

Предисловие

Благодарности

Контрольные списки

Часть I Основы разработки ПО

1 Добро пожаловать в мир конструирования ПО!

1.1. Что такое конструирование ПО?

1.2. Почему конструирование ПО так важно?

1.3. Как читать эту книгу

2 Метафоры, позволяющие лучше понять разработку ПО

2.1. Важность метафор

2.2. Как использовать метафоры?

2.3. Популярные метафоры, характеризующие разработку ПО

3 Семь раз отмерь, один раз отрежь: предварительные условия

3.1. Важность выполнения предварительных условий

3.2. Определите тип ПО, над которым вы работаете

3.3. Предварительные условия, связанные с определением проблемы

3.4. Предварительные условия, связанные с выработкой требований

3.5. Предварительные условия, связанные с разработкой архитектуры

3.6. Сколько времени следует посвятить выполнению предварительных условий?

4 Основные решения, которые приходится принимать при конструировании

4.1. Выбор языка программирования

4.2. Конвенции программирования

4.3. Волны развития технологий

4.4. Выбор основных методик конструирования

Часть II Высококачественный код

5 Проектирование при конструировании

5.1. Проблемы, связанные с проектированием ПО

5.2. Основные концепции проектирования

5.3. Компоненты проектирования: эвристические принципы

5.4. Методики проектирования

5.5. Комментарии по поводу популярных методологий

6 Классы

6.1. Основы классов: абстрактные типы данных

6.2. Качественные интерфейсы классов

6.3. Вопросы проектирования и реализации

6.4. Разумные причины создания классов

6.5. Аспекты, специфические для языков

6.6. Следующий уровень: пакеты классов

7 Высококачественные методы

7.1. Разумные причины создания методов

7.2. Проектирование на уровне методов

7.3. Удачные имена методов

7.4. Насколько объемным может быть метод?

7.5. Советы по использованию параметров методов

7.6. Отдельные соображения по использованию функций

7.7. Методы-макросы и встраиваемые методы

8 Защитное программирование

8.1. Защита программы от неправильных входных данных

8.2. Утверждения

8.3. Способы обработки ошибок

8.4. Исключения

8.5. Изоляция повреждений, вызванных ошибками

8.6. Отладочные средства

8.7. Доля защитного программирования в промышленной версии

8.8. Защита от защитного программирования

- 9 Процесс программирования с псевдокодом
  - 9.1. Этапы создания классов и методов
  - 9.2. Псевдокод для профи
  - 9.3. Конструирование методов с использованием ППП
  - 9.4. Альтернативы ППП
- Часть III Переменные
- 10 Общие принципы использования переменных
  - 10.1. Что вы знаете о данных?
  - 10.2. Грамотное объявление переменных
  - 10.3. Принципы инициализации переменных
  - 10.4. Область видимости
  - 10.5. Персистентность
  - 10.6. Время связывания
  - 10.7. Связь между типами данных и управляющими структурами
  - 10.8. Единственность цели каждой переменной
- 11 Сила имен переменных
  - 11.1. Общие принципы выбора имен переменных
  - 11.2. Именованые конкретные типы данных
  - 11.3. Сила конвенций именования
  - 11.4. Неформальные конвенции именования
  - 11.5. Стандартизованные префиксы
  - 11.6. Грамотное сокращение имен переменных
  - 11.7. Имена, которых следует избегать
- 12 Основные типы данных
  - 12.1. Числа в общем
  - 12.2. Целые числа
  - 12.3. Числа с плавающей запятой
  - 12.4. Символы и строки
  - 12.5. Логические переменные
  - 12.6. Перечислимые типы
  - 12.7. Именованные константы
  - 12.8. Массивы
  - 12.9. Создание собственных типов данных (псевдонимы)
- 13 Нестандартные типы данных
  - 13.1. Структуры
  - 13.2. Указатели
  - 13.3. Глобальные данные
- Часть IV Операторы
- 14 Организация последовательного кода
  - 14.1. Операторы, следующие в определенном порядке
  - 14.2. Операторы, следующие в произвольном порядке
- 15 Условные операторы
  - 15.1. Операторы if
  - 15.2. Операторы case
- 16 Циклы
  - 16.1. Выбор типа цикла
  - 16.2. Управление циклом
  - 16.3. Простое создание цикла — изнутри наружу
  - 16.4. Соответствие между циклами и массивами
- 17 Нестандартные управляющие структуры
  - 17.1. Множественные возвраты из метода
  - 17.2. Рекурсия
  - 17.3. Оператор goto
  - 17.4. Перспективы нестандартных управляющих структур
- 18 Табличные методы

- 18.1. Основные вопросы применения табличных методов
- 18.2. Таблицы с прямым доступом
- 18.3. Таблицы с индексированным доступом
- 18.4. Таблицы со ступенчатым доступом
- 18.5. Другие примеры табличного поиска
- 19 Общие вопросы управления
  - 19.1. Логические выражения
  - 19.2. Составные операторы (блоки)
  - 19.3. Пустые выражения
  - 19.4. Укращение опасно глубокой вложенности
  - 19.5. Основа программирования: структурное программирование
  - 19.6. Управляющие структуры и сложность
- Часть V Усовершенствование кода
- 20 Качество ПО
  - 20.1. Характеристики качества ПО
  - 20.2. Методики повышения качества ПО
  - 20.3. Относительная эффективность методик контроля качества ПО
  - 20.4. Когда выполнять контроль качества ПО?
  - 20.5. Главный Закон Контроля Качества ПО
- 21 Совместное конструирование
  - 21.1. Обзор методик совместной разработки ПО
  - 21.2. Парное программирование
  - 21.3. Формальные инспекции
  - 21.4. Другие методики совместной разработки ПО
  - 21.5. Сравнение методик совместного конструирования
- 22 Тестирование, выполняемое разработчиками
  - 22.1. Тестирование, выполняемое разработчиками, и качество ПО
  - 22.2. Рекомендуемый подход к тестированию, выполняемому разработчиками
  - 22.3. Приемы тестирования
  - 22.4. Типичные ошибки
  - 22.5. Инструменты тестирования
  - 22.6. Оптимизация процесса тестирования
  - 22.7. Протоколы тестирования
- 23 Отладка
  - 23.1. Общие вопросы отладки
  - 23.2. Поиск дефекта
  - 23.3. Устранение дефекта
  - 23.4. Психологические аспекты отладки
  - 23.5. Инструменты отладки — очевидные и не очень
- 24 Рефакторинг
  - 24.1. Виды эволюции ПО
  - 24.2. Введение в рефакторинг
  - 24.3. Отдельные виды рефакторинга
  - 24.4. Безопасный рефакторинг
  - 24.5. Стратегии рефакторинга
- 25 Стратегии оптимизации кода
  - 25.1. Общее обсуждение производительности ПО
  - 25.2. Введение в оптимизацию кода
  - 25.3. Где искать жир и патоку?
  - 25.4. Оценка производительности
  - 25.5. Итерация
  - 25.6. Подход к оптимизации кода: резюме
- 26 Методики оптимизации кода
  - 26.1. Логика
  - 26.2. Циклы

- 26.3. Изменения типов данных
- 26.4. Выражения
- 26.5. Методы
- 26.6. Переписывание кода на низкоуровневом языке
- 26.7. Если что-то одно изменяется, что-то другое всегда остается постоянным
- Часть VI Системные вопросы
- 27 Как размер программы влияет на конструирование
  - 27.1. Взаимодействие и размер
  - 27.2. Диапазон размеров проектов
  - 27.3. Влияние размера проекта на возникновение ошибок
  - 27.4. Влияние размера проекта на производительность
  - 27.5. Влияние размера проекта на процесс разработки
- 28 Управление конструированием
  - 28.1. Поощрение хорошего кодирования
  - 28.2. Управление конфигурацией
  - 28.3. Оценка графика конструирования
  - 28.4. Измерения
  - 28.5. Гуманное отношение к программистам
  - 28.6. Управление менеджером
- 29 Интеграция
  - 29.1. Важность выбора подхода к интеграции
  - 29.2. Частота интеграции — поэтапная или инкрементная?
  - 29.3. Стратегии инкрементной интеграции
  - 29.4. Ежедневная сборка и дымовые тесты
- 30 Инструменты программирования
  - 30.1. Инструменты для проектирования
  - 30.2. Инструменты для работы с исходным кодом
  - 30.3. Инструменты для работы с исполняемым кодом
  - 30.4. Инструменты и среды
  - 30.5. Создание собственного программного инструментария
  - 30.6. Волшебная страна инструментальных средств
- Часть VII Мастерство программирования
- 31 Форматирование и стиль
  - 31.1. Основные принципы форматирования
  - 31.2. Способы форматирования
  - 31.3. Стили форматирования
  - 31.4. Форматирование управляющих структур
  - 31.5. Форматирование отдельных операторов
  - 31.6. Размещение комментариев
  - 31.7. Размещение методов
  - 31.8. Форматирование классов
- 32 Самодокументирующийся код
  - 32.1. Внешняя документация
  - 32.2. Стиль программирования как вид документации
  - 32.3. Комментировать или не комментировать?
  - 32.4. Советы по эффективному комментированию
  - 32.5. Методики комментирования
  - 32.6. Стандарты IEEE
- 33 Личность
  - 33.1. При чем тут характер?
  - 33.2. Интеллект и скромность
  - 33.3. Любопытство
  - 33.4. Профессиональная честность
  - 33.5. Общение и сотрудничество
  - 33.6. Творчество и дисциплина

- 33.7. Лень
- 33.8. Свойства, которые менее важны, чем кажется
- 33.9. Привычки
- 34 Основы мастерства
- 34.1. Боритесь со сложностью
- 34.2. Анализируйте процесс разработки
- 34.3. Пишите программы в первую очередь для людей и лишь во вторую — для компьютеров
- 34.4. Программируйте с использованием языка, а не на языке
- 34.5. Концентрируйте внимание с помощью соглашений
- 34.6. Программируйте в терминах проблемной области
- 34.7. Опасайтесь падающих камней
- 34.8. Итерируйте, итерируйте и итерируйте
- 34.9. И да отделена будет религия от разработки ПО
- 35 Где искать дополнительную информацию
- 35.1. Информация о конструировании ПО
- 35.2. Не связанные с конструированием темы
- 35.3. Периодические издания
- 35.4. Список литературы для разработчика ПО
- 35.5. Профессиональные ассоциации
- Библиография
- Предметный указатель
- Об авторе
- Стив МакКоннелл
- Стив МакКоннелл