

» Idź do

- Spis treści
- Przykładowy rozdział

» Katalog książek

- Katalog online
- Zamów drukowany katalog

» Twój koszyk

- Dodaj do koszyka

» Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

» Czytelnia

- Fragmenty książek online

» Kontakt

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl
© Helion 1991-2008

Flex 3.0. Tworzenie efektywnych aplikacji

Autorzy: Juan Sanchez, Andy McIntosh

Tłumaczenie: Aleksander Lamża

ISBN: 978-83-246-2259-7

Tytuł oryginału: [Creating Visual Experiences with Flex 3.0 \(Developer's Library\)](#)

Format: B5, stron: 408



Zachwyć użytkowników nowym interfejsem!

- Jak tworzyć dynamiczny układ graficzny?
- Jak konstruować i obsługiwać „skórki”?
- Jakie możliwości dają filtry?

Czy wiesz, na co w pierwszej kolejności zwracają uwagę przeciętni użytkownicy Twojej aplikacji? Jeżeli myślisz, że na jej możliwości, jesteś w niemałym błędzie! To interfejs użytkownika jest tym, co wywiera na nich największe wrażenie! Najważniejsze, aby właśnie ten element był estetyczny, ergonomiczny, dynamiczny oraz funkcjonalny! Musi być także szybki i wzbudzać zazdrość innych programistów. Flex to najlepsza odpowiedź na Twoje potrzeby. To niezwykle narzędzie firmy Adobe pozwoli Ci na wygodne tworzenie aplikacji o dużych możliwościach i eleganckim wyglądzie.

Autorzy książki „Flex 3.0. Tworzenie efektywnych aplikacji” skupiają swoje siły na przekazaniu Ci tajników projektowania atrakcyjnego interfejsu użytkownika. W trakcie lektury dowiesz się m. in., jak rozmieszczać komponenty, programować obsługę „skórek” czy zapewniać atrakcyjny wygląd różnym elementom interfejsu. Dowiesz się także, jak dobrać właściwy krój i rozmiar czcionek oraz wykorzystać dostępne efekty wizualne. W IV części książki znajdziesz zbiór ćwiczeń wraz z rozwiązaniami – dzięki nim ugruntujesz i poszerzysz swoją wiedzę. Jeżeli chcesz zachwyć użytkowników Twojej aplikacji nowym interfejsem, to idealna lektura dla Ciebie!

- Podstawy Adobe Flex oraz Adobe AIR
- Tworzenie dynamicznego układu graficznego
- Wykorzystanie arkuszy stylów
- Tworzenie „skórek” – rozwiązania programowe oraz graficzne
- Dostosowanie komponentów wizualnych do Twoich potrzeb
- Dobór odpowiednich krojów i rozmiarów czcionek
- Zastosowanie filtrów
- Wykorzystanie efektów i przejść
- Integracja z Flashem
- Ćwiczenia praktyczne (m.in. tworzenie „skórek”, okno czatu, arkusze stylów)

Opanuj możliwości Adobe Flex w zakresie tworzenia atrakcyjnego interfejsu użytkownika!

Spis treści

Przedmowa	15
Słowo wstępne	17
Podziękowania	21
O autorach	21
Część I Wprowadzenie	23
1 Bogate aplikacje internetowe	25
Odbiorcy	26
Zawartość	26
Reakcje na działania użytkowników	28
Prezentacja	29
Kolor	29
Ruch	30
Typografia	30
Ikonografia	31
Warstwy i głębia	32
Elementy zewnętrzne	33
Podsumowanie	34
2 Adobe Flex i AIR	35
Framework Flex 3.0	36
Dynamiczny układ graficzny	36
Rozbudowa wizualnych komponentów	36
Własny wygląd	36
Potęga ruchu	37
Budowanie aplikacji we Fleksie	38
Flex 3 SDK i kompilator MXML-a	38
Pakiet Adobe Creative Suite	39
Dystrybuowanie fleksowych aplikacji	39
Podsumowanie	40

Część II Framework Flex 3	41
3 Dynamiczny układ graficzny	43
Wprowadzenie do układów graficznych we Fleksie	44
Automatyczne rozmieszczanie	45
Układ liniowy	45
Układ wielowymiarowy	46
Wypełnienie i luki	46
Wykluczanie komponentów z układu automatycznego	47
Korzystanie z kontrolki Spacer	47
Rozmieszczanie bezwzględne	47
Określanie położenia komponentu jako współrzędnych kartezjańskich	48
Określanie położenia komponentu za pomocą układu z wymuszeniami	49
Zmiana rozmiaru komponentów	51
Określanie rozmiaru w pikselach	52
Określanie rozmiaru w procentach	52
Przekazanie zadania określania rozmiaru komponentom	53
Jak wymuszenia położenia wpływają na rozmiar?	53
Korzystanie z przewijania	54
Podsumowanie	57
4 Style	59
Wprowadzenie do właściwości stylów	59
Dziedziczenie	60
Typy danych i formaty	61
Umieszczanie stylów bezpośrednio w kodzie	61
Stosowanie stylów za pomocą CSS	63
Lokalne i zewnętrzne arkusze stylów	63
Selektory CSS	64
Różnice między CSS we Fleksie i języku HTML	65
Pierwszeństwo stosowania stylów	66
Praca ze stylami w języku ActionScript	70
Tworzenie widżetów ze stylami	71
Wykorzystanie klasy StyleManager	72
Definiowanie stylów w widoku Design	73
Widok Design dla CSS	76
Motywy	77
Podsumowanie	78
5 Skórki — rozwiązania graficzne	79
Różnice między grafiką bitmapową a wektorową	81
Osadzanie grafiki we Fleksie 3	83

Stosowanie 9-polowej siatki skalowania	84
Ustawianie 9-polowej siatki	85
Konwencje nazewnictwa w procesie importowania skórek	87
Główne zasady	87
Nazywanie selektorów klas	88
Nazywanie komponentów składowych	88
Korzystanie z kreatora importowania skórek	90
Praca ze skórkami w widoku projektu arkusza CSS	92
Szablony skórek	93
Łączenie narzędzi	94
Podsumowanie	95
6 Skórki — rozwiązania programowe	97
Wprowadzenie do funkcji rysujących	98
Wyjaśnienie sposobu rysowania obrysów	98
Tworzenie skórki programowej	99
Wybór klasy bazowej	100
Implementowanie interfejsów	101
Przesłanie metod	102
Miary w skórkach programowych	104
Podsumowanie	105
7 Komponenty list i nawigacji	107
Komponenty list	108
Komponenty List, HorizontalList oraz TileList	109
Komponent Tree	110
Komponent DataGrid	110
Komponent Menu	111
Elementy renderujące i edycyjne	111
Stosowanie elementów renderujących lub edycyjnych typu drop-in	112
Wpłatanie elementów renderujących lub edycyjnych	113
Tworzenie elementu renderującego jako osobnego komponentu	114
Tworzenie elementu renderującego dla komponentów Tree i Menu	115
Kontrolki nawigacyjne	115
Kontrolka ButtonBar	118
Kontrolka ToggleButtonBar	118
Kontrolka TabBar	118
Kontrolka LinkBar	119
Kontenery nawigacyjne	119
Określanie etykiet i ikon	122
Wykorzystanie komponentów list w nawigacji	122
Podsumowanie	123

8 Wskazówki i kursory	125
Nauka na przykładzie	126
Ikony	127
Wstawianie ikon	127
Ustalanie położenia ikon	128
Ramka aktywnej kontrolki	129
Podpowiedzi	131
Tworzenie podpowiedzi	131
Podpowiedzi związane z błędami	132
Podpowiedzi związane z danymi	133
Podpowiedzi związane z paskiem przewijania	135
Modyfikowanie stylów i skórek podpowiedzi	136
Korzystanie z klasy ToolTipManager	137
Programistyczne tworzenie podpowiedzi	137
Dostosowanie zależności czasowych i ruchu podpowiedzi	138
Kursory	140
Klasa CursorManager	141
Podsumowanie	142
9 Czcionki i tekst	143
Zwróć uwagę na tekst	143
Stosowanie czcionek w aplikacji	144
Czcionki urządzenia	145
Czcionki systemowe	145
Osadzanie czcionek	146
Osadzanie lokalnej czcionki	146
Osadzanie pliku czcionki TTF lub OTF	147
Korzystanie z czcionek osadzonych w pliku SWF	147
Zaawansowane ustawienia antyaliasingu	148
Wyznaczenie zakresu znaków do osadzenia	148
Zakres osadzanych znaków w deklaracji @font-face	149
Zakres osadzanych znaków we Flashu	149
Praca z czcionkami w widoku projektowania CSS	150
Dbanie o jednolity wygląd tekstu	151
Praca z tekstem	152
Rozmieszczanie tekstu	152
Tekst formatowany znacznikami HTML	152
Przypisywanie tekstu formatowanego znacznikami HTML	153
Formatowanie za pomocą znaczników HTML	154
Arkusze stylów w HTML	156
Podsumowanie	157

10 Filtry i mieszanie kolorów	159
Filtry	159
Praca z filtrami	160
Właściwości filtrów	162
Stosowanie filtrów	163
Korzystanie ze znaczników filtrów	164
Korzystanie z tablicy filtrów	165
Stosowanie filtrów za pomocą języka ActionScript	166
Tryby mieszania kolorów	166
Stosowanie mieszania kolorów	168
Podsumowanie	169
11 Efekty i przejścia	171
Korzystanie z efektów i przejść	171
Rozważne używanie efektów	172
Przykłady wzięte z życia	173
Efekty	174
Stosowanie efektów	175
Określanie docelowych komponentów	177
Wyzwalanie efektu	177
Zdarzenia rozpoczynające i kończące efekt	178
Dostosowywanie efektu	179
Ustalanie czasu trwania efektu i liczby powtórzeń	179
Kontrola odtwarzania	180
Efekty złożone	180
Funkcje dynamiki	182
Korzystanie z funkcji dynamiki	183
Tworzenie własnej funkcji dynamiki	184
Wykorzystywanie własnej funkcji dynamiki	185
Wielokrotne wykorzystywanie efektów	186
Efekty powiązane z danymi	186
Domyślne efekty powiązane z danymi	187
Uruchamianie efektów powiązanych z danymi	187
Korzystanie z domyślnego efektu powiązanego z danymi	187
Własne efekty powiązane z danymi	188
Tworzenie własnego efektu powiązanego z danymi	189
Efekty akcji	189
Właściwości filtrów	189
Inne możliwości animacji automatycznej	191
Tweenery	191
Boostworthy AS3 Animation System	192
KitchenSync	192
Przejścia	192

Stany widoku	193
Tworzenie stanu widoku	194
Przesłonięte klasy	195
Dodawanie przejść	195
Tworzenie przejścia	196
Przesłanianie akcji	198
Przejścia i układ graficzny	199
Podsumowanie	200
Część III Nie tylko Flex	201
12 Integracja Flasha z Flexem	203
Dodatek Flex Component Kit	203
Praca z etykietami klatek	204
Wyznaczanie granic	205
Tworzenie komponentów od podstaw	206
Pliki XML z animacjami	206
Podsumowanie	207
13 Personalizacja aplikacji AIR	209
Przygotowanie prostego projektu	210
Zmiana ramki okna	211
Personalizowanie fleksowej ramki okna	212
Własna ramka okna	214
Własne ikony aplikacji	214
Podsumowanie	216
Część IV Ćwiczenia	217
4.1 Ładowanie arkusza stylów w czasie działania aplikacji	219
Podsumowanie	222
5.1 Tworzenie skórek we Flashu	223
Od czego zacząć?	224
Tworzenie grafiki	224
Konwertowanie grafiki na symbole	227
Ustawianie siatki 9-polowego skalowania	228
Umieszczanie we Fleksie elementów graficznych z Flasha	229
Efekty pracy	232
Możliwości rozwijania przykładu	232
Korzystanie z grafiki bitmapowej we Flashu	233
Szablony skórek we Flashu	234
Podsumowanie	234

5.2 Tworzenie skórek w Illustratorze	235
Od czego zacząć?	236
Tworzenie grafiki	236
Importowanie grafiki skóry do Fleksa	240
Ostateczny efekt	242
Dalsze możliwości rozbudowy	243
Tworzenie innych zasobów	243
Korzystanie z szablonów skórek	243
Podsumowanie	244
5.3 Tworzenie skórek w Fireworksie	245
Od czego zacząć?	246
Tworzenie grafiki	246
Przenoszenie skórek do Fleksa	249
Ostateczny efekt	253
Możliwości rozbudowy ćwiczenia	254
Tworzenie innych zasobów	254
Korzystanie z szablonów skórek	254
Podsumowanie	255
5.4 Tworzenie skórek w Photoshopie	257
Od czego zacząć?	258
Tworzenie grafiki	258
Importowanie grafiki skóry do Fleksa	261
Ostateczny efekt	266
Możliwości rozbudowy ćwiczenia	266
Tworzenie innych zasobów	267
Korzystanie z szablonów skórek	267
Podsumowanie	268
6.1 Tworzenie skórek programowych	269
Przygotowanie aplikacji testowej	270
Tworzenie klasy skóry	271
Nakładanie skóry na przyciski	272
Rysowanie grafiki	273
Przesłanie metody updateDisplayList	273
Rysowanie krawędzi	274
Rysowanie wypełnienia przycisku dla poszczególnych stanów	277
Stosowanie stylów	280
Ostateczny efekt	281
Możliwości rozbudowania przykładu	281
Podsumowanie	283

7.1 Tworzenie okna czatu z wykorzystaniem komponentu List	285
Od czego zacząć?	285
Testowanie aplikacji	288
Podsumowanie	292
7.2 Tworzenie galerii fotografii z wykorzystaniem komponentu TileList	293
Podsumowanie	300
9.1 Osadzanie czcionki w pliku SWF za pomocą Flasha	301
Podsumowanie	302
9.2 Umieszczanie czcionek z CSS w pliku SWF	303
Podsumowanie	305
9.3 Tworzenie arkusza stylów dla tekstu HTML	307
Podsumowanie	310
11.1 Stosowanie własnych efektów	311
Podsumowanie	313
11.2 Tworzenie przejść między stanami	315
Podsumowanie	324
12.1 Tworzenie skórek za pomocą narzędzia Flex Component Kit	325
Tworzenie struktury skórk	326
Tworzenie elementów graficznych skórk	328
Tworzenie przejść	330
Ustawianie siatki 9-polowego skalowania	333
Przenoszenie skórk do Fleksa	333
Ostateczny efekt	337
Tworzenie innych elementów	338
Podsumowanie	338
12.2 Korzystanie z szablonów skórek Flasha	339
Podsumowanie	340
12.3 Tworzenie kontenerów za pomocą narzędzia Flex Component Kit	341
Tworzenie prostego kontenera we Flashu	341
Umieszczanie kontenera we Fleksie	343
Podsumowanie	345
12.4 Korzystanie z plików XML ruchu	347
Prace we Flex Builderze	351
Podsumowanie	353

Część V Dodatki	355
A Diagramy stylów i skórek	357
B Ściąga z filtrów	377
C Przydatne zasoby i materiały	381
Projektowanie interfejsu użytkownika	381
Skórki i motywy dla Fleksa	381
Narzędzie Flex Component Kit	382
Aplikacje demonstracyjne	382
Społeczności związane z komponentami Fleksa	383
Biblioteki i frameworki dla Fleksa	383
Dokumentacja	384
Pobieranie aplikacji i dodatków	384
Społeczności	385
Blogi na temat Fleksa	385
Inne blogi	386
Prezentacje rozwiązań fleksowych i AIR	387
Zbiory ikon	388
Zbiory czcionek	389
Zbiory grafiki	389
Skorowidz	391

Adobe Flex i AIR

Adobe Flex jest frameworkiem zaprojektowanym specjalnie na potrzeby tworzenia bogatych aplikacji internetowych (RIA). Jego „sercem” jest Adobe Flash Player, który od dawna służy do dostarczania wyrazistej i zajmującej zawartości zgodnej z głównymi przeglądarkami internetowymi działającymi w kilku systemach operacyjnych. Dzięki odtwarzaczowi Flash Player we Fleksie można przygotować coś więcej niż tylko proste rozwiązania typu „link do strony” i zastosować elastyczne i ciekawe interakcje. Takie podejście pozwala użytkownikom wykonać więcej zadań za jednym razem bez niebezpieczeństwa powolnego odświeżania strony i stopniowego pobierania jej zawartości.

Framework Flex oferuje rozszerzalny zestaw wizualnych i niewizualnych komponentów podobnych do tych, które dostępne są w wielu desktopowych środowiskach dla deweloperów. Są to na przykład siatki danych, kontenery definiujące układ graficzny czy kontrolki zapewniające interakcję. Komponenty te w połączeniu z interesującą zawartością i ruchem mogą posłużyć do zaimplementowania elementów dających duży komfort pracy.

Platforma AIR (ang. *Adobe Integrated Runtime*) pozwala deweloperom przenieść rozwiązania wykorzystujące HTML, AJAX czy Flash do aplikacji desktopowych. Gdy połączy się to z Fleksem, AIR daje możliwość tworzenia aplikacji desktopowych równie łatwo, jak w przypadku bogatych aplikacji internetowych. Te cechy, dołączając jeszcze możliwość uruchamiania aplikacji z menu *Start* lub doka w Mac OS, ustawiają aplikacje AIR na tym samym poziomie, na którym znajdują się najlepsze programy, takie jak Microsoft Office, Adobe Photoshop itp.

Spółeczność skupiona wokół Fleksa gwałtownie urosła, dostarczając dodatkowe biblioteki komponentów, frameworki aplikacji oraz duży wybór innych przydatnych zasobów. Można je potraktować jako środek do rozwijania Fleksa lub początkowy punkt zupełnie nowego kierunku rozwoju. Wszystkie te czynniki sprawiają, że Flex jest atrakcyjnym narzędziem służącym do tworzenia interesujących rozwiązań w świecie RIA.

Framework Flex 3.0

Główna część frameworku Flex składa się z bogatego zestawu wizualnych komponentów, dzięki którym możliwe jest szybkie budowanie i dostarczanie bogatych aplikacji internetowych. Podczas pracy w aplikacji użytkownicy widzą te komponenty i wchodzi z nimi w interakcje. Z tego względu istotne jest, by dobrze poznać ich możliwości, a także ograniczenia. W każdym przypadku masz możliwość zupełnego dopasowania wyglądu i zachowania do oczekiwań użytkowników.

Dynamiczny układ graficzny

Jedną z bardziej przydatnych możliwości oferowanych przez framework Flex jest tworzenie dynamicznych układów graficznych. Wizualna struktura fleksowej aplikacji może przystosowywać się do zawartości i kolejnych zadań użytkowników. Komponenty mogą dynamicznie zmieniać swoje właściwości na wiele sposobów, pomagając prowadzić użytkowników właściwą drogą, zwalniać przestrzeń ekranową lub odsłaniać odpowiednie fragmenty zawartości.

Rozbudowa wizualnych komponentów

Flex jest dostarczany z wieloma wizualnymi komponentami, które mogą posłużyć do utworzenia w pełni funkcjonalnej aplikacji. Komponenty dzielą się na kontenery, które tworzą układ graficzny, oraz kontrolki, których zadaniem jest obsługa interakcji. Wykorzystanie komponentów dostarczonych z nową instalacją Fleksa zarówno do budowania prototypów, jak i „dopieszczania” aplikacji przeznaczonej do wdrożenia jest stosunkowo proste.

Komponenty Fleksa mogą być zmieniane i rozbudowywane w celu utworzenia własnych komponentów. Dzięki włożeniu odrobiny czasu i wysiłku w przygotowania utworzone komponenty mogą mieć modułową budowę i być wykorzystywane w wielu aplikacjach. Opisane możliwości społeczność Fleksa wykorzystała do tworzenia zróżnicowanych komponentów wizualnych udostępnianych wszystkim, którzy chcą z nich skorzystać. Na przykład w bibliotece FlexLib typu open source, zawierającej komponenty interfejsu graficznego, można znaleźć komponent rozszerzający kontrolkę Fleksa TabNavigator o nowe możliwości (patrz rysunek 2.1). Komponenty z biblioteki FlexLib umieszczono na stronie <http://code.google.com/p/flexlib/>.

Własny wygląd

We frameworku Flex poza rozbudową funkcjonalności wizualnych komponentów można również dopasować ich wygląd w taki sposób, by zgadzał się z projektem graficznym, z uwzględnieniem najmniejszych szczegółów. Można dostosować kursory, zdefiniować palety kolorów i przeprowadzić wiele innych operacji. Dzięki temu możliwa jest zmiana właściwości, tworzenie własnych grafik, osadzanie czcionek oraz programowe rysowanie elementów.



Rysunek 2.1. W bibliotece FlexLib dostępny jest komponent SuperTabNavigator (na górze), który rozszerza możliwości domyślnej kontrolki TabNavigator (na dole) o takie funkcje, jak przemieszczanie zakładek, zamykanie ich oraz przemieszczanie poza widoczny obszar okna

Można również wykorzystać integrację z aplikacjami z pakietu Adobe Creative Studio, takimi jak Flash, Fireworks, Photoshop czy Illustrator, aby sprostac nawet bardziej złożonym projektom. W końcu to te modyfikacje wyglądu wspierają inne aspekty dobrze zaplanowanej aplikacji, dając możliwość tworzenia zapadających w pamięć projektów. Poświęcenie czasu na dopracowanie projektu pozwala utworzyć aplikację wyróżniającą się na tle konkurencyjnych rozwiązań (patrz rysunek 2.2).



Rysunek 2.2. Dostosowanie wyglądu flexowych komponentów — w tym przypadku przycisku

Flex udostępnia dla bitmap zestaw różnorodnych efektów, nazywanych filtrami, oraz mechanizmów mieszania kolorów, dzięki którym do elementów flexowej aplikacji można zastosować operacje znane z Photoshopa. Cienie, poświaty czy efekty kolorystyczne mogą posłużyć do wyeksponowania pewnych elementów i przyciągnięcia uwagi użytkowników.

Potęga ruchu

Animacje są podstawowym elementem we Flash Playerze, więc Flex daje możliwość wykorzystania ich na wiele sposobów. Dostępne są efekty oparte na ruchu, dzięki którym komponenty mogą w dowolnym momencie interakcji z użytkownikiem rozświetlić się,

przesunąć czy przeskalować, a także na przykład odtworzyć dźwięk. Wykorzystanie ruchu może dodać do fleksowych aplikacji całkiem nowy wymiar i sprawić, by stały się jeszcze bardziej dynamiczne.

We Fleksie zabawę w odświeżanie strony i ładowanie statycznej zawartości zamieniasz na dynamiczne i ciekawe przejścia między widokami. Układy graficzne mogą się płynnie zmieniać, na właściwości wizualne mogą wpływać działania użytkownika, a widoki mogą zyskać zupełnie inne znaczenie, dzięki czemu użytkownicy łatwiej odnajdą zawartość, której szukają.

Budowanie aplikacji we Fleksie

Do definiowania różnych elementów bogatej aplikacji internetowej we Fleksie jest wykorzystywany język znaczników oparty na XML-u, nazwany MXML. Wszystko, co zostanie opisane w tym języku — od kontrolek interfejsu użytkownika po struktury danych i efekty graficzne — jest tłumaczone przez kompilator na język ActionScript, przy czym to odwzorowanie jest bardzo bezpośrednie. Przycisk opisany w MXML-u staje się w języku ActionScript instancją klasy Button, zainicjowaną parametrami określonymi w znaczniku. Wszystkie klasy utworzone w języku ActionScript również są dostępne w MXML-u.

Wykorzystanie MXML-a do budowy fleksowych aplikacji upraszcza tworzenie układu graficznego, a wiele męczących prac niezbędnych do połączenia elementów aplikacji zostaje przeniesionych na wyższy poziom abstrakcji. Z tych względów MXML jest odpowiedni zarówno do szybkiego tworzenia prototypów, jak i wysokiej jakości aplikacji przeznaczonych do wdrożenia.

Flex 3 SDK i kompilator MXML-a

Aplikacje we Fleksie można tworzyć całkowicie za darmo, korzystając z edytora tekstowego i pakietu Flex SDK (ang. *Software Development Kit*), który zawiera framework Flex oraz kompilator MXML-a (MXMLC). Za jego pomocą można połączyć kod i zasoby w działającą fleksową aplikację. Nie jest to jednak wygodna metoda pracy, zwłaszcza wtedy, gdy wymagane jest szybkie tworzenie aplikacji.

Firma Adobe udostępnia narzędzie Flex Builder, które powstało na podstawie platformy Eclipse. Jeżeli korzystasz już ze środowiska Eclipse, ze strony Adobe możesz pobrać wtyczkę Flex Buildera. Efekt obu opisanych rozwiązań jest taki sam. Narzędzie to jest dostępne na platformy Windows, Mac OS X oraz Linux. Więcej na ten temat można znaleźć na stronie <http://www.adobe.com/products/flex>.

Uwaga

Studenci mogą bezpłatnie korzystać z Flex Buildera. Poza tym firma Adobe oferuje 60-dniową wersję próbną. Więcej szczegółów można znaleźć na stronie www.adobe.com.

Flex Builder udostępnia zestaw wydajnych funkcji pomagających szybko tworzyć nie tylko flexowe aplikacje, ale również rozwiązania dla platformy AIR. Podczas pracy z kodem MXML, ActionScript i CSS dostępne są funkcje kolorowania kodu, podpowiedzi, uzupełniania składni i wiele innych. Można również przełączać się między widokiem źródła (ang. *source*) i projektu (ang. *design*), mając w ten sposób dostęp do wizualnej reprezentacji aplikacji. Jest jeszcze wiele innych funkcji ułatwiających tworzenie aplikacji i pomagających szybko realizować przedsięwzięcia.

Pakiet Adobe Creative Suite

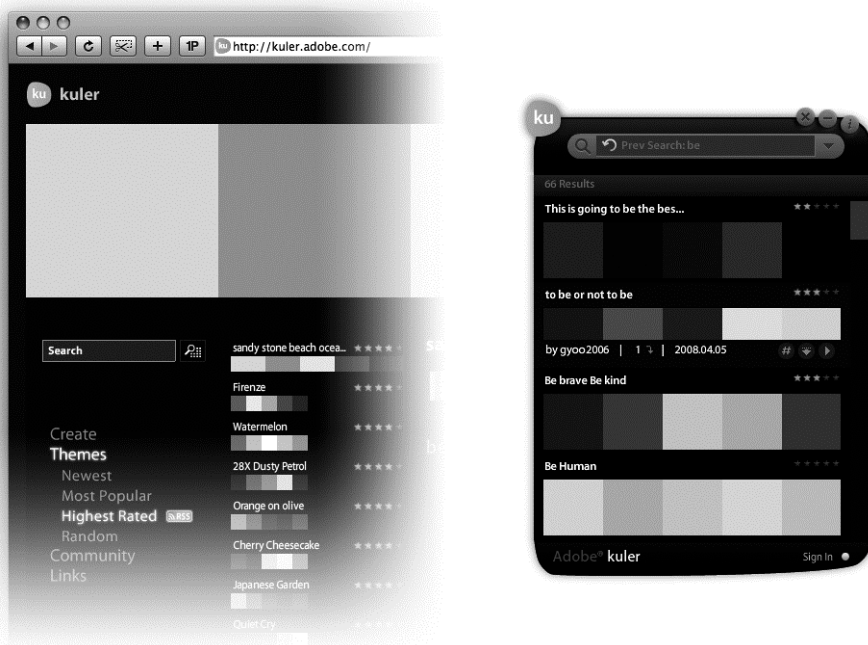
W niektórych sytuacjach będziesz chciał znaleźć rozwiązania pozwalające spersonalizować interfejs użytkownika i wygląd całej aplikacji. Przygotowanie aplikacji z oryginalnym wyglądem pozwala wyróżnić się na tle konkurencji, ugruntować markę i podarować użytkownikom projekt graficzny zwiększający komfort pracy. Zestaw elementów graficznych dla aplikacji można oczywiście utworzyć w różnych programach graficznych, jednak składniki pakietu Adobe Creative Suite dostarczają wiele funkcji bezpośrednio powiązanych z Flexem (patrz rysunek 2.3). W dalszej części książki zapoznasz się z tymi możliwościami i jeżeli chcesz, możesz spróbować popracować w tych programach, korzystając z darmowych wersji testowych dostępnych na stronie <http://www.adobe.com/downloads/>.



Rysunek 2.3. Pakiet Adobe Creative Suite złożony z narzędzi powiązanych z Flexem: Adobe Illustrator, Flash, Fireworks i Photoshop. (Materiały graficzne firmy Adobe zostały przedrukowane za zgodą firmy Adobe Systems Incorporated)

Dystrybuowanie flexowych aplikacji

Fleksowe aplikacje są dystrybuowane w postaci plików SWF uruchamianych w przeglądarce internetowej, w której są odtwarzane we Flash Playerze. Mogą zajmować całe okno lub mieszać się z inną zawartością HTML-a. Ten sam kod źródłowy może posłużyć do zbudowania desktopowej aplikacji uruchamianej za pomocą Adobe Integrated Runtime. Aplikacje AIR działają w systemach Mac OS X, Windows oraz Linux, więc wystarczy raz je utworzyć, by mogły zostać wdrożone na wielu platformach (patrz rysunek 2.4).



Rysunek 2.4. Fleksowe aplikacje mogą być uruchamiane w przeglądarce internetowej lub dostarczane w postaci aplikacji AIR z wykorzystaniem tego samego kodu źródłowego. (Materiały graficzne firmy Adobe zostały przedrukowane za zgodą firmy Adobe Systems Incorporated)

Podsumowanie

Framework Flex udostępnia wydajne środowisko tworzenia aplikacji, dzięki któremu można szybko opracować wysokiej jakości rozwiązania. Podczas tworzenia we Fleksie aplikacji można skorzystać z różnych narzędzi programistycznych i projektanckich. Gdy projekt jest już gotowy do wdrożenia, aplikacja może być uruchamiana w oknie przeglądarki lub może stać się dzięki platformie AIR typową aplikacją desktopową, opartą na tym samym kodzie źródłowym.

W kolejnym rozdziale poznasz wybrane wizualne komponenty oferowane przez Flex, zaczynając od układu graficznego. Podczas projektowania aplikacji układ graficzny jest pierwszą sprawą, o której należy pomyśleć. Określa on powiązania między składnikami aplikacji i w wielu sytuacjach odpowiada przygotowanym wcześniej modelom aplikacji i diagramom interakcji.