	289
Глобализация.....	290
Культура по умолчанию и выбор культуры	291
Локализация интерфейса приложения	293
Локализация изображений	296
Использование локализованных ресурсов в коде	297
Локализация названия приложения.....	298
Итоги.....	299
Глава 22. Базовые принципы дизайна приложений для Windows 8.....	300
Истоки нового стиля Windows 8.....	300
Принципы современного дизайна для Windows	303
Будьте искусным в деталях	303
Достигайте большего меньшими средствами	307
Делайте по-настоящему цифровым.....	311

Делайте быстрым и подвижным.....	314
Выигрывайте вместе.....	316
Итоги.....	318

Глава 23. Расстановка приоритетов, или пять первых шагов к отличному приложению для Windows 8..... 320

Жесткая расстановка приоритетов.....	321
Этап 1. Знайте своего пользователя.....	321
Этап 2. Чем ваше приложение лучше других?.....	324
Этап 3. Выделите ключевые сценарии.....	325
Этап 4. Спланируйте навигацию.....	329
Этап 5. Продумайте функциональность.....	344
Итоги.....	353

Глава 24. Размещение и продажа приложений в Windows Store 354

Устройство Windows Store.....	356
Аудитория Windows Store.....	358
Регистрация в Windows Store.....	358
Резервирование имени приложения.....	362
Создание пакета приложения для публикации в Windows Store.....	363
Демо-версии приложений.....	366
Итоги.....	368

ПРИЛОЖЕНИЯ 371

Приложение 1. Язык разметки XAML..... 373

Задание значений свойств.....	376
Использование стилей.....	378
XAML-ресурсы и ресурсные словари.....	381
Шаблоны элементов управления.....	383
Менеджеры размещения.....	384
Связывание данных.....	389
Работа с <i>DataContext</i>	392
Связывание с коллекциями.....	396
Итоги.....	398

Приложение 2. C# 5 и асинхронное программирование..... 399

Ключевые слова <i>async</i> и <i>await</i> в C# 5.....	403
Итоги.....	407

Предметный указатель 409



Глава 1

Платформа Windows 8

Несколько лет назад, еще до выпуска компанией Apple первого iPad, разработчики Windows начали проект по созданию нового поколения флагманской операционной системы, в котором требовалось решить три задачи:

1. Увеличить время автономной работы путем общей оптимизации, переноса ОС на другие процессорные архитектуры (Windows 8 работает, в том числе, на ARM-процессорах) и, наконец, выработки правил для разработчиков, с целью минимизации потребления ресурсов приложениями.
2. Оптимизировать ОС под сенсорный ввод (тач-интерфейс).
3. Создать механизмы по продаже и продвижению приложений.

В результате начала выкристаллизовываться новая версия Windows, обещающая стать самым существенным сдвигом в семействе операционных систем от Microsoft и, одновременно, самым рискованным проектом компании за всю ее историю. Со времен Windows 95 еще не было столь кардинальных изменений для пользователей и разработчиков.

Одновременно для разработчиков Windows 8 появился уникальный шанс. Ведь у каждого пользователя Windows теперь будет Windows Store — магазин, который является основным (а в некоторых случаях и единственным) источником приложений для всех устройств, будь то настольный компьютер, ноутбук или планшет.

Все приложения, работающие на Windows 7, функционируют и на Windows 8. Однако в Windows Store можно загрузить только приложения, использующие новую программную платформу Windows Runtime (WinRT). Поэтому мы будем называть их Windows Store-приложениями. Также Windows 8 предлагает новый пользовательский интерфейс. Понять его концепцию можно, взглянув на начальный экран (рис. 1.1).

Windows Runtime — это своего рода новый Windows API и замена "старого доброго" Win32 API. По типу программной платформы приложения для Windows можно разделить на две группы:

1. Классические Windows-приложения.
2. Windows Store-приложения.

Создание Windows Store-приложений на языке C# и работа с Windows Runtime — вот тема данной книги. Такие приложения представляют собой сплав новой парадигмы интерфейса, эффективного современного API и соответствующей платформы разработки.



Рис. 1.1. Начальный экран Windows 8

Windows Store-приложения

В отличие от классических, Windows Store-приложения содержат одно окно без оформления (а также без заголовка, кнопок "Закрыть", "Развернуть" и "Свернуть"), по умолчанию занимающее весь экран. Это сделано для того, чтобы не отвлекать пользователей на лишние детали (рис. 1.2).

Windows Store-приложения могут поддерживать различные компоновки и представления, чтобы обеспечить динамичное и удобное обслуживание пользователей при различных параметрах конструкции и размерах экрана устройств (от 30 дюймовых мониторов настольных компьютеров до сравнительно небольших экранов планшетов). Такие приложения могут работать в трех режимах (рис. 1.3):

- приложение развернуто на весь экран (Full Screen);
- приложение закреплено сбоку экрана (слева или справа) (Snapped). Ширина приложения в таком режиме составляет 320 пикселей;

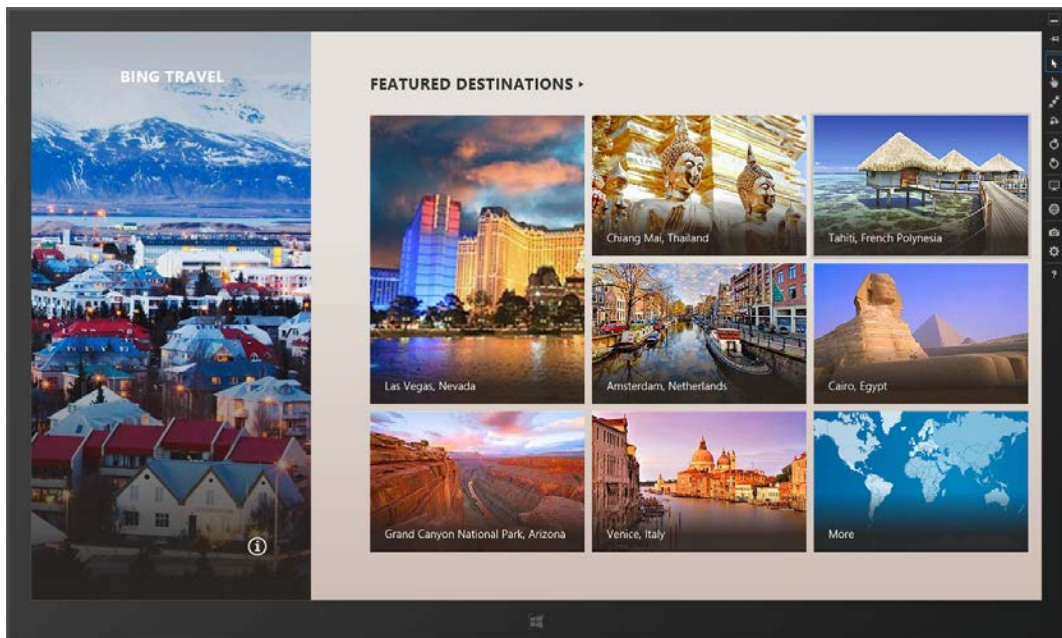


Рис. 1.2. Вид приложения в стиле Windows 8, работающего в эмуляторе

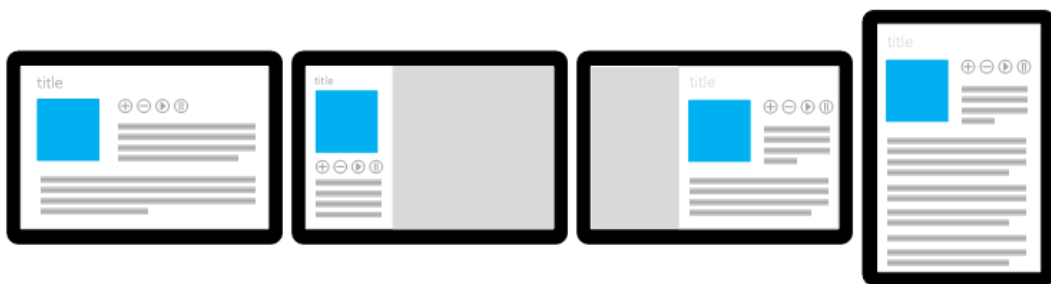


Рис. 1.3. Режимы работы приложений

- приложение работает совместно с другим закрепленным сбоку приложением и занимает все оставшееся пространство (Filled).

Таким образом, на экране одновременно могут находиться два Windows Store-приложения. Чтобы приложение можно было закрепить сбоку, экран должен иметь разрешение как минимум 1366×768 пикселей. При меньшем разрешении закрепление приложений недоступно, и одновременно может отображаться только одно Windows Store-приложение.

Кроме того, Windows Store-приложения могут работать в ландшафтной или портретной ориентации экрана (рис. 1.3), что особенно актуально для планшетов. Вертикальная ориентация, когда высота экрана больше его ширины, является портретной (Portrait) (крайний правый вариант на рис. 1.3). При ландшафтной ориентации (Landscape) ширина экрана больше его высоты.

Windows Store-приложения без проблем работают с различными устройствами ввода, включая перо, мышь, клавиатуру и сенсорный ввод. Для всех этих устройств в программной модели приложений используется единый набор событий. Также имеется набор стилей по умолчанию, гарантирующий нормальную работу элементов пользовательского интерфейса с сенсорным вводом. Раньше программный API был почти исключительно нацелен на работу с мышью и клавиатурой, а в Windows 8 одинаково хорошо поддерживаются все способы ввода, в особенности сенсорный ввод.

Windows Store-приложения содержат, кроме уже знакомых, но выполненных в другом стиле элементов управления, несколько новых элементов, повышающих эффективность взаимодействия с пользователями. Среди новинок можно отметить панель приложения (App Bar) и "чудо-кнопки" (Charms). Панель приложения — это концепция, уже знакомая многим по операционной системе Windows Phone, но претерпевшая в Windows 8 существенные изменения.

Панель приложения (рис. 1.4) размещается вне окна приложения, появляется при необходимости и служит основным командным интерфейсом (на странице может быть две панели: панель навигации сверху и панель приложения снизу). Верхняя панель удобна для навигации (переход между документами, чатами, важными разделами), нижняя — для размещения элементов команд и инструментов пользователей. По умолчанию панель приложения скрыта и появляется, когда пользователь проводит пальцем в направлении от верхнего или нижнего края экрана или щелкает правой кнопкой мыши. Так как на первом месте должен быть контент, панели приложения носят, хотя и важный, но вторичный характер и, соответственно, появляются только по запросу пользователя — явному (например, соответствующим жестом) или неявному (выделению элемента, к которому можно применить какие-то действия). Пользователь может скрыть панель тем же действием: если он проведет пальцем по краю экрана, щелкнет правой кнопкой мыши еще раз или будет взаимодействовать с приложением иным образом. Вам, как разработчикам, важно понимать принципы работы с панелью приложения, чтобы не создавать элементов управления, дублирующих ее.

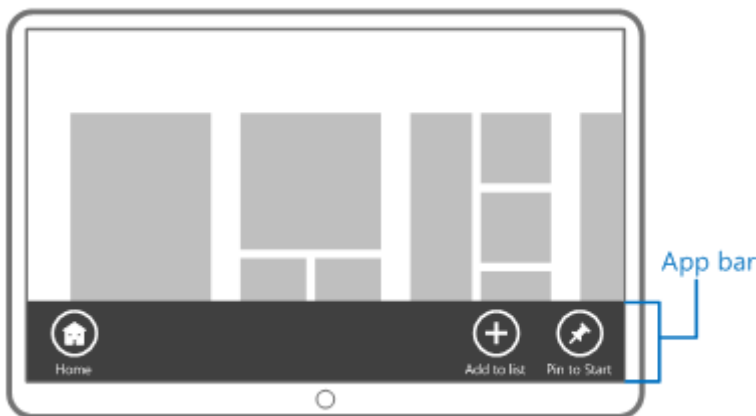


Рис. 1.4. Панель приложения

Новая концепция Windows 8 — "чудо-кнопки" (Charms) (рис. 1.5) — фиксированный набор кнопок, используемый во всех приложениях для поиска, взаимодействия между приложениями, подключения к устройствам и для настройки. Это основные рабочие сценарии, которые все пользователи выполняют практически в каждом приложении. С помощью "чудо-кнопок" пользователи могут:

- искать содержимое, размещенное в вашем или каком-либо ином приложении. Кроме того, поиск содержимого в вашем приложении возможен в любой момент, даже когда пользователь работает с другими приложениями. Приложения не должны дублировать интерфейс поиска, т. к. для этого уже предусмотрен системный механизм "чудо-кнопок";
- делиться содержимым из вашего приложения с другими приложениями стандартизированным образом. При этом вам не требуется знать, как работают другие приложения, вы только создаете источник или приемник данных. Всю остальную работу выполнит система;
- подключаться к устройствам и отправлять им содержимое, выполнять потоковую передачу мультимедийных данных, а также печатать документы;
- выполнять настройку приложений;
- кнопка "Пуск" с логотипом Windows, как можно догадаться, переключает пользователя на стартовый экран.



Рис. 1.5. "Чудо-кнопки"

Взаимодействие с "чудо-кнопками" осуществляется с помощью контрактов, работе с каждым из которых (поиск, настройки, передача данных) посвящена отдельная глава.

Когда пользователь устанавливает приложение, оно появляется на начальном экране в виде "плитки" (Tile) (см. рис. 1.1). Если нажать на плитку, приложение будет запущено. Приложения могут отображать на плитках различную информацию, как текстовую, так и графическую. Также можно обновлять данные плиток прямо из

Интернета через Web-службу, задействуя систему Push-уведомлений. С помощью механизма плиток Windows Store-приложения способны выводить на экран полезную информацию в краткой форме и при минимальном расходе заряда батареи. Вид плиток трех приложений с информацией о новостях, спорте и биржевых индексах приведен на рис. 1.6.

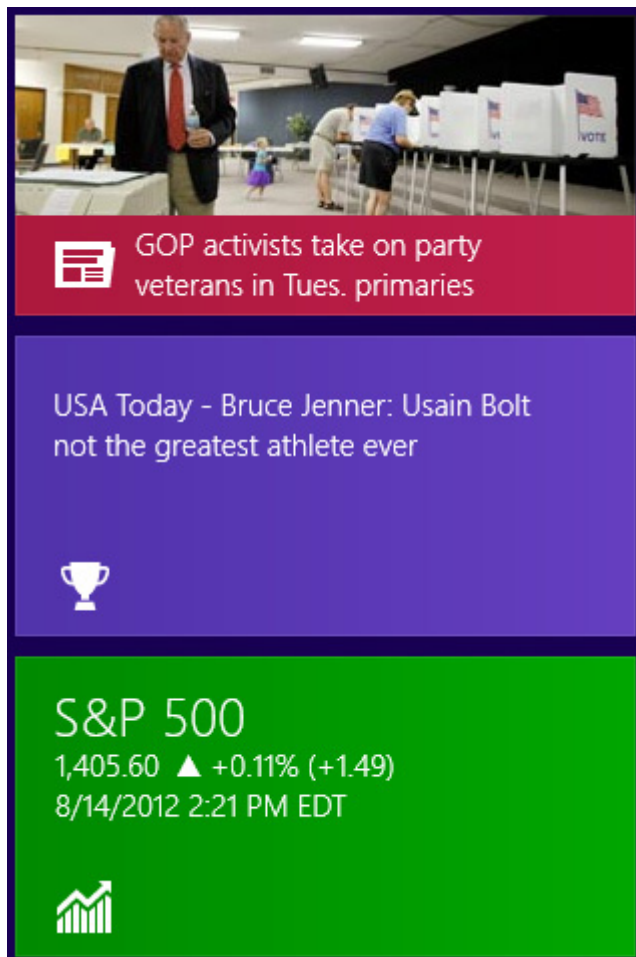


Рис. 1.6. Плитки с информацией о новостях, спорте и биржевых индексах

Благодаря возможности создания вторичных плиток (Secondary Tiles) у приложений может быть несколько плиток одновременно. При нажатии на вторичную плитку пользователь может попасть в определенный раздел приложения.

Таким образом, плитки, "чудо-кнопки" и панели приложения — это как раз те элементы интерфейса, с которыми сталкиваются пользователи и разработчики в первую очередь. Далее мы подробно рассмотрим эти и другие аспекты создания Windows Store-приложений в данной книге, а сейчас поговорим про программную платформу.

Программная платформа

До недавнего времени разработчики приложений для Windows использовали две основные группы API: неуправляемый (native) через Win32 API и управляемый (managed) через .NET Framework. Вторая группа постепенно развивалась, получая различные новые и усовершенствованные библиотеки для создания пользовательского интерфейса (например WPF, Windows Presentation Foundation), работы с данными и сервисами, дополнительные инструменты для разработки, построения исходного кода и архитектуры приложений.

Между тем, сама платформа Windows, т. е. Win32 API получала не так много настоящих толчков к развитию базовой модели разработки. Пожалуй, последним существенным нововведением был COM (Component Object Model), появившийся еще в 90-е годы. Но все это время компьютеры не стояли на месте. Появлялись всевозможные новые устройства, экраны, чувствительные к прикосновениям, возникали новые форм-факторы, такие как планшеты, и т. д. Наконец, такой параметр, как энергопотребление, становился все более важным. Если для Windows 95 энергопотребление почти не имело значения, то для Windows 8 — это один из основных показателей.

Поэтому, создавая новую версию Windows, в Microsoft понимали, что необходимо разработать и новый API, который, будучи родным (native) для операционной системы, станет отвечать новым требованиям и веяниям времени. В результате появился Windows Runtime (WinRT).

Windows Runtime — это новая модель разработки приложений, а также объектно-ориентированный языкнезависимый программный интерфейс (API), написанный на неуправляемом коде и реализующий концепции асинхронного программирования. Все функции и методы, потенциально работающие более 50 мс, реализованы асинхронно. Синхронных аналогов для них нет. Это обеспечивает лучшие характеристики и большую "отзывчивость" приложений.

ПРИМЕЧАНИЕ

Более подробно про асинхронное программирование можно узнать в *приложении 2*.

WinRT работает на основе новой оптимизированной версии COM, при этом благодаря системе метаданных и языковых проекций он может напрямую интегрироваться с управляемыми средами, такими как .NET Framework. Некоторые API, входящие в WinRT, могут быть использованы и в классических приложениях, но большая часть из них доступна только для Windows Store-приложений.

Windows Store-приложения могут создаваться на различных языках программирования: C#, JavaScript, Visual Basic и C++. Хочется отметить, что можно создавать невидимые компоненты WinRT на одном из перечисленных языков (кроме JavaScript), например на языке C#, и встраивать их в приложения, написанные на других языках программирования, например на C++ или JavaScript.

Все программные интерфейсы WinRT выглядят "родными" для каждого из поддерживаемых языков программирования. Поэтому для взаимодействия с WinRT не

придется предпринимать никаких дополнительных усилий, как, например, для взаимодействия с COM из C#. Для разработчика управляемые типы .NET Framework и типы WinRT выглядят одинаково.

Пользовательский интерфейс приложений, написанных на JavaScript, создается с помощью HTML 5, в остальных случаях применяется XAML (eXtensible Application Markup Language — расширяемый язык разметки приложений).

XAML знаком многим разработчикам по WPF и Silverlight. На XAML разрабатывают приложения для Windows Phone. А теперь XAML стал и частью Windows Runtime, а значит, и одним из базовых компонентов операционной системы Windows.

В общем случае Windows Store-приложения изолированы друг от друга. Это обеспечивает стабильность и безопасность как самих приложений, так и системы в целом. Разработчики приложений должны декларативно объявить, какие потенциально небезопасные возможности они будут использовать. Например, если приложение захочет взаимодействовать с камерой, это должно быть объявлено заранее, а при первом обращении к камере система спросит у пользователя, разрешает ли он данное действие. Аналогично обстоит дело и, например, с определением местоположения. Пользователь может явно разрешить или запретить предоставление приложению такой информации.

Архитектуру платформы Windows 8 иллюстрирует рис. 1.7.

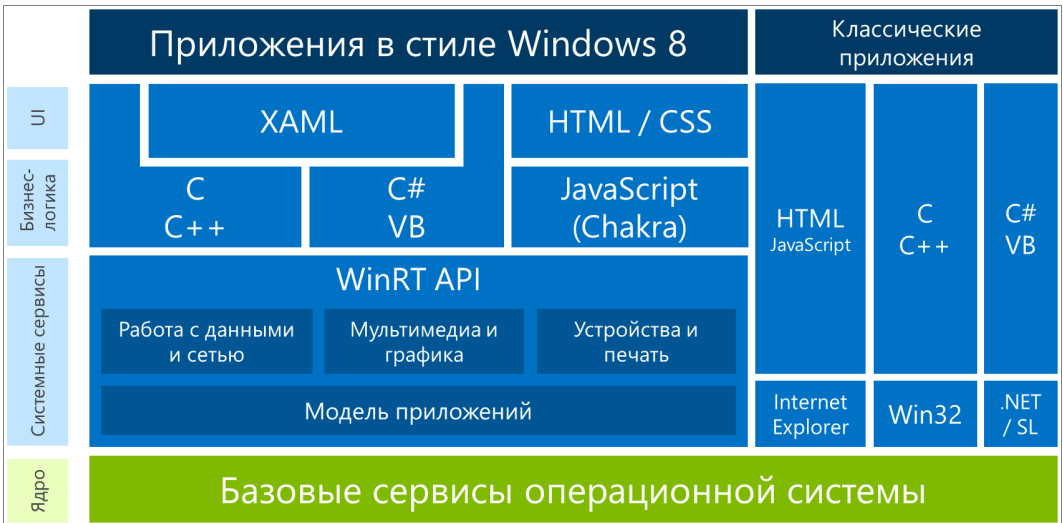


Рис. 1.7. Платформа Windows 8

В данной книге мы рассмотрим создание Windows Store-приложений на языках C# и XAML. В основной части книги мы не будем подробно останавливаться на описании этих языков. Про язык C# написано много хороших книг, например, <http://bhv.ru/books/book.php?id=188312>. Языку разметки XAML посвящено *приложение 1*.

Дизайн Windows Store-приложений

В Windows Store-приложениях используется новое направление (стиль) дизайна пользовательских интерфейсов, сегодня широко применяемое Microsoft, а также независимыми разработчиками в своих продуктах. Данный стиль реализован в Windows 8, Windows Phone, Xbox, а также Web-приложениях и сайтах компании, таких как <http://outlook.com> и др.

Принятый в Windows 8 и других платформах Microsoft дизайн основывается на идеях Баухауса (Bauhaus), швейцарского дизайна (он же International Typographic Style) и анимационного дизайна (Motion Design).

Одним из источников вдохновения разработчиков ОС послужили объекты, встречающиеся нам каждый день и позволяющие быстро сориентироваться в окружающем мире: указатели, дорожные знаки, информационные табло в аэропортах, на вокзалах и в метро (рис. 1.8). Все это характеризует быстрый, современный мобильный мир. Вряд ли человек, опаздывающий на самолет, остановится и будет восхищаться изысканным дизайном указателя со шкурками леопарда. Указатель должен помочь быстро сориентироваться, он обязан быть простым и информативным. Также и приложения призваны помогать наилучшим способом работать с содержимым, а не отвлекать пользователя лишними деталями. Информация первич-

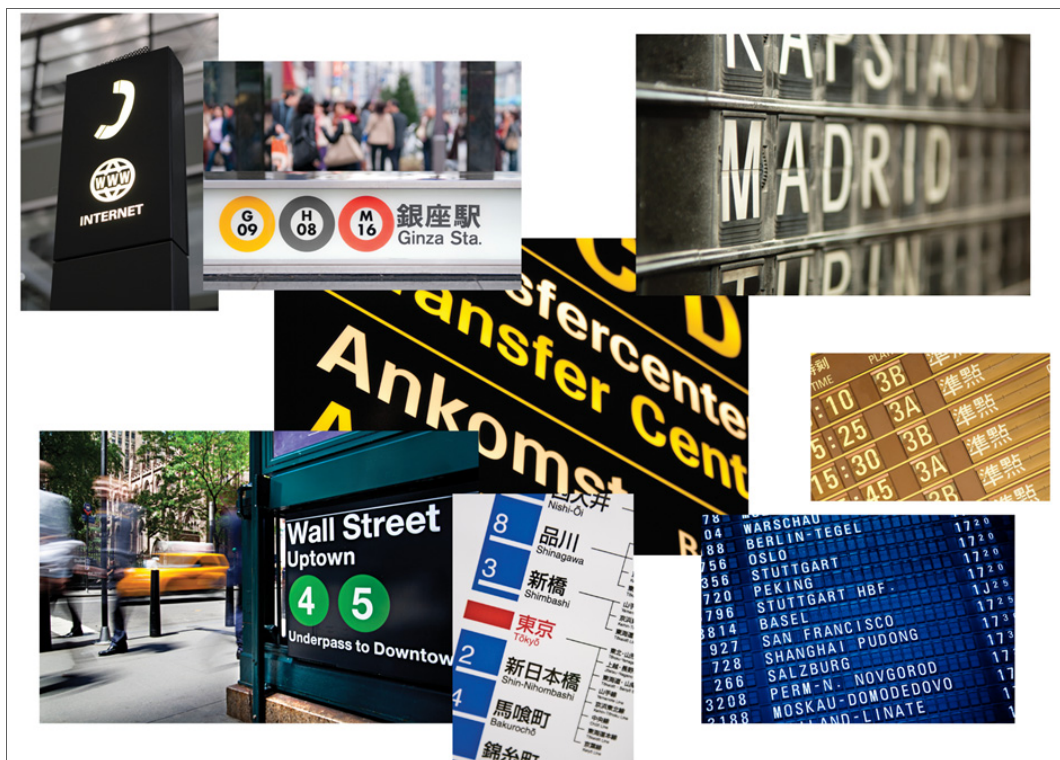


Рис. 1.8. Истоки дизайна приложений Windows 8

на, оболочка вторична. Все элементы максимально упрощены. Поэтому в Windows Store-приложениях нет обилия градиентов, теней и закруглений. Информация, содержимое, — это и есть собственно пользовательский интерфейс, а фон, кнопки и другие элементы управления — только дополнение.

Еще одним важным источником нового стиля стала качественная типографика (типографика — это и искусство, и ремесло, и набор правил, которые используют шрифты и оформительские средства для достижения одной-единственной цели: сделать текст наиболее оптимальным для восприятия). При этом текст, его шрифт (гарнитура шрифта), размер и положение сами по себе являются элементами дизайна и этот дизайн создают. Разные комбинации текста и шрифтов позволяют задать иерархию и расставить приоритеты.

Однако одних "уличных указателей" и шрифтов мало. Сами по себе они не интерактивны. Пользовательский интерфейс Windows Store-приложений подвижен, гибок и интерактивен. Движение и навигация, — один из важных принципов дизайна, применяемого в Windows 8.

Интерфейсы Windows Store-приложений по-настоящему цифровые. Это значит, что мы можем выйти за рамки метафор реального мира и отказаться от некоторых из них. К примеру, в iPad есть приложение iBooks, повторяющее с графической точки зрения вид настоящей деревянной книжной полки. Это прекрасная метафора. И в этом суть иконографического дизайна. Еще один пример иконографического стиля — традиционный интерфейс рабочего стола Windows (рис. 1.9). Но "экранное"

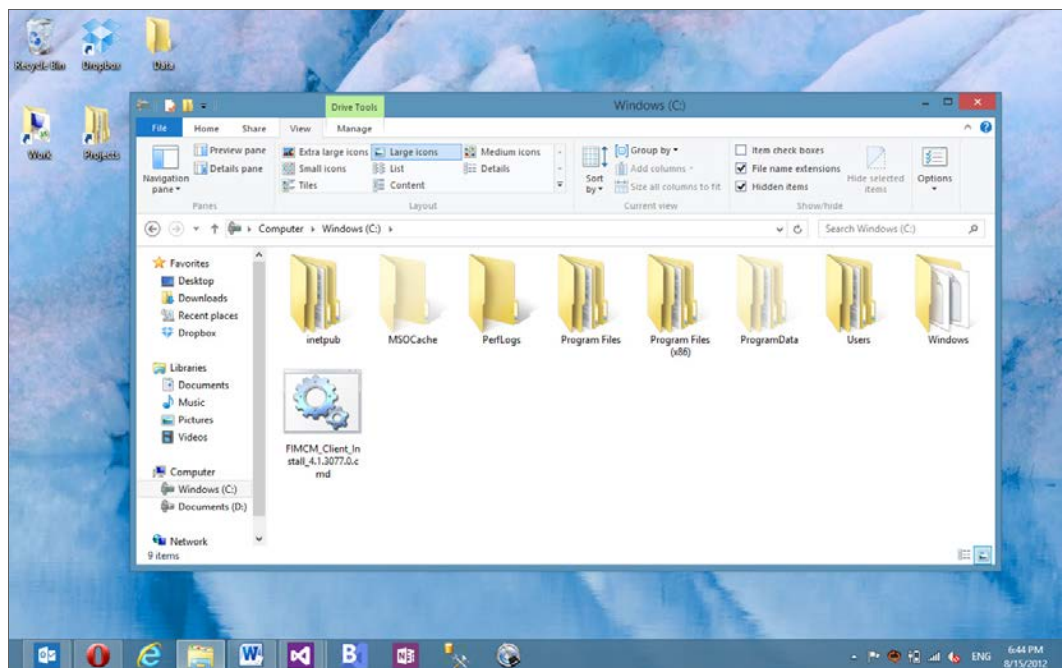


Рис. 1.9. Иконографический дизайн

дерево деревом не является, а книги не обладают весом, чтобы где-то стоять и иметь возможность падать (гравитации тоже нет). Поэтому вся оболочка в виде книжной полки не обязательна, чтобы удобно читать книги. Нужны только обложки книг, чтобы пользователь мог найти требуемую, и сам контент (содержимое книг). Дизайн Windows 8, в противовес иконографическому стилю, является *инфографическим*. Мы работаем с пикселями и не нужно пытаться повторить в цифровом мире метафоры мира реального.

Хочется отметить, что новый стиль дизайна не лучше и не хуже других, у каждого человека может быть свое мнение на этот счет. Дизайн Windows 8 просто другой, не похожий на то, что было раньше. Многие люди, в том числе и авторы данной книги, считают его красивым и удобным.

Итоги

В данной главе мы на базовом уровне рассмотрели архитектуру Windows 8, узнали про новую программную платформу под названием Windows Runtime (WinRT), на которой строятся Windows Store-приложения. Такие приложения можно загружать в магазин Windows и продавать на мировом рынке.

В Windows 8 реализован новый стиль дизайна, который существенно отличается от того, что можно увидеть на других платформах и в более ранних версиях Windows.

Пришло время установить инструменты разработки и начать создавать Windows Store-приложения, чем мы и займемся в следующей главе.



Глава 2

Среда разработки

Все необходимое для разработки приложений для Windows, включая инструменты создания Windows Store-приложений, вы можете найти в центре Windows-разработки на сайте MSDN (Microsoft Software Development Network):

<http://msdn.microsoft.com/windows>

Данный центр доступен и на русском языке по адресу:

<http://msdn.microsoft.com/ru-ru/windows>

В центре Windows-разработки вы найдете ссылки на скачивание SDK (Software Development Kit), актуальную версию документации, примеры и т. д.

Для разработки Windows Store-приложений вам потребуется x86 или x64 версия Windows 8. Разработка Windows Store-приложений на более ранних версиях Windows или на Windows RT для ARM-процессоров не поддерживается.

Если у вас пока нет Windows 8, вы можете бесплатно скачать пробную (не для коммерческого использования) 90-дневную версию. По истечении 90-дневного срока обновить эту версию нельзя, потребуется полная переустановка системы. Но, если вы хотите попробовать разработку Windows Store-приложений, благодаря пробной версии Windows это можно сделать абсолютно бесплатно.

Если вы студент, аспирант или преподаватель и ваш вуз имеет подписку DreamSpark Premium, то можно бесплатно получить полноценную версию Windows 8. Узнайте необходимые детали у администрации своего вуза.

Итак, у вас инсталлирована Windows 8, пришло время установить и все необходимое для разработки Windows Store-приложений.

Главный инструмент для создания приложений на платформе Microsoft — это Visual Studio, имеющий множество редакций. Microsoft предоставляет бесплатную Express-редакцию Visual Studio 2012, позволяющую создавать Windows Store-приложения. Полное название данной редакции — Visual Studio Express 2012 for Windows 8. Для создания Windows Store-приложений вы можете также воспользо-

ваться старшими коммерческими редакциями Visual Studio 2012, такими как Ultimate, Premium и Professional.

Если у вас не установлена одна из старших редакций Visual Studio 2012, скачайте и установите Visual Studio Express 2012. Существует русскоязычная версия данного продукта, но поскольку англоязычный оригинал значительно более популярен среди разработчиков, все примеры данной книги будут выполнены на нем.

Быстро найти ссылку на загрузку Visual Studio Express 2012 for Windows 8 вы можете через Windows Store. Во встроенном приложении Store выполните поиск Visual Studio, откройте страницу продукта (рис. 2.1), а затем нажмите на ссылку **Go to publisher's website**.



Рис. 2.1. Страница Visual Studio Express 2012 в Windows Store

Вместе с Visual Studio Express for Windows 8 в числе прочих продуктов, таких как Windows 8 SDK, будет установлен Blend for Visual Studio. Blend поставляется также и в составе старших редакций Visual Studio.

Blend — это инструмент для создания (дизайна) пользовательского интерфейса приложений. Данный продукт существует уже достаточно давно, есть различные версии Blend, предназначенные, например, для создания приложений WPF (Windows Presentation Foundation), а также приложений для Windows Phone. Blend for Visual Studio позволяет создавать Windows Store-приложения.

Разрабатывать Windows Store-приложения можно полностью в Visual Studio, ни разу не запуская Blend. Однако Blend предоставляет интерфейс, более удобный для формирования дизайна приложений, в то время как в Visual Studio удобней рабо-