

Содержание

Введение	7
Глава 1. Работа с типами данных	11
Совет 1. Используйте свойства вместо данных-членов, открытых для доступа	11
Совет 2. Отдавайте предпочтение неявным свойствам перед изменяющимися данными. .	17
Совет 3. Отдавайте предпочтение неизменяемости для типов значений	20
Совет 4. Проводите различие между типами значений и ссылочными типами.....	25
Совет 5. Удостоверьтесь, что 0 является допустимым состоянием для типов значений....	30
Совет 6. Удостоверьтесь в том, что свойства ведут себя подобно данным	33
Совет 7. Ограничивайте область видимости типов с использованием кортежей	38
Совет 8. Определяйте локальные методы на анонимных типах.....	42
Совет 9. Освойте взаимосвязи между разными концепциями равенства	47
Совет 10. Исследуйте ловушки метода GetHashCode ()	54
Глава 2. Проектирование API-интерфейсов	61
Совет 11. Избегайте присутствия операций преобразования в своих API-интерфейсах....	61
Совет 12. Используйте необязательные параметры для минимизации количества перегруженных версий методов.....	65
Совет 13. Ограничивайте видимость своих типов.....	68
Совет 14. Отдавайте предпочтение определению и реализации интерфейсов перед наследованием.....	71
Совет 15. Осмыслите, чем методы интерфейса отличаются от виртуальных методов...	78
Совет 16. Реализуйте паттерн “Событие” для уведомлений	82
Совет 17. Избегайте возвращения ссылок на объекты внутренних классов	87
Совет 18. Отдавайте предпочтение переопределениям перед обработчиками событий...	90
Совет 19. Избегайте перегрузки методов, определенных в базовых классах	93
Совет 20. Выясните, как события увеличивают связывание между объектами во время выполнения.....	97
Совет 21. Объявляйте только неvirtуальные события.....	99
Совет 22. Создавайте группы методов, которые являются ясными, минимальными и полными	103
Совет 23. Предоставляйте частичные методы в частичных классах для конструкторов, методов изменения и обработчиков событий	109
Совет 24. Избегайте реализации интерфейса ICloneable, поскольку он ограничивает варианты проектных решений	113
Совет 25. Ограничивайте параметры-массивы массивами params	117
Совет 26. Сделайте возможным немедленное сообщение об ошибках в итераторных и асинхронных методах с использованием локальных функций	120

Глава 3. Асинхронное программирование на основе задач	125
Совет 27. Используйте асинхронные методы для выполнения асинхронной работы....	125
Совет 28. Никогда не создавайте методы <code>async void</code>	129
Совет 29. Избегайте объединения синхронных и асинхронных методов	134
Совет 30. Используйте асинхронные методы, чтобы избежать размещения потоков и переключения контекста	138
Совет 31. Избегайте ненужной маршализации контекста	140
Совет 32. Формируйте асинхронную работу с использованием объектов задач	143
Совет 33. Обдумайте реализацию протокола отмены задачи	148
Совет 34. Кешируйте обобщенные возвращаемые типы в асинхронном коде	154
Глава 4. Параллельная обработка	157
Совет 35. Выясните, как PLINQ реализует параллельные алгоритмы	157
Совет 36. Создавайте параллельные алгоритмы с учетом исключений	168
Совет 37. Используйте пул потоков вместо создания потоков	172
Совет 38. Используйте класс <code>BackgroundWorker</code> для взаимодействия между потоками	177
Совет 39. Освойте вызовы между потоками в средах XAML	181
Совет 40. Используйте оператор <code>lock</code> в качестве первого кандидата для синхронизации	188
Совет 41. Используйте для дескрипторов блокировок как можно меньшую область действия	194
Совет 42. Избегайте вызова неизвестного кода в синхронизированных секциях	197
Глава 5. Динамическое программирование	201
Совет 43. Изучите достоинства и недостатки динамической типизации	201
Совет 44. Применяйте динамическую типизацию, чтобы использовать типы времени выполнения для параметров обобщенных типов	209
Совет 45. Используйте класс <code>DynamicObject</code> или интерфейс <code>IDynamicMetaObjectProvider</code> для динамических типов, управляемых данными	212
Совет 46. Выясните, как использовать API-интерфейсы выражений	220
Совет 47. Сводите к минимуму наличие динамических объектов в открытых API-интерфейсах	226
Глава 6. Участие в глобальном сообществе C#	231
Совет 48. Ищите лучший, а не самый популярный ответ	231
Совет 49. Участвуйте в разработке спецификаций и кода	233
Совет 50. Обдумайте автоматизацию практик с помощью анализаторов	234
Предметный указатель	236