

Оглавление

| | |
|--|----|
| Об авторе | 13 |
| Благодарности | 14 |
| От издательства | 14 |
| Введение | 15 |
| Что нового в четвертом издании..... | 17 |
| Целевая аудитория..... | 17 |
| Подход, использованный в этой книге..... | 18 |
| Соглашения, принятые в книге..... | 19 |
| Глава 1. Введение в объектно-ориентированные концепции | 20 |
| Фундаментальные концепции..... | 20 |
| Объекты и унаследованные системы..... | 21 |
| Процедурное программирование в сравнении с объектно-ориентированным..... | 23 |
| Переход с процедурной разработки на объектно-ориентированную..... | 27 |
| Процедурное программирование..... | 27 |
| Объектно-ориентированное программирование..... | 27 |
| Что такое объект..... | 28 |
| Данные объектов..... | 28 |
| Поведения объектов..... | 28 |
| Что такое класс..... | 32 |
| Создание объектов..... | 33 |
| Атрибуты..... | 35 |
| Методы..... | 35 |
| Сообщения..... | 35 |
| Использование диаграмм классов в качестве визуального средства..... | 36 |
| Инкапсуляция и скрытие данных..... | 36 |
| Интерфейсы..... | 37 |
| Реализации..... | 38 |
| Реальный пример парадигмы «интерфейс/реализация»..... | 38 |
| Модель парадигмы «интерфейс/реализация»..... | 39 |
| Наследование..... | 40 |
| Суперклассы и подклассы..... | 41 |
| Абстрагирование..... | 42 |
| Отношения «является экземпляром»..... | 43 |
| Полиморфизм..... | 44 |
| Композиция..... | 47 |
| Абстрагирование..... | 47 |
| Отношения «содержит как часть»..... | 48 |
| Резюме..... | 48 |
| Примеры кода, использованного в этой главе..... | 48 |

| | |
|--|----|
| Глава 2. Как мыслить объектно | 51 |
| Разница между интерфейсом и реализацией | 52 |
| Интерфейс | 54 |
| Реализация | 54 |
| Пример интерфейса/реализации | 55 |
| Использование абстрактного мышления при проектировании классов | 59 |
| Обеспечение самого минимального интерфейса пользователя из возможных | 62 |
| Определение пользователей | 63 |
| Поведения объектов | 64 |
| Ограничения, налагаемые средой | 64 |
| Определение открытых интерфейсов | 64 |
| Определение реализации | 65 |
| Резюме | 66 |
| Ссылки | 66 |
| Глава 3. Продвинутое объектно-ориентированные концепции | 67 |
| Конструкторы | 67 |
| Когда осуществляется вызов конструктора | 68 |
| Что находится внутри конструктора | 68 |
| Конструктор по умолчанию | 68 |
| Использование множественных конструкторов | 69 |
| Перегрузка методов | 70 |
| Использование UML для моделирования классов | 71 |
| Как сконструирован суперкласс | 73 |
| Проектирование конструкторов | 73 |
| Обработка ошибок | 74 |
| Игнорирование проблем | 74 |
| Проверка на предмет проблем и прерывание выполнения приложения | 75 |
| Проверка на предмет проблем и попытка устранить неполадки | 75 |
| Выбрасывание исключений | 76 |
| Важность области видимости | 78 |
| Локальные атрибуты | 78 |
| Атрибуты объектов | 80 |
| Атрибуты классов | 82 |
| Перегрузка операторов | 83 |
| Множественное наследование | 84 |
| Операции с объектами | 85 |
| Резюме | 86 |
| Ссылки | 86 |
| Примеры кода, использованного в этой главе | 87 |
| Глава 4. Анатомия класса | 88 |
| Имя класса | 88 |
| Комментарии | 90 |
| Атрибуты | 90 |
| Конструкторы | 92 |
| Методы доступа | 94 |
| Методы открытых интерфейсов | 96 |
| Методы закрытых реализаций | 97 |
| Резюме | 97 |

| | |
|---|-----|
| Ссылки | 97 |
| Примеры кода, использованного в этой главе | 98 |
| Глава 5. Руководство по проектированию классов | 100 |
| Моделирование реальных систем | 100 |
| Определение открытых интерфейсов | 101 |
| Минимальный открытый интерфейс | 101 |
| Скрытие реализации | 102 |
| Проектирование надежных конструкторов (и, возможно, деструкторов) | 103 |
| Внедрение обработки ошибок в класс | 104 |
| Документирование класса и использование комментариев | 104 |
| Создание объектов с прицелом на взаимодействие | 105 |
| Проектирование с учетом повторного использования | 106 |
| Проектирование с учетом расширяемости | 106 |
| Делаем имена описательными | 106 |
| Абстрагирование непереносимого кода | 107 |
| Обеспечение возможности осуществлять копирование и сравнение | 108 |
| Сведение области видимости к минимуму | 108 |
| Класс должен отвечать за себя | 109 |
| Проектирование с учетом сопровождаемости | 111 |
| Использование итерации в процессе разработки | 111 |
| Тестирование интерфейса | 112 |
| Использование постоянства объектов | 114 |
| Резюме | 115 |
| Ссылки | 115 |
| Примеры кода, использованного в этой главе | 116 |
| Глава 6. Проектирование с использованием объектов | 117 |
| Руководство по проектированию | 117 |
| Проведение соответствующего анализа | 121 |
| Составление технического задания | 121 |
| Сбор требований | 122 |
| Разработка прототипа интерфейса пользователя | 122 |
| Определение классов | 123 |
| Определение ответственности каждого класса | 123 |
| Определение взаимодействия классов друг с другом | 123 |
| Создание модели классов для описания системы | 123 |
| Прототипирование интерфейса пользователя | 123 |
| Объектные обертки | 124 |
| Структурированный код | 125 |
| Обертывание структурированного кода | 126 |
| Обертывание непереносимого кода | 128 |
| Обертывание существующих классов | 129 |
| Резюме | 130 |
| Ссылки | 130 |
| Глава 7. Наследование и композиция | 131 |
| Повторное использование объектов | 131 |
| Наследование | 133 |
| Обобщение и конкретизация | 135 |
| Проектные решения | 136 |

| | |
|--|------------|
| Композиция | 138 |
| Почему инкапсуляция является фундаментальной объектно-ориентированной концепцией | 141 |
| Как наследование ослабляет инкапсуляцию..... | 142 |
| Подробный пример полиморфизма | 144 |
| Ответственность объектов | 144 |
| Абстрактные классы, виртуальные методы и протоколы..... | 148 |
| Резюме | 150 |
| Ссылки..... | 150 |
| Примеры кода, использованного в этой главе | 151 |
| Глава 8. Фреймворки и повторное использование: проектирование с применением интерфейсов и абстрактных классов | 153 |
| Код: использовать повторно или нет?..... | 153 |
| Что такое фреймворк | 154 |
| Что такое контракт | 156 |
| Абстрактные классы..... | 157 |
| Интерфейсы..... | 160 |
| Связываем все воедино | 162 |
| Код, выдерживающий проверку компилятором | 165 |
| Заключение контракта..... | 165 |
| Системные «точки расширения»..... | 168 |
| Пример из сферы электронного бизнеса..... | 168 |
| Проблема, касающаяся электронного бизнеса | 168 |
| Подход без повторного использования кода | 169 |
| Решение для электронного бизнеса..... | 172 |
| Объектная модель UML | 172 |
| Резюме | 176 |
| Ссылки..... | 177 |
| Примеры кода, использованного в этой главе | 177 |
| Глава 9. Создание объектов и объектно-ориентированное проектирование | 180 |
| Отношения композиции | 180 |
| Поэтапное создание..... | 182 |
| Типы композиции | 184 |
| Агрегации | 185 |
| Ассоциации..... | 186 |
| Использование ассоциаций в сочетании с агрегациями..... | 186 |
| Избегание зависимостей | 187 |
| Кардинальность | 188 |
| Ассоциации, включающие множественные объекты..... | 190 |
| Необязательные ассоциации | 191 |
| Связываем все воедино: пример..... | 191 |
| Резюме | 193 |
| Ссылки..... | 193 |
| Глава 10. Создание объектных моделей..... | 194 |
| Что такое UML | 194 |
| Структура диаграммы класса | 195 |
| Атрибуты и методы..... | 197 |

| | |
|--|------------|
| Атрибуты | 197 |
| Методы | 197 |
| Обозначения доступа | 198 |
| Наследование | 199 |
| Интерфейсы | 200 |
| Композиция | 201 |
| Агрегации | 202 |
| Ассоциации | 202 |
| Кардинальность | 204 |
| Резюме | 206 |
| Ссылки | 206 |
| Глава 11. Объекты и переносимые данные: XML и JSON | 208 |
| Переносимые данные | 209 |
| XML | 210 |
| XML в противопоставлении с HTML | 211 |
| XML и объектно-ориентированные языки программирования | 212 |
| Обмен данными между двумя компаниями | 213 |
| Валидация документа с определением типа документа (DTD) | 214 |
| Включение определения типа документа в XML-документ | 216 |
| Использование CSS | 221 |
| JavaScript Object Notation (JSON) | 223 |
| Резюме | 228 |
| Ссылки | 228 |
| Глава 12. Постоянные объекты: сериализация, маршалинг и реляционные базы данных | 229 |
| Основные положения, касающиеся постоянных объектов | 229 |
| Сохранение объекта в плоском файле | 231 |
| Сериализация файла | 232 |
| Еще раз о реализации и интерфейсе | 234 |
| А как насчет методов? | 235 |
| Использование XML в процессе сериализации | 236 |
| Запись в реляционную базу данных | 238 |
| Резюме | 242 |
| Ссылки | 242 |
| Примеры кода, использованного в этой главе | 243 |
| Глава 13. Объекты в веб-службах, мобильных и гибридных приложениях | 246 |
| Эволюция распределенных вычислений | 246 |
| Основанные на объектах языка сценариев | 247 |
| Пример валидации с использованием JavaScript | 250 |
| Объекты на веб-странице | 253 |
| JavaScript-объекты | 253 |
| Элементы управления веб-страницы | 255 |
| Аудиопроигрыватели | 256 |
| Видеопроеигрыватели | 257 |
| Flash | 257 |
| Распределенные объекты и корпоративные вычисления | 258 |
| Common Object Request Broker Architecture (CORBA) | 259 |
| Определение веб-служб | 264 |

| | |
|---|------------|
| Код веб-служб..... | 268 |
| Representational State Transfer (ReST) | 269 |
| Резюме | 270 |
| Ссылки | 270 |
| Глава 14. Объекты и клиент-серверные приложения | 272 |
| Подходы «клиент/сервер» | 272 |
| Проприетарный подход | 272 |
| Сериализованный объектный код | 273 |
| Клиентский код | 274 |
| Серверный код..... | 276 |
| Выполнение примера «клиент/сервер» на основе проприетарного подхода..... | 277 |
| Непроприетарный подход | 278 |
| Код определения объектов | 279 |
| Клиентский код | 280 |
| Серверный код..... | 281 |
| Выполнение примера «клиент/сервер» на основе непроприетарного подхода..... | 283 |
| Резюме | 284 |
| Ссылки | 284 |
| Примеры кода, использованного в этой главе | 284 |
| Глава 15. Шаблоны проектирования | 285 |
| Зачем нужны шаблоны проектирования | 286 |
| Парадигма «Модель/Вид/Контроллер» в Smalltalk | 287 |
| Типы шаблонов проектирования | 289 |
| Порождающие шаблоны..... | 289 |
| Структурные шаблоны..... | 294 |
| Поведенческие шаблоны..... | 296 |
| Антишаблоны | 297 |
| Резюме | 299 |
| Ссылки | 299 |
| Примеры кода, использованного в этой главе | 299 |