Патрик Гецманн, Симон Хакфорт, Петер Новак

Разработка приложений для Windows Phone

Архитектура, фреймворки, API

bhv®

Microsoft Press

Patrick Getzmann, Simon Hackfort, Peter Nowak

Entwickeln für Windows Phone 7.5

Architektur, Frameworks, APIs

Patrick Getzmann, Simon Hackfort, Peter Nowak

Entwickeln für Windows Phone 7.5

Architektur, Frameworks, APIs

Microsoft Press

Патрик Гецманн
Симон Хакфорт
Петер Новак

Разработка приложений для Windows Phone

Архитектура, фреймворки, API

Санкт-Петербург
«БХВ-Петербург»
2014

УДК 004.451+004.4'2
ББК 32.973.26-018.2
Г44

Гецманн, П.

Г44 Разработка приложений для Windows Phone. Архитектура, фреймворки, API: Пер. с нем. / П. Гецманн, С. Хакфорт, П. Новак. — СПб.: БХВ-Петербург, 2014. — 880 с.: ил.

ISBN 978-5-9775-0855-1

Рассмотрена разработка приложений и игр для Windows Phone 7.5 в среде Visual Studio 2010 Express for Windows Phone, Microsoft Expression Blend for Windows Phone и XNA Game Studio 4.0. Описана архитектура Windows Phone и особенности создания пользовательского интерфейса Metro. Показана разработка приложений Silverlight и игр для Windows Phone. Рассмотрено использование и создание элементов управления приложением, специализированные API (для работы с жестами, датчиками ускорения, компаса, гироскопа, геолокации и др.). Описана работа с сетями и базами данных. Показано тестирование приложений, анализ их производительности и размещение в интернет-магазине Windows Phone Store. Уделено внимание миграции существующих игр XNA, приложений Silverlight или Windows Mobile на новую платформу.

Для программистов

УДК 004.451+004.4'2
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Игорь Шишигин*
Зав. редакцией	*Екатерина Капалыгина*
Перевод с немецкого	*Виктора Букирева*
Редакторы	*Григорий Добин, Юрий Рожко*
Компьютерная верстка	*Ольги Сергиенко*
Корректор	*Зинаида Дмитриева*
Оформление обложки	*Марины Дамбиевой*

© 2014, BHV-St. Petersburg
Authorized Russian translation of the German edition of Entwickeln für Windows Phone 7.5© 2011 O'Reilly Verlag GmbH & Co. KG (ISBN 9783866454651). This translation is published and sold by permission of O'Reilly Media, Inc.

© 2014, БХВ-Петербург
Авторизованный перевод с немецкого на русский язык издания Entwickeln für Windows Phone 7.5© 2011 O'Reilly Verlag GmbH & Co. KG (ISBN 9783866454651). Издание переведено и поступило в продажу с разрешения O'Reilly Media, Inc.

Подписано в печать 30.09.13.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 70,95.
Тираж 1000 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-3-86645-465-1 (нем.)
ISBN 978-5-9775-0855-1 (рус.)

Оглавление

Предисловие от сотрудников Microsoft **1**

Предисловие от авторов **3**
Цель и мотивация этой книги 3
Об авторах 5
Благодарности 5

ГЛАВА 1. Вступление **7**
Много нового 7
Устранение недостатков 9
Мультипрограммный режим против управляемой системы 9
Silverlight против Windows Forms 10
Управляемый код против неуправляемого кода 12
Почему XNA? 12
Открытая система против закрытой системы 13
Что требуется уже знать, перед тем как начать? 13
Далее... 14
Ошибки в этой книге 15
Путаница с номерами версий 15

ГЛАВА 2. Привет, новый мир **17**
Учимся работать 17
Microsoft Visual Studio 2010 Express для Windows Phone 18
Microsoft Expression Blend для Windows Phone 20
XNA Game Studio 4.0 21
Эмулятор 23
... и что еще? 24
Полнофункциональные и Express-версии 26

ГЛАВА 3. Из чего состоит Windows Phone **27**
Новая платформа аппаратных средств 27
Дисплей и разрешение 28
Процессор и графика 29
Память 29

Клавиатура и органы управления 29
Кнопка питания 30
Кнопки для регулирования громкости 30
Кнопка камеры 30
Кнопка *Back* 30
Кнопка *Start* 30
Кнопка *Search* 30
Аудио 30
Датчики 31
Акселерометр 31
Гироскоп 31
A-GPS 32
Компас 32
Датчик приближения 32
Датчик освещенности 32
Подключения 32
Прочие аппаратные средства 33
Итог 33
Новая платформа программного обеспечения 33
Уровень приложений 33
Модель приложений в Windows Phone 35
Модель пользовательского интерфейса в Windows Phone 37
Модель навигации 37
Оболочка фрейма 38
Облачные сервисы (Cloud Services) 39
Ядро Windows Phone 40
Дизайн Metro 40
Разработка с Windows Phone 44
NuGet 45

ГЛАВА 4. Приложения Silverlight для мобильных устройств Windows Phone 49
Разработка приложений Silverlight для Windows Phone 49
Windows Phone SDK 49
Visual Studio 2010 49
Expression Studio 4 50
"Hello World" 51
Еще раз о языке дизайна Metro 54
Панель состояния и панель приложения 55
Ориентация экрана 55
Стандартный тип шрифта 56
Push-уведомления 56
Структура графических интерфейсов 56
Индикатор хода выполнения (Progress Indicator) 56
Темы 57
Конфигурация приложения 57
Сенсорный экран и жесты сенсорного ввода 57
Аппаратные кнопки 58
Виртуальная экранная клавиатура 58

Датчики ... 59
Элементы управления ... 60
Уведомления (Notifications) Windows Phone ... 60
Форматирование текста ... 61
Методы увеличения производительности ... 62
Подвижный дизайн Metro ... 62
Фреймы и навигация между страницами ... 62
Элементы управления для Windows Phone ... 67
Button ... 67
CheckBox ... 68
RadioButton ... 70
TextBox ... 72
Grid ... 73
HyperlinkButton ... 75
Image ... 77
PasswordBox ... 78
ProgressBar ... 79
TextBlock ... 80
RichTextBox ... 82
ListBox ... 85
PhoneApplicationFrame ... 90
PhoneApplicationPage ... 90
Border ... 90
Canvas ... 92
InkPresenter ... 94
MediaElement ... 96
MessageBox ... 98
MultiScaleImage ... 99
Popup ... 102
WebBrowser ... 103
ToolTip ... 107
Slider ... 109
StackPanel ... 110
ScrollViewer ... 112
Panorama ... 114
Pivot ... 118
Bing Maps ... 120
Прочие элементы управления Silverlight для Windows Phone ... 128
Набор инструментов Silverlight for Windows Phone ... 129
ContextMenu ... 130
WrapPanel ... 131
DatePicker и *TimePicker* ... 132
ToggleSwitch ... 133
AutoCompleteBox ... 134
ListPicker ... 135
LongListSelector ... 136
MultiselectList ... 138
LockablePivot ... 138
HubTile ... 140

ExpanderView ... 142
DateTimeConverter и *LocalizedRessources* ... 144
PhoneTextBox ... 145
PerformanceProgressBar ... 147
TiltEffect ... 147
Страничные переходы ... 148
GestureListener ... 148
Coding4Fun Windows Phone Toolkit ... 149
Дополнительные компоненты и библиотеки ... 151
Пакет Rad Controls for Windows Phone от Telerik ... 151
Инструменты NetAdvantage for Windows Phone от Infragistics ... 152
Компонент Studio for Windows Phone ... 152
Введение в Expression Blend для разработчиков ... 153
Создание собственного элемента управления в Silverlight для Windows Phone ... 173
Разработка интерфейсов и их особенности (Theming) ... 182
Кисти ... 183
Цвета ... 183
Наименования шрифтов ... 183
Размеры шрифтов ... 183
Толщины ... 183
Текстовые стили ... 183
Видимость и непрозрачность ... 183
Стилизованное оформление элементов управления ... 185
Глобализация и локализация ... 189
Глобализация ... 189
Локализация ... 191
Локализация названия приложения ... 199
Интеграция в Silverlight ... 200
Интеграция в XNA ... 202
Краткое введение в привязку данных Silverlight ... 203
Рекомендации по увеличению производительности ... 214
Производительность ... 214
Память ... 215
Производительность во время загрузки ... 215
Элемент управления *ListBox* ... 215
Применение элемента управления *Panorama* ... 215
Применение элемента управления *Pivot* ... 216
Общие рекомендации по оптимизации ... 216
Файлы манифеста ... 217
Предварительная подготовка, производимая вручную ... 221
LINQ ... 226
Основы ... 227
LINQ to Objects ... 227
LINQ to XML ... 231
LINQ to JSON ... 234
LINQ to SQL ... 237
Reactive Extensions (Rx) ... 237
Шаблон Observer ... 237
Rx, шаблон Observable и LINQ ... 243

Шаблон Model-View-ViewModel (MVVM) 245
"Hello World" с MVVM 247
Шаблон MVVM — это другой подход 251
MVVM Light Toolkit 253

ГЛАВА 5. Разработка игр с XNA Game Studio 4.0 261
Мобильные игры 261
XNA Framework 262
Шаблоны проектов 262
Другие компоненты XNA Game Studio 4.0 263
Принцип работы игры 264
Какие возможности предоставляет XNA Framework? 265
Каких возможностей не предоставляет XNA Framework? 266
Структура проекта игры XNA 266
Hallo Windows Phone 272
Полноэкранный режим 272
Вывод текста 272
Графический объект 274
Ввод данных 277
Игры XNA для Windows Phone 279
Первая игра 279
Игровые ресурсы 279
Структура игры 280
Что нам поможет запрограммировать *MultiTouch*? 281
Указание директив *using* и ссылок 281
Переменные 283
Конструктор и жизненный цикл 285
Инициализация и загрузка ресурсов 287
Обновления состояния игры 288
Графическое оформление игры 293
Следующие шаги 295
Основа для другой игры 295
Суть игры 296
Разделение на компоненты 296
Класс *Game* 297
Морская звезда 299
Спрут 304
Рыбы 309
Следующие шаги 313
Основы 3D 314
3D и Windows Phone 318
Загрузка модели 3D 318
Изменение текстуры 321
Масштабирование 324
Вращение 325
Движение 328
Комбинация 330
Silverlight и XNA 334
Новые шаблоны проекта 335

Принцип работы 336
Практика 337
Непосредственное разделение 337
Смешанный вид 340
Привязка данных и раскадровки 344
Мультиплатформенная поддержка 349
Xbox LIVE 351

ГЛАВА 6. Работа с платформой интерфейсов API 353
Жизненный цикл приложения Windows Phone 353
Геопозиционирование 361
GPS и триангуляция GSM 361
Эмуляция GPS 367
Радио FM 370
Особенности класса *FMRadio* 370
Слушаем радио 371
Подробнее о классе *FMRadio* 372
Работа с жестами 372
Одиночные жесты 373
Жесты многократных касаний 373
Обработка собственных жестов разработчика 374
Жесты в Windows Phone 374
Собственные жесты Silverlight для Windows Phone 374
Жесты с одним пальцем (Touch) 374
Жесты с несколькими пальцами (Multi-Touch) 381
Дополнительно о Multi-Touch 385
Многократные касания и жесты в XNA 386
Свойства класса *TouchPanel* 386
Многократные касания и XNA 386
Жесты и XNA 388
Обработка жестов 388
Класс *TouchPanel* от XNA пригодится и в Silverlight 393
Датчик ускорения 394
Датчик компаса 403
Датчик гироскопа 408
Интерфейс *Motion-API* 411
Панель *ApplicationBar* 417
Вибратор 422
Микрофон 425
Виртуальная клавиатура 430
Работа с задачами 434
Утилита запуска *PhoneCallTask* 435
Параметры 435
Пример 436
Утилита выбора *PhoneNumberChooserTask* 436
Параметр 436
Метод 436
Событие 437
Пример 437

Объект *PhoneNumberResult* 437
Параметры 438
Пример 438
Утилита выбора *PhotoChooserTask* 438
Параметры 438
Метод 439
Событие 439
Пример 439
Объект *PhotoResult* 440
Параметры 440
Пример 441
Утилита выбора *EmailAddressChooserTask* 441
Параметр 442
Метод 442
Событие 442
Пример 442
Утилита запуска *EmailComposeTask* 443
Параметры 443
Метод 443
Пример 443
Объект *EmailResult* 443
Параметры 444
Пример 444
Утилита запуска *MediaPlayerLauncher* 444
Параметры 446
Метод 446
Пример 446
Утилита запуска *SearchTask* 446
Параметр 447
Метод 447
Пример 447
Утилита запуска *SmsComposeTask* 448
Параметры 448
Метод 448
Пример 448
Утилита запуска *WebBrowserTask* 449
Параметр 449
Метод 449
Пример 449
Класс *TaskEventArgs* 450
Свойства 450
Метод 450
Пример 450
Утилита выбора *SaveEmailAddressTask* 450
Параметр 451
Метод 451
Пример 452
Утилита выбора *SavePhoneNumberTask* 452
Параметр 453

Метод 453
Пример 454
Утилита выбора *CameraCaptureTask* 454
Метод 455
Пример 456
Утилита выбора *AddressChooserTask* 457
Параметр 457
Метод 457
Событие 458
Пример 458
Утилита запуска *BingMapsTask* 458
Параметры 459
Метод 459
Пример 460
Утилита запуска *BingMapsDirectionsTask* 460
Параметры 461
Метод 462
Пример 462
Утилита запуска *ConnectionSettingsTask* 463
Параметр 464
Метод 464
Пример 465
Утилита выбора *GameInviteTask* 465
Параметры 466
Метод 466
Событие 466
Пример 466
Утилита выбора *SaveContactTask* 467
Параметры 467
Метод 469
Событие 469
Пример 469
Утилита выбора *SaveRingtoneTask* 470
Параметры 471
Метод 471
Событие 471
Пример 471
Утилита запуска *ShareLinkTask* 472
Параметры 473
Метод 473
Пример 473
Утилита запуска *ShareStatusTask* 474
Параметр 475
Метод 475
Пример 475
Фоновые сервисы 475
AudioPlayerAgent и *BackgroundAudioPlayer* 476
AudioStreamingAgent и *BackgroundAudioPlayer* 487
ScheduledTaskAgent 488

Фоновая пересылка файлов 494
Плитки 502
Чтение информации об устройствах 506
Получение анонимного идентификатора пользователя 508
Расширение работы с изображениями 509
Предотвращение блокировки экрана 514
Ориентация изображения на дисплее 515
Реализация ориентации в Silverlight 515
Реализация ориентации в XNA 523
Работа с файловой системой 526
Создание и запись файла 527
Чтение и удаление файла 529
Создание и удаление подкаталогов 531
Список файлов и подкаталогов 532
Очистка папки Isolated Storage 536
Isolated Storage Explorer 536
Список содержимого изолированной памяти 537
Копирование изолированной памяти на персональный компьютер 537
Возврат измененных данных на устройство 538
Альтернативы 538
Работа с дополнительным содержимым приложения 538
Работа с дополнительными файлами 539
Работа с внедренными ресурсами 541
Сжатие данных 542
Как функционирует сжатие? 543
Архивация и деархивация потоков данных 543
Работа с архивами ZIP 545
Работа с локальными данными 549
Локальные настройки приложения 549
Сериализация и десериализация объектов 551
Сериализация с *XmlSerializer* 552
Сериализация с *DataContractSerializer* 554
Сериализация с использованием *DataContractJsonSerializer* 557
Работа с *FrameRateCounter* 557
Свойство *Redraw Regions* 560
Технология растрового кэша (BitmapCache) 560
Визуализация кэша 561
Доступ к камере 563
Silverlight 4 Webcam-API 564
Класс *PhotoCamera* 566
Возможности управления 567
Тип камеры 567
Автофокусировка 567
Фокус на определенную точку 567
Разрешение 567
Вспышка 568
Кнопки камеры 568
Реализация функции *PhotoCamera* 568

Работа с напоминаниями и будильниками 575
Создание напоминания....576
Создание будильника....579
Работа с данными контакта и календарными данными....580
Запрос данных контакта через *SearchAsync* 580
Запрос календарных данных....584

ГЛАВА 7. Работа с сетями 587
Сервисы сети....587
Веб-сервисы 589
Сервисы WCF....592
Сервисы WCF Data и протокол OData 601
Чтение данных сервиса WCF Data 604
Изменение данных сервиса WCF Data....609
Сохранение данных сервиса WCF Data 611
Доступ к другим веб-ресурсам 614
Классы *WebRequest* и *WebResponse*....617
Аутентификация в Windows Phone с помощью технологии OAuth 621
Push-уведомления 621
Toast-уведомления 623
Tile-уведомления....633
Raw-уведомления....641
Уведомление "облачного" приложения....647
Взаимодействие с элементами управления веб-браузера....647
Класс *NetworkInterface* 650
Класс *DeviceNetworkInformation*....653
Классы *NetworkInterfaceList* и *NetworkInterfaceInfo*....655
Работа с утилитой Fiddler....656
Сокеты 660
Соединение по протоколу TCP....660
Простые сервисы TCP/IP 661
Примеры простых сервисов TCP 662
Соединение по протоколу UDP 665
Использование UDP Unicast 666
Использование UDP-Multicast 669
Класс *SocketExtensions*....674

ГЛАВА 8. Работа с базами данных в Windows Phone 679
Первые шаги 679
Строка *DbConnectionString*....692
Изменение структуры базы данных 694
Взаимосвязи между данными....699
Связь 1:1 699
Ассоциации 701
Каскадное удаление....703
Связь 1:N 705
Связь N:M....710
Протоколирование данных SQL....721
Использование существующих баз данных....721

Повышение скорости запроса 726
Компилируемый запрос 726
Индексация 731

ГЛАВА 9. Протоколирование и модульное тестирование в Windows Phone 733
Модульное тестирование 733
Протоколирование 745

ГЛАВА 10. Работа с утилитой Windows Phone Performance Analysis 749
Запуск утилиты Performance Analysis 749
Подробный анализ выполнения 753
Подробный анализ памяти 757

ГЛАВА 11. Размещение 765
Предварительная подготовка для Windows Phone Store 765
Методы распределения 766
Public 766
Targeted 766
Beta 766
Режим Trial для собственного приложения 767
Бесплатное, сопряженное с возмещением расходов или рекламно-финансированное приложение? 769
Реклама в собственном приложении 769
Реклама в Silverlight для Windows Phone Apps 770
Реклама в XNA для игр Windows Phone 771
Альтернативный рекламный фреймворк для Windows Phone 773
Запускающие модули для Store 774
MarketplaceHubTask 774
MarketplaceDetailTask 774
MarketplaceReviewTask 775
MarketplaceSearchTask 775
Размещение 776
Правила сертификации Windows Phone Store 776
Рекомендации по содержанию 776
Рекомендации по приложению 777
Производительность и использование памяти 779
Размеры графического объекта 779
Интеграция в центр музыки и видео и расширения изображений 780
Предпосылки для инсталляционных XAP-файлов 780
Процесс сертификации 780
Профиль Dev Center 781
Активация устройства Windows Phone 781
Подача приложения 785
Шаг 1. Информация о приложении 785
Шаг 2. Загрузка приложения 787
Шаг 3. Информация о ценах 787
Шаг 4. Картографические сервисы 787
Marketplace Test Kit 787

Подключение устройств с ПО Zune-Desktop 791
Windows Phone Connect Tool 795

ГЛАВА 12. Миграция игр и приложений 797
Общие соображения 797
Миграция существующих игр XNA 799
Шаги миграции 800
Миграция проекта XNA 3.1 для ПК на платформу Windows Phone 805
Разрешение 807
Анимация и частота смены кадров 809
Управление 810
Миграция существующих приложений Silverlight 812
Миграция приложений Windows Mobile 813
Миграция компонента *View* 815
Миграция компонентов шаблона MVC *Model* и *Controller* 816
Создание компонента *ModelView* 817
Миграция приложения Windows Phone 7.0 на Windows Phone 7.5 817
Итоги 819

ПРИЛОЖЕНИЕ. Прочее 821
SketchFlow 821
Создание прототипа 823
Согласование проекта (Feedback) 828
Анализ материалов согласования 829
Советы и рекомендации по обеспечению оптимальной производительности 831
Работа с графическими объектами 832
Кэширование изображений из Интернета 832
Уменьшение размеров сборки 843
Исходный код в конструкторе и во время события *Loaded* 843
Работа с элементами *ListBox* 843
Дополнительные ресурсы 843
Разработка 844
Базы данных 844
Элементы управления 844
Инструменты для Frameworks 845
Архитектура и доступ к данным 845
XNA Framework 845
Дополнительные источники информации 846
Обновление глав 846
Пример кода для загрузки 847

Предметный указатель 849

Предисловие от сотрудников Microsoft

Это просто здорово, что члены сообщества Microsoft MVP[1] пишут книги для разработчиков Windows Phone[2]. Наличие экспертов, которые провели время с командой инженеров Windows Phone, идущих через создание приложений, является одним из лучших способов для любого разработчика узнать, как создать большие приложения Windows Phone.

Ахим Берг (Achim Berg), корпоративный вице-президент Windows Phone Marketing, корпорация Microsoft

Для меня является большой честью предложение одного из наших специалистов MVP по Windows Phone написать предисловие для этой книги. Когда Питер, Патрик и Симон в первый раз подошли ко мне с этой просьбой, я не смог достаточно быстро дать согласие. Это самое малое, что я мог бы сделать для команды, которая потратила столько времени и энергии, помогая корпорации Microsoft создать экосистему для разработки Windows Phone. Патрик и Питер оба имеют статус MVP-специалистов по Windows Phone, и это сотрудничество с Симоном является примером поддержки сообщества разработчиков Windows Phone лицами, которые очень страстны и талантливы.

Область мобильных телефонов продолжает расти с каждым годом, все больше и больше людей выходят в Интернет в первый раз, и многие из них делают это со смартфона, а не с ПК. Предвкушения мобильных клиентов несколько другие, в отличие от пользователей ПК. Их ожидания возрастают с опытом, который они приобретают с телефоном, и это под-

---

[1] *MVP* (Most Valuable Professionals, наиболее опытные профессионалы) — это статус, который корпорация Microsoft присваивает специалистам в области ИТ-технологий, которые не являются сотрудниками Microsoft, но активно принимают участие в различных сообществах, а также в блогах, форумах и т. д. Специалисты MVP являются признанными экспертами в области ИТ-технологий, часто они привлекаются корпорацией Microsoft в качестве докладчиков на семинары, форумы, конференции. Статус MVP-специалиста дает им право доступа к закрытым техническим ресурсам корпорации Microsoft, а в корпорации выделяется человек, ответственный за взаимодействие с такими специалистами. — *Ред.*

[2] *Windows Phone* — мобильная операционная система, разработанная Microsoft. — *Ред.*

питывается увеличением производительности сетей, поддерживающих телефоны, мобильные телефоны, которые клиенты используют, и операционные системы, работающие на этих мобильных телефонах.

В этих ожиданиях роста упор делается все более и более на разработчиков мобильных приложений и сервисов, чтобы было просто немного лучше. Любой пустяк может вызвать затруднения. Проблемой является все. Разработчики любят сложные задачи. Питер, Патрик и Симон предоставили для вас способ преодоления этих проблем при создании удивительных приложений для Windows Phone. Эта книга написана разработчиками для разработчиков, так что вы знаете, что все ваши проблемы они, наверное, уже предвидели. Эта книга является путеводителем через иногда бурные воды вашего путешествия, чтобы донести ваше приложение от вашего воображения до ваших клиентов.

Windows Phone это удивительный интегрированный опыт, где на первом месте стоят люди, а не предмет поклонения. Приложения работают так, как потребители ожидают этого от них, только при этом доводятся до уровня совершенства и магии на телефоне, и не могут быть обнаружены на других платформах. Это одна из причин, почему так много приложений добавляется в Windows Phone с функциями, которые невозможно найти в других крупных разработках мобильных платформ. Вы найдете так много интересного при создании приложений для Windows Phone, начиная от Live Tiles, которые появляются на начальном экране, до всеобъемлющего языка дизайна Metro и удивительных средств разработки.

Система Windows Phone 7.5 имеет слоган "Put people first" (Прежде всего люди). Наша команда ставит разработчиков в первую очередь — разработчиков, таких как Питер, Патрик и Симон, и как вы. Если у вас есть удачное приложение на других платформах, пусть эта книга поможет вам выпустить это приложение для Windows Phone. Если у вас есть новая идея приложения, подумайте, не может ли холст Windows Phone помочь вам мечтать немного больше. Каждый разработчик имеет материал для обсуждения. Каждый.

Брэндон Уотсон, старший директор Windows Phone Apps и Developer Experience, корпорация Microsoft ***http://www.manyniches.com/***

Предисловие от авторов

Цель и мотивация этой книги

У этой книги имеется несколько целей. Таким образом, основная цель — довести платформу Windows Phone до разработчиков *Silverlight*[1] и *XNA*[2] и объяснить новые возможности *Windows Phone SDK*[3] *7.1* для существующих разработчиков коротко и лаконично. В этой области есть еще огромное количество маркетинговых суждений, например, что каждый разработчик Silverlight и XNA уже является разработчиком Windows Phone, но это не совсем так. Прошлый год с публикации Windows Phone показал, в частности, через руководства *Marketplace*[4], что разработка для Windows Phone не всегда так же проста и знания исходных основ через мобильные приложения приносят пользу. Это не менялось с Windows Phone 7.5. Так или иначе: даже дилетант системы Mobile[5] с инструментами может быстро получить результат, если уже имеется в наличии опыт с Silverlight и XNA.

Скорее разработчик Silverlight редко должен разбираться с темой "Ориентация" и создавать для этого, возможно, две разработки для его приложения, которые представляются в вертикальном и горизонтальном формате. Разработчик XNA также скорее не заботится о теме

---

[1] *Microsoft Silverlight* — это название новой графической системы и технологии представления данных в Интернете, предназначенной для запуска на различных платформах. — *Ред.*

[2] *XNA* — набор инструментов с управляемой средой времени выполнения (.NET), созданный Microsoft, облегчающий разработку и управление компьютерными играми. — *Ред.*

[3] *Windows Phone SDK* — пакет средств разработчика (*SDK* — Software Development Kit) для Windows Phone предоставляет все необходимые средства для разработки приложений и игр для устройств под управлением Windows Phone. — *Ред.*

[4] *Windows Phone Marketplace* (в настоящее время, с сентября 2012 года, переименована компанией Microsoft в *Windows Phone Store*) — является интернет-магазином приложений Microsoft для мобильной платформы Windows Phone, который предназначен для скачивания и приобретения различных приложений и игр. — *Ред.*

[5] *Windows Mobile* — мобильная операционная система, разработанная Microsoft для собственных аппаратных платформ коммуникаторов Pocket PC и смартфонов. — *Ред.*

"Потребление тока", поскольку Xbox 360[1] и ПК располагают постоянным электропитанием. И уже в самом начале появляются такие вопросы, как: "Что такое SIP[2]? Почему там требуются различные размещения? Почему у меня отсутствуют в известном классе *x* методы *y* и *z*, которые я использовал раньше? И почему я должен совмещать Silverlight и XNA в приложении?"

Эта книга не только укажет различия с Silverlight и XNA Framework, но и ближе познакомит с особенностями платформы Windows Phone. Для этого предназначены, в том числе, также уведомления Push, которые еще в этом виде раньше отсутствовали в контексте Microsoft.

Кроме того, представляются не только возможности применения различных технологий. Скорее здесь философия учитывается также как идея, почему решились со стороны корпорации Microsoft сделать так, как решили. К этому принадлежит и язык дизайна Metro[3] (Metro Design Language), который нашел применение тем временем также на Xbox и под Windows 8.

В конце концов, можно сказать, что эта книга рассчитана не для начинающего разработчика Silverlight или XNA, и, безусловно, требует определенных базовых знаний. Для изучения Silverlight и XNA отдельно имеются намного лучшие и более толстые книги, которые могут способствовать необходимому знанию основ более основательно и более всеобъемлюще.

Тем не менее, имеется также вторая цель или, точнее, целевая группа читателей этой книги — разработчики Windows Mobile. Именно им придется особенно тяжело, прежде всего потому, что система Windows Phone функционирует как таковая принципиально иначе, чем Windows Mobile. Ничего из старых знаний, кроме знания C#, Visual Basic .NET и Visual Studio, здесь не осталось. Начиная уже с того, что никакого прямого доступа к аппаратным средствам не поддерживается. Разработка на C++ не поддерживается, можно разрабатывать только используя C# или Visual Basic .NET[4]. Больше не имеется привычного доступа к памяти устройств, т. к. применяется концепт *изолированного хранения* (Isolated Storage) — прежде всего, для пользы безопасности устройств. Разработчики скорее оказываются длительное время под давлением Windows Mobile. Хотя Windows Mobile все еще поддерживается и предлагается, однако, постепенно сокращается количество фирм, которые и далее применяли бы ее, так что "перенос программного обеспечения" постепенно становится необходим.

В этой книге можно найти эту и другую более подробную информацию для того, чтобы вы смогли ориентироваться в экосистеме Windows Phone.

Возможно, вам бросится в глаза, что в этой книге отсутствуют некоторые скриншоты. В частности, они отсутствуют тогда, когда речь идет о некоторых функциях или диалоговых окнах, которые функционируют только на Windows Phone и не могут представляться в эмуляторе. К сожалению, Windows Phone также и в версии 7.5 все еще не предоставляет возможность выполнять скриншоты. Это позволило бы в результате обойтись без цифровой камеры.

---

[1] *Xbox 360* — игровая приставка корпорации Microsoft, конкурирующая с Sony PlayStation 3 и позволяющая играть через Интернет, а также загружать разнообразный контент. — *Ред.*

[2] *SIP* (Session Initiation Protocol — протокол установления сеанса) — протокол передачи данных, который описывает способ установления и завершения пользовательского интернет-сеанса, включающего обмен мультимедийным содержимым. — *Ред.*

[3] *Metro Design Language* — внутреннее кодовое название дизайнерского языка компании Microsoft, ориентированного на типографское оформление. Язык разработан Microsoft для создания пользовательского интерфейса в Zune и Windows Media Center в мобильной операционной системе Windows Phone 7. — *Ред.*

[4] В Windows Phone 8 вновь вернули возможность разработки native-приложений на C++. — *Ред.*

Об авторах

Патрик Гецманн с января 2010 года является MVP-специалистом Microsoft в качестве разработчика приложений для устройств (Device Application Development), а с октября 2010 года — MVP-специалистом Microsoft в качестве разработчика приложений для Windows Phone. С начала 2000 года он занимается разработкой, концепцией и архитектурой мобильных приложений на платформах Microsoft. Патрик охотно делится знаниями в своем блоге **http://patrickgetzmann.wordpress.com**. Также с ним можно связаться по электронной почте: **patrick.getzmann@t-online.de**.

Симон Хакфорт — сертифицированный программист SCJP (Sun Certified Java Programmer). С 2002 года занимается разработкой высокопроизводительных и дружественных интерфейсов мобильной платформы Microsoft. Симон пишет статьи для различных отраслевых журналов и участвует в конференциях по мобильным технологиям. С ним можно связаться по электронной почте: **simon.hackfort@live.de** или посетить его блог в Интернете по адресу: **www.simonhackfort.net**.

Петер Новак с апреля 2008 года имеет статус Microsoft MVP в качестве разработчика приложений для устройств, а с октября 2010 года является MVP-специалистом Microsoft в качестве разработчика приложений для Windows Phone. С 2002 года он занимается разработкой, концепцией и архитектурой в области .NET. Петер соавтор книги "Программирование с .NET Compact Framework", выпущенной Microsoft Press. Он пишет статьи для различных отраслевых журналов и участвует в конференциях по мобильным технологиям. Связаться с ним можно по электронной почте: **peter.nowak@hotmail.de** или через Twitter: **@PeNoWiMo**. Его блог находится по адресу: **http://winphonedev.de/**.

Благодарности

Собственно, на обложке этой книги стоят имена только трех авторов. Строго говоря, нужно сказать, что речь здесь идет о книге сообщества, в которую многие его члены внесли долю информации.

Прежде всего, мы благодарим следующих людей, без которых эта книга не была бы возможна:

- Сандра Мишель (Sandra Michel), которая работала ранее в Microsoft Press, и, к сожалению, не смогла сопроводить этот проект до конца в связи со сменой работы.
- Томас Браун-Визхоллер (Thomas Braun-Wiesholler), Microsoft Press Division при издательстве O'Reilly, который читал курс лекций и реализовывал наши идеи.
- Стефани Рейманн (Stephanie Reimann), старший менеджер по продукции — разработчик экосистемы при корпорации Microsoft в Редмонде, который всегда поддерживал нас во всех отношениях.
- Брэндон Уотсон (Brandon Watson), старший директор Windows Phone Apps и Developer Experience при корпорации Microsoft в Редмонде, который внес свою часть предисловия к книге, а также всегда был открыт для всех.
- Клифф Симпкинс (Cliff Simpkins), старший менеджер по продукции Windows Phone Apps и Developer Experience при корпорации Microsoft в Редмонде, который отвечал нам на все мыслимые и немыслимые вопросы. Он всегда был открыт для авторов этой книги.
- Ахим Берг (Achim Berg), корпоративный вице-президент Windows Phone Marketing, предоставил несколько слов об этой книге.

- Оливер Шеер (Oliver Scheer) от Developer Platform & Strategy Group der Microsoft Deutschland GmbH, который в первом издании выручил как технический редактор и снова был доступен в этот раз.
- Ута Бергофф (Uta Berghoff) и Silja Brands за монтаж оригинала.
- Катрин Летцел (Katrin Letzel), руководитель MVP при Microsoft Deutschland GmbH. Она всегда была внимательна и оказывала помощь, где это было только возможно.
- Джинни Кои (Ginny Caughey), американский MVP-специалист, который предоставил много полезной информации о текущих событиях и помощь в развитии нескольких тестов через публикации на твиттере. Вы найдете многие из ее предложений в приложении.
- Рене Шульте (Rene Schulte), также MVP Silverlight, с его глубоким знанием Silverlight может просто помочь при первых шагах в Silverlight.
- Лоран Бюньон (Laurent Bugnion), который разработал MVVM Light Toolkit и состояние запроса и ответа многократных касаний для проекта CodePlex Silverlight 4, WPF[1] 4 и Windows Phone 7.
- Тим Шмиц (Tim Schmitz), который оказывал содействие не только по элементам управления Bing maps, но и помог уменьшить нагрузку авторам, таким образом, чтобы они могли сосредоточиться на этой книге.
- Гордон Бройер (Gordon Breuer) является разработчиком для цифровых и печатных СМИ и разработчиком .NET для веб-приложений и Windows Phone. Он создал для нас модель 3D Windows Phone для угла Roll-Nick-Gier.
- Дирк Нордхузен (Dirk Nordhusen), DiNoGames, за техническую редакцию главы XNA.
- Харальд М. Генауцк (Harald M. Genauck), редактор, который исправил рукопись и лингвистические ошибки. Писатели всегда очень рады видеть его.

Наконец, мы благодарим всех наших родных и близких, без терпения которых, покорности, принятия и понимания, эта книга никогда не была бы закончена в срок. Это не всегда было легко, но всегда находилось решение.

Патрик Гецманн, Симон Хакфорт и Петер Новак
Ноябрь 2011 года

[1] *WPF* (Windows Presentation Foundation) — система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML. — *Ред.*

Глава 1

Вступление

Много нового

Редко можно встретить, чтобы новая версия продукта была разработана таким образом, что существенно отличалась бы от базового продукта, а не являлась бы только простым его обновлением. Обычно имеется достаточно причин для сохранения значительных моментов совместимости версий программных продуктов. Так ОС Windows 7 даже сегодня содержит некоторые элементы ОС Windows 3.11.

ОС Windows Phone 7, или проще говоря — Windows Phone, представляет собой мобильную операционную систему, разработанную Microsoft и основанную на Windows Embedded CE 6.0. Другими словами: это была версия "1.0".

В предыдущей ОС Windows Mobile фирма Microsoft достигла такого момента, когда уже стали необходимы радикальные изменения. Все началось уже с подготовки платформы. Если в случае с ОС Windows Mobile в равной степени было уделено внимание только бизнес-пользователям и корпоративным пользователям, то в случае с ОС Windows Phone в центре внимания уже находятся конечные пользователи. Это будет оставаться таковым в течение длительного времени, т. к. ОС Windows Mobile в настоящее время производится в коммерческих целях под флагом "Windows Embedded Handheld". Можно предположить, однако, что новая платформа в дальнейшем будет, в конечном счете, также хорошо ориентирована на частных пользователей.

Для такого предположения имеется несколько причин: бизнес-пользователи и корпоративные пользователи имеют разные требования, чем частные клиенты. Таким образом, на этих устройствах выполняются бизнес-приложения, а в Windows Phone, если, это вообще возможно, они должны еще как-то адаптироваться. Такое положение дел привело бы к дополнительным затратам, которых деловые клиенты, естественно, хотели бы избежать, поскольку они уже вложили средства в текущую версию программного обеспечения. Таким образом, ОС Windows Embedded Handheld можно понимать как средство совместимости.

Такая двойственность ОС Windows Mobile и ОС Windows Embedded Handheld предоставляет возможность избежать прежних недостатков. Но зачем это было нужно делать?

Если посмотреть на рынок мобильных устройств, то видны два основных конкурента. С одной стороны, смартфон iPhone от компании Apple, с другой стороны различные устройства на базе свободной ОС Android. При этом продавцы придерживаются разных философий. В то время как iPhone OS является достаточно закрытой экосистемой, Android предоставляет лишь только базовую операционную систему, которую можно свободно адаптировать практически к любому аппаратному средству — для этого исходный код системы является открытым. Ориентация четко определена: ОС iPhone предназначена для частных клиентов, а Android как для частных, так и бизнес-клиентов.

С этой точки зрения Windows Mobile можно отнести к Android-категории. Как базовая операционная система Android, так и ОС Windows Mobile предоставляются производителям аппаратных средств, которые адаптируют их к своим устройствам. Поскольку при этом отсутствуют требования к аппаратным средствам, программному обеспечению и внешним устройствам, то это приводит к тому, что имеются разнообразные устройства с самыми разными форм-факторами. Производитель аппаратных средств, однако, всегда имеет тенденцию к экономии при затратах на производство. Таким образом, для увеличения продаж устройств иногда экономически целесообразно применение небольших процессоров и маломощных аккумуляторов.

Рассмотрение ОС Windows Mobile на протяжении прошедших лет приводит к выводу, что на рынок выходили устройства, для которых не была определена конкретная операционная система и поэтому зачастую представлялись пользователю как "медленные".

Но это еще не все: при отсутствии требований к программному обеспечению производителям устройств приходилось создавать собственные интерфейсы для своих устройств. Это не означает, что эти интерфейсы были плохими. Но иногда вы не можете определить, какая операционная система изготовлена заводом. Таким образом, устройство, работающее с ОС Windows Mobile, одного производителя могло выглядеть совсем другим, чем устройство другого. В дополнение к этому для собственного интерфейса даже модифицировались базовые библиотеки, так что разработчик мучился иногда с излишними ошибками в программе.

Кроме того, некоторыми производителями были предприняты попытки адаптации устройств для конкретных операторов сети мобильной связи. Таким образом устройства, поставляемые группе компаний T-Mobile (Германия), иногда явно отличались от вариантов для компании Vodafone (Великобритания). Помимо этого, с предоставлением бесплатного программного обеспечения устройства должен был подниматься его престиж, что, однако, могло отражаться на стабильности устройств и приводить к поломкам.

Другая проблема связана с обновлениями программного обеспечения. Здесь очень легко попасть в зависимость от производителя (и часто также от продавца мобильной связи). Модернизированная версия программного обеспечения для того же самого устройства от компании Vodafone необязательно функционирует на устройстве компании E-Plus, и наоборот. Иногда возникает вопрос к производителю оборудования, почему он предлагает дорогостоящие обновления для него, в то время как он мог бы продать одно новое устройство с обновленной операционной системой.

На iPhone такие проблемы отсутствуют. Имеется только одно оборудование, на котором выполняется определенная операционная система. Один и только один производитель оборудования может обеспечить непосредственно свое обновление операционной системы. Так что не удивительно, что iPhone с первых дней и также еще сегодня может эксплуатироваться с текущей iOS-версией. Также проблемы эффективности функционирования большей частью чужды iPhone, т. к. все приложения, которые предоставляются в программе iTunes

для iPhone, подвергаются ранее процессу сертификации. Этот метод гарантирует, что никакое приложение не может доставить проблемы iPhone.

С этой точки зрения ОС Windows Phone можно отнести к категории iPhone, с той разницей, что в данном случае имеется больше одного производителя аппаратных средств с такой же самой операционной системой. Тем не менее, ОС Windows Mobile также представляет закрытую экосистему. Однако чтобы были гарантии согласованности при "стыковке" операционной системы и аппаратного средства, для устройств имеются строгие условия, которые должен выполнять производитель для возможности применения ОС Windows Phone в качестве операционной системы на его устройствах.

ОС Windows Phone 7.5 один год даже не была представлена на рынке после публикации бесплатной ОС Windows Phone модернизированной версии. Версия 7.0 имела несколько сложностей для запуска, которые являются типичными для новых платформ — получилось не то, что предполагали. Первая большая модернизированная версия в марте или апреле потерпела частично крах, так что даже сегодня на некоторые устройства провайдера все еще не было установлено обновление под названием NoDo. Кроме того, веб-сайт App Hub, через который подаются новые приложения для сертификации, в начале имел множество "детских болезней". Тем не менее, в итоге все проблемы решались в кратчайшие сроки.

ОС Windows Phone приносит много улучшений, которые теперь позволяют воспринимать платформу в качестве версии "1.0". Раньше отсутствовали еще несколько функций, а имеющиеся не были упомянуты полностью. Вспомним хотя бы только о недостающих функциональных возможностях регулировки собственных звуков звонка.

Устранение недостатков

Без желания предвосхитить дальнейшее содержание книги, здесь предлагается наглядно показать большие технологические различия между "ранним" Windows Mobile и "сегодняшним" Windows Phone. Сначала необходимо понять, почему в случае с ОС Windows Phone одно или другое решение было принято так, а не иначе. Почему Windows Phone является тем, что есть, а не просто Windows Mobile 7 с теми же "недостатками", как его предшественник.

Мультипрограммный режим против управляемой системы

ОС Windows Phone является, в первую очередь, системой для частных пользователей. Они часто ожидают совсем другие смартфоны, чем бизнес-пользователи.

Частные пользователи хотят развлекаться. Для этого приложения также должны выглядеть прекрасными. В том числе эта "красота" основывается на анимациях. При переключении, например, от одной страницы к другой, плавный, анимированный переход кажется гораздо интереснее, чем внезапная подача новой страницы с полной яркостью. Также навигация через элементы в окне списка кажется просто естественнее, если элементы "передвигаются" посредством анимации.

Эта "естественность" берется из реального мира. Например, дверь не открывается внезапно. Открытие двери воспринимается чем-то похожим на "анимацию".

Зачем, однако, нужно делать анимацию закрытой системой? Однако запускаются ли Silverlight- или WPF-приложения на открытых системах, например, таких как Windows 7?

Теперь просто находится в распоряжении большее количество ресурсов (ОЗУ, более эффективный процессор, лучшие графические процессоры и т. д.). Запросто теперь может выполняться одновременно большее количество задач. Все же и устройство, работающее под Windows 7, может иметь свои ограничения: если приложения запускаются в фоне, которые расходуют очень большое процессорное время, анимации могут выглядеть "прерывистыми".

Во всяком случае этот эффект воспринимается пользователем как неестественный. Он чувствует себя расстроенным и просто не использует больше такое "хромающее" приложение или клеймит устройство как слишком маломощное.

Напротив, операционная система, которая вмешивается в исполнение приложений, может определять сама, должно ли и когда приложение останавливаться или оканчиваться, если не находится в распоряжении достаточное количество ресурсов. Именно поэтому охотно используются закрытые системы.

Silverlight против Windows Forms

Приложения, созданные на платформе *Windows Forms*, полюбившиеся многим .NET-разработчикам, между тем больше не возможны с ОС Windows Phone — на их место выдвигается теперь программная платформа Silverlight.

Хотя принципиально можно было бы позволять также выполнение Windows Forms-приложений под ОС Windows Phone, но компания Microsoft, очевидно, не хочет вкладывать дополнительные средства в эту технологию. Сегодня, в частности, потребительские приложения уже едва ли создаются при помощи Windows Forms (кроме как имеется хорошее основание для этого), поскольку для пользователя это достаточно трудно. В случае с Windows Forms речь идет именно о технологии, которая не позволяет создавать анимации.

Кажется, что время плоских серых интерфейсов с еще более темными мрачными кнопками уже прошло. Напротив, они еще достаточно часто применяются для деловых приложений, где требуются скорее определенные функциональные возможности, чем визуальная элегантность.

Приложения Silverlight и WPF основываются на иерархически разделенной системе элементов управления. Если ранее элементы управления имели только "плоский" внешний вид, то эта иерархическая система привносит, в какой-то мере, трехмерный компонент: элементы управления могут размещаться внутри других элементов управления, т. к. почти каждый элемент управления в то же время может снова действовать в качестве контейнера для других элементов управления.

Для создания гибкого и настраиваемого стиля элементов управления в Windows Forms возникли бы существенные затраты на разработку. В случае с Silverlight эта гибкость и адаптируемость были учтены уже при разработке. Виды кнопок могут изменяться теперь в основном через шаблоны, независимо от того, сколько экземпляров кнопок используется для интерфейса (рис. 1.1).

Отображение информации на экране дисплея в Windows Forms происходит в противоположность Silverlight только на основе пикселов, т. е. в виде растрового изображения. Напротив, платформа Silverlight рассчитана на векторное представление информации. Преимущество очевидно: интерфейсы приложений и их элементы можно масштабировать на любые габариты без потери при этом качества картинки. Поэтому разработчику также может безразлично, какое разрешение дисплея используется для приложения, поскольку "пиксельное" представление информации отсутствует.

Эволюционная модель платформы Silverlight напоминает больше об эволюционной модели сетевого приложения, нежели Forms-приложения. С одной стороны, это связано с разделением в процессе создания интерфейсов, так называемой разметкой (Markup), и с другой стороны, соответствующим исходным кодом (Code-Behind). Значительно проще и гибче создавать описания интерфейса на языке *XAML* (eXtensible Application Markup Language — расширяемый язык разметки приложений), чем посредством чистого исходного кода. Дизайнер сможет позаботиться о разработке интерфейсов, а разработчик одновременно о функциональных возможностях, без внесения путаницы в планы друг друга.

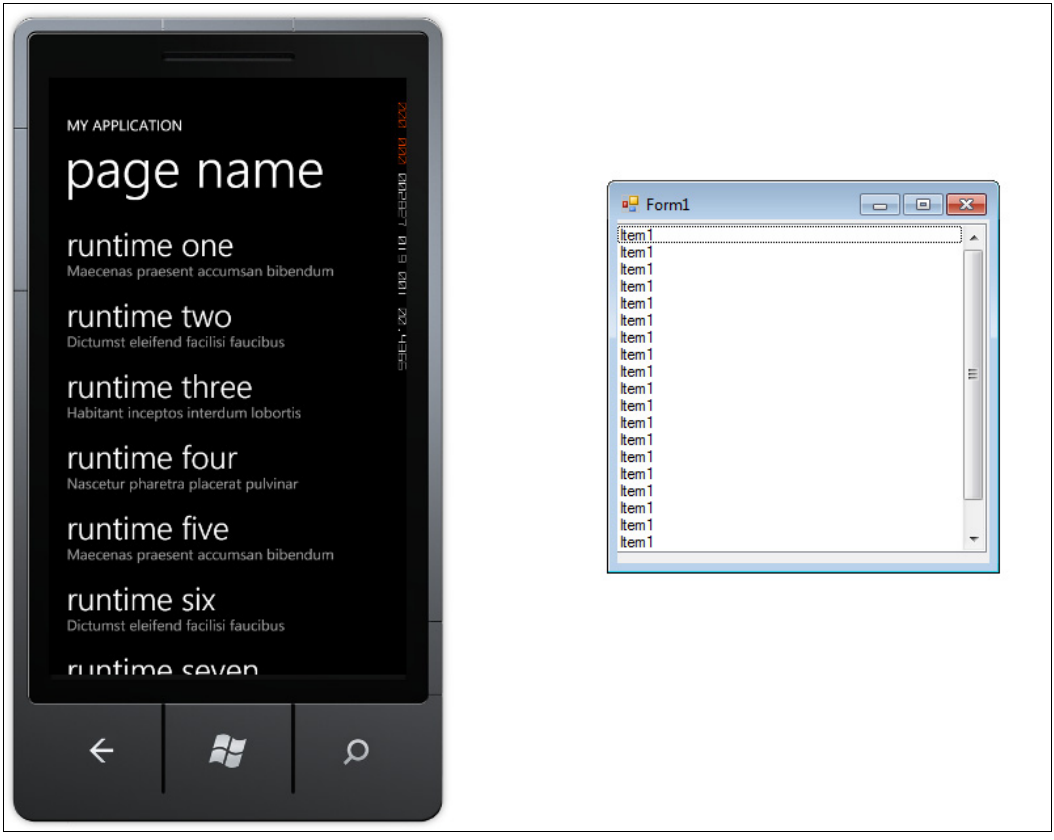

Рис. 1.1. Элемент управления окна списка в приложениях Silverlight и Windows Forms

Почему для приложения Windows Phone выбрана, тем не менее, программная платформа Silverlight как база, а не подобная система Windows Presentation Foundation (WPF)? Все очень просто: платформа Silverlight представляет только часть WPF. Она была предназначена для запуска веб-браузера в изолированной программной среде. Поэтому для проектирования закрытой системы платформа Silverlight является идеальным выбором. Преимущество ограничения части WPF в Silverlight зависит также и от того, что настройка может производиться значительно проще в условиях маломощных аппаратных средств смартфонов по сравнению с обыкновенными аппаратными средствами ПК. Начиная с версии системы Silverlight 3, в нее непосредственно встроено ускорение программного обеспечения, так что уже только поэтому требуется меньшее количество необходимых настроек.

Тем не менее, ОС Windows Phone базируются на версии Silverlight 4.

Управляемый код против неуправляемого кода

Почему больше нет возможности для создания приложения на языке C++? С ОС Windows Embedded CE происходит так же, как с Silverlight? Вроде бы, с технической точки зрения, ничего не мешает.

Язык C++, тем не менее, не является совсем простым и предназначенным для всех. Дело в том, что неопытные разработчики могут наделать много ошибок, которые, в свою очередь, могут привести не только к пресловутой потере памяти, но даже к неисправному состоянию всей операционной системы. Забытый деструктор[1] в языке C++ может привести к тому, что имеющаяся в распоряжении основная память будет уменьшаться и, наконец, устройство надо будет перезапустить, поскольку в распоряжении больше вообще не останется памяти. Устройства с ОС Windows Mobile часто страдают от этого.

Управляемый код (Managed Code) должен использоваться существенно проще. Таким образом, как принято издавна в случае с Silverlight, будет ограничено непосредственное взаимодействие с системой благодаря промежуточной прослойке IL[2]-кода. Следовательно, таким образом осуществляется более эффективный контроль за доступом к файловой системе, основной памяти или функциям устройств. Для хранения данных также вполне достаточно *изолированного хранилища* (Isolated Storage).

В то время как большинство приложений на языке C++ могут непосредственно обращаться к библиотекам операционной системы и поэтому могут нанести ей вред, то с помощью управляемого кода, пожалуй, можно легко препятствовать этому. Защитный уровень, состоящий из атрибутов, предотвращает вызов критического кода — и поэтому вы будете чувствовать себя в безопасности.

Кроме того, уже предоставляется одна очень обширная база для .NET-разработчиков, которые используют преимущественно языки C# и Visual Basic .NET. Для осуществления перехода к этой новой платформе как можно более простым, управляемый код является идеальным.

Почему XNA?

Так как в случае с ОС Windows Phone речь идет о платформе для пользователей, то соответственно игры также являются важной темой для рассмотрения. Что можно предложить более очевидное для ОС Windows Phone, чем платформу XNA Framework, если сама система уже основана на управляемом коде?

Наконец, платформа XNA не является чем-то новым. Уже достаточно давно на базе платформы XNA имеются игры, предназначенные для ПК, Xbox 360 и Zune[3]. В связи с этим платформа XNA для Windows Phone существенно облегчает жизнь разработчикам игр.

Таким образом с ОС Windows Phone 7.5 в проекте можно смешивать платформы Silverlight и XNA, которые привносят свои некоторые преимущества. Более подробная информация представлена в *главе 5*.

---

[1] *Деструктор* (от англ. destructor) — специальный метод класса, служащий для деинициализации объекта (например, освобождения памяти). — *Ред.*

[2] *IL* (сокращение от *MSIL* — Microsoft Intermediate Language) — это промежуточный язык, разработанный корпорацией Microsoft для платформы .NET Framework. IL-код не зависит от платформы, на которой выполняется приложение, т. е. компилятор .NET генерирует IL-код в независимости от того, какой язык программирования использовался для написания кода. — *Ред.*

[3] *Zune* — портативный медиаплеер, соответствующее программное обеспечение и интернет-сервис от компании Microsoft. — *Ред.*

Открытая система против закрытой системы

Так в чем же смысл закрытия системы? ОС Windows Mobile была уже открытой системой? Не признается ли таким образом пользователь совершенно неспособным?

Открытость системы является важным пунктом для бизнес-пользователей и продвинутых "опытных пользователей". Они точно знают, как работает система, и как они могут себе помочь при возникающих проблемах. А как же с простым пользователем?

Ему важно, чтобы система была надежно защищена от вредоносного ПО. Он вовсе не хочет иметь на устройстве только что-то вроде сканера вирусов. Ему также безразлично, что происходит под пользовательской оболочкой. Важно только то, что его игры и аппарат работают и что он больше не должен заботиться ни о чем. Все должно выполняться просто и быть стабильным.

Каким образом загружаются его песни или фотографии во внутреннюю память, ему абсолютно безразлично. Он не нуждается в файловой системе. Он хочет получать свои данные только в предложенный музыкальный плеер или средство просмотра фотографий. То, что он при пересылке своей фотографии по электронной почте должен был бы добавлять в письмо файл, например foto_23.jpg и именно из папки \Мои документы\Фото, для него совершенно не имеет значения. Для него важно, чтобы он мог только видеть изображение и мог прибавлять его непосредственно в электронное письмо, независимо от того, где оно действительно находится физически. Замкнутая система, или, правильнее говоря, "закрытая" система, его полностью удовлетворяет.

Что требуется уже знать, перед тем как начать?

Поскольку эта книга не озаглавлена как "Разработка приложений для ОС Windows Phone за 21 день" и не является самоучителем, то она, безусловно, предполагает следующие базовые знания:

- знакомство с операционными системами Microsoft;
- знакомство с устройствами мобильной связи/смартфонами;
- общие знания в разработке .NET-приложений;
- опыт работы с языком программирования C#;
- опыт работы с технологией Silverlight;
- опыт работы с LINQ[1] (Language Integrated Query).

Кроме того, будут очень полезны следующие знания:

- знания в разработке приложений .NET Compact Framework[2];
- работа с эмулятором Windows Mobile;
- обмен данными посредством веб-сервисов или WCF[1]-сервисов.

---

[1] *LINQ* (Language Integrated Query) — проект Microsoft по добавлению синтаксиса языка запросов, напоминающего язык структурированных запросов SQL (Structured Query Language), в языки программирования платформы .NET Framework. — *Ред.*

[2] *.NET Compact Framework* — это аппаратно-независимая среда, версия .NET Framework, которая разработана корпорацией Microsoft для запуска приложений на устройствах, основанных на платформе Windows CE, таких как карманные компьютеры, мобильные телефоны, заводские контроллеры и др. — *Ред.*

К этим пунктам имеется рекомендованная литература и материалы, которые перечисляются в *приложении*. В отдельных главах этой книги имеются более конкретные дополнительные рекомендации.

В книге используется единственный язык программирования C#. Это не связано с тем, что авторы поленились дополнительно создавать и описывать примеры, например, на Visual Basic .NET. Скорее на переднем плане было наполнение имеющихся в распоряжение книжных страниц концептуальной информацией, чем заниматься разбором различного синтаксиса разных языков.

В этой книге делается упор на англоязычные версии сред разработки и дополнительных программ. Этому имеется две причины. Первая причина состоит в том, что в случае ошибки англоязычного приложения предоставляется точный дословный текст сообщения об ошибке, который достаточно просто можно найти в Интернете, поскольку англоязычная община разработчиков является наиболее распространенной.

Вторая причина состоит в том, что англоязычные оригиналы являются более ранними, чем локализованные версии, и поэтому для них снова имеется больше информации в Интернете. К тому же в локализованные версии иногда прокрадываются небольшие ошибки, которые отсутствуют в "оригинале". Так что пользы от английской версии будет в два раза больше.

Далее...

Эта книга содержит двенадцать глав, а также названное ранее приложение.

В этой главе вы уже кое-что узнали о ОС Windows Phone.

Глава 2 предлагает обзор и введение в средства разработки, которые доступны для ОС Windows Phone.

В *главе 3* осуществлен подробный обзор операционной системы Windows Phone и соответствующих аппаратных средств. Это содержание осознанно размещено в начале книги, чтобы для понимания следующих глав заложить фундамент знаний о системе.

Глава 4 наконец-то посвящается разработке вместе с программной платформой Silverlight для ОС Windows Phone. Начинается, как обычно, с простого примера "Hello World". Позже будут обсуждены более интересные темы, такие как MVVM[2]-шаблон и локализация. Тем не менее первоначальная тема — это Silverlight для ОС Windows Phone.

В *главе 5* подробно рассматривается платформа XNA Framework и пример с созданием игр. Вы познакомитесь здесь с несколькими базовыми функциями XNA Framework и, надо надеяться, сможете после прочтения главы и рассмотрения приложенного исходного кода написать собственные игры для Windows Phone.

Глава 6 посвящена специфическим для платформы особенностям, как ориентация, файловая система, включая сжатие данных, сенсорное управление, локализацию, задачи и другие темы, которые представляют интерес как для платформ XNA, так и Silverlight для ОС Windows Phone.

Темы сети будут затронуты в основном в *главе 7*. Там рассматривается обработка простого опроса сети через WCF-сервисы и Push-уведомления (Push Notifications). Так же регистрация сетевых данных при помощи программы *Fiddler* является отдельной темой.

---

[1] *WCF* (Windows Communication Foundation) — программный фреймворк (Framework), входящий в состав .NET Framework и используемый для обмена данными между приложениями. — *Ред.*

[2] Шаблон MVVM (Model-View-ViewModel) — применяется при проектировании архитектуры приложения. — *Ред.*

В *главе 8* предоставлена тема баз данных, которые поддерживаются Windows Phone 7.5 с Linq2SQL.

Как можно разумно протестировать приложения или же протоколировать их характеристики после установки, рассмотрено в *главе 9*.

Прежде чем дойдет дело до внедренных приложений, важно рассмотреть тему настройки производительности. Как это осуществляется при помощи ПО Visual Studio с пакетом Windows Phone SDK 7.1, показано в *главе 10*.

После этого тут же в *главе 11* представлено собственно внедрение. Что требуется для установки приложений, представленных на веб-сайте Marketplace? Какие там приняты правила? Какая задача имеется специально для Marketplace?

После рассмотрения всех циклов разработки приложений в предыдущих главах, в *главе 12* речь пойдет о миграции существующих приложений и игр. Хотя сведения в этой главе в основном теоретические, но они обеспечивают существенную помощь в миграции проектов.

И последнее, но не менее важное — это *приложение*. Здесь представлены полезные ресурсы, имеющиеся в Интернете, которые вы должны, по крайней мере, хотя бы один раз увидеть, в том числе ресурсы для дизайнеров. Как получить ссылку для скачивания обновленных глав и примеры для этой книги, вы узнаете тоже в этом *приложении*. Также здесь показывается, как можно "сделать эскиз" Windows Phone-приложения при помощи Sketch Flow в качестве опытного образца.

Ошибки в этой книге

Если вы найдете орфографическую ошибку или опечатку в этой книге, то считайте, что обнаружили одну из наших многочисленных спрятанных загадок в этой книге!

Конечно, это лишь шутка. Хотя эта книга была неоднократно прочитана авторами и посторонними лицами, а также в конце еще раз редакционной коллегией, а это не легко, все же не исключено, что одна или несколько ошибок ускользнули от обнаружения.

Если ошибка присутствует в основном тексте, то, скорее, это не так уж опасно. Тем не менее, мы всерьез воспринимаем ошибку в исходных текстах примеров. Если в них была обнаружена ошибка, не бойтесь сообщить нам об этом по электронной почте. После этого мы охотно опубликуем корректуру затронутого примера в нашем блоге на сайте **http://wp7buch.wordpress.com**.

Кроме того, если у вас имеется инициативное предложение, новости, критика, пожелания, то можете передать их нам так же по электронной почте.

Путаница с номерами версий

ОС Windows Phone 7.5 с кодовым названием Mango имела, к сожалению, два номера версий. С точки зрения маркетинга Windows Phone здесь обозначается номером версии "7.5", в то время как Windows Phone OS и Windows Phone SDK получили номер версии "7.1". Это снова отражается также при составлении нового проекта, где "7.1" находится в распоряжении как новая версия. Начальная версия Windows Phone, также как появившаяся тогда как первая модернизированная версия с условным обозначением "NoDo", имеет наименование все же под номером версии "7.0", для которой могут создаваться также с Windows Phone

SDK 7.1 еще новые проекты. Тем не менее, нужно отказываться от этого, т. к. приложения для "7.0" должны обновляться раньше или позже на "7.1". Также со временем простились с цифрой "7" в названии "Windows Phone 7": теперь платформа называется попросту Windows Phone.

Поскольку эта книга представляет собой пересмотренный вариант первого издания, то вы часто наткнетесь на название "Windows Phone 7". В этом случае Windows Phone подразумевается в качестве платформы, если не указано иное.

Кроме того, следует иметь в виду, что также еще было изменено название средства разработки. Windows Phone Developer Tools теперь называется Windows Phone SDK.

Глава 2

Привет, новый мир

Как уже было — это что-то новое. Кто пытается сравнивать Windows Phone с Windows Mobile, могут сравнивать Windows Phone также одинаково с Android или Symbian[1]. Они все не имеют почти ничего общего друг с другом, разве что в обоих случаях речь идет об операционных системах для мобильных устройств.

Таким образом, эта глава посвящена первым шагам в этой технологии и описывает предполагаемые шаги для разработки мобильных приложений.

Учимся работать

У того, кто хоть раз использовал Windows Phone и немного разбирается в технике, часто вскоре возникает желание самому разрабатывать приложения.

Никогда еще не было проще начинать разработку приложения, чем с ОС Windows Phone.

В этом разделе описываются необходимые первые шаги, чтобы начать разработку. Более подробно этот вопрос будет рассмотрен в последующих главах.

Для Windows Mobile предварительно в дополнение к платному ПО Visual Studio 2008 Professional требовалось установить необходимые средства разработки. Сегодня вы получите все из одних рук. Ключом к этому является установочный файл размером почти 3 Мбайта, который можно загрузить в Интернете по адресу **http://create.msdn.com/**.

ПРИМЕЧАНИЕ

На данный момент было сделано большое сокращение: для разработки приложений для ОС Windows Phone достаточно иметь только Visual Studio, начиная с версии 2010. Приложения для старых ОС Windows Mobile и Windows Embedded Handheld можно разрабатывать только под Visual Studio 2008, а приложения, разработанные под Visual Studio 2010, для этих операционных систем не годятся.

---

[1] Symbian — операционная система для сотовых телефонов, смартфонов и коммуникаторов, первая версия которой в 1999 г. была разработана консорциумом Symbian. — *Ред.*

При этой установке программы речь идет о так называемой программе самозагрузки (Bootstrapper). Она анализирует при выполнении уже существующие приложения, которые нужны для разработки, и инсталлирует недостающие части. Для этого отсутствующие пакеты программного обеспечения загружаются из Интернета. Приятно то, что все необходимые для разработки бесплатные приложения находятся в Express-версии. Если у вас нет устройства, то это не проблема, поскольку программа инсталляции, помимо всего прочего, установит очень мощный *эмулятор*.

Требования к системе, по крайней мере с точки зрения разработчика ориентированных систем, носят очень умеренный характер. Все, что вам нужно — это компьютер под управлением Windows Vista или Windows 7. А какая должна быть память и центральный процессор? На этот вопрос трудно ответить уже с давних пор, т. к. на рынке предлагались и будут предлагаться очень разные аппаратные средства.

Таким образом, вполне можно работать с двухъядерным процессором и, по крайней мере, с 2 Гбайтами оперативной памяти. Еще будет лучше, если вы сможете использовать не менее 4 Гбайт. В любом случае следует руководствоваться девизом: *чем лучше процессор и больше памяти, тем выше будет производительность*.

Кроме того, при разработке приложения с помощью эмулятора имеются особые требования к видеокарте (графическому адаптеру). Она должна поддерживать DirectX 10, а также драйвер WDDM 1.1.

После того как аппаратные проблемы будут решены и сделана установка, на вашем компьютере появятся все необходимые компоненты.

ПРИМЕЧАНИЕ

Тем не менее, могут возникнуть ситуации, в которых веб-установка будет невозможна, поскольку нет стабильного подключения к Интернету.

В этом случае, можно получить по адресу **http://download.microsoft.eom/download/6/E/7/6E795194-EE2D-4636-AEBD-D1C42D2E06E1/ WPSDKv71_en1.iso** английский пакет установки как DVD-образ в ISO-формате. Далее этот образ может быть либо записан на DVD-диск или открыт непосредственно при помощи соответствующей программы, так что при установке догрузка компонентов из Интернета вам не потребуется.

Microsoft Visual Studio 2010 Express для Windows Phone

При желании создавать приложения, Visual Studio является исходной точкой. Здесь можно выборочно создавать свое приложение на языке C# или Visual Basic .NET.

С языком программирования предоставляются сразу девять шаблонов проектов (рис. 2.1), которые помогают на первых шагах: общий шаблон для создания приложений Windows Phone, затем шаблон проекта для создания приложений, связанных с данными, а также шаблон для создания компонентов, которые должны быть использованы в других приложениях. Эти шаблоны предназначены для создания Pivot- или Panorama-приложений (подробнее см. *главу 4*).

Шаблон для создания роскошного графического приложения является шаблоном Silverlight для Windows Phone-приложений, который уже поддерживает отрисовку графики с помощью XNA.

При помощи *агента для проигрывания аудиофайлов* (Audio Playback Agent) получаются *фоновые агенты* (Background Agents), которые воспроизводят аудиоданные на заднем плане. При помощи *агента для проигрывания аудиопотока* (Audio Streaming Agent) можно

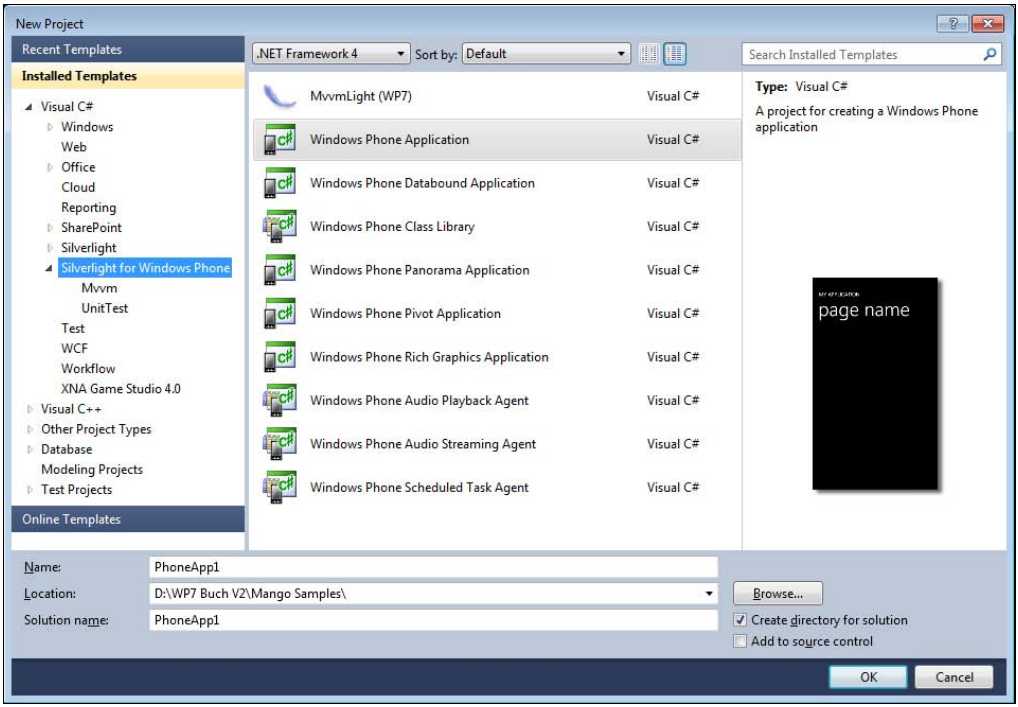

Рис. 2.1. Шаблоны проектов Silverlight для проектов Windows Phone

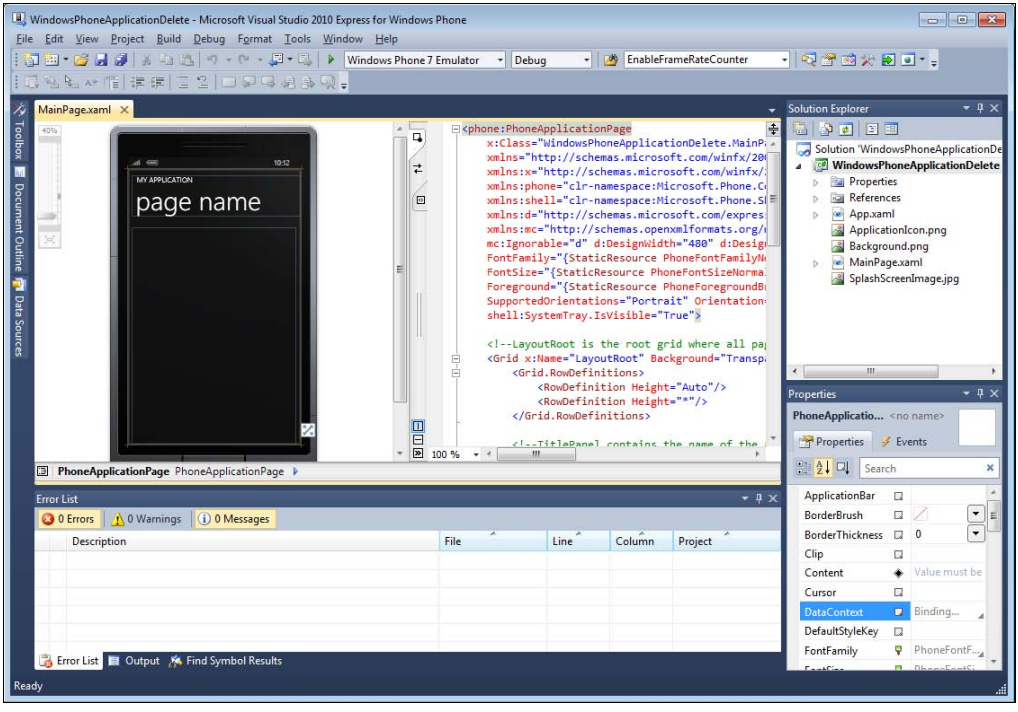

Рис 2.2. Программа Visual Studio в качестве среды разработки

создать агентов фона, которые воспроизводят поток звуковых данных в фоновом режиме. Через агента назначенного задания (Scheduled Task Agent) получаются агенты фона, которые могут выполнять ресурсоемкие или периодические задачи на заднем плане.

В качестве технологии здесь используется платформа Silverlight. Это должно быть уже известно из Интернета. Она является альтернативой программе Adobe Flash, которую многие знают, имея дело с игровыми веб-приложениями. Эта технология доказала свою ценность. Она гибка и усовершенствована, так что перенос программного обеспечения на ОС Windows Phone является только следующим логическим шагом.

То, что все шаблоны проектов Visual Studio собраны вместе, усиливает конструктор интерфейсов. При этом внешний вид интерфейса при его создании обычно отображается так, как он будет выглядеть позже на устройстве (рис. 2.2). Также в создании интерфейсов помогает язык дизайна Metro (Metro Design Language). Естественно, установки также изменяются, так что приложение не обязательно должно быть представлено в стиле Metro Design.

Дополнительная информация по теме "Metro Design" приведена в *разд. "Metro Design" главы 3*, а также в *разд. "Обзор Metro Design" главы 4*.

Microsoft Expression Blend для Windows Phone

Часто говорится, что разработчики не являются дизайнерами. Это не ложь, потому что в то время как разработчики зациклены больше на функциональных приложениях, дизайнеры имеют совершенно другие цели. Их интересует простое и захватывающее представление приложения. Здесь же шрифты, презентации, и, возможно, даже осмысленная анимация играют определенную роль, что является причиной того, что многие из них работают дизайнерами в рекламных агентствах, а не в фирме по разработке домов.

Так же как разработчики, дизайнеры — это определенный тип людей, что отражается и в приложениях. В то время как разработчик стремится скорее к тому, чтобы получить оптимально представленный исходный код и "создать дизайн" интерфейса скорее рудиментарно, то для дизайнера — это разработка от А до Я. Его не интересует, как правило, никакой исходный код, а только доступ к графическим элементам, палитре цветов и т. д.

Поэтому программа *Microsoft Expression Blend*[1], предназначенная больше для дизайнеров (рис. 2.3), и выглядит несколько по-другому. На первый взгляд, она напоминает скорее графическое приложение, такое, например, как Photoshop, а не среду разработки. На самом деле программа Expression Blend позволяет также создавать приложения для ОС Windows Phone на языке C# и Visual Basic .NET. Хотя может показаться странным, что следует выбирать язык программирования в приложении разработки, где должен создаваться "только" дизайн интерфейса приложения. Все же в этом имеется более глубокий смысл: программы Visual Studio и Expression Blend работают "рука об руку". Таким образом, исходный код для Silverlight генерируется программой Expression Blend в фоновом режиме, так что в принципе разработчики интерфейса и исходного кода могут работать над проектом одновременно. Expression Blend помимо самой разработки приложений предлагает еще несколько полезных функций для разработчика. Таким образом можно не только отображать и изменять исходный код приложения, но и запускать непосредственно разработанное приложение на эмуляторе или устройстве.

Программа Microsoft Expression Blend обеспечивает те же самые шаблоны проектов, как и Visual Studio для разработки и создания приложений.

---

[1] *Microsoft Expression Blend* — программа от корпорации Microsoft, предназначенная для разработки дизайна веб-интерфейсов и графических настольных приложений, которая совмещает в себе особенности этих двух типов приложений. — *Ред.*

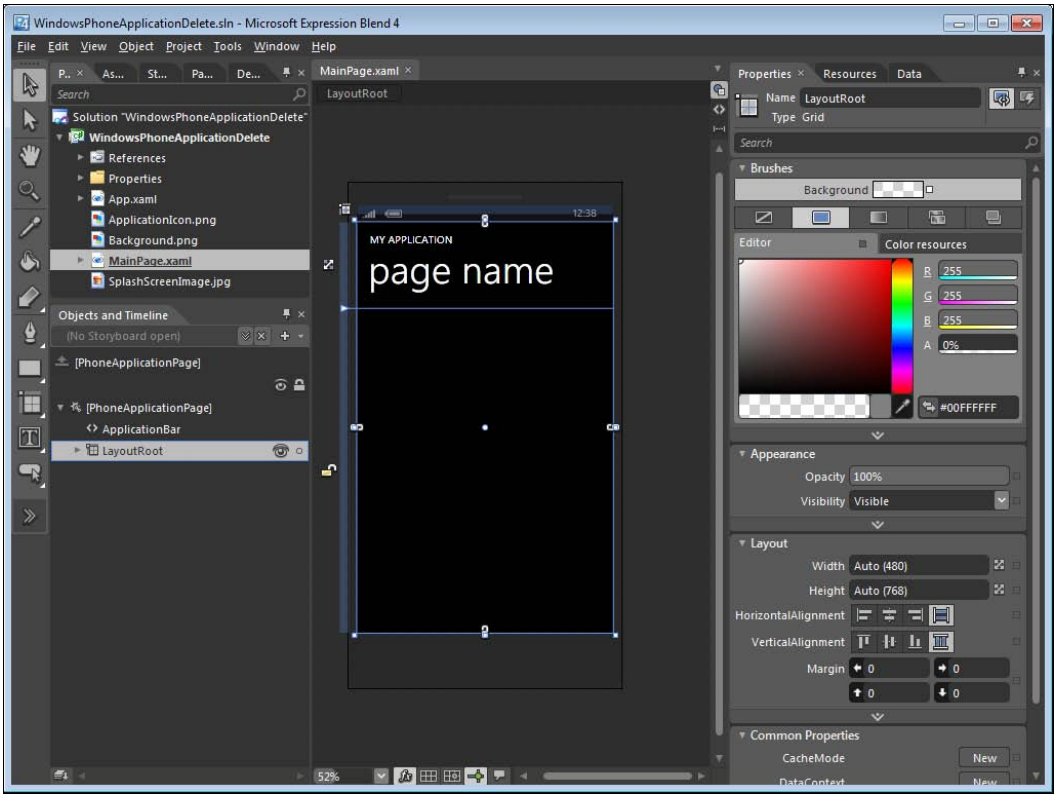

Рис. 2.3. Программа Microsoft Expression Blend является приложением для дизайнера

XNA Game Studio 4.0

Программа XNA Game Studio является *интегрированной средой разработки* (Integrated Development Environment — IDE) для разработки игр, которая явно отличается от создания Silverlight-приложений в Expression Blend и Visual Studio. В первую очередь она служит в качестве платформы для разработчиков игр, которые можно использовать для создания игр на основе XNA Framework[1].

XNA Framework, однако, преследует другой подход, чем Silverlight, хотя также поддерживает язык программирования C#. Silverlight форсирует подготовку, предоставляя разработчику в распоряжение функции, которые быстро приводят к полноценному приложению, будь то через вспомогательные классы или путем адаптации существующих элементов управления. XNA Framework — это скорее тематически ориентированный набор часто используемых функций. Здесь осуществлена поддержка разработчика для возможности сконцентрироваться на настоящем игровом плане, вместо того, чтобы заниматься утомительным рисованием содержимого. Это влечет за собой совсем другую модель программирования.

[1] *XNA Framework* — высокопроизводительный кроссплатформенный интерфейс прикладного программирования (Application Programming Interface — API) для разработки графических приложений для платформ Windows и Xbox 360. — *Ред.*

Кроме того, набор инструментов XNA Framework, далеко не нов. Он существует уже относительно давно для Xbox 360, ПК и Zune, что влечет за собой преимущество в возможности действительно простого переноса игры на этой базе между платформами. Так как Windows Phone поддерживает, как известно, также и игры, перенос программного обеспечения платформы XNA был, естественно, следующим логическим шагом.

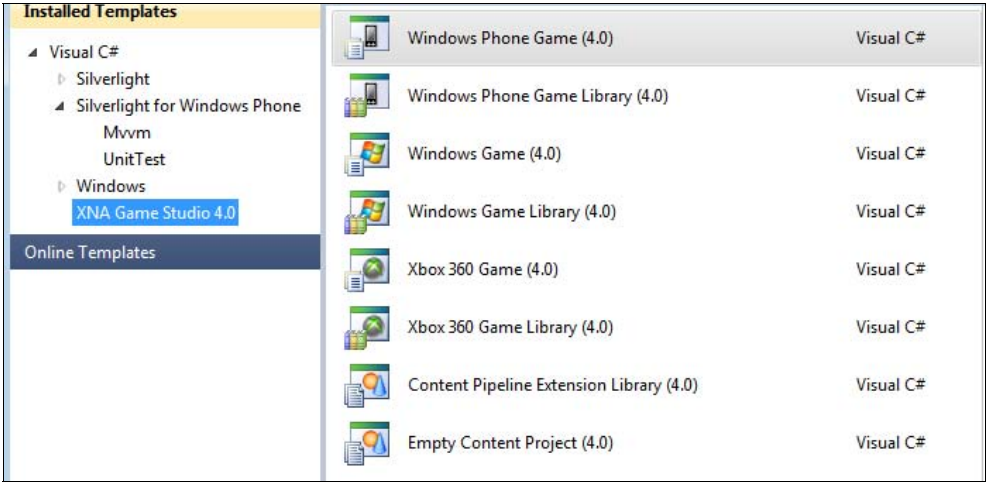

Рис. 2.4. Шаблоны проектов для XNA-проектов

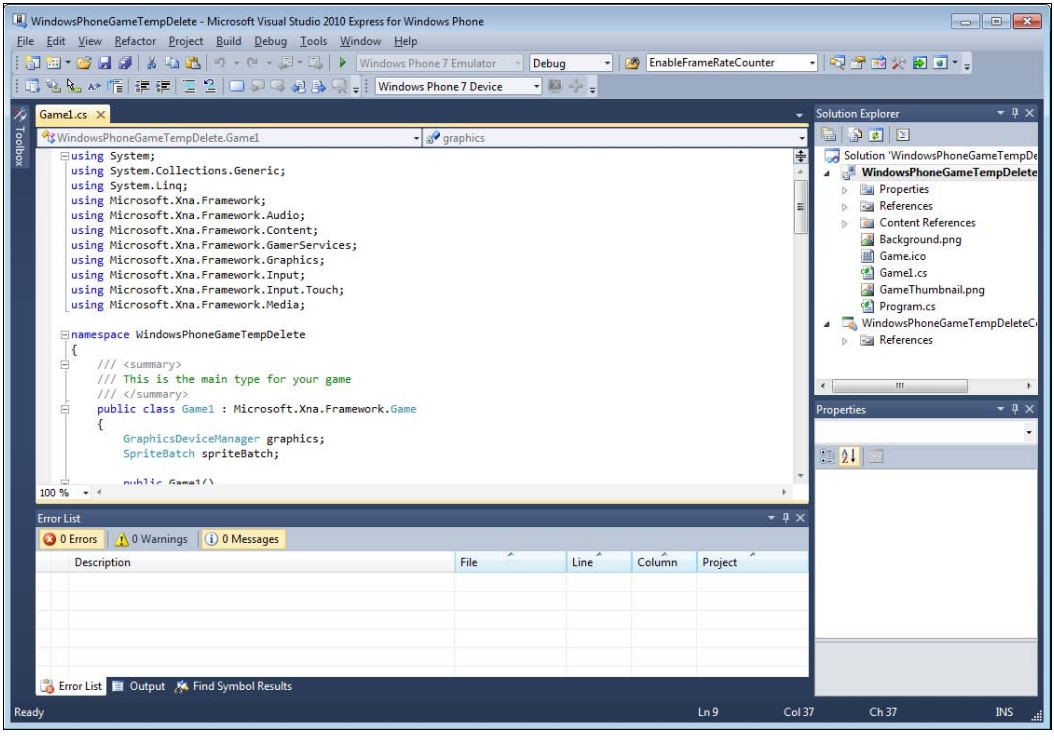

Рис. 2.5. Так выглядит новый XNA-проект

Также когда XNA Game Studio может появляться как собственное приложение (рис. 2.4), то на самом деле это специализированная версия Visual Studio (рис. 2.5), которая наряду с несколькими инструментами включает в себя еще коллекцию проектных шаблонов.

Эмулятор

Эмулятор можно запускать из всех ранее упомянутых приложений. Хотя возможен отдельный запуск эмулятора из рабочего стола с несколькими трюками.

Если создается новый проект в одном из приложений и запускается отладка посредством нажатия клавиши <F5>, то запускается эмулятор и созданное приложение. Тем не менее, также можно "свободно" передвигаться внутри эмулятора. После нажатия кнопки **Start** (Пуск) откроется стартовый экран (рабочий стол), который очень нагляден. На нем наверняка будет виден графический значок — плитка (tile) — приложения Internet Explorer (рис. 2.6). Он ведет себя так же, как и обычный компьютерный Internet Explorer, даже на мобильном устройстве. Если компьютер, на котором работает эмулятор, подключен к Интернету, то он также используется в эмуляторе. В противном случае нет никаких приложений, установленных за исключением тех, которые были предоставлены самой программой Visual Studio для устройства.

Рис. 2.6. Эмулятор выглядит спартанским, но все равно очень полезен

То, что в первый момент может показаться скорее разочаровывающим, тем не менее, почти всегда достаточно для разработки. Все функции, в которых нуждается приложение, предоставляются в распоряжение эмулятором. Также могут непосредственно использоваться акселерометр (accelerometer) и GPS-навигатор, поскольку они тоже эмулируются Windows Phone 7.1 SDK (рис. 2.7). Однако об этом более подробно будет рассказано далее в соответствующих главах.

СОВЕТ

Отсутствие поддержки прямого ввода с клавиатуры в эмуляторе очень тревожит. Вместо этого всегда предлагается только виртуальная клавиатура эмулятора, где ввод текста должен происходить посредством мыши. Далеко не всем известно, что имеется возможность переключения. Имеются следующие кнопки:

- Bild ⇧ (Break) — активизирует аппаратную клавиатуру компьютера;
- Bild ⇩ (Break) — активизирует виртуальную клавиатуру эмулятора;
- Pause — переключает между этими обоими режимами ввода туда-сюда.

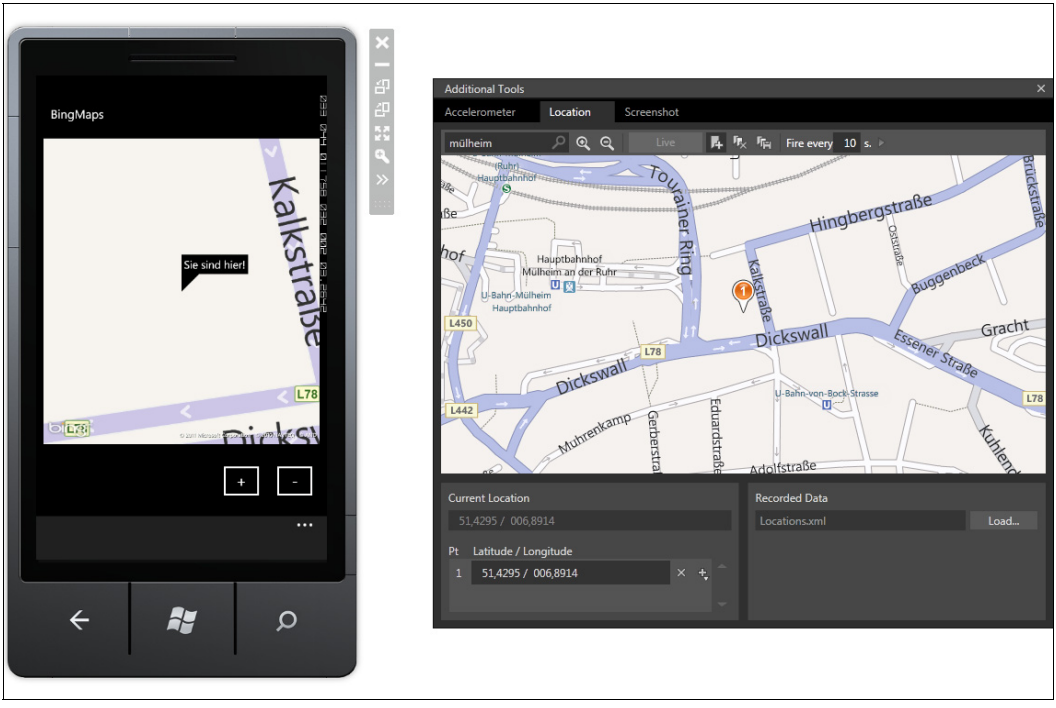

Рис. 2.7. При помощи Bing Maps также возможна эмуляция GPS-данных

... и что еще?

Наряду с представленными приложениями на компьютере после установки комплекта разработчика находятся еще два дополнительных вспомогательных приложения.

Приложение *Windows Phone Developer Registration* (рис. 2.8) позволяет осуществлять регистрацию стандартных устройств с ОС Windows Phone на сайте Marketplace для использования чужого программного обеспечения — подробнее об этом см. в *главе 11*.

Первоначально для уже разблокированных устройств Windows Phone упоминается приложение *Application Deployment*. Оно позволяет просто устанавливать инсталляционные XAP-файлы на устройстве или эмуляторе (рис. 2.9).

Рис. 2.8. Приложение для активации стандартных устройств с ОС Windows Phone в качестве разработчика устройств

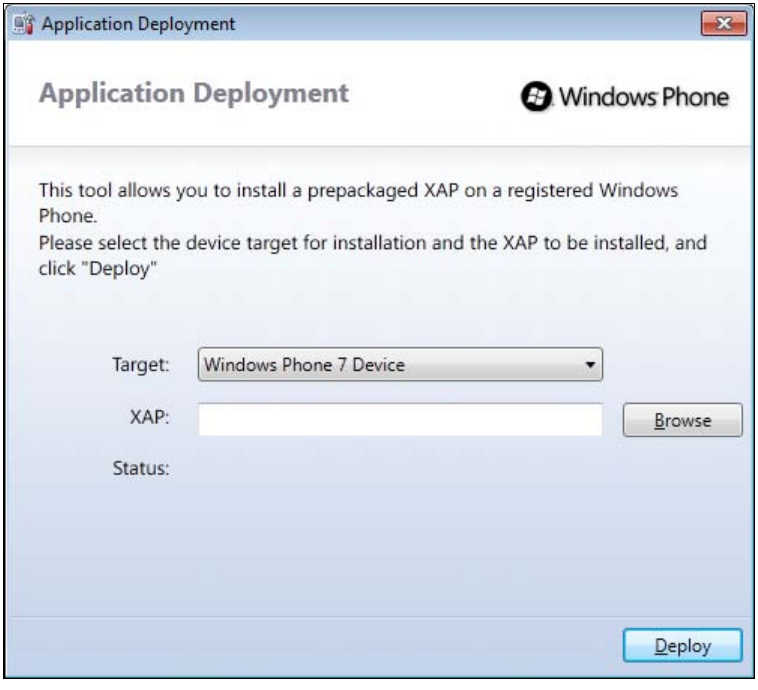

Рис. 2.9. Приложение для внедрения XAP-установок

Полнофункциональные и Express-версии

Что происходит, если комплект Visual Studio не достаточен в Express-редакции и требуется "полноценная" программа Visual Studio, например, для интеграции с Team Foundation Server? Как это сделать, если необходимо иметь только шаблоны проектов для Windows Phone-приложений в уже установленной программе Expression Blend?

Нет ничего проще, т. к. есть только одно правило, которое необходимо соблюдать. Все разработчики приложений, которые должны получать шаблоны, должны быть уже инсталлированы перед установкой вышеназванного пакета установки Express-редакции. Вот и все.

Программа установки автоматически определяет, инсталлирован ли другой вариант Visual Studio или Expression Blend, и добавляет во время установки к уже имеющимся приложениям необходимые шаблоны проектов.

Только нужно принимать во внимание то, чтобы соответственно та же самая версия языка была инсталлирована. Англоязычная Express-редакция в немецкоязычный версии Visual Studio не инсталлируется.

Глава 3

Из чего состоит Windows Phone

Новая платформа аппаратных средств

Одна из инноваций, которую привносит Windows Phone, — это спецификация аппаратных средств для производителей, желающих применять Windows Phone на своих устройствах.

Разработчику может быть практически безразлично, на каких аппаратных средствах выполняется его приложение, до тех пор, пока они отвечают соответствующим требованиям. Но именно в этот момент в прошлом часто возникали проблемы. В принципе, производители устройств были совершенно свободны при выборе аппаратных средств.

На начальном этапе появления карманных ПК разработчику нужно было работать с несколькими архитектурами процессоров, таких как MIPS, ARM, SH2/4 или *x*86. Как правило, код приложения не зависел от конкретной архитектуры процессора. Тем не менее имелся недостаток в том, что для каждого типа процессора был необходим собственный формат выходного файла, который нужно было предоставлять в соответствии с этим отдельно в специальных файлах CAB[1] или в иной форме. Эта проблема вскоре разрешилась, по меньшей мере для Pocket PC и, соответственно, для разработчика Windows Mobile, т. к. ОС Windows Mobile 2003 работала только в рамках архитектуры ARM.

Для этого устройства привлекалось все больше дополнительных компонентов аппаратных средств. Bluetooth и беспроводная ЛВС (*WLAN* — Wireless Local Area Network) были двумя типичными представителями таких компонентов. Для Bluetooth имелись в распоряжении различные интерфейсы, которые были не совместимы, к сожалению, друг с другом. Беспроводная ЛВС также была очень проблематична, поскольку в это время еще каждый производитель "варился в собственном котле" и стандарт NDIS[2] для беспроводной ЛВС был больше

---

[1] *CAB* (от англ. Cabinet) — это расширение установочного файла, упакованного в архив. Этот файл содержит программы и инструкции для операционной системы и подготовлен для использования на КПК. — *Ред.*

[2] *NDIS* (Network Driver Interface Specification) — сетевой стандарт, спецификация стандартного интерфейса сетевых адаптеров, разработанная компанией Microsoft для того, чтобы сделать коммуникационные протоколы независимыми от сетевого оборудования. — *Ред.*

подобен опциональному руководству, и не было концепции, на которую разработчик мог бы полагаться. Хотя это и были разработки в смежных областях, и только немногие разработчики могли взаимодействовать с таким многообразием аппаратных средств, но для большинства разработчиков это создавало достаточное количество проблем.

Кроме того, стало проявляться принципиальное отличие между устройствами с сенсорным экраном и устройствами без сенсорного экрана, управление которых имеет довольно существенное различие. Помимо этого появилось огромное множество устройств с различными форм-факторами и разрешениями экрана, поэтому они должны были быть учтены при разработке приложения, если конечно оно не предназначалось для конкретного устройства.

Со временем появлялось все больше более новых компонентов аппаратных средств, таких как встроенные GPS или датчики, которые могли оценивать движения или освещенность. Если GPS еще относительно просто можно было поддерживать с внедрением GPS-API (за исключением нескольких "своеобразных" реализаций драйвера от нескольких производителей), то с датчиками было уже несколько сложнее. Отсутствовали унифицированные интерфейсы прикладного программирования (*API* — Application Programming Interface). Различные производители чипов или различные версии ПЗУ к тому же самому чипу показывали разные характеристики. Кроме того, не все производители устройств всегда были информированы и в большинстве случаев напрасно искали официальные инструменты разработки ПО (*SDK* — Software Development Kit) к таким системным компонентам. Нужно было использовать так называемую утечку информации, ошибочную публикацию документов или пользоваться непосредственно декодированием. Если можно было использовать такие функции устройств, то они могли оказаться не совсем качественными, и не было гарантии, что программное обеспечение будет работать при следующем обновлении ПЗУ. Без точного знания последствий вызовов функций можно было легко повредить устройство. Кроме того, такой "сторонний" код затруднял ремонтопригодность и техническое обслуживание проекта. Представьте только себе код, который должен был опрашивать три устройства датчиков движения, к которым нужно было обращаться в разных версиях ПЗУ также по-разному. И если рассматривать связанные с этим затраты на тестирование и подготовку при поддержке таких смешанных аппаратных средств, то они были связаны непосредственно с ними.

Для преодоления этих недостатков и упрощения жизни разработчикам, со стороны компании Microsoft для производителей устройств на ОС Windows Phone были введены минимальные требования к аппаратным средствам.

Благодаря этим требованиям к аппаратному обеспечению теперь имеется надежная база, которую разработчик может использовать в своем приложении. Все же наличие этих базовых аппаратных средств — это еще не все, поскольку спецификация также предусматривает, чтобы аппаратные средства придерживались определенных руководств и располагали, прежде всего, также унифицированными интерфейсами, которые предоставляются разработчику приложения для Windows Phone посредством API. Такая общая база чрезвычайно облегчает реализацию специфических для аппаратных средств функций и, в конечном счете, помогает пользователю приложения.

В следующих разделах дан обзор минимальных требований к аппаратным средствам.

Дисплей и разрешение

В противоположность используемым до сих пор в среде Windows Mobile резистивным сенсорным экранам, которые реагируют на нажатие, устройства, предназначенные для ОС

Windows Phone, поставляются в основном только с емкостными сенсорными экранами. Емкостные сенсорные экраны реагируют не на нажатие, а на изменения электрического поля. Если палец касается дисплея, то изменяется статическое электрическое поле. Явное преимущество такого управления — это лучшее удобство управления пальцем, а также простая поддержка мультисенсорного, или иначе мультитач-управления[1]. Ввод данных при помощи обычных прикосновений больше не возможен. Разумеется, стали доступны специальные способы ввода для емкостных дисплеев. Мультисенсорный ввод может применяться, например, для изменения масштаба изображения: при увеличении расстояния между точками касания происходит увеличение изображения. Любая ОС Windows Phone поддерживает *мультисенсорное управление* и предлагает минимум четыре точки касания. Теперь при Windows Phone разрешение дисплея составляет 800×480 пикселов.

Процессор и графика

ОС Windows Phone работает на современных процессорах ARMv7 с частотой минимум 1 ГГц. Графические процессоры должны быть совместимы с DirectX 9 и поддерживать аппаратное ускорение для DirectX и Direct3D. Вследствие этого предоставляется высокая производительность для игр XNA и для сложного графического приложения Silverlight.

Память

Приложения со сложной графикой и игры расходуют большой объем памяти. Windows Phone может принимать большое количество электронных писем и, посредством интеграции Zune, большое количество файлов видео и аудио.

Для возможности сохранения таких объемов данных даже при минимальной конфигурации аппаратных средств к ним предъявляются относительно высокие требования. Оперативной памяти должно быть как минимум 256 Мбайт. Объем памяти должен быть определен на уровне не менее 8 Гбайт. Поскольку не предусмотрены взаимозаменяемые платы памяти, то перед приобретением аппаратных средств нужно определиться с необходимым для личных нужд объемом памяти.

Клавиатура и органы управления

Для многих пользователей аппаратная клавиатура является важным параметром при покупке. Некоторым пользователям, однако, устройство с клавиатурой кажется слишком тяжелым и громоздким. Другие пользователи наоборот не хотят отказываться от клавиатуры при написании электронных писем или SMS. Поэтому спецификация аппаратных средств Windows Phone предусматривает клавиатуру только как опциональный компонент.

Тем не менее, такая аппаратная клавиатура подчиняется также нескольким ограничениям. Ее назначением является только лишь ввод текста. Использование клавиатуры для управления устройством, например, внутри приложений, не предусмотрено.

Чтобы обеспечить бесперебойное обслуживание различных устройств, спецификация аппаратных средств предусматривает ряд кнопок, которые можно найти на любом устройстве Windows Phone.

---

[1] *Мультитач* (от англ. Multi-Touch — множественное прикосновение) — функция сенсорных систем ввода, осуществляющая одновременное определение координат двух и более точек касания. — *Ред.*

Кнопка питания

Кнопка питания служит для включения и выключения устройства и, соответственно, также дисплея. В зависимости от состояния устройства, а также частично от длительности нажатия на кнопку может запускаться и другая функция.

Кнопки для регулирования громкости

Кнопки громкости принципиально служат для регулирования громкости устройств в масштабах системы. Кроме того, они также активны, если проигрывается фоновая музыка и экран заблокирован. При поступлении входящего звонка телефон можно временно выключить, используя кнопки регулировки громкости мелодии звонка.

Кнопка камеры

Кнопка камеры используется для запуска приложения камеры. Кнопка может быть нажата полностью или только наполовину. В то время как при полном нажатии кнопки запускается съемка, то при "половинном" нажатии может активироваться или деактивироваться автоматическая фокусировка.

Кнопку камеры можно использовать в режиме ожидания или режиме блокировки, что позволяет пользователю делать снимки в любое время.

Кнопка *Back*

Кнопка ← **Back** (Назад) (возврат к предыдущему экрану) используется для навигации в пределах приложения и между различными приложениями. Событие нажатия на кнопку может перехватываться в собственном приложении, чтобы осуществлять переходы между страницами или выполнять закрытие меню.

Конкретная реализация этой кнопки ни в коем случае не может определять фактический тип навигации телефона или переопределять ее.

Кнопка *Start*

Благодаря кнопке **Start** (Пуск) пользователь может перейти из любого приложения к начальному экрану устройства — на рабочий стол.

Кнопка *Search*

Благодаря кнопке **Search** (Поиск) пользователь может запускать Bing-поиск в Интернете из любого приложения, несмотря на некоторые исключения. В приложении Zune запускается поиск Zune Marketplace, в области контактов — контакты, а также в области электронной почты можно просматривать электронные письма.

Аудио

Несмотря на то что Windows Phone — это телефон, вместе с тем во многих приложениях используется аудио. Но вывод звука важен не только для получения информации о событиях, таких как входящие телефонные звонки, встречи, но и для других событий — запуска приложений и срабатывания кнопок.

Также важен ввод аудиоданных. С одной стороны, это необходимо, само собой разумеется, для телефонного разговора. С другой стороны, через речевой ввод также предлагаются со-

вершенно новые функции управления приложениями. Таким образом появляется возможность управления приложениями полностью через голосовые команды. Также могут использоваться определенные шумы, которые возникают, например, когда дуют в микрофон.

Оба имеющихся в наличии аудиоэлемента устройства (динамик и микрофон) могут быть совершенно интересны для вас, как разработчика приложения. Поэтому оба интерфейса, как для микрофона, так и для динамика, доступны через API.

Дополнительный аудиоинтерфейс — это разъем для головной гарнитуры или телефонов (наушников). Собственно, наушники служат как дополнительный звуковой выход для возможности наслаждаться, например, музыкой или игрой, не создавая неудобств окружающим.

Разумеется, особую роль также играют головные телефоны, когда устройство используется для приема радио FM. Кабель наушников применяется в этом случае в качестве антенны для получения радиосигнала. Динамик может активироваться дополнительно для выхода аудиосигнала, однако, наушники требуются обязательно для получения сигнала от передатчика.

Датчики

Современное оборудование, особенно для устройств связи, таких как мобильные телефоны, в настоящее время уже не может обойтись без датчиков, по крайней мере, если оно должно соответствовать ожиданиям пользователей. Датчики представляют собой электронные комплектующие, которые могут измерять определенные физические свойства. Благодаря таким датчикам устройство может реагировать, например, на движение и повороты и регулировать яркость экрана или определять местоположение устройства.

Далее приведен обзор датчиков, которые должны присутствовать в устройстве Windows Phone.

Акселерометр

Собственно акселерометр (датчик ускорения) на самом деле измеряет только чистое ускорение и может использоваться как преобразователь перемещения. Поскольку информация о чистом ускорении в принципе относительно бесполезна, то часто в современных мобильных телефонах используются совершенные датчики гравитации, сокращенно G-датчики. Они измеряют не только ускорение, но и перемещение устройства под воздействием силы тяжести.

Это может быть использовано, например, для автоматического переключения между вертикальным и горизонтальным форматом отображения на экране, а также для управления изображением во время игры, благодаря изменению положения устройства.

Разработчик может получить доступ к этому датчику через API акселерометра.

Гироскоп

Гироскоп может присутствовать в устройствах Windows Phone 7.5 опционально. *Гироскоп* — это гироскопический прибор, с помощью которого можно достичь более точного определения положения устройства, чем при акселерометре. Для разработчика под Windows Phone 7.5 поставляется API гироскопа.