

Оглавление

Предисловие к первому изданию	24
Предисловие	27
Благодарности	30
О книге	32
Краткое содержание глав	33
Загрузка образов программного кода и используемые в книге соглашения	36
Об авторах	36
О названии книги	38
Об иллюстрации на обложке книги	39
От издательства	41

Часть I. Основы создания приложений

Глава 1. Вводный этап	44
1.1. Почему существует так много веб-приложений?	44
1.2. Содержательные интернет-приложения как решение	47
1.2.1. Это нужно всем	47
1.2.2. Содержательные интернет-приложения приходят на помощь	47
1.2.3. Как содержательные интернет-приложения позволяют добиваться таких результатов	49
1.3. Конкуренты содержательных интернет-приложений	49
1.3.1. Технология Flex от компании Adobe	50
1.3.2. Технология Silverlight от компании Microsoft	51
1.3.3. Технология JavaFX от компании Sun Microsystems	52
1.3.4. Технология AJAX — последний конкурент	52
1.4. Знакомство с Flex	55
1.4.1. Использование преимуществ Adobe Flash	56
1.4.2. Совместимость Flex и JavaScript	57
1.4.3. Экосистема Flex	57

1.5. Как работает Flex	61
1.5.1. Языки, поддерживаемые Flex	61
1.5.2. События, события, события	63
1.5.3. Ограничения	64
1.6. Что нового появилось в Flex 4	65
1.7. Заключение.	67
Глава 2. Приступаем к работе.	69
2.1. Flex «по дешевке».	69
2.1.1. Конфигурирование среды компилирования	70
2.1.2. Конфигурирование среды редактирования	72
2.1.3. Последующие шаги (если вам еще интересно)	73
2.2. Знакомимся с Flash Builder	74
2.2.1. Программные продукты и ценовая политика	75
2.2.2. Загрузка и установка Flash Builder	76
2.3. Исследуем Flash Builder	76
2.4. Представления и перспективы	79
2.4.1. Встроенные перспективы	79
2.4.2. Переключение между перспективами	79
2.4.3. Конфигурирование перспектив	80
2.5. Наш первый проект — Hello World!	81
2.5.1. Создаем проект	82
2.5.2. Печатаем код	84
2.5.3. Компилирование и запуск	84
2.5.4. Добавляем реальную функциональность	85
2.6. Использование режима Design.	85
2.7. Встроенный справочник API Reference	87
2.7.1. Объектно-ориентированные языки и их справочники API Reference.	88
2.7.2. Как получить доступ к справочнику API Reference	88
2.7.3. Чтение справочника API Reference	90
2.8. Кратко о MXML и ActionScript.	92
2.8.1. Структура MXML	92
2.8.2. Взаимоотношения MXML с ActionScript	93
2.8.3. ActionScript отвечает за обработку событий	94
2.9. Заключение	97

Глава 3. Работа с ActionScript	99
3.1. Комментарии	99
3.1.1. Встроенные комментарии	100
3.1.2. Блочные комментарии	100
3.2. Переменные	100
3.2.1. Имена переменных	101
3.2.2. Строгая типизация данных	101
3.2.3. Статическая и динамическая проверка типа	101
3.2.4. Классы верхнего уровня	102
3.2.5. Специальные типы данных	103
3.3. Циклы	103
3.3.1. For (начальное значение; соответствующее условие; инкремент)	104
3.3.2. For (имена свойств в массиве/объекте)	104
3.3.3. For each (элемент в массиве/объекте)	105
3.3.4. While (условие)	106
3.3.5. Do while (условие)	106
3.4. Условные операторы (операторы if и switch)	107
3.4.1. If..else	107
3.4.2. Switch	109
3.5. Массивы	110
3.5.1. Индексированные массивы	111
3.5.2. Ассоциативные массивы	113
3.6. Аспекты, касающиеся ActionScript	115
3.6.1. Скобки	115
3.6.2. Логические операторы и сокращения	116
3.7. Взгляд на функции, классы и пакеты	118
3.7.1. Ваши собственные функции	119
3.7.2. Отделение ActionScript-кода и размещение его в обособленных файлах.	123
3.8. Простая привязка данных	124
3.8.1. Жизнь без привязки	124
3.8.2. Добавление привязки	125
3.8.3. Тег <fx:Binding>	126
3.8.4. Делаем переменные ActionScript привязываемыми	126
3.9. Заключение	128

Глава 4. Макет и контейнеры	129
4.1. Сравнение Spark и Halo (МХ)	129
4.2. Макет, предусматривающий абсолютное позиционирование	131
4.3. Макет на основе ограничений	134
4.3.1. Базовые ограничения	135
4.3.2. Дополнительные ограничения	137
4.4. Макет, предусматривающий автоматическое позиционирование	144
4.4.1. Использование классов макетов	144
4.4.2. Установление промежутков	146
4.5. Установление переменных и фиксированных размеров	147
4.5.1. Установление переменных размеров	147
4.5.2. Установление фиксированных размеров	148
4.6. Контейнеры	148
4.6.1. Контейнер Application	149
4.6.2. Контейнер Canvas	150
4.6.3. Контейнеры Group и SkinnableContainer	151
4.6.4. Контейнер Panel	153
4.6.5. Контейнер ApplicationControlBar	154
4.6.6. Контейнеры DataGroup и SkinnableDataContainer	156
4.6.7. Контейнеры DividedBox, HDividedBox и VDividedBox	158
4.6.8. Контейнер Form.	160
4.6.9. Контейнер Grid	161
4.7. Заключение.	162
Глава 5. Отображение форм и захват вводимых пользователем данных	163
5.1. Атрибут id	164
5.2. Каталог элементов управления Flex	165
5.2.1. Текстовые элементы управления	165
5.2.2. Date-элементы управления	169
5.2.3. Элементы управления для захвата числовых значений	171
5.2.4. Обзор кнопок Flex	174
5.2.5. Элементы управления со списками выбора	180
5.3. Доступ к значению элемента управления	183
5.3.1. Передача значений функции	183
5.3.2. Передача событий функции.	184

5.3.3. Прямой доступ к свойствам	186
5.3.4. Какой подход следует предпочесть	187
5.4. Заключение	188
Глава 6. Проверка вводимых пользователем данных	189
6.1. Обзор способов проверки	189
6.2. Встроенные валидаторы	190
6.2.1. Компонент Validator	191
6.2.2. StringValidator	193
6.2.3. NumberValidator	194
6.2.4. DateValidator	195
6.2.5. EmailValidator	197
6.2.6. CreditCardValidator	199
6.2.7. CurrencyValidator	200
6.2.8. PhoneNumberValidator	201
6.2.9. RegExpValidator	202
6.2.10. SocialSecurityValidator	205
6.2.11. ZipCodeValidator	206
6.3. Проверка в реальном времени	207
6.4. Проверка при фиксации значения	208
6.5. Сквозная проверка	208
6.6. Сценарная проверка	209
6.7. Аспекты, касающиеся проверки	210
6.7.1. Всегда ли валидатор проверяет соответствие всем критериям?	211
6.7.2. Контроль над тем, что инициирует проверку	211
6.8. Заключение	212
Глава 7. Форматирование данных	213
7.1. Встроенные форматы	214
7.1.1. Formatter	214
7.1.2. NumberFormatter	215
7.1.3. CurrencyFormatter	217
7.1.4. DateFormatter	218
7.1.5. PhoneFormatter	221
7.1.6. ZipCodeFormatter	223
7.1.7. SwitchSymbolFormatter	225

7.2. Форматирование в реальном времени	226
7.3. Сценарное форматирование	227
7.3.1. Использование функции в сочетании с компонентом в виде форматера	227
7.3.2. Использование функции в сочетании с классом форматера	228
7.4. Выявление ошибок форматирования с целью их последующего устранения	230
7.5. Заключение.	231
Глава 8. MX-компоненты DataGrid, List и Tree	232
8.1. Генеалогия основанных на списке компонентов MX	232
8.1.1. Свойства, поддерживаемые ListBase и AdvancedListBase.	233
8.1.2. События, поддерживаемые MX-компонентом ListBase.	235
8.2. Понятие коллекций и dataProvider	235
8.2.1. Заполнение dataProvider	236
8.2.2. Типы коллекций	236
8.2.3. Компоненты, использующие коллекции	237
8.3. Инициализация коллекций	237
8.4. Заполнение коллекций	238
8.4.1. List	239
8.4.2. HorizontalList	242
8.4.3. TileList.	243
8.4.4. DataGrid	246
8.4.5. Tree	250
8.5. Взаимодействие с основанными на списке компонентами MX.	253
8.5.1. События, связанные с основанными на списке компонентами	253
8.5.2. Передача события функции.	254
8.5.3. Передача данных функции	256
8.5.4. Прямой доступ к выбранной строке	257
8.5.5. Привязка к выбранной строке	257
8.6. Заключение	258
Глава 9. Использование основанных на списке элементов управления Spark.	259
9.1. Генеалогия основанных на списке компонентов Spark	260
9.1.1. Идентификация соответствующего компонента с использованием пространств имен	260

9.1.2. Рендереры элементов и основанные на списке компоненты Spark	261
9.2. Основанные на списке элементы управления Spark	261
9.2.1. Элемент управления ButtonBar	262
9.2.2. Spark-элемент управления List	263
9.2.3. Элемент управления DropDownList	264
9.3. Взаимодействие с основанными на списке компонентами Spark	266
9.3.1. Отправка событий по умолчанию при выборе элемента	266
9.3.2. ОбъектIndexChangedEvent	267
9.4. Понятие архитектуры основанных на списке компонентов Spark, поддерживаемых версией Flex 4	267
9.4.1. Иерархия классов	267
9.4.2. Новые Spark-классы для основанных на списке элементов управления	268
9.5. Создание пользовательских основанных на списке компонентов на базе архитектуры Spark	270
9.5.1. Понятие классов Group и SkinnableContainer.	270
9.5.2. Создание пользовательского основанного на списке компонента Spark на базе класса List	271
9.6. Заключение	274

Глава 10. Конфигурирование основанных на списке компонентов Spark	275
10.1. Конфигурирование отображения данных	275
10.1.1. Свойство labelField	276
10.1.2. Функции метки	276
10.1.3. Типы функций метки	278
10.1.4. Использование функций метки, предназначенных для многостолбцовых основанных на списке компонентов	279
10.1.5. Для чего могут применяться функции метки	282
10.1.6. Более общая картина	282
10.2. Рендереры элементов (средства визуализации элементов).	283
10.2.1. Spark-рендереры элементов MXML	283
10.2.2. Spark-рендереры элементов MXML с использованием компонентов MX	285
10.2.3. Создание встроенного рендерера элементов	289
10.2.4. Использование вставных рендереров элементов	289
10.3. Редакторы элементов	291

10.3.1. Активация опции редактирования элементов	292
10.3.2. Создание редактора элементов	294
10.3.3. События, инициируемые во время процесса редактирования.	296
10.3.4. Объединяем силы: <code>renderersIsEditor</code>	297
10.4. Продвинутое редактирование элементов	298
10.4.1. <code>AdvancedDataGridRendererProvider</code>	299
10.4.2. Ссылка на столбец.	300
10.4.3. Охват столбцов	301
10.4.4. Охват строки целиком	302
10.5. Функции фильтрации.	303
10.6. Заключение.	305

Часть II. Поток и структура приложения

Глава 11. События.	308
11.1. Система событий.	308
11.1.1. Система событий — принцип Голливуда.	311
11.1.2. Система доставки событий.	311
11.2. Отправка и получение событий	313
11.2.1. Добавление слушателей событий на <code>ActionScript</code>	315
11.2.2. Привязка событий	317
11.2.3. Удаление слушателей событий.	320
11.3. Пользовательские события	323
11.3.1. Отправка пользовательских типов событий	324
11.3.2. Создание пользовательских событий	324
11.3.3. Добавление метаданных событий в пользовательские отправители событий	330
11.3.4. Остановка распространения событий	332
11.4. Заключение.	333
Глава 12. Навигационная функциональность приложений	335
12.1. Подготовка данных меню	335
12.1.1. Вложенные массивы	336
12.1.2. Коллекции вложенных массивов	337
12.1.3. Модели	338
12.1.4. Компонент и класс <code>XML</code>	340

12.1.5. Компонент XMLList	341
12.1.6. Компонент и класс XMLListCollection	342
12.2. Работа с меню	343
12.2.1. Создание меню	344
12.2.2. Позиционирование меню	346
12.2.3. Конфигурирование элементов меню	346
12.2.4. Взаимодействие с меню	349
12.3. Использование строки меню	351
12.3.1. Создание строки меню	351
12.3.2. Позиционирование строки меню	352
12.3.3. Конфигурирование элементов в строке меню	352
12.3.4. Обработка пользовательских взаимодействий со строками меню	353
12.4. Использование контейнера ViewStack	356
12.4.1. Создание ViewStack	356
12.4.2. Добавление навигационной функциональности в ViewStack	359
12.4.3. Обработка пользовательских взаимодействий с ViewStack	361
12.5. TabNavigator	363
12.5.1. Создание TabNavigator	363
12.5.2. Обработка пользовательских взаимодействий с TabNavigator	364
12.6. Accordion	366
12.6.1. Создание Accordion	366
12.6.2. Заполнение Accordion	367
12.6.3. Обработка пользовательских взаимодействий с Accordion	368
12.7. Заключение	368
Глава 13. Всплывающие окна	370
13.1. Создание первого всплывающего окна	370
13.1.1. Сначала о главном: создание TitleWindow	370
13.1.2. Использование PopUpManager для открытия окна	372
13.1.3. Закрытие всплывающего окна	373
13.2. Контроль над позиционированием всплывающих окон	374
13.2.1. Использование метода centerPopUp()	374
13.2.2. Расчет размещения всплывающего окна	375

13.3. Интеграция данных с использованием всплывающих окон	379
13.3.1. Извлечение данных из всплывающего окна	382
13.3.2. Отправка пользовательских событий с данными	383
13.3.3. Извлечение данных	384
13.3.4. Отправка данных во всплывающее окно	385
13.4. Использование всплывающих окон с предупреждениями Alert	388
13.4.1. Создание простого окна Alert	388
13.4.2. Дополнительные манипуляции с окнами Alert.	389
13.4.3. Более продвинутое окно Alert	390
13.4.4. Вывод на экран стилизованных окон Alert	391
13.5. Заключение	394
Глава 14. Реализация состояний представлений	395
14.1. Понятие состояний представлений	395
14.2. Состояния представлений в сфере Flex	397
14.2.1. Работа со свойствами	399
14.2.2. Работа с обработчиками событий	402
14.2.3. Использование групп состояний	405
14.2.4. Добавление и удаление компонентов	408
14.2.5. Смена родителей компонентов	413
14.2.6. События состояний	415
14.3. Применение состояний представлений в реальном приложении. . .	417
14.4. Заключение	420
Глава 15. Работа со службами данных	421
15.1. Доступ к данным на стороне сервера	421
15.1.1. Использование объекта HTTPService	423
15.1.2. Использование веб-служб с применением компонента WebService	424
15.2. Протокол Action Message Format в действии.	425
15.2.1. AMF с открытым исходным кодом.	427
15.2.2. AMF и PHP	427
15.2.3. AMF и ColdFusion	427
15.2.4. BlazeDS	428
15.2.5. LiveCycle Data Services	428
15.2.6. Прочие технологии	428
15.3. Создание ориентированных на данные приложений с использованием Flash Builder	429

15.3.1. Конфигурирование среды	429
15.3.2. Подключение к серверу	430
15.4. DCD на Flex с использованием ColdFusion.	438
15.5. DCD на Flex с использованием Java EE и BlazeDS	441
15.6. Привязка модели к представлению	442
15.6.1. Привязка данных посредством перетаскивания мышью	442
15.6.2. Генерирование формы Master-Detail.	443
15.6.3. Просмотр кода, генерируемого средой Flash Builder	445
15.7. Заключение	448
Глава 16. Объекты и классы	449
16.1. Теория объектно-ориентированного программирования за 5 минут.	449
16.1.1. Взаимоотношения между объектами и классами.	450
16.1.2. Объекты располагают свойствами и методами	451
16.1.3. Наследование	452
16.1.4. Инкапсуляция и связывание.	453
16.2. Работа с объектами	455
16.2.1. Более пристальный взгляд на объекты	455
16.2.2. Методы объектов.	457
16.2.3. Параметры методов.	457
16.2.4. Данные, возвращаемые методами	458
16.3. Создание класса	458
16.3.1. Создание файла класса	458
16.3.2. Указание пакета	459
16.3.2. Модификаторы класса	460
16.3.4. Суперклассы: расширение класса	461
16.3.5. Интерфейсы	461
16.3.6. Взгляд на код нашего класса	462
16.4. Работа со свойствами	463
16.4.1. Добавление свойств	463
16.4.2. Добавление методов получателя/установщика.	464
16.5. Создание методов для класса	465
16.6. Заключение	468
Глава 17. Пользовательские компоненты	469
17.1. Понятие компонентов Flex 4.	470
17.1.1. Архитектура компонентов Spark.	470
17.1.2. Многообразие пользовательских компонентов	471

17.2. Создание простого пользовательского компонента	475
17.2.1. Создание собственного простого элемента управления ComboBox	476
17.2.2. Простые компоненты Spark	478
17.3. Работа со скинами (темами оформления) с использованием объекта Skin	479
17.3.1. Использование метаданных для привязки скинов компонентов	481
17.3.2. Состояния представлений пользовательских компонентов	481
17.3.3. Определение элементов скина	482
17.3.4. Объявление хоста	483
17.4. Составные компоненты	484
17.4.1. Halo и Spark	484
17.4.2. Классы макетов Spark	486
17.4.3. Создание составных компонентов MXML	487
17.5. Создание продвинутых компонентов Flex 4	489
17.5.1. Использование ActionScript для создания селектора штата	489
17.5.2. Знание того, когда именно необходимо прибегать к переопределению функций	491
17.6. Обеспечение коммуникации между компонентами	496
17.6.1. Использование получателей и установщиков в качестве прокси свойств	497
17.6.2. Привязка переменных к свойствам компонента	498
17.6.3. Использование событий для передачи данных	499
17.7. Заключение	502
Глава 18. Создание пользовательских компонентов	504
18.1. Возможность повторного использования в действии	505
18.1.1. Подготовительная компонента Reveal	505
18.1.2. Теория и концепции, касающиеся возможности повторного использования компонентов Spark	505
18.1.3. Создание компонента Reveal	508
18.1.4. Создание скина для первой реализации компонента Reveal	511
18.2. Новый внешний вид компонента Reveal	514
18.3. Совместно используемые библиотеки времени выполнения	518
18.3.1. Понятие файлов SWC	518

18.3.2. Типы совместно используемых библиотек времени выполнения.	519
18.3.3. Как сделать так, чтобы приложение Flex задействовало совместно используемую библиотеку времени выполнения.	519
18.4. Заключение	523
Глава 19. Шаблоны архитектурного проектирования	524
19.1. Управляемые Flex шаблоны проектирования	525
19.1.1. Шаблон Model–View–Controller	526
19.1.2. Создание собственной архитектуры	527
19.2. Введение в микроархитектуры.	533
19.2.1. Что такое микроархитектура?	533
19.2.2. Зачем нужно использовать микроархитектуру?	535
19.2.3. Микроархитектуры первого поколения.	535
19.2.4. Микроархитектуры второго поколения.	537
19.2.5. Шаблон Inversion of Control и внедрение зависимостей	538
19.3. Использование фреймворка Robotlegs	539
19.3.1. Внедрение зависимостей с помощью Robotlegs.	540
19.3.2. Конфигурирование внедрений зависимостей с использованием утилит для сопоставления Robotlegs	541
19.4. Создание приложения с использованием MVCS-реализации Robotlegs.	548
19.4.1. Создание проекта Robotlegs.	549
19.4.2. Начальная загрузка приложения с использованием класса Context	550
19.4.3. Связывание представлений с использованием посредников	552
19.4.4. Управление с использованием команд Robotlegs	558
19.4.5. Службы как вход во внешний мир	560
19.4.6. Использование модели для управления данными и состоянием.	563
19.5. Заключение	565

Часть III. Заключительные штрихи

Глава 20. Конфигурирование взаимодействия пользователей с приложениями	568
---	------------

20.1. Принципы проектирования взаимодействия пользователя с приложением	569
20.1.1. Берем за основу пользовательские истории	569
20.1.2. Принятие во внимание контекста	571
20.1.3. Модель VIBE	574
20.2. Визуальная привлекательность	575
20.2.1. Использование имеющихся и создание новых тем	575
20.2.2. Стилизация приложений Flex 4 с использованием CSS	578
20.3. Интерактивное взаимодействие	584
20.3.1. Декларативный дизайн с использованием FXG и Flash Catalyst CS5	584
20.3.2. Улучшаем качество пользовательского взаимодействия при помощи эффектов	587
20.4. Бизнес-оптимизация	590
20.4.1. Следование наиболее оптимальным методикам с целью улучшения качества пользовательского взаимодействия	590
20.4.2. Улучшение качества взаимодействия пользователя с приложением посредством модульного тестирования	592
20.4.3. Профилирование приложений Flex 4	593
20.5. Расширяемость	594
20.5.1. Косвенная связь расширяемости с пользовательским взаимодействием	595
20.5.2. Прямая связь между расширяемостью и пользовательским взаимодействием	596
20.5.3. Для обеспечения удобства пользования следует писать аккуратный код	597
20.6. Заключение	599
Глава 21. Работа с эффектами	600
21.1. Что такое эффект?	600
21.1.1. Доступные эффекты	601
21.1.2. Составные эффекты	602
21.2. Использование эффектов	602
21.2.1. Причина и следствие	602
21.2.2. Эффекты, инициируемые событиями	603
21.2.3. Программный способ применения эффектов	605
21.2.4. Использование переходов из одного состояния в другое для инициирования эффектов	609

21.3. Создание составных эффектов	610
21.3.1. Эффекты Sequence	611
21.3.2. Эффекты Parallel	612
21.3.3. Сложные составные эффекты	613
21.4. Исследование эффектов	615
21.4.1. Эффект Animate.	616
21.4.2. Анимация фильтров при помощи AnimateFilter	625
21.4.3. Анимация пиксельных шейдеров с помощью AnimateTransitionShader	628
21.4.4. Конфигурирование замедления эффектов	632
21.4.5. Сохранение корректного отображения шрифтов при анимации	635
21.4.6. Создание звуковых эффектов	637
21.5. Заключение	638

Глава 22. Перетаскивание элементов при помощи мыши 640

22.1. Перетаскивание при помощи мыши	641
22.2. Реализация функциональности, связанной с перетаскиванием при помощи мыши, в компонентах Flex	644
22.2.1. Компоненты с исконной поддержкой перетаскивания при помощи мыши	645
22.2.2. Активация возможности перетаскивания с помощью мыши при использовании компонента List	645
22.2.3. Перемещение в противопоставление копированию	647
22.2.4. Использование перетаскивания при помощи мыши для осуществления управляемой пользователем сортировки	648
22.2.5. Перетаскивание при помощи мыши сразу нескольких элементов за один раз.	650
22.2.6. Двустороннее перетаскивание при помощи мыши	652
22.3. Знакомство с DragManager	654
22.3.1. Свойства и методы DragManager	654
22.3.2. Принятие или отказ от принятия перетаскиваемого элемента.	655
22.3.3. Внедрение собственной логики, связанной с перетаскиванием элементов при помощи мыши.	659
22.4. Добавление возможности перетаскивания элементов при помощи мыши в не основанные на списке компоненты	662
22.4.1. Исследуем пример	662
22.4.2. Инициирование перетаскивания	663

22.4.3. Использование пользовательского прокси перетаскивания	666
22.4.4. Обработка помещения перетаскиваемого элемента в цель перетаскивания	668
22.5. Конфигурирование взаимодействия, обеспечиваемого перетаскиванием элементов при помощи мыши	672
22.5.1. Изменение значков прокси перетаскивания	672
22.5.2. Использование скинов для компонентов List, поддерживающих перетаскивание элементов при помощи мыши	673
22.5.3. Перетаскивание элементов при помощи мыши между компонентами Spark и Halo	675
22.6. Заключение	676

Глава 23. Исследование возможностей Flex

по созданию диаграмм	678
23.1. Введение в создание диаграмм	678
23.1.1. Составные части диаграммы	679
23.1.2. Обзор типов диаграмм	680
23.2. Серии данных	680
23.3. Создание диаграмм	682
23.3.1. Создание диаграммы	682
23.3.2. Добавление легенды	683
23.3.3. Изменение типа диаграммы	684
23.3.4. Фильтрация данных для диаграммы	687
23.4. Диаграммы с поддержкой накопления	688
23.5. Исследование типов диаграмм	690
23.5.1. Диаграммы областей	690
23.5.2. Горизонтальные и обычные гистограммы	691
23.5.3. Линейные диаграммы	694
23.5.4. Пузырьковые диаграммы	695
23.5.5. Диаграммы «Свечи» и диаграммы HLOC	697
23.5.6. Круговые диаграммы	700
23.5.7. Точечные диаграммы	703
23.6. Конфигурирование диаграмм	706
23.6.1. Обводки для серий данных	706
23.6.2. Заливка для серий данных	707
23.7. Заключение	708

Глава 24. Тестирование и отладка	710
24.1. Отладка	711
24.1.1. Установка Flash Debug Player	711
24.1.2. Использование функции trace()	712
24.1.3. Использование отладчика, встроенного в Flash Builder	714
24.1.4. Мониторинг сетевой активности	719
24.2. Профилировщик Flex	720
24.3. Автоматизированное тестирование приложений	725
24.3.1. Модульное тестирование	725
24.3.2. Функциональное тестирование	732
24.4. Заключение	735
Глава 25. Завершение проекта	736
25.1. Конфигурирование HTML-обертки	736
25.1.1. Файлы обертки	737
25.1.2. Шаблон HTML	738
25.1.3. SWFObject 2	738
25.2. Развертывание	741
25.2.1. Создание производственной сборки	741
25.2.2. Размещение файлов, относящихся к стороне клиента	743
25.2.3. Размещение файлов, относящихся к стороне сервера	743
25.2.4. Тестирование приложения	744
25.3. Заключение	745