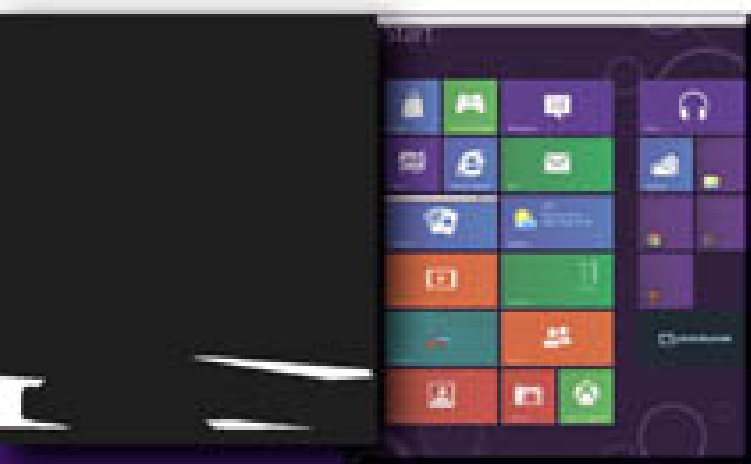


Windows 8:

разработка **Metro-приложений** для мобильных устройств



ПРИМЕНЕНИЕ «СВЯЗКИ»
HTML+CSS+JAVASCRIPT

ЭЛЕМЕНТЫ УПРАВЛЕНИЯ

РАЗМЕТКА ИНТЕРФЕЙСА

ГРАФИКА И МУЛЬТИМЕДИА

РАБОТА С ФАЙЛАМИ

РАБОТА С
ИНТЕРНЕТ-СЕРВИСАМИ

РАБОТА СО ВСТРОЕННЫМИ
ФОТО- И ВИДЕОКАМЕРАМИ

ПОКАЛИЗАЦИЯ
ПРИЛОЖЕНИЙ

СОЗДАНИЕ ПЛАТНЫХ
ПРИЛОЖЕНИЙ

УДК 681.3.06
ББК 32.973.26-018.2
Д75

Дронов В. А.

Д75 Windows 8: разработка Metro-приложений для мобильных устройств. — СПб.: БХВ-Петербург, 2012. — 528 с.: ил. — (Профессиональное программирование)

ISBN 978-5-9775-0832-2

Книга посвящена разработке Metro-приложений — нового класса приложений, работающих под управлением платформы Metro, входящей в состав Windows 8. Описана разработка приложений, предназначенных для устройств с сенсорными экранами, в частности планшетных компьютеров. Рассказано о разработке приложений на языках HTML, CSS и JavaScript, широко применяемых в Web-программировании. Рассмотрены элементы управления и их использование, разметка интерфейса приложений, вывод графики и мультимедиа, работа с файлами, удаленными интернет-сервисами, флэш-дисками, встроенными фото- и видеокерами. Показаны способы реализации обмена данными между приложениями, вывода информации на плитки меню **Пуск**, создания локализованных и платных приложений. Описан процесс публикации готовых приложений в магазине Windows Store.

Для программистов

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Елена Васильева</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Марины Дамбиевой</i>

Подписано в печать 30.06.12.
Формат 70×100¹/₁₆. Печать офсетная. Усл. печ. л. 42,57.
Тираж 1200 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

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

ISBN 978-5-9775-0832-2

© Дронов В. А., 2012
© Оформление, издательство "БХВ-Петербург", 2012

Оглавление

Введение	1
Планшетный бум.....	1
Windows + планшет = Metro	2
Чем будем пользоваться.....	3
Типографские соглашения	3
Благодарности.....	4
ЧАСТЬ I. ОСНОВЫ METRO-ПРОГРАММИРОВАНИЯ.....	5
Глава 1. Что такое платформа Metro	7
Платформа Metro как средство завоевания рынка планшетов.....	7
Ключевые особенности планшетов и их поддержка платформой Metro	8
Сенсорный экран.....	8
Ограниченные системные ресурсы.....	9
Ограниченный объем долговременной памяти	10
Иная целевая группа пользователей	11
Другие особенности платформы Metro.....	12
Достоинства и недостатки платформы Metro.....	13
Как создаются Metro-приложения.....	14
Что дальше?.....	16
Глава 2. Знакомство с Visual Studio. Первое Metro-приложение.....	17
Средства разработки Metro-приложений.....	17
Знакомство с Visual Studio	17
Главное окно и его элементы	18
Создание нового приложения	20
Проект и решение.....	21
Проект	21
Решение.....	22
Панель <i>SOLUTION EXPLORER</i>	23
Окна документов	24
Содержимое типичного проекта Metro-приложения	25
Первое Metro-приложение, часть первая	27
Добавление файла в проект	27
Открытие файла.....	27
Создание интерфейса	28
Сохранение файлов	29
Запуск Metro-приложения	29
Как выполняются Metro-приложения	29
Первое Metro-приложение, часть вторая	31
Перезапуск Metro-приложения	32
Завершение Metro-приложения.....	32
Закрытие решения	33
Первое Metro-приложение, часть третья	33
Открытие решения	33
Создание логики.....	34
Создание оформления.....	36
Закрытие файла	37
Удаление файла из проекта	37

Выявление и исправление ошибок	39
Что дальше?.....	40
Глава 3. Интерфейс и оформление Metro-приложения.....	41
Интерфейс приложения. Язык HTML.....	41
Теги	41
Атрибуты тегов	43
Порядок вывода элементов интерфейса. Блочные и встроенные элементы	44
Вложенность тегов.....	46
Служебные теги. Структура HTML-файла.....	47
Комментарии HTML.....	48
Оформление приложения. Каскадные таблицы стилей CSS.....	49
Стили и атрибуты стилей	49
Разновидности стилей. Привязка стилей	51
Таблицы стилей и их привязка.....	54
Объединение стилей. Правила каскадности.....	55
Комментарии CSS	58
Что дальше?.....	58
Глава 4. Логика Metro-приложения: основные понятия.....	59
Введение в язык JavaScript.....	59
Типы данных JavaScript.....	61
Переменные.....	62
Именованые переменных	62
Объявление переменных.....	63
Операторы	63
Арифметические операторы.....	63
Оператор объединения строк.....	64
Операторы присваивания	65
Операторы сравнения	65
Логические операторы.....	66
Оператор получения типа <i>typeof</i>	67
Совместимость и преобразование типов данных	68
Приоритет операторов.....	68
Сложные выражения JavaScript.....	70
Блоки	70
Условные выражения	70
Условный оператор ?.....	71
Выражения выбора.....	72
Циклы.....	73
Цикл со счетчиком	73
Цикл с постусловием	75
Цикл с предусловием	75
Прерывание и перезапуск цикла	76
Функции.....	76
Объявление функций	76
Локальные переменные	78
Вызов функций.....	78
Функция как значение. Анонимные функции.....	79
Встроенные функции	80
Массивы.....	80
Ссылки	82
Объекты и экземпляры объектов.....	83
Понятия объекта и экземпляра объекта	83
Получение экземпляра объекта.....	84

Работа с экземпляром объекта	86
Простые типы как объекты	87
Объект <i>Object</i> и использование его экземпляров	88
Правила написания выражений	89
Комментарии JavaScript	90
Что дальше?.....	90
Глава 5. Логика Metro-приложения: приемы программирования.....	91
Где хранится код логики Metro-приложения. Файлы логики	91
Объектная модель документа.....	92
Как получить доступ к элементу интерфейса из кода логики	94
Получение доступа к элементу по его имени.....	94
Получение доступа к элементам по имени создающего их тега	95
Получение доступа к элементам по наименованию привязанного к ним стилевого класса.....	96
Получение доступа к элементам по сложному критерию.....	97
События и их обработка.....	97
События	97
Обработчики событий.....	99
События, поддерживаемые всеми элементами интерфейса	100
Получение сведений о событии	101
Структура файла default.js.....	103
Служебные строки и выражения.....	103
Куда помещается собственно код логики	104
Инициализация Metro-приложения	105
Объект-обязательство.....	106
Пространства имен	107
Понятие пространства имен	107
Анонимное пространство имен.....	108
Перечисления	108
Что дальше?.....	109
ЧАСТЬ II. СОЗДАНИЕ ИНТЕРФЕЙСА METRO-ПРИЛОЖЕНИЙ.....	111
Глава 6. Элементы управления HTML.....	113
Основные элементы управления HTML.....	113
Кнопки	114
Простая кнопка.....	114
Сложная кнопка.....	115
Поле ввода	115
Работа с полями ввода	115
Проверка корректности введенных данных.....	117
События клавиатуры.....	120
Флажок.....	121
Переключатель	121
Область редактирования.....	122
Список.....	123
Регулятор	125
Вспомогательные элементы управления HTML.....	126
Индикатор прогресса	126
Надпись	128
Группа	129
Простые всплывающие подсказки.....	129
Пример приложения: арифметический калькулятор	130
Что дальше?.....	135

Глава 7. Элементы управления Metro	136
Введение в элементы управления Metro	136
Основные принципы работы с элементами управления Metro	137
Создание элементов управления Metro	137
Инициализация элементов управления Metro	137
Работа с элементами управления Metro	138
Основные элементы управления Metro	138
Элемент для ввода даты	139
Элемент для ввода времени	139
Переключатель	140
Элемент для ввода рейтинга	141
Панель вывода	143
Пример 1: усовершенствованный видеопроигрыватель	143
Пример 2: калькулятор значений даты	145
Что дальше?	148
Глава 8. Вывод и форматирование текста	149
Структурирование текста	149
Абзацы	149
Заголовки	150
Списки	151
Цитаты	152
Адреса	153
Разрывы строк	153
Выделение фрагментов текста	153
Оформление текста	154
Задание параметров шрифта	154
Параметры фона	157
Контейнеры. Встроенные контейнеры	158
Параметры абзацев и списков	158
Вставка недопустимых символов. Литералы	160
Создание таблиц	162
Формирование таблиц	162
Объединение ячеек таблиц	165
Что дальше?	167
Глава 9. Разметка	168
Блочные контейнеры, или блоки	168
Сеточная разметка	169
Создание сетки разметки	169
Позиционирование элементов интерфейса в ячейках сетки разметки	170
Выравнивание элементов в ячейках сетки разметки	172
Пример: окончательная версия арифметического калькулятора	173
Гибкая разметка	176
Создание гибкой разметки	177
Дополнительные параметры гибкой разметки	178
Пример: прототип приложения для чтения каналов RSS	180
Дополнительные параметры элементов интерфейса	182
Параметры размеров	182
Параметры отступов	183
Параметры рамки	185
Параметры отображения	186
Параметры выравнивания	187
Свободно позиционируемые элементы интерфейса	187
Программное управление местоположением, размерами и видимостью элементов интерфейса	189
Программная привязка стилевых классов	190
Программное указание параметров оформления	191

Получение местоположения и размеров элементов интерфейса	193
Пример: дальнейшее совершенствование видеопроигрывателя	194
Дополнительные инструменты разметки	196
Верстка текста в несколько колонок	197
Реализация прокрутки.....	199
Пример: усовершенствованный прототип приложения для чтения каналов RSS	200
Что дальше?.....	202
Глава 10. Программное формирование элементов интерфейса.....	203
Программное создание элементов интерфейса	203
Простой способ: прямое указание HTML-кода	203
Сложный способ: сборка элементов интерфейса	204
Собственно создание элемента интерфейса.....	204
Создание текстового содержимого.....	205
Задание параметров элемента интерфейса.....	205
Вывод созданного элемента интерфейса на экран	206
Программное удаление элементов интерфейса.....	206
Пример: усовершенствованный прототип приложения для чтения каналов RSS.....	207
Что дальше?.....	210
ЧАСТЬ III. РАБОТА С ГРАФИКОЙ И МУЛЬТИМЕДИА	211
Глава 11. Вывод графических изображений.....	213
Графические форматы, поддерживаемые платформой Metro	213
Средства HTML для вывода графических изображений	214
Реализация масштабирования графики с помощью жестов.....	215
Реализация масштабирования	215
Дополнительные возможности прокрутки.....	218
Создание графического фона.....	219
Что дальше?.....	221
Глава 12. Мультимедиа	222
Базовые средства для воспроизведения мультимедиа	222
Воспроизведение звука	222
Воспроизведение видео	224
Программное управление воспроизведением мультимедиа.....	225
Свойства.....	225
Методы.....	226
События	227
Поддержка видеофайлов с несколькими звуковыми дорожками	229
Пример: усовершенствованный видеопроигрыватель	230
Что дальше?.....	237
ЧАСТЬ IV. СОЗДАНИЕ СЛОЖНЫХ ЭЛЕМЕНТОВ ИНТЕРФЕЙСА	239
Глава 13. Фрагменты	241
Введение во фрагменты.....	241
Создание фрагментов	242
Содержимое вновь созданного фрагмента	243
Организация фрагментов	244
Создание интерфейса и оформления фрагмента	245
Создание логики фрагмента.....	245
Инициализация фрагмента	245
Обеспечение взаимодействия между фрагментом и основным приложением	246
Загрузка фрагмента.....	248
Удаление фрагмента	249

Пример: прототип "многооконного" приложения для чтения каналов RSS.....	249
Что дальше?.....	254
Глава 14. Списки Metro	255
Замечания о создании и инициализации списков Metro.....	255
Список Metro, выводящий несколько позиций	256
Создание списка, выводящего несколько позиций	256
Создание пунктов списка.....	258
Подготовка массива данных.....	258
Создание источника данных, получение адаптера и привязка его к списку	258
Использование шаблонов для оформления пунктов списка.....	259
Фильтрация пунктов списка.....	261
Сортировка пунктов списка	262
Группировка пунктов списка	263
Получение выбранных пунктов	265
Реализация правки данных, выводимых в списке	267
Прочие возможности	268
Пример: прототип приложения для чтения каналов RSS, использующий список Metro	269
Список-слайдшоу Metro	273
Создание списка-слайдшоу	273
Работа со списком-слайдшоу	274
Что дальше?.....	275
Глава 15. Панели инструментов, всплывающие элементы и меню.....	276
Панели инструментов Metro	276
Создание панелей инструментов, содержащих только кнопки.....	277
Создание самих панелей инструментов	277
Создание обычных кнопок	278
Создание кнопок-выключателей.....	279
Создание разделителей	279
Создание универсальных панелей инструментов.....	280
Работа с панелями инструментов и кнопками	280
Пример: видеопроигрыватель, использующий панели инструментов.....	281
Всплывающие элементы Metro.....	288
Создание всплывающего элемента	288
Работа со всплывающими элементами	289
Пример: вывод сведений о видеофайле в приложении видеопроигрывателя.....	290
Вывод всплывающего элемента после нажатия кнопки на панели инструментов	291
Меню.....	292
Что дальше?.....	293
ЧАСТЬ V. РАБОТА С ФАЙЛАМИ И СЕТЬЮ	295
Глава 16. Работа с файлами	297
Диалог открытия файла.....	297
Подготовка диалога открытия файла	297
Вывод диалога и получение выбранного в нем файла	299
Вывод выбранного пользователем файла на экран.....	300
Чтение из файла	301
Открытие файла для чтения	301
Получение потока чтения и читателя данных	302
Собственно чтение из файла	303
Чтение текстовых файлов.....	303
Чтение двоичных файлов	303
Закрытие читателя.....	305
Запись в файл	305
Диалог сохранения файла.....	305

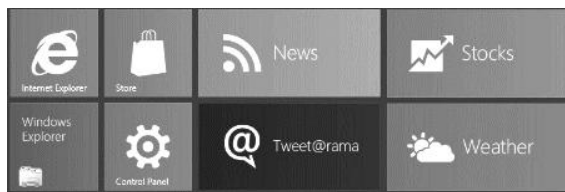
Открытие файла для записи, получение потока записи и писателя данных	307
Собственно запись в файл	308
Запись строковых данных.....	308
Запись двоичных данных.....	308
Завершение операции записи и закрытие читателя.....	310
Пример: простейший текстовый редактор.....	310
Получение списка файлов и папок	313
Замечания о программном доступе к библиотекам. Права приложения	313
Получение доступа к библиотекам	316
Диалог выбора папки	317
Собственно получение списка файлов и папок	317
Получение простого списка файлов и папок	317
Получение списка файлов с сортировкой	318
Получение списка файлов с группировкой	319
Получение сведений о файлах и папках.....	321
Получение основных сведений о файле	321
Получение миниатюры файла	322
Получение сведений о папке и миниатюры содержимого папки.....	324
Пример: приложение для просмотра графических файлов	324
Действия над файлами и папками	328
Действия над файлами	328
Действия над папками.....	332
Хранилища приложения.....	334
Что дальше?.....	336
Глава 17. Работа с каналами новостей RSS и Atom.....	337
Подготовка интернет-адреса.....	337
Создание клиента новостей и загрузка содержимого канала	338
Получение сведений о канале новостей.....	339
Получение отдельных новостей	339
Пример: окончательная версия приложения для чтения каналов RSS.....	341
Что дальше?.....	346
Глава 18. Загрузка данных из сети	347
Вывод Web-страниц.....	347
Фреймы	347
Пример: окончательная версия приложения для чтения каналов новостей.....	348
Фоновая загрузка файлов	349
Подготовительные действия	350
Загрузка файла.....	350
Приостановка, возобновление и прерывание загрузки	351
Получение сведений о прогрессе загрузки	352
Получение сведений о возникшей ошибке	352
Возобновление загрузок, оставшихся после предыдущего запуска приложения.....	353
Пример: приложение для загрузки файлов.....	354
Взаимодействие с удаленными Web-сервисами	358
Формирование запроса	358
Отправка запроса и получение ответа	359
Обработка ответа.....	359
Введение в Bing API.....	360
Параметры запроса.....	360
Содержание ответа.....	361
Пример: приложение для поиска изображений по ключевому слову.....	362
Гиперссылки.....	366
Что дальше?.....	368

ЧАСТЬ VI. ОБМЕН ДАННЫМИ И РАБОТА С УСТРОЙСТВАМИ	369
Глава 19. Обмен данными между Metro-приложениями	371
Передача данных	371
Подготовительные действия	371
Собственно передача данных	372
Передача текста	372
Передача интернет-адреса	373
Передача графических изображений	373
Передача файлов	374
Задание параметров передаваемых данных	375
Отложенная передача и передача по требованию	376
Отложенная передача	376
Передача по требованию	377
Получение данных	378
Подготовительные действия	378
Указание прав приложения на получение данных	378
Отслеживание активизации приложения	379
Определение вида полученных данных	380
Собственно получение данных	381
Получение текста	381
Получение интернет-адреса	381
Получение графического изображения	382
Получение набора файлов	383
Получение параметров принятых данных	383
Завершение получения данных	384
Отложенное получение данных	384
Пример: реализация передачи и получения данных	385
Модификация приложения для просмотра изображений	385
Добавление существующего проекта в решение	386
Модификация приложения — текстового редактора	387
Создание нового проекта в составе решения	388
Создание тестового приложения, принимающего данные	388
Что дальше?	390
Глава 20. Работа с флэш-дисками и камерами	391
Работа с флэш-дисками	391
Указание прав приложения на доступ к флэш-дискам и набора поддерживаемых им команд	391
Собственно работа с флэш-дисками	393
Пример: доработка приложения видеопроигрывателя для поддержки AutoPlay	394
Получение фото	395
Подготовительные действия	395
Собственно получение фото	395
Задание параметров получаемых фотографий	396
Получение видео	398
Подготовительные действия	398
Собственно получение видео	398
Задание параметров получаемых видеороликов	398
Что дальше?	399
ЧАСТЬ VII. ПРОЧИЕ ВОЗМОЖНОСТИ METRO	401
Глава 21. Работа с плитками меню Пуск и всплывающими уведомлениями	403
Вывод информации на плитки меню Пуск	403
Выбор шаблона для плитки	404

Заполнение шаблона данными	405
Вывод информации на плитку	406
Вывод информации на плитки разных размеров	406
Задание параметров информации, выводимой на плитку	408
Сброс плитки	409
Наклейки	409
Пример: вывод на плитку имени файла, выбранного в приложении для просмотра графики, и общего количества файлов	410
Вторичные плитки	412
Создание вторичных плиток	412
Обработка нажатий на вторичные плитки	414
Вывод информации и наклеек на вторичные плитки	414
Работа с вторичными плитками	415
Удаление вторичных плиток	416
Всплывающие уведомления	417
Активизация функции вывода всплывающих уведомлений	417
Выбор шаблона для всплывающего сообщения и заполнение его данными	417
Задание воспроизводимого звука	418
Вывод всплывающего уведомления	419
Реализация отслеживания нажатий на всплывающее уведомление	420
Дополнительные возможности всплывающих уведомлений	420
Пример: вывод сообщения об ошибке открытия видеофайла в приложении видеопроигрывателя	421
Что дальше?	422
Глава 22. Управление жизненным циклом Metro-приложения	423
Жизненный цикл Metro-приложения	423
Проблема потери и устаревания рабочих данных	424
Сохранение и восстановление рабочих данных	425
Сохранение рабочих данных	425
Восстановление рабочих данных	426
Обновление загруженных данных	426
Активизация и деактивизация приложения в среде Visual Studio	427
Пример 1: сохранение и восстановление рабочих данных в приложении видеопроигрывателя	427
Пример 2: перезагрузка содержимого канала новостей после активизации приложения	429
Что дальше?	430
Глава 23. Создание настраиваемых Metro-приложений	431
Как пользователь будет настраивать Metro-приложение	431
Хранилища настроек	432
Сохранение настроек	433
Сохранение простых значений	433
Создание составных значений	434
Создание вложенных контейнеров настроек	434
Считывание настроек	435
Отслеживание изменения настроек, сохраненных в переносимом хранилище	436
Пример: реализация настроек в приложении видеопроигрывателя	437
Что дальше?	440
ЧАСТЬ VIII. КОММЕРЦИАЛИЗАЦИЯ И РАСПРОСТРАНЕНИЕ METRO-ПРИЛОЖЕНИЙ 441	
Глава 24. Локализация Metro-приложений	443
Как создаются Metro-приложения для международного рынка	443
Процесс локализации Metro-приложения	444
Создание папок для хранения языковых ресурсов	445

Создание файлов с языковыми ресурсами	445
Указание ссылок на языковые ресурсы в HTML-коде	447
Локализация графических изображений	447
Локализация строковых значений	448
Инициализация языковых ресурсов.....	449
Пример: локализация приложения для чтения каналов новостей	449
Что дальше?.....	451
Глава 25. Адаптация Metro-приложений для устройств с различными параметрами экрана	452
Экраны устройств, работающих под Windows 8, и их типичные параметры	452
Режимы работы Metro-приложения	453
Когда следует адаптировать приложения под различные экраны	454
Медиазапросы CSS	455
Введение в медиазапросы	455
Создание медиазапросов	456
Написание условий для медиазапросов.....	457
Пример: адаптация приложения для чтения каналов новостей под портретную ориентацию устройства.....	460
Адаптация изображений под разные значения плотности пикселей.....	461
Программное определение размеров экрана и ориентации устройства	462
Тестирование приложений при помощи симулятора	464
Что дальше?.....	466
Глава 26. Создание коммерческих Metro-приложений.....	467
Бесплатные, условно-бесплатные и платные приложения	467
Коммерческое предложение Microsoft.....	469
Создание условно-бесплатного приложения	469
Лицензия приложения	469
Получение сведений о лицензии	470
Получение сведений о приложении из магазина Windows Store	471
Реализация покупки приложения	472
Отслеживание изменений в лицензии	473
Реализация покупки отдельных функций приложения	473
Тестирование условно-бесплатных приложений	475
Пример 1: условно-бесплатное приложение текстового редактора	480
Пример 2: приложение видеопроигрывателя с платной функцией	484
Перевод приложения из тестового в рабочий режим	487
Что дальше?.....	488
Глава 27. Распространение Metro-приложений.....	489
Требования к публикуемым приложениям	489
Задание общих параметров приложения	490
Окончательное тестирование приложения	493
Задание параметров дистрибутивного пакета.....	493
Создание дистрибутивного пакета	493
Установка приложения из дистрибутивного комплекта	495
Тестирование приложения с помощью Windows App Cert Kit.....	496
Регистрация в магазине Windows Store.....	499
Публикация приложения	499
Заключение	503
Предметный указатель	505

ГЛАВА 1



Что такое платформа Metro

В последнее время компьютерный рынок претерпевает коренные изменения. Продажи традиционных ПК падают, и на их место приходят другие устройства — более мобильные и ориентированные на неподготовленного пользователя. Это планшеты, или, говоря иначе, мобильные устройства с большим сенсорным экраном.

Планшеты оккупируют полки компьютерных магазинов. Планшеты не сходят со страниц компьютерной прессы. Планшеты — это модно. Планшеты — это стильно. Планшеты — это фетиш нового времени.

Платформа Metro как средство завоевания рынка планшетов

Существуют три операционные системы, под управлением которых работают планшеты. Первая — Apple iOS, которую можно назвать старожилом рынка. Вторая — Android, появившаяся позже. Третья — Microsoft Windows 8, точнее, входящая в ее состав платформа Metro; напористый новичок, пока не столь популярный, как первые две системы, но уверенно сокращающий отставание.

Metro — программная платформа, предназначенная для создания и выполнения мобильных приложений. Она предназначена именно для планшетов и поддерживает все ключевые особенности этого класса устройств. Вероятно, это первая разработка Microsoft подобного рода, и совсем не похоже, чтобы первый блин вышел комом.

Армия владельцев Windows-планшетов велика, и потребности ее неизмеримы. Говоря проще, им требуется масса разнообразных приложений, работающих под управлением платформы Metro. И если рынок традиционных Windows-приложений уже давно перенасыщен, то рынок приложений под платформу Metro (*Metro-приложений*) пока еще очень и очень далек от насыщения.

Здесь раздолье для независимых разработчиков. Так что начинайте создавать Metro-приложения прямо сейчас! Вы потесните конкурентов, завоюете симпатии

пользователей и обзаведетесь массой поклонников. Вы успеете раньше всех, займете свою рыночную нишу и станете пионерами, легендами индустрии. Наконец, вы сможете заработать, продавая свои приложения, — и Microsoft поддержит вас в этом.

А начать собственно разработку Metro-приложений вам поможет эта книга.

Ключевые особенности планшетов и их поддержка платформой Metro

Но чем так хороша платформа Metro? Вообще, что она поддерживает?

Основные возможности Metro мы будем рассматривать вместе с ключевыми особенностями планшетов как класса мобильных устройств. Они неотделимы друг от друга.

Сенсорный экран

Первая ключевая особенность планшетов — разумеется, сенсорный экран. Более того, это обязательная составная часть любого подобного устройства; планшетов без сенсорного экрана просто не существует.

Понятно, что работа с планшетом выполняется путем манипуляций с сенсорным экраном. Пользователь тыкает пальцем в кнопки, листает списки, перемещает регуляторы, прокручивает текст и выполняет все остальные действия, поддерживаемые приложением (их еще называют *жестами*). А если возникает необходимость набрать текст, он делает это с помощью экранной клавиатуры, опять же, пальцами.

На традиционных ПК для этой цели применяются не менее традиционные клавиатура и мышь. В чем-то они удобнее, в чем-то — нет, но самое главное — они имеют принципиальные отличия от сенсорного экрана.

С одной стороны, мышью можно указать в конкретное место экрана с максимальной точностью, буквально ткнуть ей в единичный пиксел. Это позволяет разработчикам приложений создавать очень компактные элементы интерфейса, скажем, служебные кнопки небольшого размера, узкие заголовки окон, крошечные регуляторы, находящиеся в строке статуса, и т. п. Пользователь в любом случае не промахнется.

В случае сенсорного экрана так сделать не получится. Человеческий палец имеет слишком большие размеры. Так что, если пользователь попытается нажать какую-либо маленькую кнопку, он либо не попадет по ней, либо заодно нажмет сразу все соседние кнопки. Понятно, что ничего хорошего из этого не получится.

С другой стороны, с помощью сенсорного экрана можно реализовать поддержку жестов, выполняемых сразу несколькими пальцами. Например, пользователь может приложить к экрану два пальца и раздвинуть их, чтобы увеличить размер изображения (кстати, этот жест реально применяется в планшетах Apple).

Мышью же в каждый конкретный момент можно указать только в одну точку экрана. Поэтому о "многопальцевых" жестах придется забыть.

Из этого следуют два важных вывода.

- Интерфейс мобильных приложений, предназначенных для планшетов, придется делать с учетом того, что с ними будут работать пальцами. Это значит, что элементы интерфейса должны иметь достаточно большие размеры, и промежутки между ними следует увеличить, чтобы пользователь случайно не нажал сразу две кнопки.
- В мобильных приложениях можно предусмотреть поддержку жестов, в том числе и "многопальцевых". Если, конечно, приложение от этого выиграет...

Платформа Metro изначально поддерживает и "пальцевый" ввод, и жесты. В большинстве случаев разработчикам даже не придется реализовывать это специально — за них все сделает сама эта платформа.

Однако Metro поддерживает и традиционные устройства ввода — клавиатуру и мышь. Так что пользователь сможет подключить к своему планшету USB-клавиатуру и набирать текст со всеми удобствами.

Ограниченные системные ресурсы

Аппаратные платформы, на основе которых создаются планшеты, не столь мощны, как платформы традиционных ПК. Центральные процессоры, применяемые в планшетах, имеют существенно меньшее быстродействие, и объем установленной в них оперативной памяти также невелик.

Далее, все без исключения планшеты питаются от аккумулятора. Этот аккумулятор имеет вполне конечную емкость и неприятную особенность разряжаться в самый "интересный" момент. Если же запитать планшет от сети, то он перестанет быть мобильным устройством и, следовательно, лишится своего главного преимущества по сравнению с традиционным ПК — мобильности.

Иначе говоря, системные ресурсы планшетов ограничены.

Вспомним, как работает пользователь традиционного ПК. Он запускает сразу несколько различных приложений — Web-обозреватель, клиенты электронной почты и системы мгновенных сообщений, текстовый процессор — и переключается между ними. Конечно, каждое запущенное приложение загружает центральный процессор и отнимает часть оперативной памяти, но, поскольку процессор имеет достаточное быстродействие и объем оперативной памяти весьма велик, производительность если и падает, то совсем неощутимо. А, раз компьютер питается от сети, пользователь не рискует внезапно оказаться наедине с сообщением "Ваш аккумулятор разряжен".

С планшетами так не разгуляешься... Как уже говорилось, центральный процессор планшета относительно слаб, и памяти в нем также мало. Следовательно, запустив несколько приложений, пользователь столкнется с существенным падением производительности. При этом, раз нагрузка на процессор и память возросла, аккумулятор будет разряжаться интенсивнее и, следовательно, истощится быстрее.

Выход из этого положения таков:

- все неактивные приложения принудительно приостанавливаются. При этом они продолжают отнимать оперативную память, но, по крайней мере, не будут загружать центральный процессор;
- что касается задач, которые должны выполняться даже будучи неактивными (например, фоновая загрузка файлов), то здесь возможны два сценария:
 - сама система может очертить круг задач, выполняемых в неактивном состоянии, и они будут выполняться, даже если запустившее их приложение неактивно и, следовательно, приостановлено;
 - приложение, которое должно выполняться в неактивном состоянии, может "попросить" систему не приостанавливать его. Естественно, пользователь должен быть в курсе;
- приложения, с которыми пользователь давно не работал, могут принудительно завершаться и выгружаться из памяти, чтобы освободить системные ресурсы.

Платформа Metro поступает именно так. Она приостанавливает неактивные приложения, выгружает из памяти те, что давно не использовались, и может выполнять определенные задачи даже в неактивном состоянии. В общем, делает все, чтобы продлить время "жизни" аккумулятора.

Ограниченный объем долговременной памяти

Традиционные ПК имеют вместительные жесткие диски. Настолько вместительные, что даже после установки нескольких десятков приложений, в том числе пары современных игр, и записи тысяч фотографий, сотен музыкальных альбомов и десятков фильмов на них все еще остается немало места.

В планшетах жестких дисков нет. В них применяется другой тип долговременной памяти — flash-память. Хотя она лучше подходит для мобильных устройств (более устойчива к механическим воздействиям и имеет меньшее энергопотребление, чем жесткие диски), ее удельная стоимость существенно выше.

Так что много долговременной памяти в планшеты не ставят.

Из чего следует, что количество приложений, которые могут быть установлены на планшет, крайне ограничено. (Ведь пользователь может занять память не только приложениями, но и теми же фотографиями, музыкой и фильмами.) И вполне может наступить момент, когда очередное жизненное необходимое приложение пользователь установить уже не сможет.

Есть два способа решить эту проблему: попросить пользователей умерить аппетиты (что маловероятно) или попытаться сделать приложения как можно более компактными (а это уже вполне реализуемо).

Компактными их можно сделать довольно просто — переложить максимум типичных задач на операционную систему. К таким задачам можно отнести, например, обработку новостей RSS, получение изображения с фото- или видеокамеры, загруз-

ку файлов, "общение" с магазином приложений (о нем — чуть позже) и т. п. В результате разработчику, чтобы реализовать, скажем, получение изображения со встроенной фотокамеры, достаточно будет написать всего одну команду, которая займет в коде готового приложения очень мало места.

Платформа Metro может похвастаться всеми перечисленными ранее "умениями". И уже в *главе 2*, создавая наше первое приложение для данной платформы, мы в этом убедимся.

Иная целевая группа пользователей

Традиционные ПК, что бы ни говорили, предназначены для более-менее подготовленного пользователя, который способен самостоятельно найти нужное ему приложение, установить его и удалить, если оно ему больше не нужно. А самое главное — пользователь должен быть в состоянии решать проблемы, которые могут быть вызваны конфликтами этого приложения с другими, уже установленными на компьютере, и с программами, входящими в состав операционной системы. И хорошо, если пользователь будет способен убирать постоянно появляющийся "мусор", в частности временные файлы, в изобилии оставляемые некорректно работающими приложениями.

Планшеты, как уже говорилось, рассчитаны на пользователей, не знакомых с компьютерами. Их можно рассматривать как бытовую технику, которая начинает нормально работать сразу после извлечения из коробки и, в идеале, без чтения инструкции.

Стало быть, операционные системы для планшетов должны удовлетворять следующим требованиям.

- Пользователь должен быстро и без проблем найти нужное ему приложение и при этом быть уверенным, что оно полностью функционально, не содержит ошибок и вредоносного кода.
- Обычно все приложения, предназначенные для планшетов, публикуются на особом Web-сайте — *магазине приложений*. Все приложения, предлагаемые к публикации в таком магазине, обязательно проходят проверку, по крайней мере, на функциональность и отсутствие вредоносного кода. При этом пользователь может установить на планшет только приложения, загруженные из магазина; установить приложение, полученное из другого источника, как правило, невозможно.
- Установка приложения должна быть максимально простой и не требовать вмешательства пользователя. Наилучший вариант — когда от пользователя требуется всего лишь нажать кнопку **Установить**.
- Устанавливаемые приложения (по крайней мере, прикладные) не должны вносить изменений в саму операционную систему.
- Приложения (по крайней мере, прикладные) не должны конфликтовать ни друг с другом, ни с операционной системой. Или, говоря на жаргоне профессиональ-

ных программистов, приложения должны быть максимально изолированы друг от друга.

- ❑ Обновление приложения должно выполняться максимально просто и прозрачно для пользователя. От пользователя должно требоваться лишь согласие на обновление.
- ❑ Удаление ненужного приложения должно быть максимально простым и не требовать никакого вмешательства пользователя, кроме нажатия кнопки **Удалить**.
- ❑ Удаление приложения должно сопровождаться удалением всех сохраненных им служебных данных: настроек, временных файлов и пр. Разумеется, созданные в приложении документы удаляться не должны!
- ❑ Если уж приложения "мусорят", и с этим ничего не поделаешь, система сама должна время от времени "подметать" за ними.

Платформа Metro все это делает. Она поддерживает магазин приложений, обеспечивает простую установку и надежное удаление приложений, надежно изолирует приложения друг от друга и убирает за ними "мусор". Так что пользователь-"чайник" будет доволен!

Другие особенности платформы Metro

А теперь рассмотрим прочие особенности платформы Metro, о которых нам следует знать.

- ❑ Начать следует с того, что Metro — не полноценная операционная система. Она представляет собой один из компонентов Windows 8, работающий совместно с другими ее компонентами и, вместе с тем, достаточно независимый от них.
- ❑ Вместе с тем, Metro не является очередной надстройкой над традиционными интерфейсами программирования Windows (к таким надстройкам относится, в частности, популярная платформа .NET). Ее можно рассматривать как совершенно независимый интерфейс программирования, работающий на том же "уровне", что и традиционный. Благодаря этому достигается высокое быстродействие мобильных приложений и скромные требования к системным ресурсам.
- ❑ Metro входит в состав только Windows 8. Выпуск отдельной редакции этой платформы для более старых версий Windows не планируется.
- ❑ Наряду с процессорной архитектурой Intel x86, на которой основаны процессоры традиционных ПК, Windows 8 впоследствии получит поддержку архитектуры ARM. (Вообще, это первая в истории версия Windows, которая будет ее поддерживать.)
- ❑ Для распространения Metro-приложений служит магазин приложений *Windows Store*. Этот магазин позволяет публиковать платные, условно-бесплатные и полностью бесплатные приложения (в том числе и с открытым исходным кодом), производить покупки платных приложений, выполнять обновления и даже распространять бета-версии приложений среди независимых тестеров.

- ❑ Приложения для платформы Metro могут выполняться не только на планшетах, но и на любых компьютерах, на которых установлена Windows 8, в том числе и на традиционных ПК.
- ❑ На планшете (и вообще, любом компьютере) с Windows 8 могут одновременно исполняться как приложения, написанные для платформы Metro, так и традиционные Windows-приложения. При этом последние будут выполняться как обычно, без приостановок и принудительного удаления из памяти. Так что у владельцев Windows-планшетов появляется возможность работать с Microsoft Office и играть в Call of Duty (разумеется, подключив клавиатуру и мышь).

Как видим, Metro по-своему уникальна и сильно выделяется среди других планшетных платформ. Это своего рода мост между вселенной традиционных ПК и миром планшетов, призванный всемерно облегчить переход между ними.

Достоинства и недостатки платформы Metro

Теперь перечислим достоинства платформы Metro и не забудем о ее недостатках. Начнем с достоинств.

- ❑ Значительно бóльшая открытость в сравнении с Apple iOS. Windows 8 может быть установлена на любой планшет от любого производителя.
- ❑ Возможность выполнения традиционных Windows-приложений. (Правда, это относится не к самой Metro, а к Windows 8, частью которой она является.)
- ❑ Возможность публикации в магазине Windows Store бесплатных приложений с открытым исходным кодом, которая может привлечь многих разработчиков и еще большее количество потребителей. (Для сравнения: Apple запрещает публикацию таких приложений в своем магазине.)
- ❑ Простота разработки Metro-приложений. Даже начинающий программист может создать вполне функциональное приложение буквально за пять минут.
- ❑ Дружелюбность к разработчикам. Microsoft предлагает для создания Metro-приложений на выбор целых три технологии, причем все они уже давно присутствуют на рынке и прекрасно обкатаны. (Эти технологии мы рассмотрим чуть позже.)
- ❑ Дружелюбность к производителям оборудования. Чтобы обеспечить поддержку платформой Metro какого-либо устройства, например фотокамеры, производителю достаточно выпустить для него драйвер, который подойдет к любому компьютеру, на котором установлена Windows 8, в том числе и любому планшету.
- ❑ В конце концов, это Windows — самая популярная на данный момент операционная система.

Существенных недостатков же автор у платформы Metro не нашел, сколь ни искал. Это говорит о том, что Microsoft проделала большую работу и создала весьма впечатляющую платформу.

Как создаются Metro-приложения

Как уже говорилось, для создания Metro-приложения Microsoft предлагает на выбор три технологии. Давайте их рассмотрим.

Первая технология известна всем, кто создавал Web-страницы. Она включает в себя язык разметки *HTML* (HyperText Markup Language, язык гипертекстовой разметки), каскадные таблицы стилей *CSS* (Cascading Style Sheet) и язык программирования *JavaScript*. Язык *HTML* используется для описания интерфейса Metro-приложений, таблицы стилей *CSS* — для его оформления, а язык *JavaScript* — для создания программной логики.

Достоинства первой технологии таковы:

- Простота освоения, в том числе и такими начинающими разработчиками, как мы. Языки *HTML*, *CSS* и *JavaScript* исключительно просты и, вместе с тем, позволяют сделать довольно многое.
- Широкая распространенность. *HTML*, *CSS* и *JavaScript* знают очень и очень многие; в конце концов, это традиционные интернет-технологии.
- Metro-приложения, созданные с применением этой технологии, будут успешно выполняться на процессорах, основанных на любой архитектуре, что поддерживается *Windows 8*, — *Intel x86* и *ARM*.

Теперь перечислим недостатки первой технологии.

- Невысокое быстродействие созданных с ее помощью Metro-приложений.
- Несколько ограниченный набор возможностей по созданию интерфейса приложений.
- Любое, даже самое простое Metro-приложение, созданное с применением этой технологии, будет состоять из множества файлов. В некоторых случаях это может быть критично.

С применением *HTML*, *CSS* и *JavaScript* рекомендуется создавать только самые простые Metro-приложения, которым не требуется развитый интерфейс, а быстродействие не является критичным. Кроме того, эта технология — идеальный выбор для начинающих Metro-разработчиков.

Вторая технология знакома всем *.NET*-программистам. Это язык *XAML* (eXtensible Application Markup Language, расширяемый язык разметки приложений) и языки *C#*, *C++* *.NET* и *Visual Basic .NET*. На языке *XAML* описывается интерфейс и оформление приложения, а языки *C#*, *C++* *.NET* и *Visual Basic .NET* служат для создания его логики. Можно сказать, что эта технология является подмножеством платформы *.NET*.

Достоинств у нее побольше...

- Более высокое быстродействие готовых приложений.
- Большие возможности в плане создания интерфейса приложений и его оформления.

- ❑ Подавляющее большинство Metro-приложений, созданных с помощью этой технологии, будет состоять всего из одного файла.
- ❑ Metro-приложения, созданные с помощью этой технологии, также будут успешно выполняться на любых процессорах — и Intel x86, и ARM.

Недостатки:

- ❑ не самое высокое быстродействие готовых приложений. Так, применять эту технологию для создания сложных трехмерных игр не рекомендуется;
- ❑ освоить эту технологию значительно сложнее, чем первую, особенно начинающим разработчикам.

С помощью данной технологии уже можно создавать более сложные приложения, имеющие более развитый интерфейс и достаточно критичные в плане быстродействия. Также это наилучший выбор для переноса созданных ранее .NET-приложений на платформу Metro.

Третья технология — выбор для самых квалифицированных разработчиков. Это язык C++ и технология DirectX. Интерфейс, оформление и логика приложения описываются на языке C++, а DirectX применяется для вывода на экран как самого интерфейса приложения, так и результатов его работы.

Достоинства этой технологии весьма существенны:

- ❑ максимально возможное быстродействие готовых приложений;
- ❑ богатейшие возможности по созданию интерфейса приложений;
- ❑ вполне возможно создать приложение, состоящее всего из одного файла (хотя это, скорее всего, будет очень простое приложение).

Недостатки, увы, тоже...

- ❑ Интерфейс и все его элементы придется создавать заново для каждого из приложений. Готовых элементов интерфейса данная технология не предоставляет.
- ❑ Необходимо создавать две редакции каждого приложения — для каждой из процессорных архитектур, поддерживаемых Windows 8 (Intel x86 и ARM). При этом редакция, предназначенная для процессоров Intel x86, не будет работать на процессорах ARM, и наоборот.
- ❑ Освоить эту технологию очень сложно. Как уже говорилось, она рассчитана на самых квалифицированных программистов.

Данная технология рекомендуется для создания сложных трехмерных игр или графических приложений. Применять ее для разработки приложений другого назначения нецелесообразно.

Поскольку мы только начинаем свой путь в Metro-программирование, давайте выберем первую из рассмотренных технологий — "связку" HTML, CSS и JavaScript. С одной стороны, она очень проста для изучения, а с другой, позволяет создавать весьма впечатляющие приложения.

И на этом закончим с теоретической частью, чтобы поскорее приступить к практике.

Что дальше?

В этой главе мы познакомились с платформой Metro, являющейся частью грядущей Windows 8 и предназначенной для создания и выполнения "планшетных" приложений. Мы выяснили, насколько полно она поддерживает ключевые особенности планшетов, перечислили ее преимущества и безуспешно попытались найти хотя бы один недостаток. Напоследок мы узнали, какие технологии применяются для создания приложений, предназначенных для этой платформы. В общем, чистая "беллетристика"...

В следующей главе мы начнем знакомиться со средствами разработки Metro-приложений, создадим наше первое, пока еще совсем простое Metro-приложение, а позднее усовершенствуем его. По ходу дела мы узнаем, что собой представляют и как работают Metro-приложения. Так что скучать нам не придется!

ГЛАВА 2



Знакомство с Visual Studio. Первое Metro-приложение

Предыдущая глава была чисто ознакомительной. Мы познакомились с платформой Metro, узнали, как она поддерживает все ключевые особенности планшетов, и поговорили о самой этой платформе.

А еще мы познакомились с тремя технологиями, которые могут применяться для создания Metro-приложений, и выбрали для себя первую технологию — "связку" языков HTML, CSS и JavaScript, как самую простую в изучении.

Что ж, инструменты выбраны. Начнем работу!

Средства разработки Metro-приложений

На данный момент Microsoft для создания Metro-приложений предлагает два средства разработки:

- ❑ *Microsoft Visual Studio 11 Express for Windows 8* — очередную версию своего флагманского пакета Visual Studio. Это мощный программный пакет, предназначенный, в основном, для программистов;
- ❑ *Microsoft Expression Blend 5* — очередную версию пакета для разработки интерфейсов приложений, который предназначен, в основном, для специалистов, занимающихся дизайном интерфейсов; к сожалению, для программистов этот пакет не очень удобен.

Для разработки первых Metro-приложений мы выберем первое средство разработки — Microsoft Visual Studio 11 Express for Windows 8. (В дальнейшем для краткости будем называть его *Visual Studio*.) Особо сложных интерфейсов мы создавать не будем, а специфические "программистские" инструменты этого программного пакета нам очень и очень помогут.

Знакомство с Visual Studio

Перед тем как приступить к написанию нашего первого Metro-приложения в Visual Studio, давайте хотя бы поверхностно рассмотрим интерфейс этого пакета.

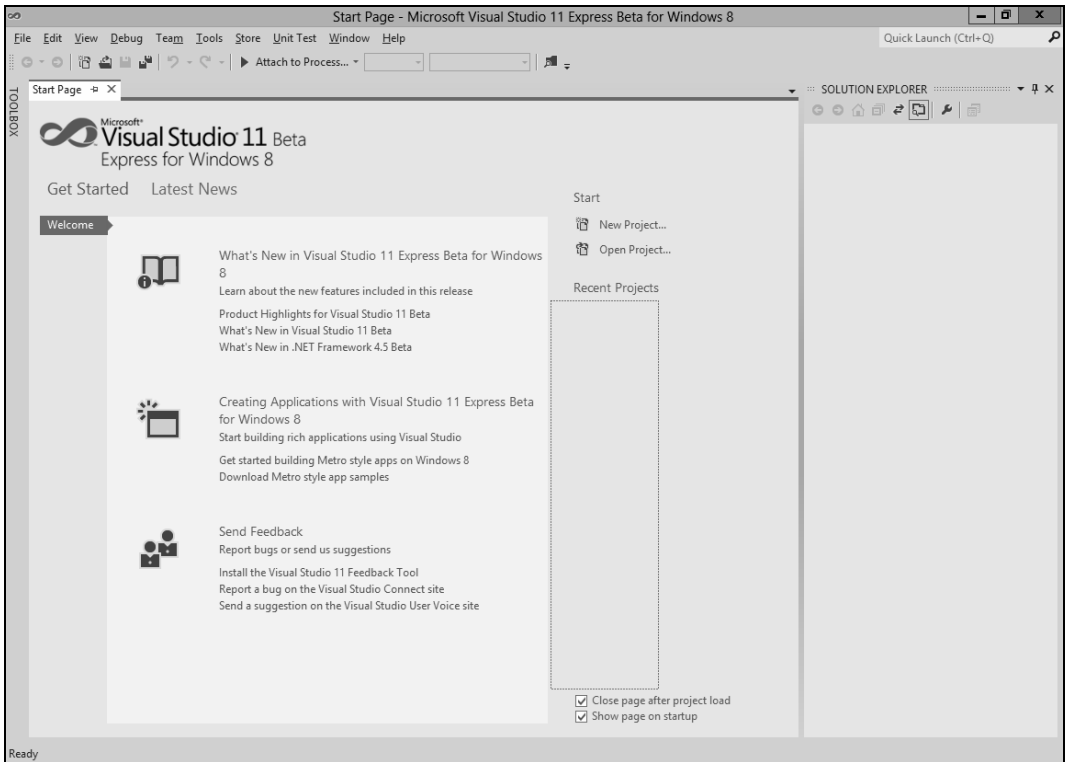


Рис. 2.1. Главное окно Visual Studio сразу после запуска пакета

Выйдем в меню **Пуск** (Start) Windows 8 и щелкнем на плитке¹ **Microsoft Visual Studio 11 Express for Windows 8**. Windows тотчас переключится на традиционный рабочий стол, и через некоторое время на экране появится главное окно Visual Studio (рис. 2.1).

ВНИМАНИЕ!

Сразу после первого запуска Visual Studio выведет окно-предупреждение, предлагающее установить лицензию разработчика Metro-приложений. Необходимо ответить положительно, нажав кнопку **I Agree**, т. к. без этой лицензии мы не сможем разрабатывать приложения такого типа. Впоследствии нам придется положительно ответить на появившееся на экране предупреждение системы УАС и выполнить вход на сервер лицензий под своим учетным именем, зарегистрированным в службе Microsoft Account.

Установка лицензии разработчика Metro-приложений будет выполнена всего один раз. В дальнейшем выполнять это действие нам больше не придется.

Главное окно и его элементы

Вдоль верхнего края главного окна тянется *главное меню*, из которого доступны все команды, что поддерживаются пакетом. Чуть ниже находится *панель инструмен-*

¹ Также эти значки называют *тайлами* от англ. *tile*. — *Ред.*

тов — длинная и узкая полоса с кнопками; такие панели инструментов позволяют получить доступ к наиболее часто используемым командам. Подробно рассматривать все это мы не будем — и главное меню, и панели инструментов неоднократно встречались нам в других программах.

Лучше обратим внимание на правую часть главного окна. Там мы видим этакое "окошко", находящееся прямо в главном окне. Сходство с полноценным окном усиливается из-за наличия заголовка с названием и кнопками закрытия и сворачивания и даже небольшой панели инструментов. Что это такое?

Это одна из *панелей* Visual Studio. Такие панели обычно содержат всевозможные списки, например список файлов, из которых состоит разрабатываемое приложение. Эти списки, разумеется, важны, чтобы постоянно держать их перед глазами, но не настолько жизненно необходимы, чтобы отводить под них клиентскую область главного окна (о ней — чуть позже).

Панели можно перемещать с места на место, буксируя их мышью за заголовок. При этом если переместить панель к левому или правому краю главного окна, она автоматически пристыкуется к нему. Так, на рис. 2.1 изображена панель, пристыкованная к правому краю главного окна.

Пристыкованные панели могут накладываться друг на друга. В этом случае в нижней их части появится панель вкладок, перечисляющая накладываются панели; чтобы переключиться на нужную панель, достаточно будет щелкнуть на вкладке, где написано название этой панели.

Если же переместить панель в какое-либо место экрана, достаточно удаленное от краев главного окна, панель станет плавающей. Плавающая панель отображается в собственном окне, независимом от главного окна пакета.

Наконец, ненужную панель можно вообще закрыть и тем самым убрать ее с экрана. Для этого следует щелкнуть на кнопке закрытия, расположенной в правой части ее заголовка и имеющей вид крестика.

А чтобы снова вывести на экран закрытую панель, придется воспользоваться главным меню. Далее в этой книге мы рассмотрим пункты главного меню, предназначенные для вывода всех нужных нам панелей.

В нижней части главного окна находится *строка статуса* — узкая серая полоса, на которой выводится различная служебная информация. Она нам также знакома по другим программам.

Остальная часть главного окна, не занятая главным меню, панелями инструментов, обычными панелями и строкой статуса, называется *клиентской областью*. В этой области будут выводиться окна документов, в которых вводится содержимое открытых файлов, и эти окна мы рассмотрим позже.

А сейчас в клиентской области отображается так называемая *стартовая страница* Visual Studio. Она содержит, в основном, краткую справочную информацию по данному пакету. Как только мы откроем хоть один файл, стартовая страница пропадет.