

Содержание

Введение	17
Часть I. Основы разработки объектно-ориентированного программного обеспечения	23
Глава 1. Введение в объектно-ориентированное программирование	24
Что такое объектно-ориентированное программирование	24
Преимущества объектно-ориентированного подхода	25
Конкретный пример	26
Основные принципы и понятия ООП	27
Классы	27
Объекты	28
Наследование	37
Интерфейсы	47
Инкапсуляция	49
Изменения ООП в PHP 5	50
Резюме	51
Глава 2. Унифицированный язык моделирования UML	52
Определение требований	52
Интервьюирование клиента	53
Диаграммы прецедентов	54
Диаграммы классов	56
Моделирование предметной области	56
Отношения между классами	58
Реализация	60
Диаграммы видов деятельности	63
Диаграммы последовательностей	64
Диаграммы состояний	67
Диаграммы компонентов и развертывания	68
Резюме	70
Глава 3. Объектный подход в действии	71
Создание менеджера контактов	71
Диаграммы UML для адресной книги	72
Класс PropertyObject	77
Типы контактной информации	79
Класс DataManager	82
Классы Entity, Individual и Organization	84
Использование системы	91
Резюме	92

Глава 4. Шаблоны проектирования	94
Шаблон Composite	95
Реализация	96
Обсуждение	99
Шаблон Observer	100
Элементы управления	101
Обсуждение	106
Шаблон Decorator	106
Реализация	109
Использование шаблона Decorator	110
Обсуждение	111
Шаблон Facade	112
Шаблон Builder	113
Реализация	114
Обсуждение	117
Резюме	117
Часть II. Разработка повторно используемого набора объектов: простые служебные классы и интерфейсы	119
Глава 5. Класс Collection	120
Назначение класса Collection	121
Проектирование класса Collection	122
Основы класса Collection	122
Метод addItem()	123
Методы getItem() и removeItem()	124
Другие методы	124
Использование класса Collection	125
Реализация позднего инстанцирования	126
Обратные вызовы	127
Метод setLoadCallback() класса Collection	131
Использование класса Collection	135
Усовершенствование класса Collection	140
Резюме	140
Глава 6. Класс CollectionIterator	141
Интерфейс Iterator	142
Класс CollectionIterator	143
Интерфейс IteratorAggregate	145
Защита содержимого объекта Iterator с помощью оператора clone	146
Резюме	148
Глава 7. Класс GenericObject	149
Класс GenericObject	150
Когда нужно использовать класс GenericObject	150
Что позволяет делать класс GenericObject	150
Преимущества использования	151
Типичная реализация GenericObject	153

Встреча с предком	155
Взаимодействие класса GenericObject с базой данных	157
Методы и свойства класса GenericObject	159
Преимущества использования класса GenericObject	162
Класс GenericObjectCollection	162
Традиционная реализация	163
Когда традиционная реализация оказывается неудачной	164
Принципы, положенные в основу класса GenericObjectCollection	164
Исходный код класса GenericObjectCollection	165
Типичное использование класса GenericObjectCollection	167
Тестирование класса UserHome	168
Как это работает	168
Класс GenericObjectCollection: подведение итогов	171
Резюме	171
Глава 8. Уровни абстракции базы данных	172
Что такое уровень абстракции базы данных	173
Простая реализация	173
Конфигурационный файл	174
Установка соединения	174
Выборка данных	175
Изменение информации	175
Использование класса Database	177
Уровень абстракции PEAR DB	179
Подключение к базе данных с помощью класса DB	180
Извлечение информации	181
Другие полезные функции	183
Получение дополнительной информации	185
Завершенный уровень абстракции базы данных	185
Поддержка транзакций	189
Шаблон проектирования Singleton	190
Резюме	192
Глава 9. Интерфейс Factory	193
Шаблон проектирования Factory	193
Пример интерфейса Factory	194
Старый школьный подход	194
Применение интерфейса Factory	195
Использование интерфейса Factory на уровне абстракции базы данных	196
Многократное применение шаблона Factory	198
Усовершенствование существующих классов	199
Резюме	199
Глава 10. Управление событиями	200
Что такое события	201
Использование ООП для управления событиями	202
Проектное решение по управлению событиями	202
Реализация решения	204
Обеспечение безопасности	208

Остановитесь и подумайте	210
Резюме	211
Глава 11. Регистрация событий и отладка	212
Создание механизма регистрации событий	212
Запись в файл	212
Пример записи в файл	213
Класс Logger	214
Расширение класса Logger	218
Создание механизма отладки	227
Резюме	230
Глава 12. Протокол SOAP	231
SOAP и PHP 5	232
Расширение SOAP PHP 5	232
Создание SOAP-клиента	235
За кулисами	238
Обработка исключений в клиенте SOAP	242
Создание сервера SOAP	243
Резюме	245
Часть III. Разработка повторно используемого набора объектов: сложные (но не слишком) служебные классы	247
Глава 13. Модель, вид, контроллер	248
Знакомство с архитектурой MVC	249
Модель	250
Вид	250
Контроллер	250
Инфраструктура	250
MVC в Web-приложениях	250
MVC в языке PHP	251
Минимальный набор классов для реализации MVC	253
Знакомство с набором	253
Использование предложенного набора	263
Применение разработанного набора	268
Реальные шаблоны	269
Реализация шаблонов средствами PHP	269
Реализация шаблонов на основе пакета Smarty	270
Установка пакета Smarty	270
Использование пакета Smarty	271
Расширенные возможности пакета Smarty	275
Когда лучше использовать пакет Smarty, а не традиционные шаблоны	277
Резюме	277
Глава 14. Общение с пользователями	279
Для чего общаться	279
Причины для общения с пользователями	280
За пределами Web-браузера	282

Типы связи	282
Общие свойства взаимодействия	282
Свойства конкретных типов сообщений	283
Информация о получателе	283
Представление взаимодействия в виде иерархии классов	284
Класс Recipient: быстрая проверка объектного мышления	284
Класс Communication	287
Переписка с пользователями по электронной почте	289
Создание тестовой версии	289
Получение сообщения	292
Применение шаблонов	298
Использование MIME	299
Другие подклассы Communication	300
Передача SMS-сообщений	300
Факс	300
Резюме	300
Глава 15. Сеансы и аутентификация	301
Знакомство с сеансами	302
Краткий экскурс в историю протокола HTTP	302
Определение сеанса	304
Сохранение сеанса	304
Безопасность сеанса	306
Реализация сеансов в PHP	312
Сеансы в PHP	312
Ограничения базовых сеансов PHP	315
Создание класса аутентификации	315
Подключение механизма управления сеансами к базе данных	315
Знакомство с классом UserSession	316
Схема базы данных	317
Код класса UserSession.phpm	318
Код модульного теста для класса UserSession	321
Как работает класс UserSession	324
Использование класса UserSession	326
Резюме	326
Глава 16. Каркас для модульного тестирования	328
Методология и терминология	328
Разработка интерфейса класса	329
Создание пакета тестирования для класса	330
Реализация класса	331
Повторный запуск	331
Знакомство с PHPUnit	331
Установка PHPUnit	332
Использование PHPUnit	332
Тестовые классы	332
Тестовый пакет	334
Зачем беспокоиться?	335
Возвратное тестирование	335

Удобство использования каркаса	336
Гарантия качества	336
Упрощение функционального тестирования	336
Пример из жизни	337
Резюме	341
Глава 17. Конечные автоматы и файлы конфигурации	342
Знакомство с конечными автоматами	343
Простой конечный автомат: калькулятор для обратной польской записи	343
Теоретические реализации конечных автоматов	345
Реализация конечных автоматов на PHP	345
Пример калькулятора для обратной польской записи	347
Реальные примеры конечных автоматов	350
Пользовательские конфигурационные файлы	351
Использование PHP	351
Использование XML	352
Использование INI-файлов	352
Класс Config пакета PEAR	354
Рекомендации для работы с конфигурационными файлами	355
Резюме	356
Часть IV. Учебный пример: автоматизация работы торгового предприятия	357
Глава 18. Знакомство с проектом	358
Компания Widget World	358
Компания Widget World изнутри	359
Технический уровень	360
Финансовый уровень	360
Политический уровень	361
Разработчик	361
Действительно ли дело в технологии	361
Подход к разработке	361
Что это означает	363
Технология	365
Резюме	365
Глава 19. Методологии управления проектами	366
Выполните домашнее задание	367
Почему возникает новый проект	367
Для кого проект предназначен	367
Какую предысторию имеет проект	369
Каковы исходные условия	369
Разработка описания проекта	370
Формулировка бизнес-требований	370
Определение границ	372
График выполнения работ	373
Бюджет	374
Коммерческие условия	375
Планы на будущее	376

“Внешний облик” приложения	376
Технология	377
Поддержка	377
Что делать дальше	377
Написание предложения	378
Предложение или счет	378
Предложение или спецификация	378
Кто должен участвовать в написании предложения	379
Когда следует двигаться дальше	379
Когда следует сказать “нет”	380
Структурирование предложения	380
Выбор персонала	382
Менеджер проекта	382
Исполнительный директор	382
Главный архитектор	382
Разработчики и кодировщики	383
Разработчики клиентской части приложения	383
Главный дизайнер	383
Художники	383
Совмещение ролей	384
Организация работ	384
Роль заказчика	384
Резюме	384
Глава 20. Проектирование системы	385
Выбор процесса	385
Каскадный процесс	385
Спиральный процесс	386
Принятие решений	388
Общие рекомендации	388
Фаза разработки спецификации	388
Фаза проектирования	390
Карта страниц	391
Фаза реализации	391
Фаза тестирования	392
Передача проекта	392
Методология программирования	393
Разработка на основе тестирования	393
Экстремальное программирование	393
Управление изменениями	395
Модификация требований	396
Изменение требований после подписания спецификации	396
Конфликт из-за разницы толкования	396
Дефекты, обнаруженные клиентом	396
Резюме	397
Глава 21. Архитектура системы	398
Что такое системная архитектура	398
Почему это важно	398
Что нужно сделать	399

Эффективная реализация требований	399
Хостинг, соединения, серверы и сеть	400
Надежность и избыточность	400
Поддержка	401
Безопасность	401
Проектирование среды	401
Хостинг и соединения	401
Вычисление параметров канала	401
Серверы	403
Сеть	404
Дополнительная память	405
Поддержка	405
Безопасность	405
Резюме	406
Глава 22. Разработка средства автоматизации торговли	407
Начало проекта: понедельник	407
Слушайте внимательно	408
Оценка трудоемкости реализации сценариев	410
Планирование процесса разработки	416
Начало работы	417
Детали реализации сценария 9	418
Создание тестов	418
RNPUnit	419
Создание страницы регистрации	425
Следующий сценарий	427
Повторная оценка проекта	437
Чистка кода	438
Рефакторинг кода	438
Завершение итерации	444
Сценарий 14 – “Сохранение данных при изменении недели”	444
Сценарий 15 – “Еженедельное добавление данных в отчет о контактах с покупателями”	445
Отчет о командировочных расходах	453
Элемент командировочных расходов	453
Вычисление командировочных расходов за неделю	457
Последние штрихи	460
Дополнительные тесты для отчета о еженедельных расходах	461
Реализация класса еженедельного отчета о командировочных расходах	463
Окончательный отчет по командировочным расходам	474
Объекты-имитаторы	488
Резюме	492
Глава 23. Обеспечение качества	493
Основы анализа качества	493
Почему нужно ставить высокие цели	494
Что такое качество	495
Мера качества	495

Тестирование	497
Модульное тестирование	497
Функциональное тестирование	498
Тестирование нагрузки	499
Тестирование удобства использования	500
Отслеживание ошибок	500
Отслеживание ошибок с помощью системы Mantis	501
Несколько заключительных слов о системе Mantis	507
Резюме	508
Глава 24. Развертывание	509
Организация среды разработки	509
Сервер разработки	509
Поэтапная разработка	510
Среда поэтапного развертывания	511
Рабочая среда	512
Разработка баз данных	512
Процесс развертывания	513
Автоматическое извлечение данных из хранилища контроля версий	514
Утилита gsync	516
Синхронизация серверов с помощью утилиты gsync	518
Резюме	518
Глава 25. Разработка надежной системы генерации отчетов	519
Рабочие данные	519
Понимание потребностей заказчика	520
Управление запросами заказчика	520
Данные отчета	521
Разработка отчета	521
Архитектура генерации отчетов	524
Генерация отчетов в фоновом режиме	525
Страница Reports	526
Страница newreport	527
Сценарий обработки отчета	530
Обработка	531
Сценарии обработчиков	532
Страница Мои отчеты	533
Сценарии преобразования отчетов	533
Пример использования генератора отчетов	534
Визуализация	535
Резюме	535
Глава 26. Что дальше	536
Мотивация	536
Ваша карьера как разработчика	537
За пределами Web-приложений	537
Жизненный опыт	537
Академические навыки	537
Жизнь в социуме	538
Резюме	538

Часть V. Приложения	539
Приложение А. Зачем использовать контроль версий	540
Принципы контроля версий	540
Параллельное и исключающее управление версиями	541
Топология контроля версий	546
Программное обеспечение контроля версий	548
Microsoft Visual SourceSafe	548
CVS	550
RCS	551
Другие системы контроля версий	552
Дополнительные приемы контроля версий	552
Ветвление	552
Тегирование	553
Комментарии	553
Двоичные файлы	553
Резюме	553
Приложение Б. Интегрированные среды разработки для языка PHP	554
Выбор IDE	555
Zend Studio Client	555
Komodo	565
Другие IDE и редакторы	569
Резюме	570
Приложение В. Настройка производительности PHP	571
Проблемы производительности	571
Типы узких мест, связанных с производительностью	572
Причины недостаточной производительности	574
Поиск узких мест	574
Повышение производительности	576
Устранение временных задержек при работе с базой данных	576
Устранение узких мест в коде	577
Тестирование	579
Предупреждение неприятностей	580
Рекомендации по реализации высокопроизводительной архитектуры	580
Тестирование нагрузки	581
Резюме	582
Приложение Г. Практические советы по установке PHP	583
Введение в установку PHP	583
Выбор платформы	584
Лучший Web-сервер	585
Лучшая база данных	586
Инсталляция	586
Загрузка и установка СУБД PostgreSQL	587
Установка дополнительных библиотек	589
Установка PHP и Apache	590
Тестирование рабочего окружения	591
Когда использовать систему Windows	592

16 Содержание

Модификация путей	593
Странные различия	593
Внешние библиотеки	593
Использование пакета PEAR	593
Резюме	594
Предметный указатель	595