

Дмитрий Альберт
Елена Альберт



ActionScript 2.0

- Детальное описание синтаксиса и возможностей языка
- Динамические эффекты и интерактивная анимация
- Объектно-ориентированное программирование в стиле ActionScript 2.0

**Наиболее
полное
руководство**

+CD



В ПОДЛИННИКЕ®

**Дмитрий Альберт
Елена Альберт**

ActionScript 2.0

Санкт-Петербург

«БХВ-Петербург»

2005

УДК 681.3.06
ББК 32.973.26-018.1
А56

Альберт Д. И., Альберт Е. Э.

А56 ActionScript 2.0. — СПб.: БХВ-Петербург, 2005. — 1136 с.: ил.
ISBN 5-94157-567-X

Книга является подробным руководством по языку сценариев ActionScript 2.0 для среды Macromedia Flash MX 2004. Описаны основные возможности, синтаксис и семантика языка, его базовые элементы: переменные, типы данных, циклы, условные операторы, массивы, функции. Рассмотрены событийная модель и процедура трансляции событий, встроенные классы и технология объектно-ориентированного программирования в стиле ActionScript 2.0. Изложены основанный на этой технологии подход к трехмерному моделированию и процедура создания пользовательских компонентов.

Книга написана ясным и логичным языком. Материал организован так, чтобы ее можно было использовать и как обучающее руководство, и как справочное пособие. В начале каждой темы приводятся теоретические сведения, которые затем иллюстрируются конкретными практическими примерами. В книгу включено более 100 примеров, что позволит читателю приобрести практические навыки работы с языком. Все исходные файлы примеров помещены на сопровождающий компакт-диск.

Для программистов

УДК 681.3.06
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Татьяна Лапина</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн обложки	<i>Игоря Цырульников</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 17.06.05.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 91,59.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ОАО "Техническая книга"

190005, Санкт-Петербург, Измайловский пр., 29.

ISBN 5-94157-567-X

© Альберт Д. И., Альберт Е. Э., 2005

© Оформление, издательство "БХВ-Петербург", 2005

Оглавление

Введение	1
Часть I. Основы ACTIONSCRIPT	3
Глава 1. Представление о Flash	5
Компоненты программы Flash MX 2004 Professional	5
Процесс создания Flash-проекта	6
Системные требования программы Flash MX 2004 Professional	7
Обзор интерфейса	8
Основы работы с векторным редактором Flash MX 2004	15
Понятие о векторной графике	15
Иерархия графических объектов Flash	17
Операции над объектами и трансформация объектов	21
Настройка глобальных параметров фильма	23
Работа с цветом	24
Работа с растровой графикой	26
Структура монтажной линейки	28
Работа со слоями	29
Работа с кадрами	31
Символы и экземпляры	35
Типы символов и операции с ними	36
Экземпляр символа и его свойства	38
Работа с библиотекой	40
Работа с общими библиотеками (Shared Libraries)	41
Разрешение вопросов политики безопасности при импорте внешних данных из других доменов	44
Анимация	45
Покадровая анимация (Frame by frame animation)	47
Анимация формы (Shape tweening)	47
Анимация движения (Motion tweening)	49
Использование звука в фильме	50
Публикация исходного документа	54
Формат Flash (SWF)	56
Формат HTML	59
Формат Windows Projector (exe)	64
Глава 2. Новые возможности ActionScript	65
ActionScript 2.0	65
Зависимость от регистра	65

Строгий контроль типа данных	66
Формальный синтаксис классов	66
Новая архитектура компонентов	67
Массивы	67
Работа с клипами	68
Связывание символа с классом	68
Компилированные клипы	68
Контроль заполнения уровней стека	68
Свойство <i>_lockroot</i>	69
Метод <i>getSWFVersion()</i>	69
Работа с текстом	69
Поддержка CSS (Cascading Style Sheets)	69
Возможность загрузки изображений и внешних SWF-документов в текстовый блок	70
Прокрутка текста с помощью колеса мыши	70
Контроль статического текста	70
Класс <i>MovieClipLoader</i> . Организация процесса загрузки	71
Класс <i>Sound</i> . Поддержка ID3-тегов второй версии	71
Динамическая загрузка видео	72
Управление контекстным меню проигрывателя	72
Объект <i>System.capabilities</i>	73
Печать из фильма	73
Политика безопасности	74
Глава 3. Введение в ActionScript. Среда разработки	75
Представление о сценариях ActionScript	75
Использование редактора ActionScript для создания сценариев	76
Раздел <i>Script pane</i> (Окно сценария)	78
Раздел <i>Actions toolbox</i>	87
Раздел <i>Script navigator</i>	89
Носители сценариев AS	89
Синтаксические нормы и правила	92
Предложения	92
Комментарии	93
Имена	94
Функции	95
Функция <i>trace()</i>	96
Инициирование и последовательность выполнения сценариев	
Системные и пользовательские события	97
Последовательность выполнения сценариев кадров	98
Сценарии кнопок и клипов	99
Представление об объектно-ориентированном программировании	101
Создание сценария во внешнем файле	103
Применение директивы <i>#include</i>	103
Проекты. Панель <i>Project</i>	104
Работа с проектами	104

Глава 4. Переменные	113
Понятие переменной	113
Создание и типизация переменной.....	114
Явное объявление переменной.....	114
Неявное объявление переменной.....	117
Динамическое создание переменной.....	117
Присваивание, извлечение и изменение значения переменной.....	118
Присваивание значения.....	118
Автоматическая типизация переменной.....	120
Извлечение значения.....	121
Область видимости и адресация переменных	121
Вопросы области видимости.....	121
Локальные переменные	123
Глобальные переменные.....	126
Адресация переменных.....	129
Время жизни переменной	135
Удаление переменной.....	136
Глава 5. Элементарные типы данных	138
Элементарные и ссылочные типы данных	138
Числовой тип. Работа с числами.....	140
Литералы чисел.....	141
Специальные числовые значения.....	142
Операции над числами	144
Встроенные методы для работы с числами.....	147
Строковый тип. Работа со строками.....	148
Литералы строк.....	148
Операции над строками.....	151
Встроенные свойства и методы для работы со строками	153
Булев тип.....	164
Логические операции.....	165
Встроенные методы для работы с булевыми значениями	167
Типы <i>null</i> и <i>undefined</i>	168
Значение <i>null</i>	168
Значение <i>undefined</i>	168
Явное и автоматическое преобразование элементарных типов данных	169
Явное преобразование типов данных.....	169
Автоматическое преобразование типов данных.....	172
Глава 6. Выражения и операторы	176
Приоритет операторов	176
Ассоциативность.....	180
Модификация значения	180
Простое присваивание	181
Составное присваивание	182
Инкрементирование/декрементирование	182

Побитовые операции	184
Битовое представление числа	185
Применение двоичных чисел.....	187
Логические побитовые операции	188
Побитовые операции сдвига	193
Глава 7. Управляющие структуры: Операторы ветвления. Циклы	198
Ветвление кода	198
Предложение <i>if</i>	199
Предложение <i>else</i>	202
Предложение <i>else if</i>	206
Предложение <i>switch</i>	210
Многократное выполнение кода	214
Понятие программного цикла	215
Цикл <i>while</i>	216
Цикл <i>do while</i>	218
Цикл <i>for</i>	219
Цикл <i>for...in</i>	222
Контроль выхода из цикла	224
Вложенные циклы.....	225
Циклы монтажной линейки.....	226
Глава 8. Пользовательские функции	231
Общие сведения о функциях	231
Создание функции	232
Объявление функции.....	232
Использование литерала функции	233
Параметры функции	234
Контроль типа параметров.....	235
Класс <i>Arguments</i> . Контроль фактических параметров функции	236
Возвращение значения	240
Контроль типа возвращаемого значения.....	243
Вызов функции.....	245
Вызов функции по имени.....	245
Вызов функции с помощью ссылки	246
Вызов функции с помощью метода <i>apply()</i>	248
Вызов функции с помощью метода <i>call()</i>	250
Периодический вызов функции. Глобальные функции <i>setInterval()</i> и <i>clearInterval()</i>	250
Вопросы области видимости и времени существования функций	255
Область видимости функции	255
Доступ к функции	258
Время жизни функции.....	259
Рекурсивные функции	259

Глава 9. Массивы	262
Понятие массива	262
Создание массива	263
Конструктор массива	263
Литерал массива	264
Глобальная функция <i>Array()</i>	265
Вывод массива в панель <i>Output</i>	266
Обращение к элементам массива	266
Операции над элементами массива	267
Определение количества элементов в массиве	267
Присваивание значения элементу массива	268
Добавление элементов в массив	269
Строковое представление массива	274
Сортировка массива	276
Извлечение части массива	288
Удаление элементов массива	289
Ассоциативные массивы	290
Многомерные массивы	292
 Часть II. БАЗОВЫЕ ВСТРОЕННЫЕ КЛАССЫ ACTIONSCRIPT	299
 Глава 10. Класс <i>Button</i>. Работа с кнопками	301
Создание кнопки в рабочей среде	302
Кнопка как объект <i>AS</i>	304
Имя экземпляра	305
Режимы кнопки	306
Обработка событий кнопки	307
События кнопки	307
Обработчик событий кнопки	308
Контроль нажатия клавиатурной клавиши	311
Область видимости обработчика событий кнопок	313
 Глава 11. Обработка событий клипов. Событийная модель ActionScript	315
Обработчик событий <i>on()</i> . Использование клипов в качестве кнопок	316
Создание клипа-кнопки	316
Область видимости обработчика событий <i>on()</i> клипа	317
События клипов	319
Событие <i>enterFrame</i>	320
Событие <i>load</i>	321
Событие <i>unload</i>	322
Событие <i>data</i>	322
Событие <i>mouseMove</i>	323
Событие <i>mouseDown</i>	323
Событие <i>mouseUp</i>	323
Событие <i>keyDown</i>	324

Событие <i>keyUp</i>	324
Область видимости обработчика <i>onClipEvent()</i>	325
Применение методов обработчиков событий	325
Понятие метода обработчика событий	326
Вопросы области видимости методов обработчиков событий кнопок и клипов	329
Удаление методов обработчиков событий	331
Применение методов обработчиков событий с использованием адресации	332
Трансляция событий	333
Передача и обработка событий	334
Прекращение трансляции событий	337
Создание источников событий. Объект <i>AsBroadcaster</i>	337
Инициализация источника и трансляция событий	338
Применение объекта <i>AsBroadcaster</i>	340
Глава 12. Класс <i>MovieClip</i>. Работа с клипами	344
Встроенные свойства класса <i>MovieClip</i>	344
Координаты. Свойства <i>_x</i> и <i>_y</i>	345
Размеры. Свойства <i>_width</i> , <i>_height</i> , <i>_xscale</i> и <i>_yscale</i>	347
Поворот. Свойство <i>_rotation</i>	352
Прозрачность. Свойство <i>_alpha</i>	352
Имя. Свойство <i>_name</i>	353
Отображение. Свойство <i>_visible</i>	354
Координаты мыши. Свойства <i>_xmouse</i> , <i>_ymouse</i>	355
Кадры. Свойства <i>_totalframes</i> , <i>_currentframe</i> и <i>_framesloaded</i>	358
Свойства клипов-кнопок: <i>useHandCursor</i> , <i>enabled</i> , <i>trackAsMenu</i> , <i>hitArea</i>	361
Создание выпадающего списка	363
Прочие свойства	369
Адресация с помощью предложения <i>with</i>	374
Управление воспроизведением	376
Остановка и запуск воспроизведения	376
Переход между кадрами	378
Управление воспроизведением монтажной линейки	383
Организация навигации внутри фильма	385
Программный реверс анимации	386
Загрузка сетевого ресурса	388
Организация навигации между Web-страницами	391
Динамическое создание клипов	391
Стеки графических объектов Flash	392
Стек документов проигрывателя	394
Присоединение экземпляра. Метод <i>attachMovie()</i>	394
Дублирование экземпляра. Метод <i>duplicateMovieClip()</i>	400
Контроль уровней стека	406
Создание пустого клипа. Метод <i>createEmptyMovieClip()</i>	409
Удаление динамически созданных клипов	409

Проверка состояния загрузки	411
Метод <i>getBytesTotal()</i>	411
Метод <i>getBytesLoaded()</i>	412
Создание предварительного загрузчика	412
Загрузка внешних Flash-документов и изображений	415
Загрузка на уровень стека проигрывателя	415
Организация навигации с использованием динамической загрузки внешних документов	417
Особенности загрузки в клип	418
Создание интерактивного слайд-шоу	420
Использование абсолютных ссылок во внешних документах	423
Загрузка переменных из внешнего источника	425
Удаление загруженных документов и содержимого клипов	427
Преобразование координат и контроль границ клипа	428
Преобразование в глобальные координаты	428
Преобразование в локальные координаты	430
Контроль границ клипа	431
Равномерное распределение клипов	431
Перетаскивание объектов	436
Перетаскивание с помощью метода <i>startDrag()</i>	437
Прекращение перетаскивания	438
Контроль наложения клипов	440
Контроль пересечения ограничивающих рамок клипов	440
Контроль попадания точки в заданную область	443
Глава 13. Динамическое рисование	446
Рисование прямолинейных сегментов	447
Задание начальной позиции рисования	447
Определение атрибутов контура	448
Соединение точек прямолинейным отрезком	448
Удаление графики	449
Создание инструмента "карандаш"	450
Применение сплошной заливки	454
Рисование многоугольников и звезд	457
Применение градиентной заливки	463
Определение матрицы трансформации градиента	465
Рисование криволинейных сегментов	479
Динамическое маскирование	483
Глава 14. Применение динамического и пользовательского текста	488
Создание и настройка параметров текстового блока в рабочей среде	489
Настройка параметров текстового блока в рабочей среде	490
Встроенные и машинно-независимые шрифты	495
Использование текстового блока, созданного в рабочей среде	496
Использование шрифтовых символов	497

Динамическое создание и форматирование текста	500
Класс <i>TextField</i> . Настройка параметров текстового блока	501
Класс <i>TextFormat</i> . Свойства форматирования текста.....	510
Создание всплывающих подсказок	517
Получение информации о параметрах форматирования текста	522
Получение информации о метриках шрифта.....	523
Проверка наличия у пользователя используемого шрифта	526
Организация прокрутки текста	527
Автоматическая прокрутка текста.....	527
Свойства управления вертикальной прокруткой текста.....	528
Свойства управления горизонтальной прокруткой текста.....	529
Контроль прокрутки содержимого текстового блока.....	531
Примеры реализации текстовой прокрутки	532
Контроль выделения и передачи фокуса. Класс <i>Selection</i>	540
Передача фокуса.....	540
Обнаружение носителя фокуса	542
Контроль передачи фокуса.....	543
Определение границ выделения и положения курсора	544
Управление выделением	547
Управление порядком переноса фокуса с помощью клавиши табуляции.....	548
Обработка событий текстового блока	550
Замена содержимого текстового блока	554
Удаление текстового блока.....	557
Форматирование текста с помощью HTML.....	558
Активизация поддержки HTML и вывод текста с применением HTML-форматирования	559
Теги HTML, используемые для форматирования текста.....	560
Поддержка и использование CSS. Класс <i>StyleSheet</i>	572
Создание таблицы стилей.....	573
Определение стилей	576
Добавление стилей в таблицу.....	581
Получение информации об используемых стилях	593
Очистка таблицы стилей.....	595
Загрузка текстовых данных из XML-документа	595
Контроль статического текста. Класс <i>TextSnapshot</i>	601
Создание объекта <i>TextSnapshot</i>	602
Чтение и поиск в статическом тексте	603
Выделение статического текста цветом	605
Глава 15. Класс <i>Math</i>. Выполнение математических вычислений	611
Встроенные свойства. Математические константы	611
Возведение числа в степень и извлечение корня	612
Вычисление логарифма числа.....	613
Определение наибольшего и наименьшего значений.....	614
Определение модуля числа.....	615
Округление значений.....	616
Математическое округление.....	616

Округление до большего целого	617
Округление до меньшего целого	617
Тригонометрические функции.....	617
Применение генератора случайных чисел.....	624
Эффект падающего снега	625
Глава 16. Класс <i>Date</i>. Контроль времени и даты	629
Всемирное скоординированное время и машинное время	629
Создание экземпляра класса <i>Date</i>	631
Вызов конструктора без параметров	631
Вызов конструктора с передачей ему машинного времени.....	631
Вызов конструктора с передачей ему параметров в формате даты.....	632
Определение и установка параметров даты.....	633
Определение местного времени и даты	634
Определение UTC-времени и даты	636
Установка местного времени и даты.....	636
Установка UTC-времени и даты.....	638
Работа с машинным временем.....	638
Создание календаря	639
Глобальная функция <i>getTimer()</i>	644
Глава 17. Классы <i>Mouse</i> и <i>Key</i>. Контроль мыши и клавиатуры.....	648
Контроль отображения курсора и обработка событий мыши.....	648
Управление отображением курсора.....	649
Обработка событий мыши.....	649
Контроль клавиатуры.....	655
События клавиатуры	655
Определение характеристик и проверка состояния клавиши	656
Глава 18. Класс <i>Color</i>. Работа с цветом	664
Цветовая модель RGB.....	665
Цветовая модель HSB	668
Перевод цвета из модели RGB в модель HSB и наоборот	670
Создание объекта <i>Color</i>	677
Окрашивание клипов. Методы <i>setRGB()</i> и <i>getRGB()</i>	678
Формирование каталога цветов на основе безопасной Web-палитры.	
Сортировка по тону	681
Цветовые трансформации. Методы <i>setTransform()</i> и <i>getTransform()</i>	686
Типы цветовых трансформаций	686
Применение методов <i>setTransform()</i> и <i>getTransform()</i>	687
Создание панели синтеза цвета на основе модели HSB.....	691
Создание интерактивного цветового редактора.....	696
Глава 19. Класс <i>Sound</i>. Управление звуком.....	698
Создание объекта <i>Sound</i>	698
Присоединение звука.....	700

Управление воспроизведением звука	701
Запуск воспроизведения	701
Остановка воспроизведения	702
Обработка окончания звука	704
Определение длительности и текущей позиции воспроизведения звука	704
Загрузка внешнего звукового файла	706
Загрузка звука в потоковом и событийном режимах	706
Контроль состояния загрузки звука	707
Проверка окончания загрузки звука	709
Поддержка ID3-тегов	710
Управление уровнем звука	714
Управление балансом каналов	716
Управление выводом каналов	718
Пример программного управления звуком. Интерактивная перкуссия	720
 Глава 20. Управление процессом загрузки внешних документов и изображений.	
Взаимодействие между SWF-документами	727
Класс <i>MovieClipLoader</i>	727
Создание экземпляра класса	728
Методы класса <i>MovieClipLoader</i>	728
Обработка событий	730
Начало загрузки. Событие <i>onLoadStart</i>	731
Поступление данных. Событие <i>onLoadProgress</i>	734
Окончание загрузки. Событие <i>onLoadComplete</i>	734
Инициализация загруженного документа. Событие <i>onLoadInit</i>	735
Ошибка при загрузке. Событие <i>onLoadError</i>	736
Пример управления загрузкой нескольких документов	737
Вопросы политики безопасности, связанные с загрузкой внешних SWF-документов	742
Класс <i>LocalConnection</i> . Взаимодействие между SWF-документами	744
Механизм взаимодействия	744
Инструментарий класса <i>LocalConnection</i>	746
Объявление методов объекта <i>LocalConnection</i> в документе-получателе	746
Открытие соединения. Метод <i>connect()</i>	747
Отправка данных получателю	748
Закрытие соединения	751
Пример организации взаимодействия документов	752
Вопросы политики безопасности, связанные с взаимодействием документов	754
Доменная безопасность	755
Безопасность протоколов	758
 Глава 21. Класс <i>SharedObject</i>. Хранение информации на стороне клиента	759
Понятие об объекте <i>SharedObject</i>	759
Ограничения локального хранения информации	760
Вопросы безопасности локального хранения информации	762

Создание объекта <i>Shared Object</i>	763
Запись и чтение данных в объекте <i>Shared Object</i>	766
Сохранение объекта <i>Shared Object</i>	768
Определение объема локального объекта	771
Удаление локальных данных.....	771
Пример применения объекта <i>Shared Object</i> для хранения пользовательских настроек интерфейса.....	772
Глава 22. Контроль параметров документа и проигрывателя.	
Информация о системе и оборудовании	779
Класс <i>Stage</i> . Контроль размеров и масштабирования окна фильма.....	779
Управление масштабированием фильма.....	779
Управление выравниванием фильма.....	782
Контроль размеров сцены и окна проигрывателя	784
Управление отображением меню проигрывателя	789
Управление автономным проигрывателем. Функция <i>fscommand()</i>	790
Класс <i>ContextMenu</i> . Управление контекстным меню проигрывателя	792
Механизм управления контекстным меню.....	792
Управление стандартным содержимым меню.....	794
Добавление и настройка пользовательских элементов меню.....	798
Примеры управления контекстным меню.....	801
Информация о системе и оборудовании. Объект <i>System.capabilities</i>	804
Глава 23. Класс <i>PrintJob</i>. Печать из Flash-фильма	810
Механизм создания и передачи задания на печать.....	811
Вызов окна с настройками параметров печати.....	812
Отправка задания в спулер печати.....	814
Вывод содержимого на печать.....	816
Применение класса <i>PrintJob</i>	816
Глава 24. Динамическая загрузка и воспроизведение видео.....	824
Формат <i>Flash Video</i> . Конвертирование видео в формат FLV.....	827
Механизм выполнения загрузки FLV-файлов	832
Создание объекта <i>Video</i> . Методы и свойства видеообъекта.....	833
Открытие локального соединения.....	834
Контроль загрузки и воспроизведения FLV-видео. Класс <i>NetStream</i>	835
Инициирование выполнения загрузки	835
Контроль процесса загрузки	836
Установка размера буфера.....	837
Управление воспроизведением	838
Ручной поиск, перемотка	839
Контроль звукового сопровождения.....	840
Обработка событий объекта <i>NetStream</i>	842
Прекращение загрузки.....	846
Создание видеопроигрывателя с динамической загрузкой FLV-файла	846

Часть III. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ.**СОЗДАНИЕ КОМПОНЕНТОВ855****Глава 25. Создание пользовательских классов 857**

Файлы классов. Понятие <i>classpath</i>	858
Объявление класса	861
Создание экземпляра класса.....	863
Пакеты.....	864
Объявление свойств.....	865
Установка слежения за значением свойства	869
Объявление конструктора.....	871
Объявление методов.....	874
Регулирование доступа к членам класса.....	876
Статические члены класса.....	880
Организация считывания и установки значений свойств.	
Определение <i>get/set</i> -методов.....	882
Динамические классы.....	886
Суперклассы и подклассы.....	888
Представление о внутреннем механизме реализации наследования	893
Связывание символа с пользовательским классом.....	895
Связывание символа с классом с помощью окна <i>Linkage Properties</i>	896
Связывание символа с классом с помощью метода <i>Object.registerClass()</i>	898
Директивы <i>#initclip</i> и <i>#endinitclip</i>	899
Интерфейсы	901
Пример создания и применения пользовательских классов	903

Глава 26. Трехмерное моделирование и анимация 918

Понятие о матрицах.....	918
Определение матрицы и ее представление в <i>ActionScript</i>	919
Умножение матриц	920
Класс <i>Calculus</i>	921
Изображение и преобразование трехмерных точек.....	924
Проецирование трехмерных точек на плоскость.....	928
Создание класса <i>Vertex3D</i>	932
Класс <i>Vertex3D</i> . Трехмерные преобразования и визуализация точек	933
Пример реализации: управление центрированным кубом	
со срезанным углом с помощью клавиатуры	945
Пример реализации: управление трехмерной плоскостью	
с произвольным числом точек с помощью мыши	952

Глава 27. Создание пользовательских компонентов 959

Представление о компонентах.....	960
Общий процесс создания компонента.....	962
Ключевые этапы создания компонента.....	962
Проектирование компонента.....	964
Создание клипа компонента.....	967

Объявление класса компонента.....	969
Объявление свойств.....	970
Применение тегов метаданных.....	973
Тег <i>Inspectable</i> . Описание параметров внешнего интерфейса компонента.....	973
Тег <i>InspectableList</i> . Список внешних параметров.....	978
Тег <i>Event</i> . Описание событий компонента.....	979
Тег <i>IconFile</i> . Создание пиктограммы компонента.....	979
Объявление <i>get/set</i> -методов.....	980
Трансляция и обработка событий компонентов.....	988
Инициализация источника событий.....	988
Трансляция события. Событийный объект.....	990
Обработка событий компонентов.....	991
Трансляция пользовательского события <i>onUpdate</i>	997
Инициализация компонента.....	998
Изменение размеров компонента.....	1000
Динамическое присоединение графических элементов компонента.....	1001
Компоновка составляющих элементов.....	1003
Обеспечение функциональности графических элементов компонента.....	1004
Использование режима предварительного просмотра (Live Preview).....	1007
Распространение компонента.....	1008
Компиляция клипа компонента.....	1009
Экспортирование SWC-файла.....	1009
Создание MXP-пакета расширения.....	1010

Глава 28. Обработка исключительных ситуаций. Отладка сценариев.

Тестирование проекта	1014
Обработка исключительных ситуаций.....	1014
Тестирование проекта.....	1028
Управление отображением и воспроизведением фильма.....	1028
Эмуляция загрузки.....	1029
Отчет о размерах элементов фильма.....	1034
Средства отладки сценариев.....	1035
Панель <i>Output</i>	1035
Применение отладчика (панель <i>Debugger</i>).....	1039

ПРИЛОЖЕНИЯ 1051

Приложение 1. Справочник: методы, свойства и обработчики событий основных встроенных классов ActionScript	1053
Класс <i>Array</i> . Создание и обработка массивов.....	1053
Класс <i>Button</i> . Работа с кнопками.....	1054
Класс <i>Color</i> . Работа с цветом.....	1056
Класс <i>ContextMenu</i> . Управление контекстным меню проигрывателя.....	1057
Класс <i>ContextMenuItem</i> . Пользовательские элементы контекстного меню.....	1057
Класс <i>Date</i> . Контроль времени и даты.....	1058

Класс <i>Error</i> . Исключительные ситуации.....	1060
Класс <i>Key</i> . Контроль клавиатуры.....	1060
Класс <i>LocalConnection</i> . Взаимодействие между фильмами.....	1060
Класс <i>Math</i> . Выполнение математических вычислений.....	1061
Класс <i>Mouse</i> . Контроль мыши.....	1063
Класс <i>MovieClip</i> . Работа с клипами.....	1063
Класс <i>MovieClipLoader</i> . Управление процессом загрузки внешних документов и изображений.....	1069
Класс <i>NetConnection</i> . Установка локального соединения для динамической загрузки FLV-файлов.....	1070
Класс <i>NetStream</i> . Динамическая загрузка и воспроизведение FLV-файлов.....	1070
Класс <i>PrintJob</i> . Печать из Flash-фильма.....	1071
Класс <i>Selection</i> . Контроль выделения и передачи фокуса.....	1071
Класс <i>SharedObject</i> . Хранение информации на стороне клиента.....	1072
Класс <i>Sound</i> . Управление звуком.....	1072
Класс <i>Stage</i> . Контроль параметров документа.....	1074
Класс <i>StyleSheet</i> . Поддержка и использование CSS.....	1074
Класс <i>System</i> . Информация о системе и оборудовании.....	1075
Класс <i>TextField</i> . Контроль состояния и параметры текстового блока.....	1075
Класс <i>TextFormat</i> . Динамическое форматирование текста.....	1079
Класс <i>TextSnapshot</i> . Контроль статического текста.....	1080
Класс <i>Video</i> . Работа с видеообъектами.....	1081
Приложение 2. Описание компакт-диска.....	1082
Приложение 3. Виртуальные коды клавиш.....	1088
Приложение 4. Горячие клавиши инструкций ActionScript.....	1090
Приложение 5. Интернет-ресурсы, посвященные Flash.....	1092
Предметный указатель.....	1094

Введение

Если бы строители строили здания так же,
как программисты пишут программы,
первый залетевший дятел разрушил бы цивилизацию.

Второй закон Вейберга

Авторам этой книги часто приходится слышать от своих студентов, начинающих обучение по курсу Macromedia Flash, вопрос о том, насколько велика роль программирования в создании Flash-проектов и является ли владение ActionScript необходимым условием для профессиональной реализации специалиста в области Flash-технологии. Как правило, получив на свой вопрос утвердительный ответ, задающий его вскоре сам понимает, насколько широкие возможности открывает ActionScript, и с увлечением (порой смешанным с некоторой опаской) погружается в эту неизведанную область.

Характерной особенностью Flash-технологии является тесная взаимосвязь графического содержания и программной функциональности. Создание Flash-проекта выполняется как бы в двух разных плоскостях.

В одной плоскости лежит работа, связанная с разработкой визуальных образов, рисованием и созданием анимации. Эта часть работы является прерогативой художника и дизайнера.

В другой плоскости лежит работа по проектированию и реализации функциональности и интерактивного наполнения проекта. Эта часть работы входит в обязанности разработчика, программиста.

Несмотря на разный характер этих видов деятельности, они имеют общие точки соприкосновения, поскольку направлены на формирование единого конечного продукта, успех которого в равной степени определяется и визуальным представлением, и интерактивным содержанием. Поэтому профессионал, который обладает навыками работы с графикой и анимацией и владеет таким мощнейшим средством, как ActionScript, способен самостоятельно решать задачи широкого диапазона и тем самым повышает свою конкурентоспособность как Flash-специалист.

Настоящая книга посвящена изучению новейшей версии языка сценариев ActionScript 2.0 и ориентирована на читателей, имеющих опыт работы с Flash MX 2004 (Professional), которые желают освоить возможности программирования в этой среде.

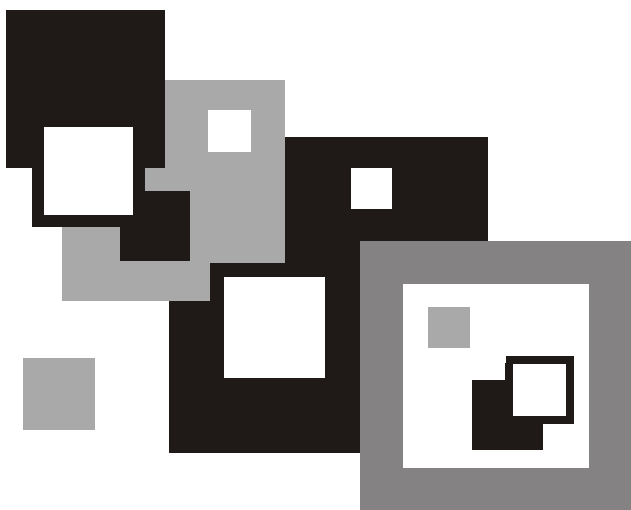
В *главе 1* книги проведен краткий обзор базовых возможностей рабочей среды, связанных с созданием графики, анимации и структурированием документа. Однако в остальных главах при изложении материала предполагается, что читатель уверенно владеет базовыми навыками работы в Flash. Если в процессе чтения читатель столкнется с трудностями, вызванными незнакомыми понятиями или неизученными возможностями рабочей среды Flash, он может обратиться для разъяснения к нашей

предыдущей книге, посвященной основам работы в Flash MX 2004, которая вышла в издательстве "БХВ-Петербург" в 2004 году (серия "Самоучитель").

В настоящей книге приводится детальное описание основных возможностей новейшей версии ActionScript, даются подробные описания приемов и методов эффективной работы, пошаговые инструкции, приводятся многочисленные практические примеры. При написании книги авторы учли свой многолетний опыт преподавания Flash и направили все усилия на то, чтобы сделать повествование полным и вместе с тем предельно ясным и доступным не только профессиональным программистам, но и читателям, не имеющим опыта программирования, но желающим освоить возможности ActionScript и приобрести необходимые навыки работы. Материал в книге организован таким образом, чтобы ее можно было использовать одновременно и как обучающее руководство, и как справочное пособие. Последовательность тем сформирована по принципу от общего к частному. Несмотря на то, что многие элементы языка тесно связаны друг с другом, авторы попытались обеспечить максимальную последовательность изложения и по возможности избежать перекрывания тем и "забегания" вперед. В основном при изложении материала используется принцип, в соответствии с которым в начале каждой темы приводятся подробные теоретические сведения, которые затем иллюстрируются конкретными практическими примерами. В книгу включено большое количество практических примеров, демонстрирующих как модели применения элементов языка, так и создание конкретных проектов. Это даст читателю возможность не просто получить формальные сведения о языке, но и приобрести практические навыки работы с ним. Все исходные файлы практических примеров, обладающих самостоятельной ценностью, помещены на сопровождающий книгу компакт-диск. В тексте глав листинги таких примеров обладают заголовком, выведенным на сером фоне. Все программные коды, приведенные в качестве примеров, снабжены подробными комментариями.

Мы надеемся, что данная книга окажется полезной и поможет читателю освоить, возможно, пока неизвестную, но увлекательнейшую область Flash-технологии и уверенно овладеть таким мощным и гибким инструментом, как язык сценариев ActionScript.

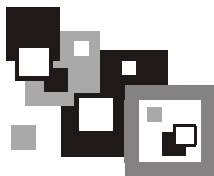
Мы будем признательны вашим отзывам, комментариям и пожеланиям, которые можно направлять по адресу: **book_flash@mail.ru**.



ЧАСТЬ I

ОСНОВЫ ACTIONSCRIPT

Глава 1



Представление о Flash

Если ничто другое не помогает, прочтите, наконец, инструкцию.

Аксиома Кана

Компоненты программы Flash MX 2004 Professional

Рабочую среду Flash MX 2004 условно можно подразделить на следующие компоненты: векторный редактор, монтажная линейка (Timeline), интегрированная среда разработки (IDE) ActionScript.

□ **Векторный редактор** Flash MX 2004 представляет собой инструментарий, позволяющий создавать векторные графические изображения в среде разработки и элементы интерфейса, обеспечивающие удобный доступ к различным инструментам. Среди инструментов векторного редактора можно выделить несколько групп, объединенных общим назначением:

- *инструменты рисования* — **Oval** (Овал), **Rectangle** (Прямоугольник), **Line** (Линия), **Polystar** (Многоугольник), **Pencil** (Карандаш), **Brush** (Кисть), **Pen** (Перо), — с помощью которых непосредственно создаются графические объекты;
- *инструменты выделения* — **Selection** (Выделение), **Lasso** (Лассо), **Subselection** (Частичное выделение), — позволяющие организовать доступ к отдельным элементам или совокупностям графических объектов для их дальнейших преобразований;
- *инструменты работы с цветом* — **Ink Bottle** (Чернильница), **Paint Bucket** (Ведро заливки), **Fill Transform** (Трансформация заливки), **Eyedropper** (Пипетка), — задающие цвет и различные параметры окрашивания графических объектов и их отдельных элементов;
- *инструменты трансформации* — **Free Transform** (Свободное трансформирование) — многофункциональный инструмент, позволяющий выполнять различные преобразования графических объектов;
- *инструмент работы с текстом* **Text** (Текст), — позволяющий снабдить проект информационным содержанием и задать параметры его применения;

- *вспомогательные инструменты* — **Zoom** (Линза), **Hand** (Рука), — с помощью которых осуществляется изменение масштаба просмотра содержимого и перемещение по экрану.

Кроме перечисленных групп инструментов векторный редактор включает в себя целый набор средств (панели, команды меню, диалоговые окна), дающих широкие возможности манипуляции графическими объектами.

- ❑ **Монтажная линейка** (Timeline) — это средство, позволяющее организовать изменение состояния графических объектов во времени, то есть это инструмент, с помощью которого, собственно, и создается анимация. Монтажная линейка Flash включает в себя также палитру слоев, что позволяет решить множество вопросов, связанных с организацией структуры фильма (в частности, с использованием разных планов: переднего, среднего, заднего) и одновременным выполнением нескольких анимационных процессов.
- ❑ **Интегрированная среда разработки ActionScript** — это совокупность инструментария, позволяющего создавать программный код, интегрировать его в структуру документа и осуществлять отладку и тестирование работы программы. Среда разработки ActionScript включает в свой состав следующие элементы.
 - Панель **Actions** — представляет собой редактор ActionScript, который применяется для создания сценариев, включаемых непосредственно в исходный FLA-документ в рабочей среде. Панель **Actions** содержит множество удобных возможностей, таких как всплывающие подсказки, подсветка элементов языка, проверка синтаксиса и анализ ошибок, удобную и разветвленную справочную систему и т. д.
 - Автономный редактор ActionScript — предназначен для создания программного кода, который включается в результирующий SWF-документ на этапе публикации. Автономный редактор применяется при создании пользовательских классов и элементов кода, которые включаются в фильм с помощью директивы `#include`. Интерфейс и функциональные возможности автономного редактора схожи с панелью **Actions**.
 - Панель **Output** и панель **Debugger** — средства отладки программного кода, позволяющие проследить ход выполнения программы и диагностировать ошибки в ее выполнении.

Процесс создания Flash-проекта

В данной книге продукт, созданный в рабочей среде Flash, условимся называть *фильмом*. Это название отражает основную направленность Flash — разработку динамических приложений с использованием анимации. Таким образом, любой проект, будь то анимированный баннер, мультимедиа-презентация или интерактивная заставка, условно будем называть Flash-фильмом.

Технология разработки Flash-фильма включает в себя несколько этапов. В рабочей среде разработчик создает так называемый авторский документ. Этот документ сохраняется с расширением `fla`. FLA-файл содержит графику, звук информацию об

организации внутренней структуры фильма и применяемых разработчиком технологиях, а также включенные в него сценарии ActionScript. Исходный файл всегда имеет существенно больший объем, чем объем результирующего, конечного файла, и никогда не покидает пределы команды разработчиков, если, конечно, нет особой договоренности с клиентом.

Для того чтобы на основе исходного FLA-документа получить "конечный продукт", предназначенный для дальнейшего распространения среди пользователей, необходимо выполнить процедуру, которая называется *публикация*. В результате публикации Flash транслирует исходный документ в один или сразу несколько результирующих форматов.

В первую очередь, это формат SWF — что называется, "родной" для Flash. Файл обладает всей функциональностью, которой был снабжен исходный FLA-файл, и имеет существенно меньший размер. Это достигается за счет реорганизации данных, осуществляемой на этапе публикации (генерирования конечного файла). В ходе этой реорганизации упрощается структура фильма, неиспользованные элементы библиотеки отбрасываются, растровые изображения и звуки подвергаются компрессии, исходный программный код подвергается компиляции и преобразуется в более простую форму — байткод (bytecode). Кроме того, сам конечный файл также может быть подвергнут компрессии. Для распространения Flash-фильма по Сети его, как правило, необходимо разместить на HTML-странице. SWF-файл связывается с HTML-страницей с помощью специальных тегов, описывающих параметры его позиционирования и воспроизведения, после чего оба файла могут быть размещены в сети Интернет или интранет (intranet).

В результате публикации можно также автоматически создать HTML-документ, на котором будет размещен результирующий SWF-файл. Все настройки, на основании которых будут сгенерированы соответствующие HTML-теги, задаются в настройках публикации с помощью диалоговой формы; таким образом, разработчику не нужно заботиться о создании соответствующего HTML-кода вручную.

При создании проекта, который должен распространяться как независимое приложение (например, электронная презентация или обучающая программа), удобно опубликовать исходный файл как исполняемый файл с расширением exe. Такой файл называется проектор (projector) и включает в себя проигрыватель Flash Player, с помощью которого данный файл можно воспроизвести. Это удобная возможность, когда нет уверенности в том, что в файловой системе у конечного зрителя установлен проигрыватель Flash Player.

Системные требования программы Flash MX 2004 Professional

В табл. 1.1 приведены рекомендуемые системные требования Flash MX 2004 Professional для платформ Windows и Macintosh.

Таблица 1.1. Системные требования Flash MX 2004

Системные требования	Платформа	
	Windows	Macintosh
Процессор	600 МГц Intel Pentium III	500 МГц PowerPC G3
Операционная система	Windows 98 SE, Windows 2000, Windows XP	Mac OS 10.2.6
Оперативная память	128 Мбайт (желательно 256 Мбайт)	128 Мбайт (желательно 256 Мбайт)
Свободное дисковое пространство	190 Мбайт	190 Мбайт

Для воспроизведения Flash-фильмов на HTML-страницах проигрывателем Flash Player 7 в зависимости от операционной системы необходимо использовать следующие браузеры (табл. 1.2).

Таблица 1.2. Совместимость браузеров и платформ ОС Windows для Flash Player 7

Браузер	Платформа			
	Windows 98	Windows ME	Windows 2000	Windows XP
Microsoft Internet Explorer 5.x	+	—	+	—
Microsoft Internet Explorer 5.5	+	+	—	—
Microsoft Internet Explorer 6.0	+	—	—	+
Netscape 4.7	+	+	+	—
Netscape 7.x	+	+	+	+
CompuServe 7	—	—	+	+
Mozilla 1.x	+	+	+	+
AOL 8	+	+	+	+
Opera 7.11	+	+	+	+

Обзор интерфейса

Внешний вид окна документа Flash MX 2004, содержащего основные элементы интерфейса, представлен на рис. 1.1. В качестве основных элементов интерфейса можно выделить следующие составляющие: Главное меню, Стандартная панель, Рабочая область, Вспомогательная область, Полоса редактирования, Служебные панели, Па-

нель инструментов, Инспектор свойств, Монтажная линейка, Контроллер, Библиотека, Палитра истории.

- ❑ **Главное меню** включает десять разделов, содержащих все основные команды Flash. Некоторые разделы содержат подменю, позволяющие выбрать необходимое действие.

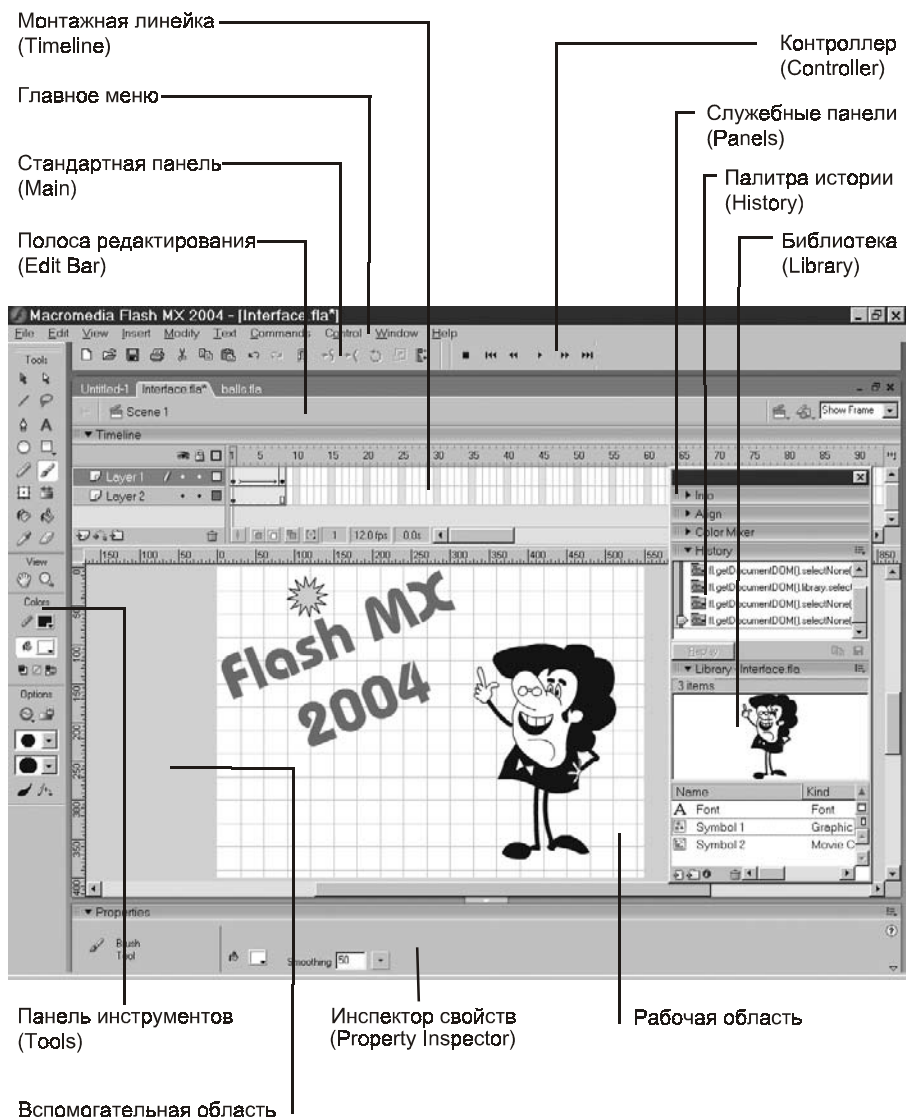


Рис. 1.1. Интерфейс Flash MX 2004

Под главным меню расположены закладки, позволяющие быстро переключаться между открытыми документами Flash. Закладки видны только в случае, если окно документа развернуто.

- ❑ **Стандартная панель.** Чтобы включить отображение стандартной панели, нужно выполнить команду меню **Window>Toolbars>Main**. Стандартная панель содержит ряд кнопок, позволяющих выполнить типовые процедуры (создание, открытие, сохранение, печать документа, команды редактирования, отмены произведенных действий и некоторые операции трансформации графических объектов).
- ❑ **Рабочая область (WorkArea)** по умолчанию представляет собой прямоугольник белого цвета. Все объекты, размещенные в пределах рабочей области, будут видны в конечном фильме. Размер и цвет рабочей области могут быть заданы произвольно.
- ❑ **Вспомогательная область** — область серого цвета, расположенная в рабочем окне за пределами рабочей области. Так же, как и на рабочей, на вспомогательной области можно размещать графические объекты и применять к ним все имеющиеся в распоряжении приемы работы. Однако объекты, находящиеся в пределах вспомогательной области, как правило, не видны в конечном фильме (это зависит от параметров публикации). Вспомогательная область используется для размещения элементов, которые не должны быть видны зрителю (например, объекты, содержащие сценарий), или для имитации эффекта маневрирования камерой (наезд, панорамирование и т. д.), когда в пределах рабочей области находится только часть целого изображения.

В совокупности рабочая область и вспомогательная область называются *сценой* (Stage).

Примечание

Во Flash термин "сцена" имеет несколько значений. Причиной этого является то, что два понятия, используемых в официальной терминологии, — Stage и Scene — переводятся на русский язык совершенно одинаково — "сцена". Конечно, можно было бы найти соответствующий эквивалент, например, "подмости", однако он несколько уводит от основного значения слова. Термин Stage используется, когда имеется в виду физическое размещение объектов в пределах рабочей и вспомогательной областей, а термин Scene применяется, когда речь идет о структурной организации основной монтажной линейки.

- ❑ **Полоса редактирования (Edit Bar)** служит для ориентации в структуре рабочего проекта. На ней отображается информация о том, в какой среде происходит работа в данный момент (сцена, режим редактирования группы, среда редактирования символа). Кроме того, полоса редактирования содержит элементы управления, позволяющие перемещаться между структурными единицами проекта (сценами), получать доступ к редактированию символов и задавать произвольный масштаб просмотра.
- ❑ **Служебные панели.** В рабочей среде Flash MX 2004 имеется целый набор рабочих панелей (Panels), служащих для разнообразных целей дизайна и разработки проектов. Вызов любой служебной панели осуществляется с помощью главного меню **Window**.

Здесь все панели организованы в три группы в соответствии с их назначением.

- **Design Panels** — панели, с помощью которых выполняются задачи по созданию, трансформации и организации графики.
- **Development Panels** — панели, используемые при разработке интерактивности.
- **Other Panels** — остальные панели, позволяющие оптимизировать рабочий процесс создания проекта.

Для того чтобы открыть ту или иную панель, необходимо установить флажок рядом с ее названием. Снятие флажка, соответственно, закрывает панель. Чтобы сразу закрыть все панели, можно использовать команду **Window>Hide Panels** или ее клавиатурный эквивалент <F4>. Повторное выполнение этой команды возвращает все панели на место.

В связи с тем, что число панелей достаточно высоко, и они занимают большую площадь экрана, держать открытыми их все одновременно представляется крайне нецелесообразным. Существует несколько возможностей, позволяющих оптимизировать схему размещения панелей.

Панели могут быть плавающими, парковаться и объединяться в группы.

Для того чтобы открепить панель, необходимо взяться за пиктограмму с изображением двух вертикальных полос в ее левом верхнем углу и перетащить в сторону. Закрыть плавающую панель можно, щелкнув по пиктограмме крестика в ее правом верхнем углу.

Чтобы припарковать панель к границе сцены, нужно взяться за ту же пиктограмму и подтащить панель к одной из границ. Появляющаяся черная рамка укажет на то, что панель будет закреплена. Для включения запрета на прикрепление панелей можно установить флажок **Disable panel docking** меню **Edit>Preferences**.

Для группировки панели необходимо просто припарковать ее к другой панели.

Стандартный вид панели представлен на рис. 1.2. Любая служебная панель может быть развернута и свернута. У развернутой панели черный треугольник в заголовке обращен вправо. Для того чтобы свернуть панель, необходимо щелкнуть по полосе с заголовком в ее верхней части. В свернутом состоянии черный треугольник в левом верхнем углу панели обращен вниз. Повторный щелчок позволяет вновь развернуть панель. Два щелчка по верхней полосе окна (полосе захвата) открепленной панели позволяют минимизировать панель. Для возврата в первоначальное состояние необходимо повторно дважды щелкнуть по полосе захвата.

Любая служебная панель содержит собственное контекстное меню, содержащее набор команд как специфических только для нее, так и стандартных для всех остальных панелей. Чтобы вызвать контекстное меню панели, нужно щелкнуть по пиктограмме в ее правом верхнем углу. Панель при этом должна быть развернута. Контекстное меню каждой панели содержит три стандартных команды **Help** (Вызов справки), **Maximize Panel** (Развернуть панель), **Close Panel** (Закрыть панель).

Для того чтобы развернуть окно сцены по ширине на весь экран, нужно либо закрыть все припаркованные справа панели, либо щелкнуть по пиктограмме стрелки,

расположенной на середине правой грани сцены. Повторный щелчок вернет сцену в исходное положение. Развернуть окно сцены также можно с помощью команды главного меню **Window>Hide Panels** или ее клавиатурного эквивалента <F4>.

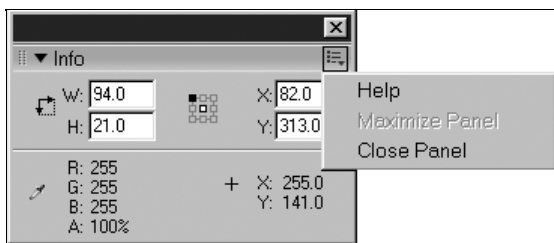


Рис. 1.2. Стандартный вид рабочей панели с ее контекстным меню

Для того чтобы каждый раз не тратить время на организацию панелей, можно сохранить схему их размещения с помощью команды **Window>Save Panel Layout....** В поле **Name** одноименного диалогового окна нужно ввести имя схемы, таким образом можно создать несколько различных схем на все случаи жизни.

Команда **Window>Panel Sets** позволяет выбрать и применить одну из стандартных или пользовательских схем размещения.

❑ **Панель инструментов (Tools)** — содержит инструментарий, предназначенный для создания графики и работы с ней. Панель инструментов состоит из четырех разделов (рис. 1.3).

- **Tools** (Инструменты) — в этом разделе содержатся инструменты для создания графики и работы с ней.
- **View** (Вид) — раздел, содержащий вспомогательные инструменты, используемые для изменения масштаба просмотра и перемещения по экрану.
- **Colors** (Цвета) — блок управления цветом, с помощью которого могут быть заданы цвета элементов векторного объекта.
- **Options** (Модификаторы) — в этом разделе содержатся модификаторы (**Modifiers**), позволяющие задать дополнительные параметры или режимы работы соответствующих инструментов.

Для того чтобы включить (или убрать с экрана) панель инструментов, нужно выполнить команду главного меню **Window>Tools** или воспользоваться ее клавиатурным эквивалентом <Ctrl>+<F2>.

❑ **Инспектор свойств (Property Inspector)** обеспечивает быстрый доступ к различным параметрам настройки и атрибутам выделенного объекта как на сцене, так и на монтажной линейке. С помощью Инспектора свойств можно задавать значения тех или иных параметров объекта или документа, не прибегая к использованию дополнительных меню или панелей (рис. 1.4).

Для управления открытием или закрытием Инспектора свойств используется команда главного меню **Window>Properties** или сочетание клавиш <Ctrl>+<F3>.

Selection (V)
выделение объекта

Line Линия (N)
рисование прямых

Pen Перо (P)
рисование ломаных
и кривых

Oval Овал (O) рисование
овалов и кругов

Pencil Карандаш (Y)
рисование прямых и
кривых линий

Free Transform Свободное
трансформирование (Q)

Ink Bottle Чернильница (S)
окраска и стиль контура
фигуры

Eyedropper Пипетка (I)
копирование заливки и
стиля линии и применение
их к другой фигуре

Hand Рука (H)
передвижение по
служебной области



Subselection

Пустая стрелка (A)
частичное выделение

Lasso Лассо (L)
выделение площади
произвольной формы

Text Текст (T) ввод текста

Rectangle Прямоугольник (R)/

PolyStar рисование
звезд и многоугольников

Brush Кисть (B)

Fill Transform

редактирование заливки (F)

Paint Bucket

Заливка (K)
окраска фигур и
незамкнутых контуров

Eraser

Ластик (E)
стирание линий,
заливок, фигур

Zoom

Лупа (M,Z)
масштаб просмотра
изображения

Блок управления
цветом

Модификаторы
настройки параметров
инструмента

Рис. 1.3. Панель инструментов **Tools**

В левом верхнем углу Инспектора свойств указывается тип выделенного объекта. В зависимости от того, какой именно объект выделен в данный момент, инспектор свойств содержит информацию и настройки текущего документа, контура, векторной формы, группы, растрового изображения, видеоролика, текста, кадра, символа или звукового фрагмента. Если в области выделения одновременно находятся несколько разнородных объектов, то Инспектор свойств указывает на смешанное выделение (**Mixed**).

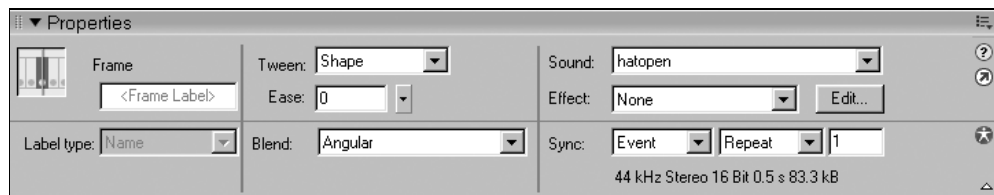


Рис. 1.4. Инспектор свойств при установке свойств кадра

Как и вспомогательные панели, Инспектор свойств может парковаться к краю окна или становиться плавающим; его также можно минимизировать или развернуть. Для выполнения этих операций используются те же приемы, что и для вспомогательных панелей.

Инспектор свойств может иметь более компактный вид или разворачиваться, открывая дополнительные параметры. Для того чтобы развернуть его, необходимо щелкнуть по белому треугольнику в правом нижнем углу, соответственно, эту же пиктограмму можно использовать для приведения к более компактному виду.

- ❑ **Монтажная линейка (Timeline)** — это инструмент, с помощью которого создается анимация. Монтажная линейка включает в свой состав палитру слоев (**Layers**) и кадры (**Frames**), размещенные в пределах соответствующих слоев. Любой объект, созданный в рабочей среде, размещается в определенном кадре слоя монтажной линейки. При создании нового документа монтажная линейка практически пуста, она содержит один слой и один кадр.

Для включения (отключения) монтажной линейки используется команда главного меню **Window>Timeline** или сочетание клавиш **<Ctrl>+<T>**.

Монтажная линейка может быть плавающей или парковаться к границам сцены. Ее можно минимизировать или развернуть.

- ❑ **Контроллер (Controller)** — панель, содержащая кнопки, с помощью которых осуществляется управление монтажной линейкой непосредственно в рабочей среде. Контроллер содержит стандартные команды воспроизведения (**Playback**).

Для включения (отключения) контроллера используется команда главного меню **Window>Toolbars>Controller**. Кроме того, для управления воспроизведением фильма можно использовать главное меню **Control**.

Контроллер может парковаться или становиться плавающим. Для того чтобы закрепить его, нужно взяться курсором за верхнюю грань над элементами управления или два раза щелкнуть по ней и оттащить в сторону. Вернуть панель на место можно, два раза щелкнув по ее заголовку.

- ❑ **Библиотека (Library)** представляет собой хранилище символов (**Symbols**), а также всех импортированных объектов (растровой графики, видеороликов, звуков). Любой внешний объект, будучи импортированным во Flash, попадает в библиотеку. Библиотека позволяет организовать эти элементы, сгруппировав их с помощью папок. Используя возможности библиотеки, можно задать параметры оптимизации растровых изображений и звуков.