



# ОГЛАВЛЕНИЕ

<b>Предисловие .....</b>	<b>13</b>
<b>Об авторах .....</b>	<b>15</b>
<b>Введение .....</b>	<b>16</b>
Для кого предназначена эта книга .....	16
Предполагаемые знания .....	17
Для кого не предназначена эта книга.....	17
Организация материала .....	18
Принятые соглашения .....	19
Требования к системе .....	19
Примеры кода .....	20
Установка примеров кода.....	20
Использование примеров кода .....	21
Благодарности .....	21
Замеченные опечатки и поддержка книги .....	22
Нам важно ваше мнение .....	22
Оставайтесь на связи .....	23
<b>Глава 1. Общие сведения и подход C++ AMP ...</b>	<b>24</b>
Что означает GPGPU? Что такое гетерогенные вычисления? ...	24
История роста производительности.....	25
Гетерогенные платформы.....	26
Архитектура ГП.....	29
Кандидаты на повышение производительности за счет распараллеливания .....	30
Технологии распараллеливания вычислений на ЦП .....	34
Векторизация.....	34
OpenMP .....	37
Система Concurrency Runtime (ConcRT) и библиотека Parallel Patterns Library.....	39
Библиотека Task Parallel Library.....	41
WARP – Windows Advanced Rasterization Platform.....	41
Технологии распараллеливания вычислений на ГП.....	41
Что необходимо для успешного распараллеливания .....	43

Подход C++ AMP .....	45
C++ AMP вводит GPGPU (и не только) в обиход.....	45
C++ AMP – это C++, а не C .....	46
Для использования C++ AMP нужны только знакомые вам инструменты .....	47
C++ AMP почти целиком реализована на уровне библиотеки....	48
C++ AMP порождает переносимые исполняемые файлы с прицелом на будущее .....	50
Резюме .....	52
<b>Глава 2. Пример: программа NBody .....</b>	<b>53</b>
Необходимые условия для запуска примера.....	53
Запуск программы NBody.....	55
Структура программы.....	59
Вычисления на ЦП .....	60
Структуры данных .....	60
Функция wWinMain .....	62
Обратный вызов OnFrameMove .....	62
Обратный вызов OnD3D11CreateDevice.....	63
Обратный вызов OnGUIEvent .....	65
Обратный вызов OnD3D11FrameRender .....	66
Классы NBody для вычислений на ЦП .....	66
Класс NBodySimpleInteractionEngine.....	67
Класс NBodySimpleSingleCore .....	67
Класс NBodySimpleMultiCore .....	68
Функция NBodySimpleInteractionEngine:: BodyBodyInteraction...	68
Вычисления с применением C++ AMP .....	70
Структуры данных .....	70
Функция CreateTasks .....	72
Классы NBody в версии для C++ AMP .....	74
Функция NBodyAmpSimple::Integrate .....	74
Функция BodyBodyInteraction .....	76
Резюме .....	77
<b>Глава 3. Основы C++ AMP .....</b>	<b>79</b>
Тип array<T, N> .....	79
accelerator и accelerator_view .....	82
index<N> .....	85
extent<N> .....	86
array_view<T, N>.....	86
parallel_for_each.....	91
Функции, помеченные признаком restrict(amp).....	94

Копирование между ЦП и ГП.....	96
Функции из математической библиотеки .....	98
Резюме .....	99
<b>Глава 4. Разбиение на блоки .....</b>	<b>100</b>
Назначение и преимущества блоков.....	101
Блочно-статическая память .....	102
Тип tiled_index<N1, N2, N3> .....	105
Преобразование простого алгоритма в блочный .....	106
Использование блочно-статической памяти .....	108
Барьеры и синхронизация .....	113
Окончательный вариант блочного алгоритма .....	116
Влияние размера блока.....	117
Выбор размера блока.....	120
Резюме .....	122
<b>Глава 5. Пример: блочный вариант программы NBody.....</b>	<b>124</b>
Насколько разбиение на блоки повышает производительность программы NBody?.....	124
Блочный алгоритм решения задачи N тел .....	126
Класс NBodyAmpTiled.....	127
Метод NBodyAmpTiled::Integrate.....	127
Визуализатор параллелизма .....	133
Выбор размера блока.....	140
Резюме .....	144
<b>Глава 6. Отладка .....</b>	<b>145</b>
Первые шаги .....	145
Выбор режима отладки: на ЦП или на ГП .....	146
Эталонный ускоритель .....	150
Основы отладки на ГП.....	154
Знакомые окна и подсказки.....	154
Панель инструментов Debug Location.....	155
Обнаружение состояний гонки .....	156
Получение информации о нитях .....	158
Маркеры нитей .....	159
Окно GPU Threads.....	159
Окно Parallel Stacks .....	161
Окно Parallel Watch .....	163
Пометка, группировка и фильтрация нитей .....	165

Дополнительные способы контроля .....	168
Заморозка и разморозка нитей .....	168
Выполнение блока до текущей позиции .....	170
Резюме .....	172
<b>Глава 7. Оптимизация.....</b>	<b>173</b>
Подход к оптимизации производительности .....	173
Анализ производительности.....	174
Измерение производительности ядра.....	175
Использование визуализатора параллелизма.....	178
Использование пакета SDK визуализатора параллелизма .....	185
Способы оптимизации доступа к памяти .....	187
Совмещение и вызовы <code>parallel_for_each</code> .....	187
Эффективное копирование данных в память ГП и обратно .....	191
Эффективный доступ к глобальной памяти ускорителя .....	198
Массив структур или структура массивов.....	202
Эффективный доступ к блочно-статической памяти.....	205
Константная память .....	210
Текстурная память.....	211
Занятость и регистры.....	211
Оптимизация вычислений .....	213
Избегайте расходящегося кода.....	213
Выбор подходящей точности .....	218
Оценка стоимости математических операций .....	220
Развертывание циклов .....	220
Барьеры синхронизации .....	222
Режимы очереди .....	226
Резюме .....	227
<b>Глава 8. Пример: программа Reduction .....</b>	<b>229</b>
Постановка задачи .....	229
Отказ от ответственности.....	230
Структура программы.....	231
Инициализация и рабочая нагрузка.....	233
Маркеры визуализатора параллелизма.....	234
Функция <code>TimeFunc()</code> .....	235
Накладные расходы .....	237
Алгоритмы на ЦП .....	238
Последовательный алгоритм .....	238
Параллельный алгоритм .....	238
Алгоритмы с использованием C++ AMP .....	239
Простой алгоритм .....	240

Простой алгоритм с <code>array_view</code> .....	242
Простой оптимизированный алгоритм .....	244
Наивный блочный алгоритм .....	246
Блочный алгоритм с разделяемой памятью .....	248
Минимизация расхождения .....	254
Устранение конфликтов банков .....	256
Уменьшение числа простаивающих нитей .....	257
Развертывание цикла .....	258
Каскадная редукция .....	263
Каскадная редукция с развертыванием цикла .....	265
Резюме .....	266
<b>Глава 9. Работа с несколькими ускорителями ... 268</b>	
Выбор ускорителей .....	269
Перебор ускорителей .....	269
Ускоритель по умолчанию .....	272
Использование нескольких ГП .....	274
Обмен данными между ускорителями .....	279
Динамическое балансирование нагрузки .....	285
Комбинированный параллелизм .....	288
ЦП как последнее средство .....	290
Резюме .....	292
<b>Глава 10. Пример: программа Cartoonizer .... 294</b>	
Необходимые условия .....	295
Запуск программы .....	295
Структура программы .....	299
Конвейер .....	301
Структуры данных .....	301
Метод <code>CartoonizerDlg::OnBnClickedButtonStart()</code> .....	303
Класс <code>ImagePipeline</code> .....	304
Стадия мультипликации .....	309
Класс <code>ImageCartoonizerAgent</code> .....	309
Реализации интерфейса <code>IFrameProcessor</code> .....	312
Использование нескольких ускорителей, совместимых с C++ AMP .....	321
Класс <code>FrameProcessorAmpMulti</code> .....	321
Разветвленный конвейер .....	324
Класс <code>ImageCartoonizerAgentParallel</code> .....	325
Производительность мультипликатора .....	328
Резюме .....	331

**Глава 11. Интероперабельность с графикой ... 333**

Основы .....	334
Типы norm и unorm .....	334
Типы коротких векторов .....	336
Тип texture<T, N> .....	340
Сравнение текстур и массивов.....	349
Использование текстур и коротких векторов .....	351
Встроенные функции HLSL .....	355
Интероперабельность с DirectX .....	356
Интероперабельность представления ускорителя и устройства Direct3D.....	357
Интероперабельность array и буфера Direct3D .....	358
Интероперабельность texture и текстурного ресурса Direct3D .....	359
Практическое использование интероперабельности с графикой .....	363
Резюме .....	365

**Глава 12. Советы, хитрости и рекомендации... 367**

Решение проблемы несоответствия размеру блока.....	368
Дополнение до кратного размеру блока.....	369
Отсечение блоков .....	371
Сравнение разных подходов .....	375
Инициализация массивов.....	376
Объекты-функции и лямбда-выражения .....	377
Атомарные операции.....	378
Дополнительные возможности C++ AMP Features в Windows 8.....	382
Обнаружение таймаутов и восстановление .....	384
Предотвращение TDR .....	385
Отключение TDR в Windows 8.....	386
Обнаружение TDR и восстановление.....	387
Поддержка вычислений с двойной точностью.....	388
Ограниченная поддержка двойной точности .....	388
Полная поддержка двойной точности.....	389
Отладка в Windows 7 .....	389
Конфигурирование удаленной машины.....	390
Конфигурирование проекта .....	390
Развертывание и отладка проекта .....	392
Дополнительные отладочные функции .....	392
Развертывание .....	393

Развертывание приложения .....	393
Запуск C++ AMP на сервере .....	394
C++ AMP и приложения для Windows 8 в магазине Windows Store .....	397
Использование C++ AMP из управляемого кода .....	397
Из приложения .NET, приложения для Windows 7, Windows Store или библиотеки.....	397
Из приложения для C++ CLR.....	398
Из проекта для C++ CLR .....	398
Резюме .....	399
<b>Приложение. Другие ресурсы .....</b>	<b>400</b>
Другие публикации авторов этой книги .....	400
Сетевые ресурсы Microsoft .....	400
Скачивайте руководства по C++ AMP.....	401
Исходный код и поддержка.....	401
Обучение .....	402
<b>Предметный указатель .....</b>	<b>403</b>



# ПРЕДИСЛОВИЕ

На протяжении большей части истории развития компьютеров мы были свидетелями экспоненциального роста производительности скалярных процессоров. Но теперь этому пришел конец. Мы находимся в начале эры гетерогенных параллельных вычислений. В мире, где всем приложениям нужно больше вычислительных мощностей, а мощность всех вычислительных систем – от мобильных устройств до облачных кластеров – ограничена, будущие вычислительные платформы просто обязаны стать гетерогенными. Так, самые мощные в мире суперкомпьютеры все чаще строятся как кластеры на базе комбинации ЦП и ГП (графический процессор). И хотя программные интерфейсы первого поколения, такие как CUDA и OpenCL, позволили приступить к созданию новых библиотек и приложений для подобных систем, выявилась настоятельная необходимость в средствах, которые обеспечили бы гораздо более высокую продуктивность при разработке гетерогенного параллельного ПО.

Основная трудность заключается в том, что любой программный интерфейс, повышающий продуктивность в этой области, должен еще и предоставлять программисту средства, необходимые для достижения требуемой производительности. Интерфейс C++ AMP, предложенный Microsoft, – крупный шаг на пути решения этой проблемы. Это простое и элегантное расширение языка C++, призванное устранить два самых заметных недостатка прежних интерфейсов. Во-первых, прежние подходы плохо сочетаются с общепринятой практикой построения программ на C++, так как модели параллельного программирования на основе ядра трудно увязать с организацией классов в приложении. Во-вторых, унаследованная от C индексация динамически выделяемых массивов затрудняет управление локальностью.

Я с радостью обнаружил, что для решения первой проблемы C++ AMP поддерживает в параллельном коде циклические конструкции и объектно-ориентированные средства C++, а для решения второй – конструкцию *array\_view*. Подход на основе *array\_view* – это задел на



будущее, позволяющий уже сейчас создавать приложения, которые сумеют воспользоваться всеми преимуществами грядущих архитектур с объединенным адресным пространством. Многие программисты, имеющие опыт работы с CUDA и OpenCL, находят программирование в стиле C++ AMP интересным, элегантным и эффективным.

Не менее важен, на мой взгляд, и тот факт, что интерфейс C++ AMP открывает возможности для широкого распространения многочисленных новаторских способов преобразования программы компилятором, в частности, выбора размещения данных и гранулярности потоков. Он также позволяет реализовать оптимизацию перемещения данных во время выполнения. Благодаря таким усовершенствованиям продуктивность программиста резко возрастет.

В настоящее время интерфейс C++ AMP реализован только в Windows, но спецификация открыта и, скорее всего, будет реализована и на других платформах. Заложенный в C++ AMP потенциал раскроется по-настоящему, когда поставщики других платформ начнут предлагать его реализацию (если это произойдет).

Издание этой книги знаменует важный этап в развитии гетерогенных параллельных вычислений. Я ожидаю, что теперь количество разработчиков, способных продуктивно работать в такой среде, существенно увеличится. Я горжусь выпавшей мне честью написать предисловие к этой книге и принять участие в этом движении. И, что еще более важно, отдаю должное группе инженеров Microsoft, создавших C++ AMP и тем самым поспособствовавших движению в этом направлении.

*Вэнь-Мей Ху  
Профессор, кафедра имени Сандерса (AMD),  
факультет электротехники и вычислительной техники  
университета штата Иллинойс в Урбане и Шампейне,  
технический директор компании MulticoreWare, Inc.*



## ОБ АВТОРАХ

**Эйд Миллер** в настоящее время работает главным системным архитектором в компании Microsoft Studios. Ранее занимал различные должности в Microsoft, в том числе являлся руководителем проекта по платформам «больших данных», над которым работал совместно с группой разработки Windows HPC Server. Был также руководителем команд гибкой разработки в группе «Patterns & Practices». Интересуется главным образом параллельными и распределенными вычислениями, а также методами улучшения качества разработки ПО за счет правильной организации работ.

Эйд – соавтор книг «Parallel Programming with Microsoft .NET» и «Parallel Programming with Microsoft Visual C++». Он много пишет и выступает на тему параллельных вычислений и своего опыта гибкой разработки ПО в Microsoft и других компаниях.

**Кэйт Грегори** программирует на C++ уже больше двадцати лет и хорошо известна как преподаватель, лектор и автор. Управление проектами, обучение, составление технической документации и выступления на различных мероприятиях отнимают большую часть ее времени, но тем не менее она умудряется писать код каждую неделю. Кэйт – автор более десятка книг, она регулярно выступает на конференциях DevTeach, TechEd (в США, Европе и Африке), TechDays и других. Кэйт удостоена звания C++ MVP, является спонсором-основателем группы пользователей .NET в Торонто, основателем группы пользователей в Восточном Торонто и преподает по временному контракту в университете Трент в Питерборо. С января 2002 года является региональным директором Microsoft в Торонто, а в январе 2004 года удостоена звания Microsoft Most Valuable Professional по Visual C++. В июне 2005 стала региональным директором года, а в феврале 2011 – Visual C++ MVP 2010 года. Ее компания, Gregory Consulting Limited, расположенная в сельской местности в районе озера Онтарио, помогает заказчикам осваивать новые технологии и адаптироваться к изменяющимся условиям ведения бизнеса.



# ВВЕДЕНИЕ

C++ Accelerated Massive Parallelism (C++ AMP) – разработанная корпорацией Microsoft технология ускорения написанных на C++ приложений за счет исполнения кода на оборудовании с распараллеливанием по данным, например, на графических процессорах (ГП). Она рассчитана не только на современную параллельную аппаратуру в виде ГП и APU (Accelerated Processing Unit – ускоренный процессорный элемент), но и на поддержку будущего оборудования – с целью защитить вложения в разработку кода. Спецификация C++ AMP открыта. Microsoft реализовала ее поверх DirectX, обеспечив тем самым переносимость на различные аппаратные платформы. Но другие реализации могут опираться на иные технологии, поскольку в спецификации DirectX нигде не упоминается.

Модель программирования в C++ AMP основана на библиотеке, устроенной по образцу STL, и двух расширениях языка C++, интегрированных в компилятор Visual C++ 2012. Она в полной мере поддерживается инструментами Visual Studio, в том числе IntelliSense, отладчиком и профилировщиком. Благодаря C++ AMP свойственная гетерогенному оборудованию производительность становится доступна широким кругам программистов.

В этой книге показано, как воспользоваться всеми преимуществами C++ AMP в собственных приложениях. Помимо описания различных черт C++ AMP, приведены примеры различных подходов к реализации некоторых общеупотребительных алгоритмов в реальных приложениях. Полный код примеров, равно как и код, представленный в отдельных главах, можно скачать и изучить.

## Для кого предназначена эта книга

Цель этой книги – помочь программистам на C++ в освоении технологии C++ AMP, от базовых концепций до более сложных средств. Если вас интересует, как воспользоваться преимуществами гетерогенного оборудования для повышения производительности существующих функций приложения или добавления совершенно новых,

которые раньше не удавалось реализовать из-за ограничений на быстродействие, то эта книга для вас.

Прочитав ее, вы будете лучше понимать, как и где лучше всего применить C++ в своем приложении. Вы узнаете, как работать с имеющимися в Microsoft Visual Studio 2012 средствами отладки и профилирования для поиска ошибок и оптимизации производительности.

## **Предполагаемые знания**

Предполагается, что читатель знаком с разработкой ПО в среде Windows C++, с концепциями объектно-ориентированного программирования и со стандартной библиотекой C++ (которую часто называют STL по названию ее предтечи – библиотеки Standard Template Library). Знакомство с общими понятиями параллельной обработки было бы не лишним, но не обязательно. В некоторых примерах используется DirectX, но ни для их использования, ни для понимания входящего в них кода C++ AMP, опыт работы с DirectX не требуется.

Общие сведения о языке C++ можно получить из книги Бьярна Страуструпа «Язык программирования C++» (Бином, 2012). В настоящей книге используются многие языковые и библиотечные средства, появившиеся только в стандарте C++11 и пока еще не нашедшие отражения в печатных изданиях. Хороший обзор имеется в работе Скотта Мейерса «Presentation Materials: Overview of the New C++ (C++11)», которую можно приобрести через Интернет у издательства Artima Developer по адресу [http://www.artima.com/shop/overview\\_of\\_the\\_new\\_cpp](http://www.artima.com/shop/overview_of_the_new_cpp). Хорошее введение в стандартную библиотеку приведено в книге Nicolai M. Josuttis's The C++ Standard Library: A Tutorial and Reference (2nd Edition) (Addison-Wesley Professional, 2012).

В примерах также нередко используются библиотеки Parallel Patterns Library и Asynchronous Agents Library. Та и другая неплохо описаны в книге «Parallel Programming with Microsoft Visual C++» (Microsoft Press, 2011) Колина Кэмпбелла (Colin Campbell) и Эйда Миллера (Ade Miller). Бесплатная версия этой книги имеется также на сайте MSDN по адресу <http://msdn.microsoft.com/en-us/library/gg675934.aspx>.

## **Для кого не предназначена эта книга**

Эта книга не предназначена для изучения языка C++ или его стандартной библиотеки. Предполагается, что читатель владеет тем и

другим на рабочем уровне. Книга также не является общим введением в параллельное или многопоточное программирование. Если вы не знакомы с этими темами, рекомендуем сначала прочитать книги, упомянутые в предыдущем разделе.

## Организация материала

Книга состоит из 12 глав, каждая из которых посвящена одной из сторон программирования с помощью C++ AMP. Кроме того, в книгу включены три примера, демонстрирующих применение основных возможностей C++ AMP в реальных программах. Полный код примеров и фрагменты из других глав можно скачать с сайта CodePlex.

<b>Глава 1</b> «Общие сведения и подход C++ AMP»	Введение в графические процессоры, гетерогенные вычисления, организацию параллелизма на ЦП. Обзор того, как C++ AMP позволяет задействовать всю мощь современных гетерогенных систем.
<b>Глава 2</b> «Пример: программа NBody»	Моделирование задачи N тел с помощью C++ AMP
<b>Глава 3</b> «Основы C++ AMP»	Краткое описание библиотеки и изменений в языке, составляющих C++ AMP, а также некоторых правил, которым должна следовать программа.
<b>Глава 4</b> «Разбиение на блоки»	Рассматривается вопрос о разбиении вычисления на группы потоков, называемые блоками (tile), которые имеют общий доступ к сверхбыстрому программируемому кэшу.
<b>Глава 5</b> «Пример: блочный вариант программы NBody»	Описывается вариант программы NBody из главы 2 с использованием разбиения на блоки.
<b>Глава 6</b> «Отладка»	Обзор технических приемов и средств отладки приложений на базе C++ AMP в Visual Studio.
<b>Глава 7</b> «Оптимизация»	Дополнительные сведения о факторах, которые влияют на производительность приложений на базе C++ AMP, и о том, как добиться максимального быстродействия.
<b>Глава 8</b> «Пример: программа Reduction»	Демонстрируются различные подходы к реализации простого вычисления и их влияние на производительность.
<b>Глава 9</b> «Работа с несколькими ускорителями»	Как использовать несколько ГП для достижения максимальной производительности. Рассматривается комбинированный параллелизм и применение ЦП для обеспечения максимального эффективного использования ГП.

<b>Глава 10</b> «Пример: программа Cartoonizer»	Комплексный пример, в котором сочетается параллелизм на уровне ЦП с параллелизмом в духе C++ AMP и поддержкой нескольких ускорителей.
<b>Глава 11</b> «Интероперабельность с графикой»	Использование C++ AMP в сочетании с DirectX.
<b>Глава 12</b> «Советы, хитрости и рекомендации»	Описываются менее распространенные ситуации и среды и объясняется, как разрешать некоторые типичные проблемы.
<b>Приложение</b> «Другие ресурсы»	Онлайновые ресурсы, техническая поддержка и учебные курсы для желающих обогатить свои знания о C++ AMP.

## Принятые соглашения

В этой книге применяются следующие соглашения.

- Текст, заключенный в рамочку и помеченный, например, словом «**Примечание**», содержит дополнительные сведения или описание альтернативных способов выполнить действие.
- Знак + между названиями двух клавиш, например **Alt+Tab**, означает, что эти клавиши нужно нажать одновременно.
- Вертикальная черта между двумя или более пунктами меню (например, **File | Close**), означает, что сначала нужно выбрать первое меню или пункт меню, затем следующее и т. д.

## Требования к системе

Для сборки и запуска примеров из этой книги необходимо следующее оборудование и программное обеспечение.

- Операционная система Microsoft Windows 7 с пакетом обновлений Service Pack 1 или Windows 8 (x86 или x64). Примеры должны также собираться и запускаться в системах Windows Server 2008 R2 (x64) и Windows Server 2012 (x64), но тестирование для них не производилось.
- Любое издание Visual Studio 2012 (для профилирования, описанного в главах 7 и 8, необходимо издание Professional или Ultimate).
- Для сборки программы NBody понадобится DirectX SDK (версия от июня 2010).

- Компьютер, оснащенный процессором с тактовой частотой 1,6 ГГц или выше. Рекомендуется четырехъядерный процессор.
- Оперативная память объемом не ниже 1 ГБ (для 32-разрядных ОС) или 2 ГБ (для 64-разрядных ОС).
- 10 ГБ свободного места на жестком диске (для установки Visual Studio 2012).
- Жесткий диск с частотой вращения 5400 об/мин.
- Видеокарта с поддержкой DirectX 11 (для примеров использования C++ AMP) и монитор с разрешением 1024×768 или выше (для Visual Studio 2012).
- Накопитель DVD-ROM (если Visual Studio 2012 устанавливается с DVD-диска).
- Соединение с Интернетом для скачивания ПО и примеров.

## Примеры кода

Почти во всех главах имеются примеры, позволяющие интерактивно осваивать изложенный в тексте новый материал. Исходный код примеров можно скачать в виде ZIP-файла со страницы <http://go.microsoft.com/fwlink/?Linkid=260980>.

**Примечание.** Помимо примеров кода, вы должны установить Visual Studio 2012 и DirectX SDK (версия от июня 2010). Устанавливайте последнюю доступную версию каждого продукта.

## Установка примеров кода

Для установки примеров кода выполните следующие действия.

1. Скачайте ZIP-файл с исходным кодом со страницы книги на сайте CodePlex: <http://ampbook.codeplex.com/>. Последняя рекомендуемая версия находится на вкладке Downloads.
2. Прочитайте лицензионное соглашение с конечным пользователем (если будет предложено). Если вы согласны с его условиями, отметьте флажок **Ассерт** и нажмите кнопку **Next**.
3. Распакуйте архив в любую папку и откройте файл `BookSamples.sln` в Visual Studio 2012.

**Примечание.** Если лицензионное соглашение не появилось, с ним можно ознакомиться по адресу <http://ampbook.codeplex.com/license>. Копия соглашения включена также в архив с исходным кодом.

## Использование примеров кода

После распаковки архива создается папка Samples, содержащая три подпапки.

- **CaseStudies:** содержит все три примера из глав 2, 8 и 10. Каждый пример находится в отдельной папке.
  - **NBody:** гравитационная модель для задачи N тел.
  - **Reduction:** несколько реализаций алгоритма редукции с целью демонстрации различных подходов к повышению производительности.
  - **Cartoonizer:** приложение для обработки изображений, которое мультиплицирует изображения, загруженные с диска или снятые видеокамерой.
- **Chapter 4, 7, 9, 11, 12:** содержат код примеров из соответствующей главы.
- **ShowAmpDevices:** простая утилита для вывода списка всех поддерживающих C++ AMP устройств в данном компьютере.

Папка верхнего уровня Samples содержит файл решения для Visual Studio 2012, BookSamples.sln, включающий все перечисленные выше проекты. Решение должно компилироваться без ошибок и предупреждений в конфигурациях Debug и Release для платформ Win32 и x64. Для каждого проекта имеется также собственный файл решения – на случай, если вы захотите загрузить их по отдельности.

## Благодарности

Любая книга – плод усилий не одного человека. У этой книги два автора, но у нас было еще и много помощников. Мы выражаем благодарность группе C++ AMP в Microsoft, сотрудники которой далеко не ограничивались рецензированием черновиков и ответами на многочисленные вопросы: Амиту Агарвалу (Amit Agarwal), Дэвиду Кэллахану (David Callahan), Чарльзу Фу (Charles Fu), Джерри Хиггинсу (Jerry Higgins), Йосси Леванони (Yossi Levanoni), Дону Маккреди (Don McCrady), Лукашу Мендакевичу (Łukasz Mendakiewicz), Дэниэлу Моту (Daniel Moth), Бхарату Майсору Нанджундаппа (Bharath Mysore Nanjundappa), Пуджа Нагпалу (Pooja Nagpal), Джеймсу Рэппу (James Rapp), Саймону Выбрански (Simon Wybranski), Линь Ли Чжану (Lingli Zhang) и Вэй Рон Чжу (Weirong Zhu) (корпорация Microsoft).