



# ОГЛАВЛЕНИЕ

<b>ПРЕДИСЛОВИЕ .....</b>	<b>11</b>
<b>ВСТУПЛЕНИЕ .....</b>	<b>13</b>
<b>БЛАГОДАРНОСТИ .....</b>	<b>15</b>
<b>ОБ АВТОРЕ .....</b>	<b>16</b>
<b>ГЛАВА 1. Введение в Windows Phone .....</b>	<b>17</b>
Новый вид телефона .....	17
Интеграция с телефоном.....	21
Технические характеристики телефона .....	22
Варианты взаимодействия.....	24
Дизайн с учетом сенсорного ввода .....	25
Аппаратные кнопки .....	26
Клавиатуры.....	26
Датчики .....	28
Жизненный цикл приложения.....	28
Разработка с применением служб.....	29
Живые иконки .....	30
Marketplace .....	32
Распространение приложений через Marketplace .....	32
Отправка приложения в Marketplace .....	33
Правила для приложений.....	35
Правила для содержимого.....	37
Где мы оказались?.....	38
<b>ГЛАВА 2. Создание первого приложения для телефона ...</b>	<b>40</b>
Подготовка компьютера .....	40
Создание проекта .....	42
Visual Studio .....	42
XAML .....	45
Конструирование интерфейса в Blend .....	49
Добавление кода.....	56
Работа с событиями.....	58
Отладка в эмуляторе.....	58
Отладка в устройстве.....	59
Касания .....	62
Работа с телефоном.....	65
Где мы оказались?.....	68

<b>ГЛАВА 3. Обзор XAML .....</b>	<b>69</b>
Что такое XAML? .....	69
Свойства объектов XAML .....	70
Пространства имен XAML .....	72
Именованние в XAML .....	73
Визуальные контейнеры .....	74
Основные визуальные элементы .....	78
Фигуры .....	78
Кисти .....	79
Цвета .....	80
Текст .....	81
Изображения .....	82
Преобразования и анимации .....	84
Преобразования .....	84
Анимация .....	86
Стилизация в XAML .....	88
Ресурсы .....	89
Стили .....	91
Где мы оказались? .....	94
<b>ГЛАВА 4. Элементы управления .....</b>	<b>95</b>
Элементы управления в Silverlight .....	95
Простые элементы управления .....	97
Однодетные элементы управления .....	101
Списковые элементы управления .....	102
Элементы управления, предназначенные специально для телефона .....	103
Элемент управления Panorama .....	103
Элемент управления Pivot .....	106
Привязка к данным .....	107
Простая привязка к данным .....	108
Использование DataTemplate .....	110
Повышение производительности прокрутки .....	111
Форматирование во время связывания .....	113
Связывание элементов .....	114
Конвертеры .....	114
Ошибки привязки к данным .....	116
Шаблоны элементов управления .....	118
Набор инструментов Silverlight for Windows Phone Toolkit .....	122
Элемент управления autoCompleteBox .....	122
Элемент управления ContextMenu .....	124
Элементы управления DatePicker и TimePicker .....	125
Элемент управления ListPicker .....	126
Элемент управления LongListSelector .....	128
Элемент управления PerformanceProgressBar .....	132
Элемент управления ToggleSwitch .....	132
Элемент управления ExpanderView .....	133
Элемент управления PhoneTextBox .....	134
Контейнер WrapPanel .....	136

Где мы оказались? .....	137
<b>ГЛАВА 5. Дизайн приложений для телефона.....</b>	<b>138</b>
Третий экран .....	138
Это ведь телефон, не так ли? .....	141
Определение типа приложения .....	142
Панорама.....	144
Элемент Pivot.....	145
Простые страницы .....	147
Microsoft Expression Blend .....	147
Создание проекта .....	147
Знакомство с Blend.....	148
Основы Blend .....	155
Макетирование.....	155
Кисти .....	160
Создание анимаций.....	164
Поведения .....	167
Дизайн для телефона .....	169
Панель ApplicationBar в Blend .....	170
Элемент управления Panorama в Blend .....	172
Элемент управления Pivot в Blend .....	174
Предварительный просмотр приложения.....	176
Где мы оказались? .....	177
<b>ГЛАВА 6. Разработка для телефона.....</b>	<b>178</b>
Жизненный цикл приложения.....	178
Навигация.....	180
Захоронение.....	184
Учет особенностей телефона .....	189
Ориентация .....	189
Дизайн с учетом сенсорного ввода .....	191
Клиентская область приложения.....	199
Панель приложения .....	200
Обнаружение бездействия.....	202
Эффект смещения .....	203
Где мы оказались? .....	204
<b>ГЛАВА 7. Интеграция с телефоном.....</b>	<b>205</b>
Вибрация .....	205
Движение .....	206
Имитация движения.....	209
Звук .....	211
Воспроизведение звука с помощью MediaElement.....	211
Использование библиотек XNA .....	212
Воспроизведение звука с помощью XNA .....	213
Управление воспроизведением .....	214
Запись звука .....	215
Контакты и встречи .....	218
Контакты.....	218
Встречи .....	223

Сигналы и напоминания .....	224
Создание сигнала .....	225
Создание напоминания .....	227
Доступ к существующим оповещениям .....	227
Задачи .....	228
Пускатели .....	230
Селекторы .....	238
Разделы Media и Picture .....	246
Доступ к музыке .....	246
Воспроизведение музыки .....	248
Доступ к фотографиям .....	250
Сохранение фотографий .....	252
Интеграция с витриной Pictures .....	253
Интеграция с витриной Music+Videos .....	255
Работа с камерой .....	259
Использование класса PhotoCamera .....	259
Прямой доступ к оборудованию .....	263
API буфера обмена .....	265
Живые иконки .....	266
Главная живая иконка .....	267
Дополнительные иконки .....	268
Двусторонние живые иконки .....	270
API определения местоположения .....	271
Разрешение позиционирования .....	271
Доступ к информации о местоположении .....	272
Эмуляция информации о местоположении .....	277
Где мы оказались? .....	280
<b>ГЛАВА 8. Базы данных и иные системы хранения .....</b>	<b>281</b>
Хранение данных .....	281
Изолированное хранилище .....	282
Сериализация .....	284
Локальные базы данных .....	289
Приступая к работе .....	289
Оптимизация класса контекста .....	295
Ассоциации .....	298
Использование существующей базы данных .....	304
Изменение схемы .....	305
Безопасность базы данных .....	307
Где мы оказались? .....	308
<b>ГЛАВА 9. Многозадачность .....</b>	<b>309</b>
Многозадачность .....	309
Фоновые агенты .....	310
Периодические агенты .....	312
Ресурсоемкие агенты .....	319
Агенты аудио .....	321
Фоновая служба передачи данных .....	330
Требования и ограничения .....	330

Запрос передачи данных .....	331
Мониторинг запросов .....	333
Где мы оказались? .....	338
<b>ГЛАВА 10. Службы .....</b>	<b>339</b>
Сетевые средства .....	339
Класс WebClient .....	339
Доступ к информации о сети .....	342
Работа с JSON .....	345
JSON-сериализация .....	346
Разбор JSON.....	348
Веб-службы.....	351
Потребление служб по протоколу OData.....	355
Как работает протокол OData.....	356
URI-адрес .....	357
Использование протокола OData в приложениях для телефона.....	364
Генерация ссылки на службу для OData-канала .....	364
Запрос данных.....	365
Обновление данных .....	367
Использование push-уведомлений.....	369
Требования к push-уведомлениям .....	370
Подготовка приложения к получению push-уведомлений .....	370
Настройка сервера для отправки push-уведомлений .....	372
Простые уведомления .....	375
Отправка уведомлений-тостов.....	383
Создание живых иконок .....	386
Обработка ошибок при доставке push-уведомлений .....	390
Где мы оказались? .....	392
<b>ГЛАВА 11. Магазин Marketplace .....</b>	<b>393</b>
Что такое Marketplace? .....	393
Как это работает .....	394
Взимание платы за приложения.....	397
Получение оплаты.....	399
Подача заявки на размещение приложения.....	401
Подготовка приложения.....	401
Процедура подачи заявки .....	406
После подачи заявки.....	411
Модификация приложения.....	414
Что делать после отклонения заявки? .....	415
Реклама в приложениях.....	417
Где мы оказались? .....	418
<b>ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ .....</b>	<b>419</b>



## ПРЕДИСЛОВИЕ

Когда Шон попросил меня написать предисловие к книге о разработке для Windows Phone, я испытал разные чувства. Во-первых, мне пришла в голову мысль, что раз уж меня попросили что-то написать, значит, выскребли все сусеки и больше никого не нашлось. Ведь людей, которые действительно способствуют выводу продукта на рынок, так много, а по заслугам им никогда не воздают. И хотя я польщен просьбой, причина которой – моя публичная роль в команде, настоящие герои – инженеры, спроектировавшие и реализовавшие это чудо. Сам продукт замечателен, но платформа, которая позволяет создавать потрясающие приложения с интерфейсом Metro и игры, – целиком заслуга разработчиков. Я пишу, чтобы воздать им должное.

Потом я подумал о том гигантском вкладе, который Шон внес в экосистему Microsoft. Шон является обладателем восьми номинаций MVP и звания Silverlight Insider, его ценят не только за содержательные идеи, но и за умение донести их до широкой аудитории. Когда Шон выступает, можете быть уверены, что он говорит для разработчиков, – он разрабатывает разработчиков. Без таких людей, как Шон, корпорации Microsoft было бы очень трудно (а то и вовсе невозможно) построить ту экосистему разработки, которая была создана за последние тридцать лет. Я пишу, чтобы воздать ему должное.

И, наконец, я испытал панику. Раньше мне не доводилось писать предисловия к книгам, поэтому я не совсем понимал, что говорить. Полагаю, что если вы решите купить эту книгу, то сделаете это по собственной воле, а не под влиянием того, что я собираюсь сказать. Но если вы читаете это предисловие, желая увериться в том, что позиции Windows Phone прочны, то я готов это подтвердить. Я пишу, чтобы воздать должное вам.

Выпуск первой версии Windows Phone и последовавшая за ним кооперация с Nokia знаменует инвестиции Microsoft в создание третьей экосистемы, ориентированной на разработчиков ПО для мобильных устройств. Среда, в которой строятся приложения для Windows Phone, не похожа ни на какую другую платформу и дает программисту

возможность создавать блистательные приложения, уделяя больше внимания удобству работы пользователя, а не борьбе с внутренними механизмами запутанной инфраструктуры. Интерфейс Metro оживает на экране, а с помощью технологии живых иконок (Live Tiles) вы сможете писать поистине захватывающие приложения.

Windows Phone 7.5 – это обновленная версия (кодовое название – «Mango»), девизом которой является «Put people first» (Люди – в первую очередь). Точно так же мы относимся к платформе разработки. На первое место мы ставим разработчиков. Книга, которую вы держите в руках, может стать первым шагом на пути к созданию приложений для Windows Phone. И, быть может, немного освежит ваши знания. Как бы то ни было, можно точно сказать, что под руководством Шона вы почувствуете, какие грандиозные перспективы открываются для желающих писать программы для мобильных устройств на платформе Windows Phone, и убедитесь, что Microsoft в этом случае действительно думала в первую очередь о разработчиках. Важен каждый разработчик. Каждый. Все. Без исключения.

*Брэндон Уотсон*  
Корпорация Microsoft



## ВСТУПЛЕНИЕ

У меня никогда не было PalmPilot. Но были КПК и смартфоны. Я писал программы для самых разных устройств, но особенно не углублялся. Проблемой для меня было отсутствие какой-либо системы в способах загрузки программ в телефоны, а, кроме того, я не видел, как с помощью маркетинговых уловок можно превратить приложение для телефона в успешный продукт. В последние годы я увлекся разработкой для Silverlight и веб. И не уделял особого внимания революции на рынке смартфонов. Я с радостью занимался связыванием данных, разработкой бизнес-приложений и преподаванием XAML.

Нет сомнений, что начало революции положило появление iPhone. И что мне в iPhone кажется особенно интересным, так это магазин App Store, а не сам смартфон. Это, конечно, замечательное устройство, но именно App Store все изменил, предоставив простой способ публиковать приложения для этих карманных «мастеров на все руки», продвигать их на рынок и зарабатывать на них деньги. Разумеется, Apple вовсе не рассчитывала на это. Выводя на рынок первую версию iPhone, Apple недвусмысленно заявила, что средой разработки будет Safari (созданный ей веб-браузер). Но под давлением сообщества разработчиков для OS X Apple уступила и как-то случайно совершила революцию в области распространения приложений.

Поняв, что упустил нечто важное, я снова начал внимательно присматриваться к разработке программ для телефонов. В то время у меня был Android, с него я и начал. Освоиться с языком Java и средой Eclipse оказалось не слишком сложно, но создание ПО для телефона все же оставалось кропотливой работой. Просто сами средства разработки были не такими удобными, как привычные мне Visual Studio и Blend. Примерно в то же время я приобрел Mac и попробовал написать что-нибудь для iPhone с помощью Objective-C и Xcode. Из этой переделки я вышел израненным и перебинтованным. Я хотел писать приложения, но, поскольку это было лишь побочным занятием, то у меня не хватило сил бороться с капризами инструментария для Android и iPhone, и я решил отложить это до лучших времен.



Вскоре после моих экспериментов с iPhone и Android корпорация Microsoft сорвала покровы с новой платформы: Windows Phone 7. Лично для меня настоящим открытием стала среда разработки. Я преподавал и писал о Silverlight еще с тех пор, когда эта технология называлась WPF/E, поэтому шанс совместить интерес к разработке мобильных приложений со знаниями в области Silverlight показался мне идеальным решением.

Мне доставляла истинное удовольствие возможность применить к телефону опыт разработки приложений для ПК и веб на основе Silverlight. С помощью Visual Studio и Blend я мог создавать красивые пользовательские интерфейсы и быстро переходить от прототипа к готовому приложению, а технология работы с применением этих инструментов и XAML показалась мне гораздо проще, чем на других платформах.

Пока я учился, Microsoft продолжала совершенствовать платформу и сначала анонсировала, а затем и выпустила Windows Phone 7.5 (кодовое название – Mango). Передо мной встал вопрос: то ли закончить книгу о Windows Phone 7, то ли попытаться отразить в ней все новые возможности, появившиеся в Windows Phone 7.5. Как я решил эту дилемму, вы, наверное, догадываетесь.

Не так-то просто было правильно выбрать сюжет этой книги, чтобы она была полезна как начинающим, так и опытным разработчикам на Silverlight. Я всегда стремлюсь к тому, чтобы читатель мог приступить к созданию приложений как можно быстрее, но при этом включаю сведения, позволяющие писать по-настоящему хорошие программы. Миникомпьютеры, которые мы носим в карманах, сравнительно невелики по размеру, поэтому чтобы приложение получилось хорошим, нужно знать, когда остановиться. Как вы поймете, прочитав эту книгу, моя цель – научить вас писать хорошие, а не всеобъемлющие приложения. Это означает, что я буду придерживать вас, не давая слишком увлекаться, но при этом подвергну испытанию ваши предположения о том, как строится процесс написания программ для телефона.



# БЛАГОДАРНОСТИ

Написание книги – коллективный труд. Всякий, кто думает, что для создания книги нужно расположиться в темной комнате и плести словесное кружево, которое словно по волшебству обернется денежными реками, текущими с Amazon, наверняка не бывал в горячем цеху, где пишут книги. Я, быть может, неплохо укладываю слова на виртуальную бумагу, но во всех остальных частях процесс полный профан. Необходим опытный редактор, который знает, когда поощрить, а когда и прищипить. Нужны технические рецензенты, которые не боятся гладить против шерсти. Нужно производственников, которые умеют превратить созданную в Visio бестолковщину, которую вы называли рисунками, в нечто, понятное читателю. Наконец, не обойтись без целой армии людей, готовых слушать ваши сетования о том, что невозможно написать толковую книгу, основываясь на бета-версии продукта... и не устают отвечать на постоянные докучные вопросы. Поэтому я хочу поблагодарить всех, кто внес реальный вклад (не придерживаясь какого-то определенного порядка).

Прежде всего, я выражаю признательность своему редактору из издательства Addison-Wesley, Джоан Мэррей (Joan Murray). С таким автором, как я, работать непросто, но она неизменно заставляла меня выдерживать сроки и побуждала принимать правильные решения, не обязательно самые легкие. Все прочие сотрудники издательства Addison-Wesley, с которыми я имел удовольствие работать, – также замечательные профессионалы. Особо хочу отметить Кристофера Кливленда (Christopher Cleveland), который принял обязанности научно-производственного редактора в середине работы над книгой, и блестяще проявил себя в ходе всего процесса.

Я хочу поблагодарить длинную череду участников списков рассылки Silverlight Insiders и Windows Phone 7 Advisors за терпеливое отношение к моим бесконечным вопросам и высокопарным речам. Все вы помогли формированию этой книги, пусть даже неосознанно. На всем протяжении работы читатели моего блога и мои последователи в сетях Facebook и Twitter были референтной группой. Опросы и

открытые вопросы, которые я задавал, помогли решить, что включать в книгу, а что опустить. За это я в долгу перед вами.

Хочу также поблагодарить своих блистательных рецензентов. Джереми Лайкнесс (Jeremy Likeness), Амброз Литтл (Ambrose Little) и Брюс Литтл (Bruce Little) не только помогли найти кучу мест, где я написал откровенную чушь, но и возвращали на путь истинный, когда я сбивался с него и не мог найти важнейший кусочек головоломки. Особое спасибо Амброзу за неуступчивость в вопросах дизайна. Он не позволял мне прививать разработчикам дурные привычки, из-за которых пользователям было бы неудобно работать с программой.

Всем, кого я забыл упомянуть, приношу свои извинения.

*Шон Уилдермут*

Ноябрь 2011

<http://wilderdmuth.com>

@shawnwilderdmuth

## ОБ АВТОРЕ

За двадцать пять лет работы в области разработки ПО **Шон Уилдермут** занимался самыми разными вещами. Этот многообразный опыт помог ему сформировать свое понимание технологии. Шон – девятикратный обладатель звания Microsoft MVP, член INETA Speaker's Bureau и автор нескольких книг по платформе .NET. Он также привлечен к работе в корпорации Microsoft в качестве Silverlight Insider и Data Insider. Шон выступал на многих международных конференциях, включая TechEd, MIX, VSLive, OreDev, SDC, WinDev, DevTeach, DevConnections и DevReach. Он автор десятков статей в различных журналах и на веб-сайтах, в том числе MSDN, DevSource, InformIT, *CoDe Magazine*, ServerSide.NET и MSDN Online. В настоящее время он проводит семинары в США, которые организует его компания AgiliTrain (<http://agilitrain.com>).



# ГЛАВА 1.

## Введение в Windows Phone

Для одних сотовый телефон – досадная необходимость, для других – устройство, без которого жизнь невысказима. Возможность звонить из любого места полностью изменила способы общения. За последние несколько лет технология телефонов сделала гигантский шаг вперед. С появлением iPhone и Android рынок постоянно подключенных устройств, которые могут выходить в Интернет, исполнять приложения и звонить, трансформировал взаимоотношения человека с телефоном. Поднялась также планка для устройств потребительского уровня. Современный потребитель ожидает, что телефон можно будет использовать как GPS-приемник, игровое устройство и планшет с выходом в Интернет. Для некоторых людей телефон стал основным средством подключения к Интернету, заменив настольный компьютер и ноутбук. Перед нами, как разработчиками, стоит задача создания наиболее удобного для пользователей интерфейса. Windows Phone предоставляет платформу, а Silverlight является механизмом для претворения наших идей в жизнь.

### Новый вид телефона

Когда Microsoft впервые объявила о Windows Phone 7, многие скептики думали, что это будет просто попытка догнать Apple и Google. Однако у Microsoft были другие планы. Новая операционная система для телефона оказалась совершенно непохожа на продукты других производителей мобильных операционных систем (прежде всего, Apple, Research in Motion и Google). Вместо того чтобы копировать структуру экрана со значками приложений, которая так полюбилась авторам iPhone и Android, Microsoft пошла другим путем. Дизайн приложений и операционной системы описывается на новом языке, получившем кодовое название Metro<sup>1</sup>. На этом *языке дизайна* опреде-

<sup>1</sup> UI Design and Interaction Guide for Windows Phone: <http://shawnw.me/wpmetroguide>

ляются рекомендации и стили для создания приложений для Windows Phone. Дизайн начального экрана, описанный на языке Metro, похож на дизайн других смартфонов в том смысле, что содержит ряд значков. Но вместо того чтобы разносить значки по страницам, Windows Phone позволяет пользователю перелистывать значки. Windows Phone отличается от других смартфонов еще и тем, что каждый значок может включать сведения о приложении. Эти значки называются **живыми иконками (Live Tile)** (см. рис. 1.1).

### Что такое язык дизайна?

Программисты привыкли рассматривать язык как последовательность текстовых выражений, описывающих машинные операции. Для дизайнера же язык – это набор правил, описывающих внешний облик некоторого множества приложений (или, как в данном случае, всей операционной системы). В википедии он определяется более общо: «... всеобъемлющая схема или стиль, определяющий дизайн набора продуктов или архитектурных параметров».

Начальный экран должен быть тем местом, где пользователь может быстро оценить состояние телефона. Живые иконки предоставляют сведения о количестве пропущенных вызовов, непрочитанных SMS и писем, а также информацию от сторонних поставщиков, например, о погоде. При разработке собственных приложений вы можете либо создать простой значок для размещения на начальном экране, либо живую иконку.

Что касается приложений, то экран Windows Phone разбит на три области, в которых осуществляется взаимодействие пользователя с телефоном: системная область (system tray), клиентская область и панель приложения (рис. 1.2).

Системной областью управляет операционная система телефона. Здесь отображается время, информация об уровне сигнала и уведомления.



Рис. 1.1. Начальный экран Windows Phone

Как правило, приложения оставляют эту область видимой. Некоторые программы (например, игры) могут ее скрывать, но поступать так следует лишь в случае, когда без этого приложение, на ваш взгляд, много потеряет.

Ваше приложение будет занимать клиентскую область. В ней отображается пользовательский интерфейс, данные и все точки взаимодействия.

На панели приложения показываются параметры вашего приложения. Использовать ее необязательно, но это общепринятая практика, поскольку таким образом пользователь может получить доступ к настройкам и меню приложения. Например, на рис. 1.3 изображено простое приложение для ввода заметок, в котором панель позволяет создать новую заметку или показать меню (обратите внимание, что по значку многоточия можно щелкнуть – откроется список пунктов меню).

У многих приложений, встроенных в Windows Phone, имеется общая отличительная особенность – использование **витрины** (hub). Основная идея витрины – предоставить отправную точку для удовлетворения естественного любопытства пользователя, желающего знать, что умеет делать приложение. Обычно витрина предстает в форме приложения, которое по размеру больше экрана. Вместо приложений со страничной структурой, типичных для смартфонов, в руководстве по стилю оформления Metro вводится понятие **панорамного приложения**. В таких приложениях телефон выступает в роли окна, перемещающегося по поверхности интерфейса



**Рис. 1.2.** Структура экрана телефона



**Рис. 1.3.** Панель приложения в действии

приложения. На рис. 1.4 показано, что большую часть клиентской области экрана занимает содержательная информация, однако справа виден кусочек следующей секции панорамного приложения, так что пользователь понимает, что показанным его функциональность не исчерпывается.



Рис. 1.4. Панорамное приложение

Когда пользователь перемещается внутри панорамного приложения, в окне отображаются разные части виртуального пространства. Так, на рис. 1.5 показано, что после сдвига приложения влево становится видна правая часть панорамы.

Интерфейс панорамных приложений прост, но очень эффективен и интуитивно понятен пользователю.



Рис. 1.5. Последняя секция панорамного приложения

Придерживаясь рекомендаций Metro, вы сможете создавать приложения, согласованные с общим дизайном телефона. Однако ничто не мешает писать приложения любого вида. В этом смысле Metro помогает разработчику, формулируя основные принципы дизайна приложения для Windows Phone, позволяющие избежать стилистического разнobia. Но при этом Metro явно указывает, что можно занять под пользовательский интерфейс весь экран и не пользоваться базовым оформлением Metro, так что у вас есть выбор: придумать нестандартный интерфейс или писать приложения, органично сочетающиеся с телефоном.

На рис. 1.6 показаны примеры приложений с применением и без применения оформления.

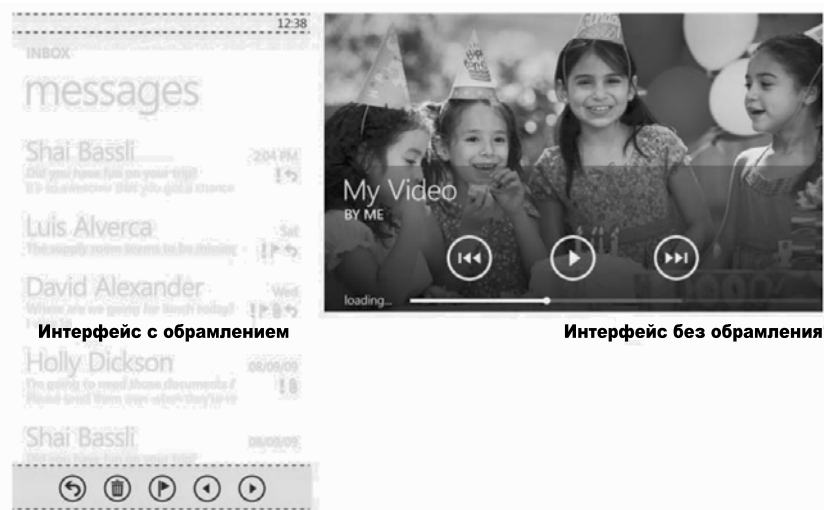


Рис. 1.6. Использовать оформление Metro или нет

## Интеграция с телефоном

Одна из основных целей телефона – предложить интегрированную платформу, на которой можно разрабатывать приложения, взаимодействующие друг с другом и с базовыми средствами телефона. Это означает, что можно писать приложения, интегрирующиеся с телефоном ранее невиданными способами. В табл. 1.1 перечислены некоторые из основных подсистем, с которыми может взаимодействовать пользователь телефона.



Таблица 1.1. Интеграция с телефоном

Подсистема	Описание
People (Люди)	Всё относящееся к людям: контакты, принятые вызовы и SMS-сообщения
Office (Офис)	Интеграция с электронной почтой, а также файлами Word, Excel и PowerPoint
Music+Videos (Музыка+видео)	Мультимедийные файлы в устройстве
Marketplace	Возможность опробовать приложение, купить и установить в телефон
Pictures (Фото)	Просмотр и обмен фотографиями, фотографирование с помощью устройства
Games (Игры)	Возможность играть на телефоне, включает интеграцию с Xbox Live

Вам, как разработчику, может нравиться или не нравиться способ кодирования традиционных приложения. Но поскольку вы можете писать приложения, запускаемые традиционным способом, то интеграция с имеющимися подсистемами позволяет взаимодействовать с ними и даже встраиваться в них. А это означает, что ваше приложение может расширять и обогащать стандартные подсистемы.

## Технические характеристики телефона

При создании Windows Phone от корпорации Microsoft требовалось не только умение создавать программное обеспечение, но и способность подвигнуть партнеров на создание самого телефона. Наученная опытом предшествующей платформы Windows Mobile, Microsoft решила очень точно специфицировать оборудование, чтобы, с одной стороны, пользователям было приятно работать с устройством, а, с другой, оставить производителям достаточную гибкость при выборе функциональности, позволяющую конкурировать друг с другом. В табл. 1.2 перечислены требования к оборудованию.

Дополнительно Windows Phone налагает некоторые физические ограничения. Самое очевидное из них то, что телефон должен иметь семь стандартных элементов взаимодействия, показанных на рис. 1.7.

Эти аппаратные элементы описаны в табл. 1.3.

Таблица 1.2. Технические характеристики оборудования

Категория	Требование
Разрешение экрана	WVGA (480 x 800)
Емкостный сенсорный экран	Поддержка не менее четырех точек касания
Память	ОЗУ 256 МБ, флэш 8 ГБ
Датчики	A-GPS, акселерометр, компас, датчик близости, датчик освещенности, гироскоп
ЦП	ARM7 Scorpion/Cortex или более мощный (как правило, не менее 1 ГГц)
Графический процессор	Ускоритель DirectX 9
Камера	Минимум 5 мегапикселей, вспышка обязательна
Bluetooth	Bluetooth 2.1 + EDR; обязательные профили Bluetooth – с гарнитурой (Hand-Free Profile – HFP), с наушниками (Headset Profile – HSP), расширенный профиль распространения аудио (Advanced Audio Distribution Profile – A2DP) и с доступом к телефонной книге (Phone Book Access Profile – PBAP)
Мультимедиа	Обязательное ускорение кодеков; поддержка DivX 4, 5 и 6, а также профиля H.264 High Profile (используется в технологии Blu-ray)
Wi-Fi	Обязательная поддержка стандарта 802.11g
Радио	Обязательное наличие радиоприемника FM-диапазона



Рис. 1.7. Семь стандартных элементов взаимодействия

Таблица 1.3. Аппаратные элементы взаимодействия

	Элемент	Ожидаемое поведение
1	Кнопка питания	Если устройство выключено, то долгое нажатие включает питание. Если устройство включено и экран активен, то нажатие выключает экран и блокирует устройство. Если экран выключен, то нажатие включает экран и выводит интерфейс разблокирования.
2	Управление громкостью	Кулисный переключатель регулирует громкость для звукового профиля текущей операции (например, громкость звонка во время телефонного разговора отключает звонок. Регулировка громкости в момент, когда нет текущей операции, позволяет переключаться между разными звуковыми профилями.
3	Сенсорный экран	Емкостный сенсорный экран с поддержкой не менее четырех точек касания.
4	Кнопка «Камера»	Долгое нажатие запускает приложение для работы с камерой.
5	Кнопка «Назад»	Активирует операцию «возврата». Это может быть возврат в конкретное приложение или переход от текущего приложения к предыдущему в соответствии со страничным API.
6	Кнопка «Пуск»	Возвращает пользователя на начальный экран.
7	Кнопка «Поиск»	Запускает операцию поиска по устройству.

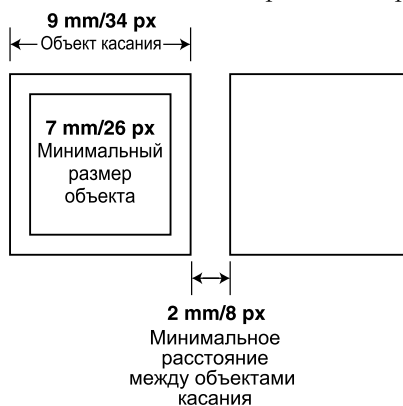
Познакомившись с отдельными элементами телефона, посмотрим, как с ним взаимодействует пользователь.

## Варианты взаимодействия

Вы разработчик. И хотите, чтобы пользователям нравилось работать с вашими приложениями. А, значит, вы должны учитывать различные способы взаимодействия пользователя с телефоном. При создании приложения для веб или настольного компьютера мы имеем дело преимущественно с клавиатурой и мышью. Но когда речь заходит о телефоне, приходится менять свое представление о способах ввода информации и принимать во внимание, что пользователь может взаимодействовать с приложением разными способами. Это и касание, и клавиатуры (аппаратная и программная), и аппаратные кнопки, и датчики.

## Дизайн с учетом сенсорного ввода

Язык дизайна Metro спроектирован в предположении, что касание является полноценной операцией, а для работы с интерфейсом не требуется обучение (то есть он интуитивно очевиден). Включив в язык дизайна средства для определения элементов сенсорного интерфейса, Microsoft облегчила разработчикам построение таких интерфейсов. В спецификации языка дизайна имеются рекомендации по поводу поддерживаемых жестов, а также указания о том, каковы должны быть размеры и промежутки между элементами, рассчитанными на касание пальцем. На рис. 1.8 приведен пример из спецификации Metro, в котором определяются минимальные размеры точек касания и расстояния между ними.



**Рис. 1.8.** Размеры интерактивных элементов в Metro

Metro определяет также типы взаимодействия (например, жесты), поддерживаемые устройством. Большинство из них привычны по таким сенсорным устройствам, как iPhone, Zune HD и Android. Перечислим их:

- Одиночное касание:
  - тычок;
  - двойной тычок;
  - сдвиг;
  - пролистывание (flick);
  - касание с удерживанием
- Множественное касание:
  - масштабирование/растяжение/поворот.

Помимо типов жестов, в языке Metro специфицированы варианты применения различных видов взаимодействия. Например, двойной тычок предназначен для увеличения и уменьшения размера. Здесь мы видим явное отличие от интерпретации, привычной разработчику Windows-приложений для настольных ПК. Но чтобы исключить разночтения, спецификация Metro описывает, что должен ожидать пользователь от разных видов взаимодействия. Конечно, реальные

пользователи не будут читать спецификацию Metro, но она является основой для встроенных в телефон приложений. Ваше приложение должно взаимодействовать с пользователем так, как все прочие, – следуя принципу минимального удивления. Кстати, это еще и указание на то, что дизайн телефона не должен основываться на соглашениях о работе Windows, он должен быть интуитивно более понятным. Сенсорный интерфейс очень отличается от мыши, и разработчики системы надеялись (как я понимаю), что пользователи смогут разобраться во взаимодействии с телефоном без специального обучения.

## **Аппаратные кнопки**

Windows Phone требует, чтобы у каждого телефона на передней стороне было три аппаратных кнопки. Выше, в разделе «Технические характеристики телефона» было отмечено, что действия этих кнопок дискретны. На практике вас будет интересовать лишь кнопка «Назад». Она не только должна позволять пользователю вернуться из вашего приложения в ранее запущенное (поведение по умолчанию), но и переходить из одного состояния вашего приложения в другое. Разрабатывая приложения для телефона, помните о том, какого поведения пользователь ожидает от кнопки «Назад». Это сделает вашу программу интуитивно более понятной.

## **Клавиатуры**

Не всякое взаимодействие сводится к простому жесту, необходимо еще поддерживать ввод текста. Поэтому в «Руководстве по проектированию пользовательского интерфейса и взаимодействий» (UI Design and Interaction Guide) оговорено, что для любой операции ввода текста должна быть доступна программная клавиатура (или панель функционального ввода, SIP), и даже если телефон оборудован аппаратной клавиатурой, у пользователя должна быть возможность набирать текст на экране. Клавиатуры по умолчанию предоставляются операционной системой. Когда пользователь пытается ввести текст (например, касается пальцем поля ввода), операционная система отображает программную клавиатуру. На рис. 1.9 показана клавиатура, предлагаемая по умолчанию.

В руководстве по стилю оформления Metro говорится также, что клавиатуры должны контекстно зависеть от типа вводимого текста. Например, на рис. 1.10 показаны клавиатуры для ввода адресов электронной почты и чисел.