

*Gotowe rozwiązania i przykłady aplikacji
dla urządzeń iPhone, iPad i iPod touch*



iOS 5 Programowanie

Receptury



O'REILLY®

Vandad Nahavandipoor

Tytuł oryginału: iOS 5 Programming Cookbook

Tłumaczenie: Robert Górczyński

ISBN: 978-83-246-4994-5

© Helion S.A. 2012.

Authorized Polish translation of the English edition of iOS 5 Programming Cookbook, 1st Edition, ISBN 9781449311438 © 2012 Vandad Nahavandipoor.

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Wydawnictwo HELION dołożyło wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie bierze jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Wydawnictwo HELION nie ponosi również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/ios5re>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzje.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Wprowadzenie	11
1. Podstawy	17
1.0. Wprowadzenie	17
1.1. Utworzenie prostej aplikacji iOS w Xcode	18
1.2. Zrozumienie modułu Interface Builder	21
1.3. Kompilacja aplikacji iOS	24
1.4. Uruchomienie aplikacji iOS w symulatorze	26
1.5. Uruchomienie aplikacji iOS w urządzeniu iOS	28
1.6. Przygotowanie aplikacji iOS do dystrybucji	30
1.7. Deklarowanie zmiennych w Objective-C	35
1.8. Alokacja i tworzenie ciągu tekstowego	37
1.9. Porównywanie wartości w Objective-C za pomocą polecenia if	40
1.10. Implementacja pętli za pomocą polecenia for	43
1.11. Implementacja pętli while	45
1.12. Tworzenie własnej klasy	48
1.13. Zdefiniowanie funkcjonalności klasy	51
1.14. Definiowanie dwóch lub wielu metod o tej samej nazwie	55
1.15. Alokacja i inicjalizacja obiektów	58
1.16. Dodawanie właściwości do klasy	60
1.17. Przejście od ręcznego do automatycznego zarządzania licznikiem użycia	63
1.18. Rzutowanie i technologia Automatic Reference Counting	68
1.19. Delegacja zadań za pomocą protokołów	71
1.20. Ustalenie, czy dostępna jest dana metoda klasy lub egzemplarza	76
1.21. Ustalenie, czy klasa jest dostępna podczas działania aplikacji	79
1.22. Alokacja i używanie liczb	80
1.23. Alokacja i używanie tablic	82
1.24. Alokacja i używanie słowników	86
1.25. Alokowanie i używanie zbiorów	89
1.26. Tworzenie tzw. pakietu	91
1.27. Wczytywanie danych z pakietu głównego	92

1.28. Wczytywanie danych z innych pakietów	95
1.29. Wysyłanie powiadomień za pomocą NotificationCenter	98
1.30. Nasłuchiwanie powiadomień wysyłanych przez NotificationCenter	100
2. Implementacja kontrolera i widoku	105
2.0. Wprowadzenie	105
2.1. Wyświetlanie komunikatów za pomocą UIAlertView	106
2.2. Tworzenie i używanie przełączników za pomocą klasy UISwitch	114
2.3. Wybór wartości za pomocą klasy UIPickerView	117
2.4. Wybór daty i godziny za pomocą UIDatePicker	124
2.5. Implementacja możliwości wyboru wartości z zakresu za pomocą UISlider	129
2.6. Grupowanie opcji za pomocą UISegmentedControl	132
2.7. Prezentowanie widoków i zarządzanie nimi za pomocą klasy UIViewController	138
2.8. Implementacja nawigacji za pomocą klasy UINavigationController	142
2.9. Operacje na tablicy kontrolerów widoku	147
2.10. Wyświetlenie obrazu na pasku nawigacyjnym	148
2.11. Dodawanie przycisku do paska nawigacyjnego przy użyciu UIBarButtonItem	149
2.12. Wyświetlenie wielu kontrolerów widoku za pomocą UITabBarController	156
2.13. Wyświetlanie tekstu statycznego za pomocą UILabel	162
2.14. Akceptacja tekstowych danych wejściowych użytkownika wprowadzanych przez UITextField	166
2.15. Wyświetlanie długich wierszy tekstu w UITextView	173
2.16. Dodawanie przycisków do interfejsu użytkownika za pomocą klasy UIButton	178
2.17. Wyświetlanie obrazów za pomocą UIImageView	182
2.18. Utworzenie przewijanej treści za pomocą UIScrollView	186
2.19. Wczytywanie stron internetowych za pomocą UIWebView	191
2.20. Przedstawianie widoków typu główny-szczegółowy za pomocą UISplitViewController	195
2.21. Włączenie stronicowania w UIPageViewController	201
2.22. Wyświetlanie okna typu Popover za pomocą UIPopoverController	205
2.23. Wyświetlanie paska postępu za pomocą UIProgressView	214
2.24. Nasłuchiwanie powiadomień klawiatury i reagowanie na nie	216
3. Konstruowanie i używanie widoku tabeli	231
3.0. Wprowadzenie	231
3.1. Utworzenie widoku tabeli	231
3.2. Przypisanie delegata widokowi tabeli	233
3.3. Wypełnianie widoku tabeli danymi	235
3.4. Pobieranie i obsługa zdarzeń widoku tabeli	238
3.5. Używanie w komórce widoku tabeli różnego rodzaju przycisków pomocniczych	240
3.6. Tworzenie własnych przycisków pomocniczych w komórce widoku tabeli	242
3.7. Wyświetlanie danych hierarchicznych w widoku tabeli	244
3.8. Usunięcie komórki widoku tabeli za pomocą gestu machnięcia	245
3.9. Utworzenie nagłówek i stopek w widoku tabeli	247
3.10. Wyświetlanie menu kontekstowego w komórce widoku tabeli	255
3.11. Przenoszenie komórek i sekcji w widoku tabeli	259
3.12. Usuwanie komórek i sekcji z widoku tabeli	265

4. Storyboard	275
4.0. Wprowadzenie	275
4.1. Utworzenie projektu z funkcją Storyboard	276
4.2. Dodanie kontrolera nawigacyjnego do aplikacji, w której wykorzystano funkcję Storyboard	278
4.3. Przekazywanie danych pomiędzy ekranami	286
4.4. Dodanie funkcji Storyboard do istniejącego projektu	289
5. Współbieżność	291
5.0. Wprowadzenie	291
5.1. Tworzenie obiektu bloku	296
5.2. Uzyskanie dostępu do zmiennych obiektu bloku	300
5.3. Wywoływanie obiektu bloku	305
5.4. Przekazywanie zadań do technologii Grand Central Dispatch	306
5.5. Wykorzystanie GCD do wykonywania zadań związanych z interfejsem użytkownika	307
5.6. Synchroniczne wykonywanie zadań niepowiązanych z interfejsem użytkownika	311
5.7. Asynchroniczne wykonywanie zadań niepowiązanych z interfejsem użytkownika	313
5.8. Wykonywanie zadań w GCD po upływie pewnego czasu	318
5.9. Tylko jednorazowe wykonanie zadania za pomocą technologii GCD	321
5.10. Grupowanie zadań w GCD	322
5.11. Tworzenie własnej kolejki w technologii GCD	326
5.12. Synchroniczne wykonywanie zadań wraz z operacjami	328
5.13. Asynchroniczne wykonywanie zadań wraz z operacjami	334
5.14. Tworzenie zależności pomiędzy operacjami	340
5.15. Tworzenie licznika czasu	342
5.16. Zapewnienie współbieżności przy użyciu wątków	346
5.17. Wywoływanie metod w tle	351
5.18. Istniejące wątki i liczniki czasu	352
6. Core Location i mapy	355
6.0. Wprowadzenie	355
6.1. Utworzenie widoku mapy	357
6.2. Obsługa zdarzeń widoku mapy	359
6.3. Zaznaczanie na mapie położenia urządzenia iOS	360
6.4. Wyświetlanie pinezek na mapie	362
6.5. Wyświetlanie w widoku mapy pinezek w różnych kolorach	365
6.6. Wyświetlanie własnych pinezek w widoku mapy	370
6.7. Konwersja adresu na współrzędne geograficzne	373
6.8. Konwersja współrzędnych geograficznych na rzeczywisty adres	374
7. Implementacja rozpoznawania gestów	377
7.0. Wprowadzenie	377
7.1. Wykrywanie gestu machnięcia	379
7.2. Wykrywanie gestu obrotu	381
7.3. Wykrywanie gestów przesunięcia i przeciągnięcia	384

7.4. Wykrywanie gestu długiego naciśnięcia	386
7.5. Wykrywanie gestu stuknięcia	389
7.6. Wykrywanie gestu uszczyknięcia	390
8. Sieć, JSON, XML i Twitter	393
8.0. Wprowadzenie	393
8.1. Asynchroniczne pobieranie danych za pomocą klasy NSURLConnection	393
8.2. Obsługa przekroczenia czasu oczekiwania dla połączeń asynchronicznych	396
8.3. Synchroniczne pobieranie danych za pomocą klasy NSURLConnection	397
8.4. Modyfikacja żądania URL za pomocą klasy NSMutableURLRequest	399
8.5. Wysyłanie żądań HTTP GET za pomocą klasy NSURLConnection	400
8.6. Wysyłanie żądań HTTP POST za pomocą klasy NSURLConnection	402
8.7. Wysyłanie żądań HTTP DELETE za pomocą klasy NSURLConnection	404
8.8. Wysyłanie żądań HTTP PUT za pomocą klasy NSURLConnection	406
8.9. Serializacja tablic i słowników do obiektu JSON	408
8.10. Deserializacja obiektu JSON na postać słownika i tablicy	410
8.11. Integracja funkcji serwisu Twitter w Twojej aplikacji	413
8.12. Przetwarzanie danych XML za pomocą klasy NSXMLParser	418
9. Audio i wideo	425
9.0. Wprowadzenie	425
9.1. Odtwarzanie plików audio	425
9.2. Obsługa zakłóceń podczas odtwarzania treści audio	427
9.3. Nagrywanie treści audio	428
9.4. Obsługa zakłóceń podczas nagrywania treści audio	434
9.5. Odtwarzanie treści audio podczas odtwarzania innych dźwięków	435
9.6. Odtwarzanie plików wideo	438
9.7. Tworzenie zrzutu ekranu z odtwarzanego pliku wideo	442
9.8. Uzyskanie dostępu do biblioteki muzycznej	444
10. Książka adresowa	451
10.0. Wprowadzenie	451
10.1. Pobranie odniesienia do książki adresowej	452
10.2. Pobranie informacji z wszystkich kontaktów w książce adresowej	455
10.3. Pobranie właściwości kontaktu książki adresowej	456
10.4. Wstawienie kontaktu do książki adresowej	460
10.5. Wstawianie grupy do książki adresowej	463
10.6. Dodawanie kontaktów do grupy	466
10.7. Przeszukiwanie książki adresowej	469
10.8. Pobieranie i ustawianie obrazu kontaktu w książce adresowej	473
11. Aparat i biblioteka zdjęć	479
11.0. Wprowadzenie	479
11.1. Wykrywanie i sprawdzanie możliwości kamery	481
11.2. Wykonywanie zdjęć aparatem	485

11.3. Nagrywanie filmów kamerą	488
11.4. Przechowywanie zdjęć w bibliotece aplikacji Zdjęcia	491
11.5. Przechowywanie filmów w bibliotece aplikacji Zdjęcia	493
11.6. Pobieranie zdjęć i filmów z biblioteki aplikacji Zdjęcia	495
11.7. Pobieranie zasobów z biblioteki zasobów	496
11.8. Edycja pliku filmu w urządzeniu iOS	502
12. Wielozadaniowość	507
12.0. Wprowadzenie	507
12.1. Wykrywanie obsługi wielozadaniowości	508
12.2. Wykonywanie długotrwałych operacji w tle	509
12.3. Otrzymywanie w tle powiadomień lokalnych	512
12.4. Odtwarzanie dźwięku w tle	519
12.5. Obsługa w tle zmian położenia urządzenia	521
12.6. Zapis i odczyt informacji o stanie w aplikacjach wykorzystujących wielozadaniowość	524
12.7. Obsługa połączeń sieciowych w tle	527
12.8. Obsługa powiadomień dostarczanych aktywowanej aplikacji	530
12.9. Udzielanie odpowiedzi na zmiany w aplikacji Ustawienia	532
12.10. Rezygnacja z wielozadaniowości	534
13. Core Data	537
13.0. Wprowadzenie	537
13.1. Tworzenie modelu Core Data za pomocą Xcode	540
13.2. Generowanie plików klas dla encji Core Data	543
13.3. Tworzenie i zapisywanie danych za pomocą Core Data	545
13.4. Odczyt danych z Core Data	547
13.5. Usuwanie danych z Core Data	549
13.6. Sortowanie danych w Core Data	552
13.7. Uzyskanie dostępu do danych w widoku tabeli	553
13.8. Implementacja związków w Core Data	564
14. Daty, kalendarze i wydarzenia	571
14.0. Wprowadzenie	571
14.1. Pobieranie listy kalendarzy	574
14.2. Dodawanie wydarzeń do kalendarzy	576
14.3. Uzyskanie dostępu do treści kalendarzy	578
14.4. Usuwanie wydarzeń z kalendarza	582
14.5. Umieszczanie w kalendarzach powtarzających się wydarzeń	587
14.6. Pobieranie uczestników wydarzenia	589
14.7. Dodawanie alarmu do kalendarza	594
14.8. Obsługa powiadomień informujących o zmianie wydarzenia	596
14.9. Wyświetlenie kontrolera widoku wydarzenia	598
14.10. Wyświetlanie kontrolera widoku wydarzenia umożliwiające edycję właściwości wydarzenia	603

15. Grafika i animacja.....	607
15.0. Wprowadzenie	607
15.1. Wyświetlenie listy czcionek i ich wczytywanie	614
15.2. Wyświetlanie tekstu	616
15.3. Tworzenie, konfigurowanie i używanie koloru	617
15.4. Wyświetlanie obrazów	622
15.5. Rysowanie linii	626
15.6. Tworzenie ścieżek	631
15.7. Rysowanie prostokątów	635
15.8. Dodawanie cienia do kształtu	638
15.9. Rysowanie gradientów	642
15.10. Zmiana położenia kształtów narysowanych w kontekście graficznym	649
15.11. Skalowanie kształtów w kontekście graficznym	652
15.12. Obracanie kształtów rysowanych w kontekście graficznym	654
15.13. Animacja i przesuwanie widoków	655
15.14. Animacja i skalowanie widoku	663
15.15. Animacja i obrót widoku	665
16. Core Motion	667
16.0. Wprowadzenie	667
16.1. Wykrywanie dostępności przyśpieszeniomierza	668
16.2. Wykrywanie dostępności żyroskopu	670
16.3. Pobieranie danych z przyśpieszeniomierza	671
16.4. Wykrywanie gestu potrząśnięcia w urządzeniu iOS	674
16.5. Pobieranie danych z żyroskopu	677
17. iCloud	681
17.0. Wprowadzenie	681
17.1. Konfiguracja aplikacji w celu zapewnienia obsługi usługi iCloud	682
17.2. Przechowywanie i synchronizacja słowników w iCloud	686
17.3. Tworzenie katalogów i zarządzanie nimi w usłudze iCloud dla aplikacji	691
17.4. Wyszukiwanie plików i katalogów w usłudze iCloud	697
17.5. Przechowywanie dokumentów użytkownika w usłudze iCloud	705
17.6. Zarządzanie stanem dokumentów w usłudze iCloud	718
17.7. Obsługa błędów w dokumentach iCloud	721
Skorowidz	731

4.0. Wprowadzenie

Jako programiści iOS jesteśmy przyzwyczajeni do wykorzystywania kontrolerów widoku. Doskonale wiemy, jak używać kontrolerów nawigacyjnych w celu umieszczania i usuwania ze stosu kontrolerów widoku. Inżynierowie Apple uznali jednak, że to zadanie można wykonywać znacznie łatwiej i w ten oto sposób zaoferowali nową funkcję o nazwie *Storyboard*. Funkcja Storyboard to nowy sposób definiowania połączeń pomiędzy różnymi ekranami w aplikacji. Jeśli przykładowo w aplikacji masz 20 unikalnych kontrolerów widoku opracowanych rok wcześniej i teraz ponownie analizujesz kod źródłowy tej aplikacji, musisz rozszyfrować połączenia pomiędzy tymi kontrolerami widoku. Ponadto musisz spróbować zapamiętać, który będzie używany po podjęciu określonego działania przez użytkownika. To może być bardzo trudne zwłaszcza wtedy, kiedy niezbyt dokładnie udokumentowałeś kod źródłowy. W takich sytuacjach z pomocą przychodzi funkcja Storyboard. Dzięki niej na pojedynczym ekranie możesz przejrzeć lub utworzyć cały interfejs użytkownika aplikacji i połączenia pomiędzy kontrolerami widoku.

Aby wykorzystać możliwości oferowane przez funkcję Storyboard, musisz użyć modułu Interface Builder. Nie przejmuj się, to również jest tematem tego rozdziału.

Podczas używania funkcji Storyboard cały ekran zawierający treść jest nazywany *sceną*. Związek pomiędzy sceną i funkcją Storyboard na platformie iOS można porównać do widoku i kontrolera widoku. W scenie całą treść umieszczasz na ekranie i w całości prezentujesz ją użytkownikowi. Użytkownikowi iPada można zaprezentować więcej niż tylko jedną scenę, ponieważ urządzenie ma znacznie większy ekran.

Funkcja Storyboard zapewnia obsługę przejścia z jednej sceny do innej. W funkcji Storyboard odpowiednikiem umieszczenia jednego kontrolera widoku na innym w kontrolerze nawigacyjnym jest tzw. *segue*. Inny rodzaj przejścia to modalny kontroler widoku, który wjeżdża na scenę od dołu i chwilowo wypełnia ekran. W iPadzie ekran modalny najczęściej jest wyświetlany na środku ekranu, przy czym nienależące do niego fragmenty ekranu są ściemniane. Dzięki temu uwaga użytkownika zostaje skoncentrowana na ekranie modalnym.

4.1. Utworzenie projektu z funkcją Storyboard

Problem

W Xcode chcesz utworzyć projekt wykorzystujący możliwości funkcji Storyboard.

Rozwiązanie

W oknie dialogowym tworzenia nowego projektu zaznacz pole wyboru *Use Storyboard* (zobacz rysunek 4.1) i upewnij się, że tworzona jest aplikacja uniwersalna.



Rysunek 4.1. Opcja *Use Storyboard* w oknie dialogowym tworzenia nowego projektu

Analiza

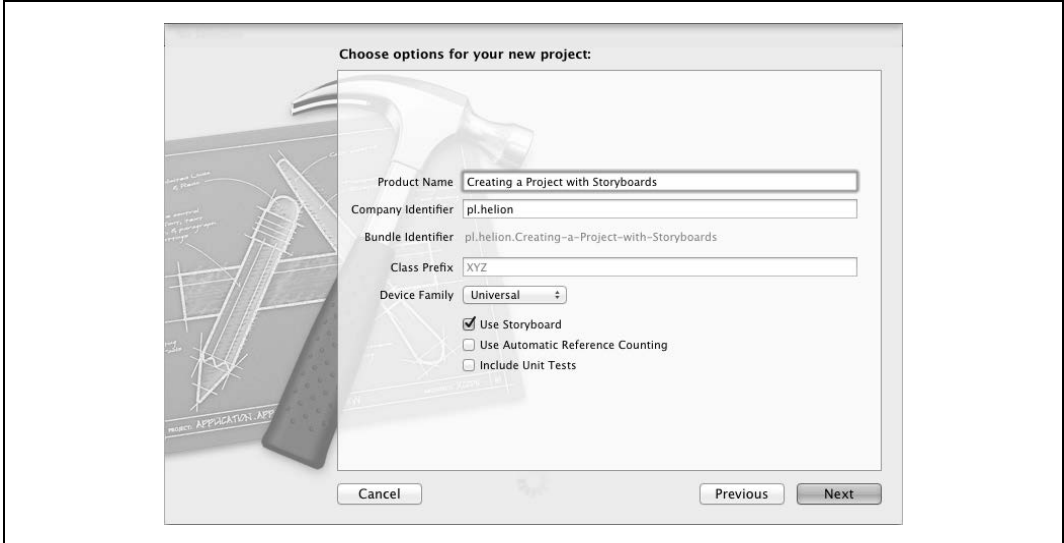
Jeżeli chcesz utworzyć projekt wykorzystujący możliwości funkcji Storyboard, po prostu wykonaj przedstawione poniżej kroki.

1. W Xcode wybierz opcję menu *File/New/New Project...*
2. W wyświetlonym oknie dialogowym upewnij się, że w panelu *iOS* została wybrana kategoria *Application*. Następnie w prawym panelu wybierz *Single View Application* (zobacz rysunek 4.2) i naciśnij przycisk *Next*.



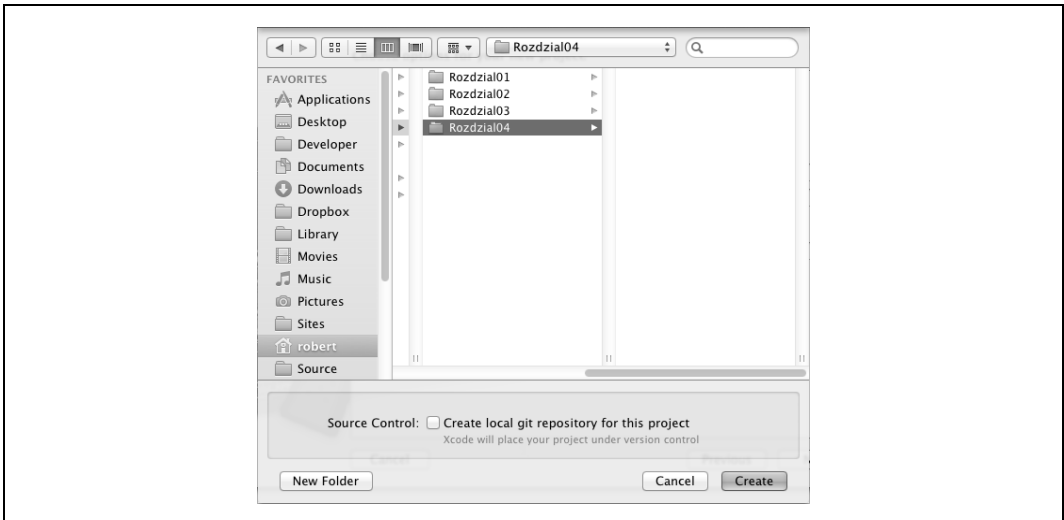
Rysunek 4.2. Tworzenie nowej aplikacji wraz z funkcją Storyboard

3. Teraz podaj nazwę projektu i upewnij się, że tworzysz aplikację uniwersalną. Firma Apple dąży do tego, aby programiści opracowywali aplikacje uniwersalne, jeśli tylko jest taka możliwość. W ten sposób użytkownicy iPada uzyskują dostęp do tych samych aplikacji, którymi cieszą się użytkownicy iPhone'a i iPod touch. W wyświetlonym oknie dialogowym upewnij się, że zaznaczone jest pole wyboru *Use Storyboard*, tak jak na rysunku 4.3. Następnie naciśnij przycisk *Next*.



Rysunek 4.3. W projekcie ma zostać użyta funkcja Storyboard

4. Zostaniesz poproszony o wskazanie katalogu, w którym będzie zapisany projekt (zobacz rysunek 4.4). Po wybraniu odpowiedniego katalogu naciśnij przycisk *Create*. W ten sposób powstał projekt używający funkcji Storyboard.



Rysunek 4.4. Zapis tworzonego projektu na dysku

Jeżeli spojrzysz na pliki utworzone przez narzędzie Xcode (zobacz rysunek 4.5), dostrzeżesz dwa pliki z rozszerzeniem *.storyboard*. Ponieważ aplikacja ma być uniwersalna, narzędzie Xcode utworzyło pliki *.storyboard* zarówno dla iPhone'a, jak i iPada. W ten sposób możesz decydować o wyglądzie aplikacji w obu rodzajach urządzeń.



Rysunek 4.5. Dwa pliki typu Storyboard w aplikacji uniwersalnej

4.2. Dodanie kontrolera nawigacyjnego do aplikacji, w której wykorzystano funkcję Storyboard

Problem

W aplikacji, w której używana jest funkcja Storyboard, chcesz mieć możliwość zarządzania wieloma kontrolerami widoku.

Rozwiązanie

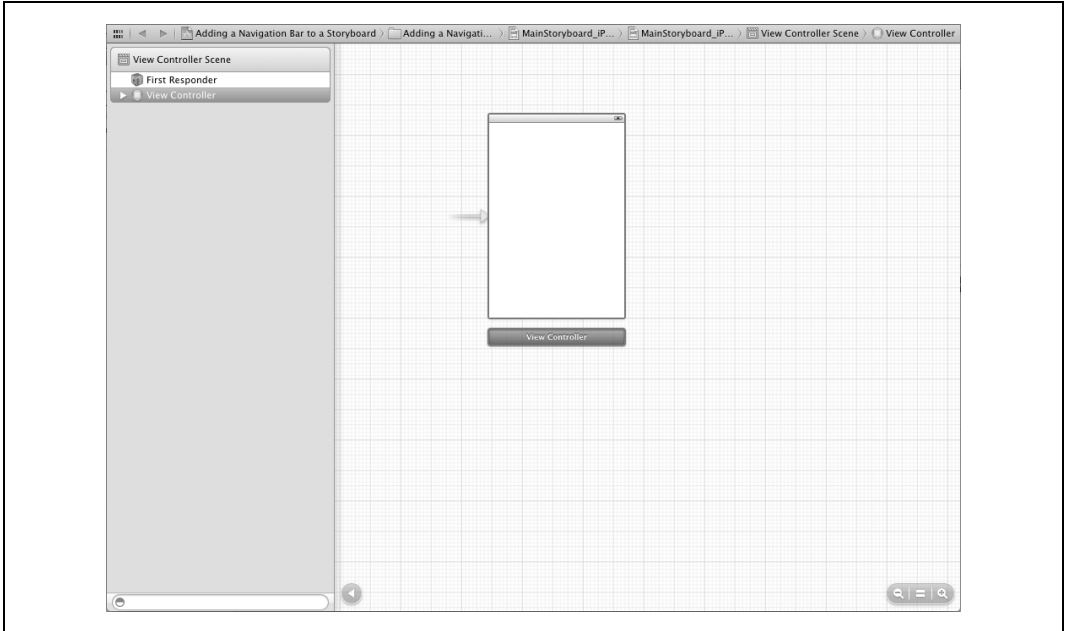
W pliku *.storyboard* ustaw kontroler nawigacyjny jako początkowy kontroler widoku.

Analiza

Jeżeli wykonałeś kroki przedstawione w podrozdziale 4.1 i uruchomisz aplikację w symulatorze iPhone'a, zobaczysz wyświetlony jedynie biały ekran bez żadnego paska nawigacyjnego na górze. Wynika to z faktu, że początkowym kontrolerem widoku pliku *.storyboard* jest kontroler widoku, a nie kontroler nawigacyjny. W celu dodania kontrolera nawigacyjnego do aplikacji z funkcją Storyboard, wykonaj przedstawione poniżej kroki.

1. Kliknij plik *.storyboard* dla iPhone'a, który został utworzony przez szablon projektu w Xcode. W omawianym tutaj przykładzie projekt nosi nazwę *Adding a Navigation Bar to a Storyboard*, natomiast wspomniany plik ma nazwę *Main-Storyboard_iPhone.storyboard*. (Warto w tym miejscu wspomnieć, że domyślna nazwa pliku *.storyboard* tworzonego przez Xcode zależy od używanej wersji narzędzia). Po kliknięciu pliku *.storyboard* zostaniesz przeniesiony do modułu Interface Builder.

2. Po otwarciu pliku *.storyboard* w module Interface Builder po prostu dwukrotnie kliknij pustą przestrzeń na obszarze Storyboard — w ten sposób treść zostanie pomniejszona i uzyskasz większą ilość przestrzeni roboczej (zobacz rysunek 4.6).

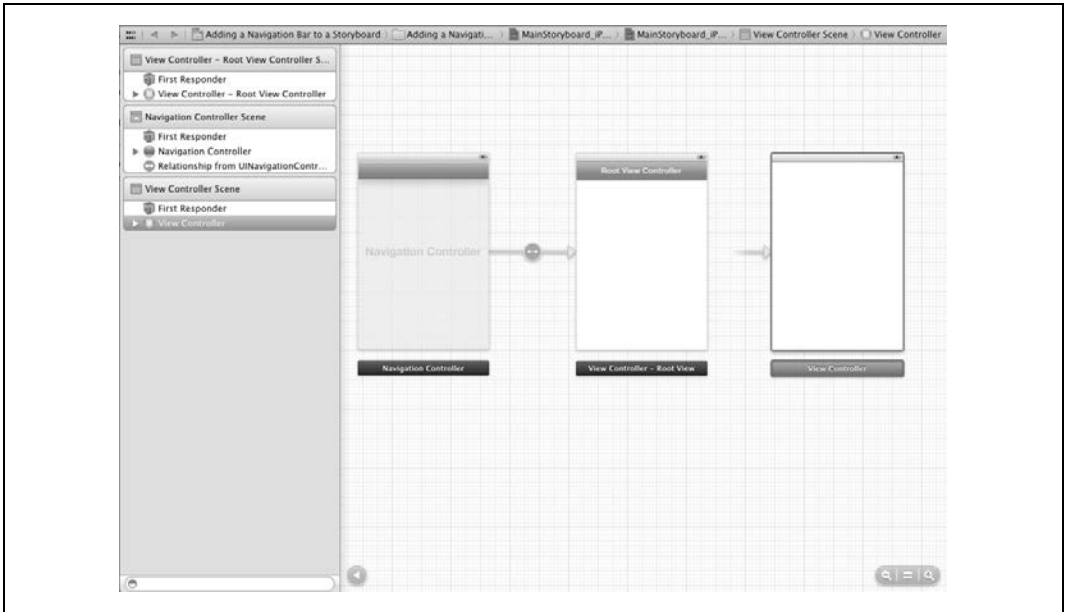


Rysunek 4.6. Pomniejszony widok obszaru Storyboard dla iPhone'a

3. Wybierz w menu opcję *View/Utilities/Show Object Library*.
4. W wyświetlonej bibliotece obiektów wyszukaj obiekt kontrolera nawigacyjnego (zobacz rysunek 4.7), a następnie przeciągnij go i upuść po lewej stronie istniejącego kontrolera widoku pokazanego na rysunku 4.6. Powinieneś otrzymać wynik, podobny do pokazanego na rysunku 4.8.

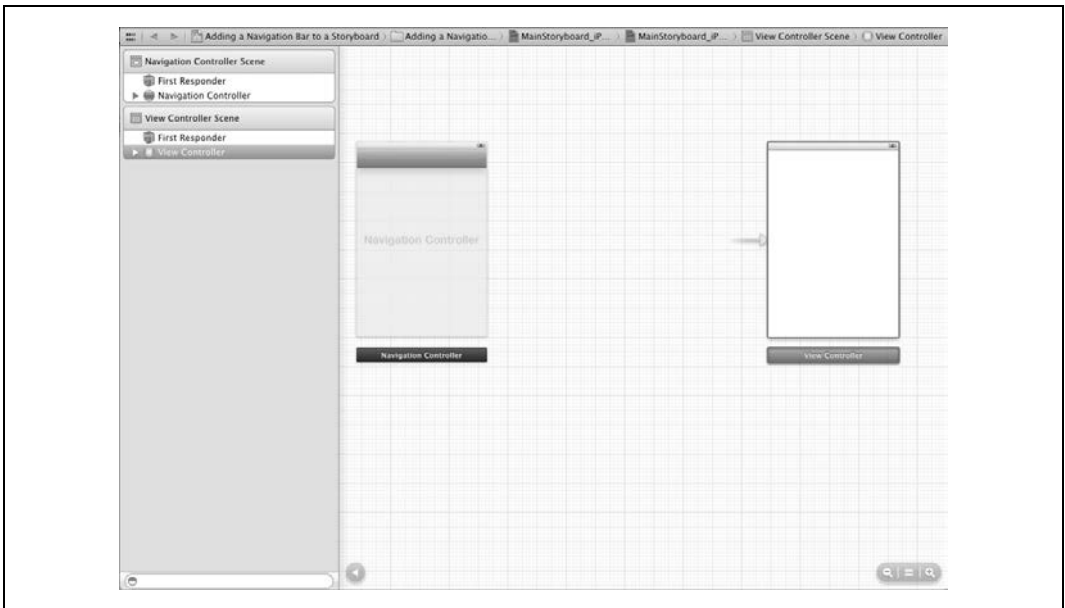


Rysunek 4.7. Obiekt kontrolera nawigacyjnego znajdujący się w bibliotece obiektów



Rysunek 4.8. Kontroler nawigacyjny zawiera własny główny kontroler widoku

5. Jak możesz zobaczyć na rysunku 4.8, kontroler nawigacyjny spowodował dodanie kolejnego kontrolera widoku do interfejsu użytkownika. Twoim zadaniem jest po prostu usunięcie tego dodatkowego kontrolera widoku. W tym celu kliknij go, a następnie naciśnij klawisz *Delete* na klawiaturze. W ten sposób pozostał Ci kontroler nawigacyjny po lewej stronie oraz początkowy kontroler widoku po prawej (zobacz rysunek 4.9).

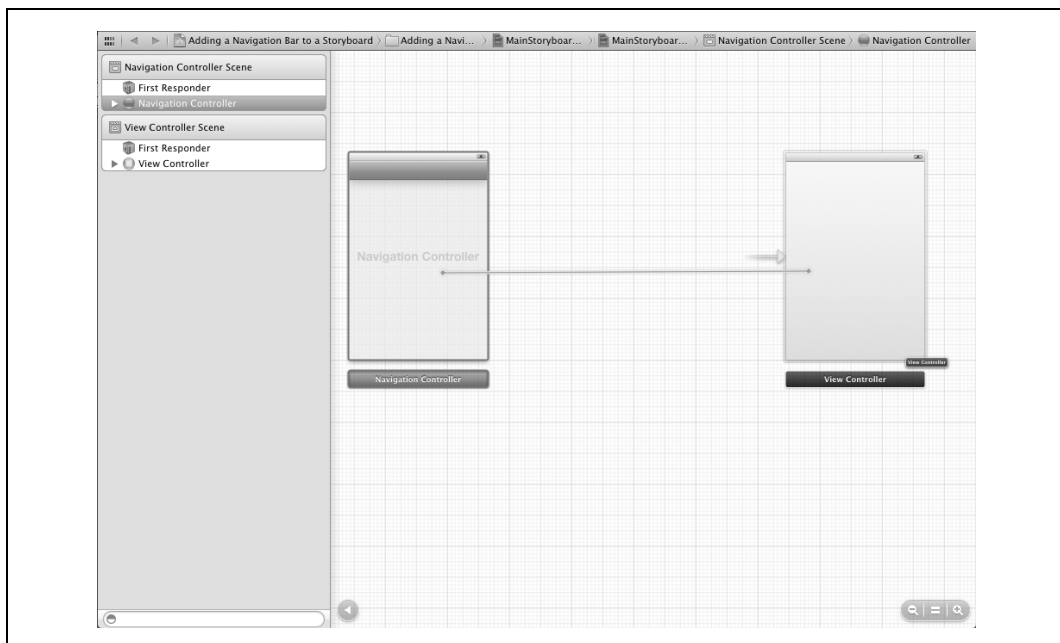


Rysunek 4.9. Główny kontroler widoku, który pojawił się z kontrolerem nawigacyjnym, został usunięty



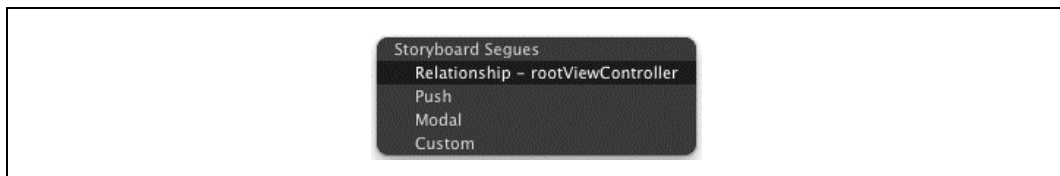
Projekt utworzony w podrozdziale 4.1 bazuje na szablonie *Single View Application*. Ten rodzaj aplikacji domyślnie *nie zawiera* kontrolera nawigacyjnego z oczywistego powodu — to przecież aplikacja z pojedynczym widokiem. Dlatego też, aby zmienić strukturę aplikacji, trzeba do pliku *.storyboard* ręcznie dodać kontroler nawigacyjny.

6. Teraz kliknij obiekt kontrolera nawigacyjnego na scenie Storyboard. Po zaznaczeniu obiektu naciśnij i przytrzymaj klawisz *Control*, a następnie lewym przyciskiem myszy przeciągnij do kontrolera nawigacyjnego znajdującego się po prawej stronie, który początkowo znajdował się w projekcie. W ten sposób od kontrolera nawigacyjnego do kontrolera widoku zostanie narysowana linia, co pokazują na rysunku 4.10.



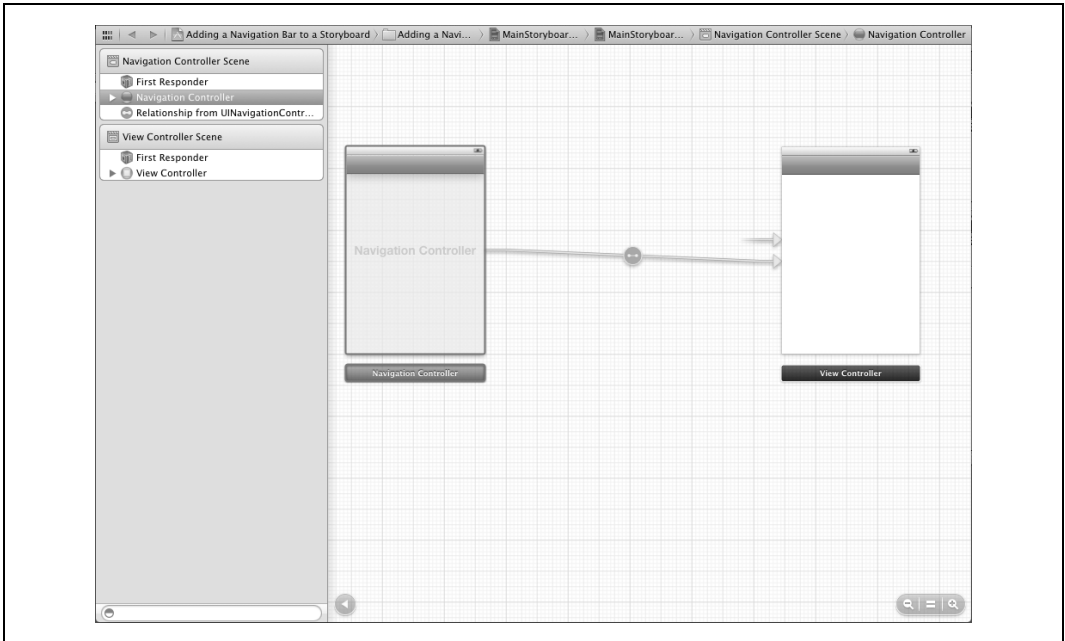
Rysunek 4.10. Połączenie kontrolera nawigacyjnego z początkowym kontrolerem widoku

7. Zwolnij przycisk myszy. Na ekranie zobaczysz menu kontekstowe pozwalające na wybór tworzonych połączenia pomiędzy kontrolerami nawigacyjnym i widoku. W menu kontekstowym wybierz opcję *Relationship - rootViewController* (zobacz rysunek 4.11), po prostu ją klikając.



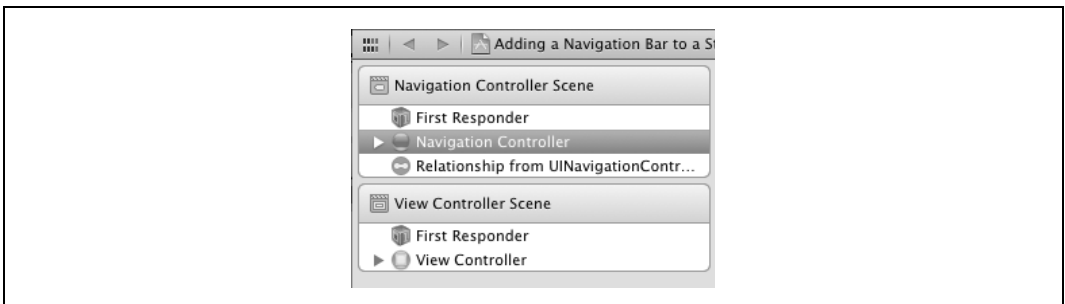
Rysunek 4.11. Ustawienie kontrolera widoku jako głównego kontrolera nawigacyjnego

8. Po wybraniu wymienionej opcji dostrzeżesz, że kontroler nawigacyjny jest dosłownie połączony z początkowym kontrolerem widoku (zobacz rysunek 4.12).



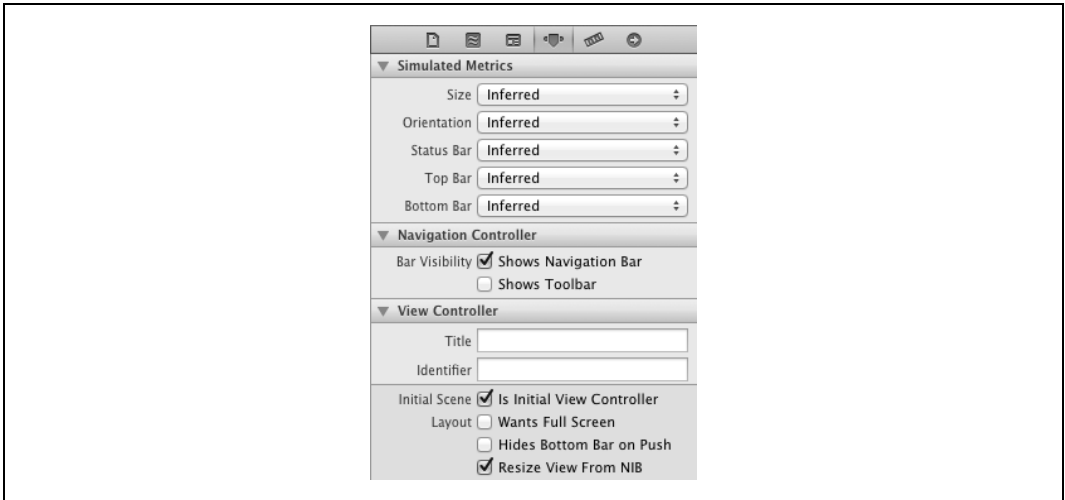
Rysunek 4.12. Kontroler nawigacyjny jest połączony z początkowym kontrolerem widoku

9. Ostatni, ale prawdopodobnie *najważniejszy krok* polega na tym, by sprawić, że kontroler nawigacyjny stanie się pierwszym wyświetlanym przez aplikację kontrolerem widoku. Jeżeli tego nie zrobisz, pierwszym wyświetlanym przez aplikację kontrolerem widoku będzie ten, który początkowo znajdował się w projekcie. Spójrz ponownie na rysunek 4.12. Czy widzisz strzałkę wyświetloną przy lewej krawędzi kontrolera widoku znajdującego się po prawej stronie? Wskazuje ona początkowy kontroler widoku. Aby takim kontrolerem stał się kontroler nawigacyjny, po prostu zaznacz element *Navigation Controller* w panelu *Navigation Controller Scene* w module Interface Builder (zobacz rysunek 4.13). Następnie wybierz opcję menu *View/Utilities>Show Attributes Inspector*. Po wyświetleniu panelu inspektora atrybutów przejdź do sekcji *View Controller* i zaznacz znajdujące się tam pole wyboru *Is Initial View Controller* (zobacz rysunek 4.14).



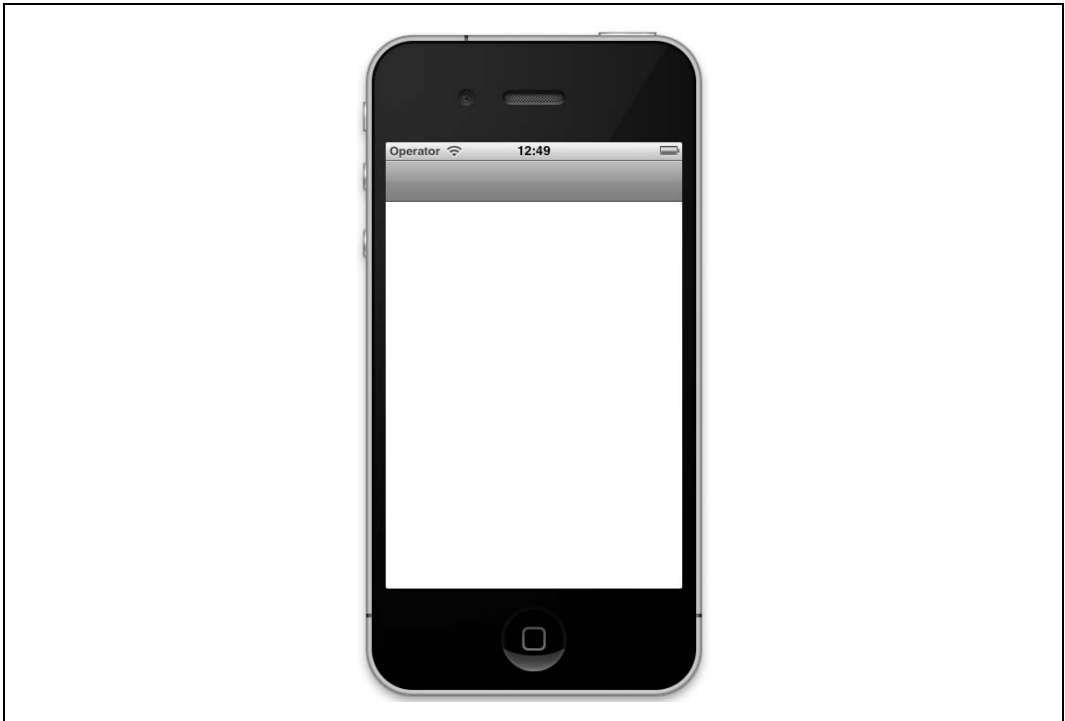
Rysunek 4.13. Zaznaczenie kontrolera nawigacyjnego w module Interface Builder

Jak widzisz, strzałka jest teraz wyświetlona przy lewej krawędzi kontrolera nawigacyjnego, a nie kontrolera widoku położonego po prawej stronie. Jeśli uruchomisz aplikację, przekonasz



Rysunek 4.14. Ustawienie kontrolera nawigacyjnego jako pierwszego kontrolera widoku wyświetlanego w aplikacji

się, że pierwszy wyświetlany kontroler widoku posiada na górze pasek nawigacyjny (zobacz rysunek 4.15). Oznacza to, że pierwszym wyświetlanym kontrolerem widoku jest teraz kontroler nawigacyjny. W kolejnym podrozdziale dowiesz się, jak wykorzystać kontroler nawigacyjny do wyświetlania nowych scen na ekranie.



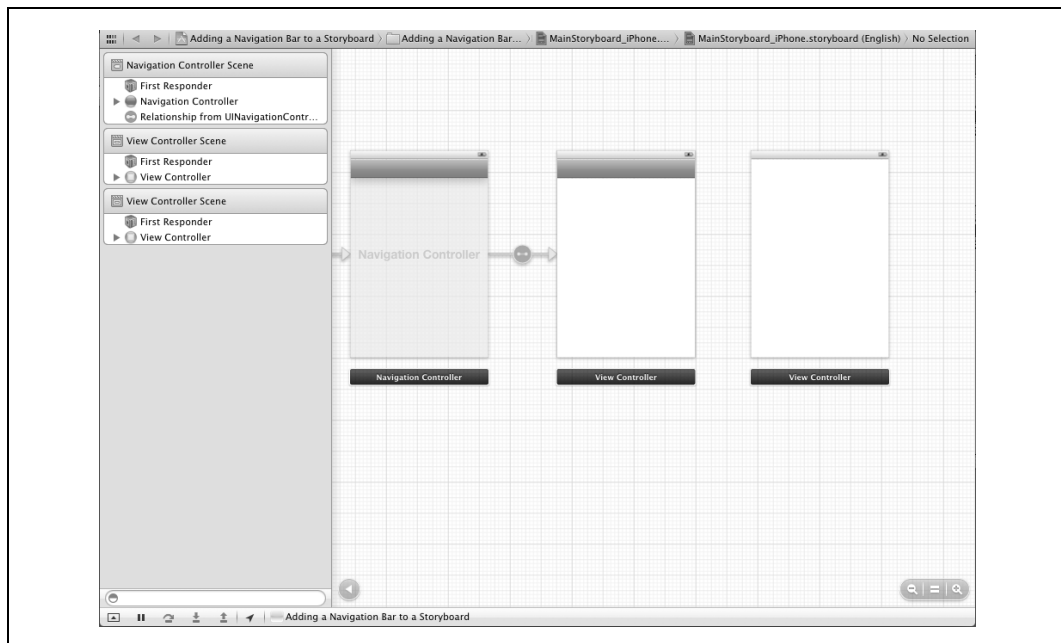
Rysunek 4.15. Pasek nawigacyjny w kontrolerze widoku utworzony przy użyciu funkcji Storyboard

Na tym etapie mamy więc kontroler nawigacyjny wraz z osadzonym kontrolerem widoku. Naszym celem jest wywołanie akcji, a następnie przejście z jednego kontrolera widoku do innego — tę operację przejścia inżynierowie Apple określili mianem *segue*. A zatem w kontrolerze widoku umieścimy przycisk, którego naciśnięcie spowoduje umieszczenie na stosie nowego kontrolera widoku. Brzmi nieźle, prawda? Zabieramy się do pracy.

1. Powróć do pliku *.storyboard*.
2. W bibliotece obiektów znajdź obiekt kontrolera widoku (zobacz rysunek 4.16), a następnie przeciągnij go na scenę i upuść po prawej stronie istniejącego kontrolera widoku (zobacz rysunek 4.17).

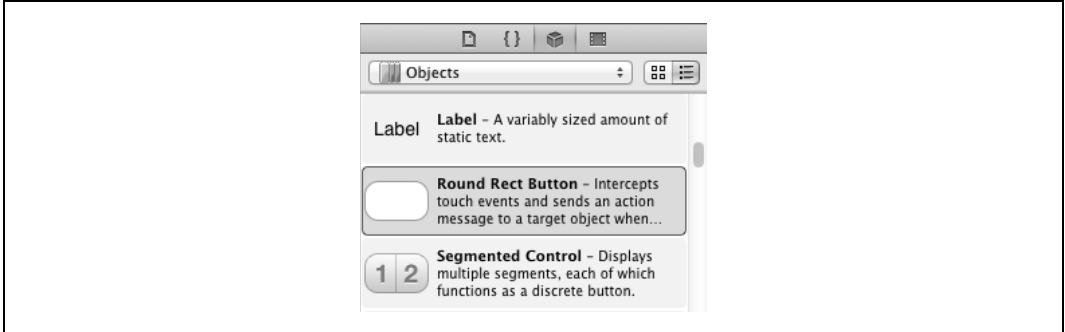


Rysunek 4.16. Obiekt kontrolera widoku (View Controller) w bibliotece obiektów

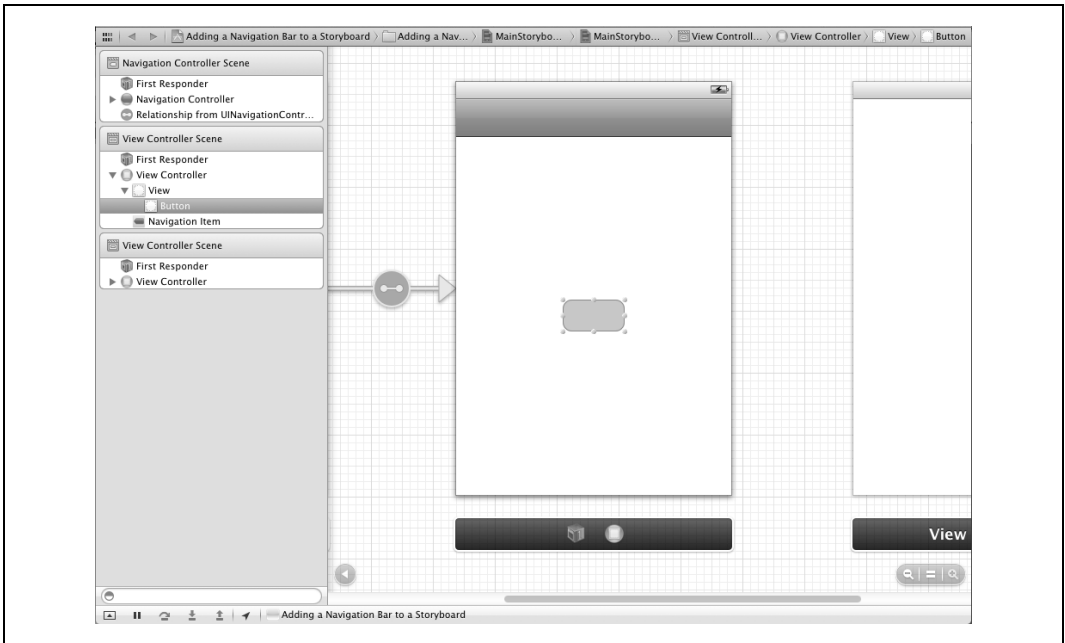


Rysunek 4.17. Umieszczenie na scenie nowego kontrolera widoku

3. W bibliotece obiektów znajdź obiekt przycisku (zobacz rysunek 4.18), a następnie przeciągnij go i upuść w pierwszym kontrolerze widoku (zobacz rysunek 4.19). Zwróć uwagę, że po zmniejszeniu powiększenia moduł Interface Builder nie pozwala na umieszczenie przycisku w kontrolerze widoku. Musisz dwukrotnie kliknąć pusty obszar sceny, zwiększając w ten sposób stopień powiększenia, zanim Interface Builder pozwoli Ci na umieszczanie komponentów interfejsu użytkownika w kontrolerach widoku.

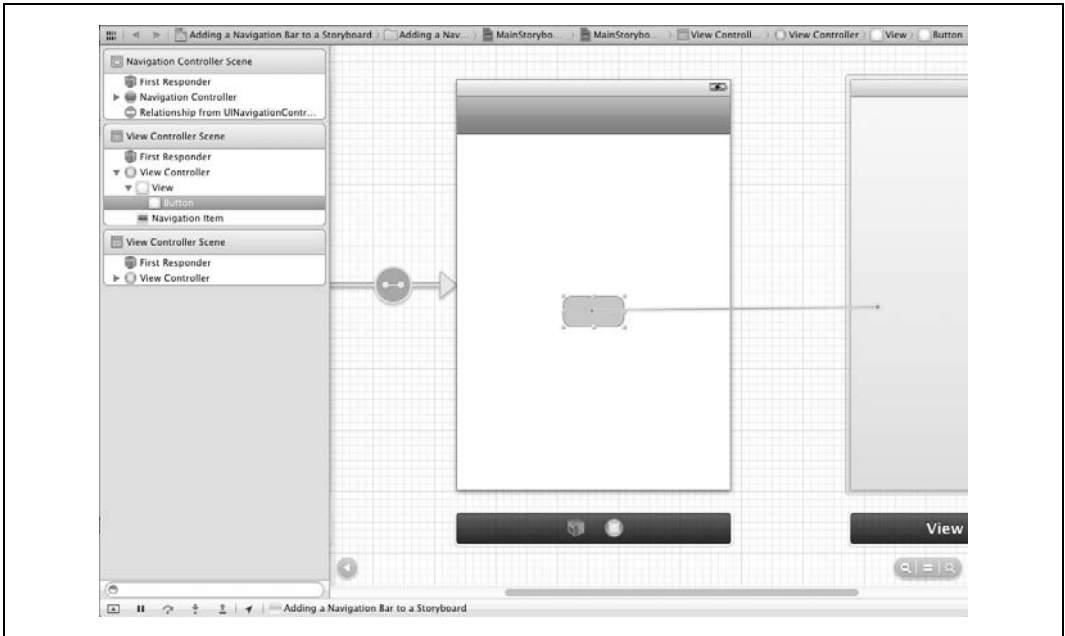


Rysunek 4.18. Obiekt przycisku (Round Rect Button) w bibliotece obiektów

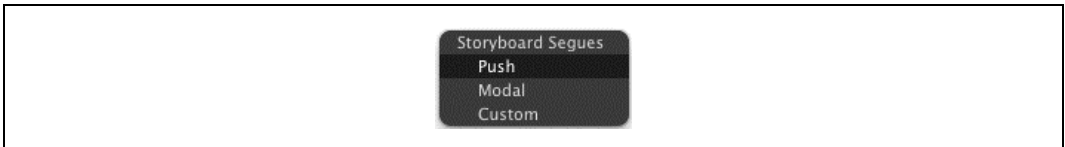


Rysunek 4.19. Przycisk został umieszczony w pierwszym kontrolerze widoku

4. Zaznacz przycisk, naciśnij i przytrzymaj klawisz *Control*, a następnie przeciągnij lewym przyciskiem myszy od dodanego przed chwilą przycisku w kontrolerze widoku do drugiego kontrolera widoku (zobacz rysunek 4.20).
5. Zwolnij przycisk myszy oraz klawisz *Control*. Na ekranie zobaczysz menu kontekstowe, podobne do pokazanego na rysunku 4.21. Kliknij opcję *Push*.



Rysunek 4.20. Połączenie przycisku z drugim kontrolerem widoku



Rysunek 4.21. Utworzenie przejścia typu segue

Jeżeli teraz spojrzysz na scenę, przekonasz się, że pierwszy kontroler widoku jest połączony z drugim (zobacz rysunek 4.22).

Po uruchomieniu aplikacji i naciśnięciu przycisku wyświetlanego przez pierwszy kontroler widoku nastąpi automatyczne umieszczenie drugiego kontrolera widoku na stosie kontrolerów widoku. Gdy na ekranie pojawi się drugi kontroler widoku, na pasku nawigacyjnym zobaczysz przycisk pozwalający na powrót do pierwszego kontrolera widoku. Naciśnięcie tego przycisku nawigacyjnego spowoduje powrót do pierwszego kontrolera widoku.

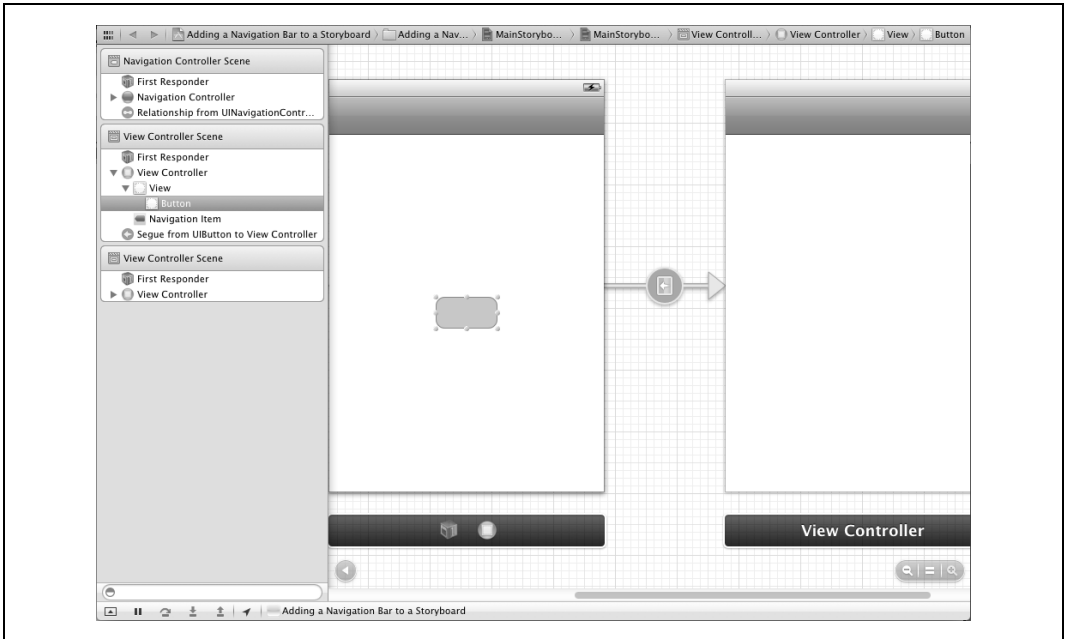
Zobacz także

Podrozdział 4.1.

4.3. Przekazywanie danych pomiędzy ekranami

Problem

Za pomocą funkcji Storyboard chcesz przekazywać dane pomiędzy ekranami.



Rysunek 4.22. Pierwszy kontroler widoku został połączony z drugim

Rozwiązanie

Użyj obiektów przejścia *segue*.

Analiza

Obiekt przejścia *segue* to obiekt podobny do innych obiektów stosowanych w języku Objective-C. W celu przeprowadzenia przejścia pomiędzy ekranami podczas działania aplikacji środowisko uruchomieniowe Storyboard tworzy obiekt będący egzemplarzem klasy `UIStoryboardSegue`. Aby rozpocząć operację przejścia, bieżący kontroler widoku (ten, który zostanie usunięty ze stosu przez operację przejścia) otrzymuje wiadomość `prepareForSegue:sender:`. Jej parametr `prepareForSegue` jest typu `UIStoryboardSegue`. Jeżeli chcesz przekazać jakiegokolwiek dane pomiędzy bieżącym kontrolerem widoku i tym, który zostanie następnie wyświetlony na ekranie, musisz to zrobić w metodzie `prepareForSegue:sender:`.



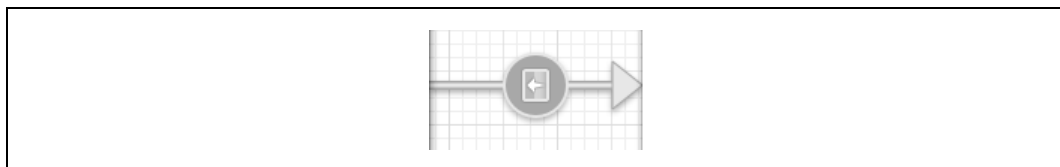
Aby omawiany tutaj przykład miał sens, wcześniej musisz wykonać kroki przedstawione w podrozdziale 4.2 i utworzyć dwa kontrolery widoku wewnątrz kontrolera nawigacyjnego.

W pierwszym kontrolerze widoku zaimplementujemy metodę `prepareForSegue:sender:`.

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    NSLog(@"Kontroler źródłowy = %@", [segue sourceViewController]);
    NSLog(@"Kontroler docelowy = %@", [segue destinationViewController]);
    NSLog(@"Identyfikator Segue = %@", [segue identifier]);
}
```

Jeżeli teraz uruchomisz aplikację, w oknie konsoli zobaczysz wyświetlone komunikaty. Zwróć uwagę na wartość `nil` identyfikatora przejścia *segue*. Każde przejście *segue* posiada własny, unikalny identyfikator. Ponieważ na scenie może być zdefiniowanych więcej przejść *segue*, dobrym rozwiązaniem jest nadawanie im unikalnych identyfikatorów w celu łatwego wykrywania tych przejść w kontrolerach widoku i podejmowania odpowiednich działań.

Obiekt przejścia *segue* w module Interface Builder to połączenie pomiędzy dwoma ekranami. Na rysunku 4.23 pokazuję przejście *segue* w postaci strzałki umieszczonej pomiędzy pierwszym kontrolerem widoku znajdującym się po lewej stronie i drugim kontrolerem widoku, który jest po prawej stronie sceny.



Rysunek 4.23. Obiekt przejścia *segue* w module Interface Builder

Aby utworzyć identyfikator przejścia *segue*, wykonaj przedstawione poniżej kroki.

1. W module Interface Builder zaznacz obiekt przejścia *segue*, po prostu go klikając.
2. Wybierz opcję menu *Show/Utilities/Show Attributes Inspector*.
3. W panelu inspektora atrybutów odszukaj pole tekstowe zatytułowane *Identifier*, a następnie wprowadź w nim identyfikator, który chcesz nadać przejściu *segue*.

Podczas działania aplikacji środowisko uruchomieniowe wywołuje metodę `prepareForSegue:` ↪ `sender:` w bieżącym kontrolerze widoku w celu przygotowania się do przeprowadzenia operacji przejścia. Docelowy kontroler widoku zawiera już zainicjalizowany obiekt przejścia *segue*. Teraz masz możliwość przekazania dowolnych danych do docelowego kontrolera widoku. Dane możesz określić bezpośrednio we właściwości docelowego kontrolera widoku lub przekazać je poprzez wywołanie metody w tym kontrolerze widoku. Wybór należy do Ciebie. W omawianym tutaj kodzie drugi kontroler widoku jest klasą `SecondViewController`, a identyfikator przejścia *segue* to `SimpleSegueToSecondViewController`.

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    NSLog(@"Kontroler źródłowy = %@", [segue sourceViewController]);
    NSLog(@"Kontroler docelowy = %@", [segue destinationViewController]);
    NSLog(@"Identyfikator Segue = %@", [segue identifier]);

    if ([[segue identifier] isEqualToString:@"SimpleSegueToSecondViewController"])
    {
        SecondViewController *viewController = [segue destinationViewController];
        viewController.dataModel = ...; /* Miejsce na kod wymaganych operacji. */
    }
}
```

W powyższym fragmencie kodu `dataModel` to hipotetyczna właściwość zdefiniowana i zaimplementowana w docelowym kontrolerze widoku dla tworzonego przejścia *segue*. Docelowy kontroler widoku jest egzemplarzem klasy `SecondViewController` utworzonej w projekcie. Celem przykładu było pokazanie, jak możesz przygotować kontrolery widoku do przeprowadzania przejść *segue* i przekazywania danych docelowemu kontrolerowi widoku.

Zobacz także

Podrozdział 4.2.

4.4. Dodanie funkcji Storyboard do istniejącego projektu

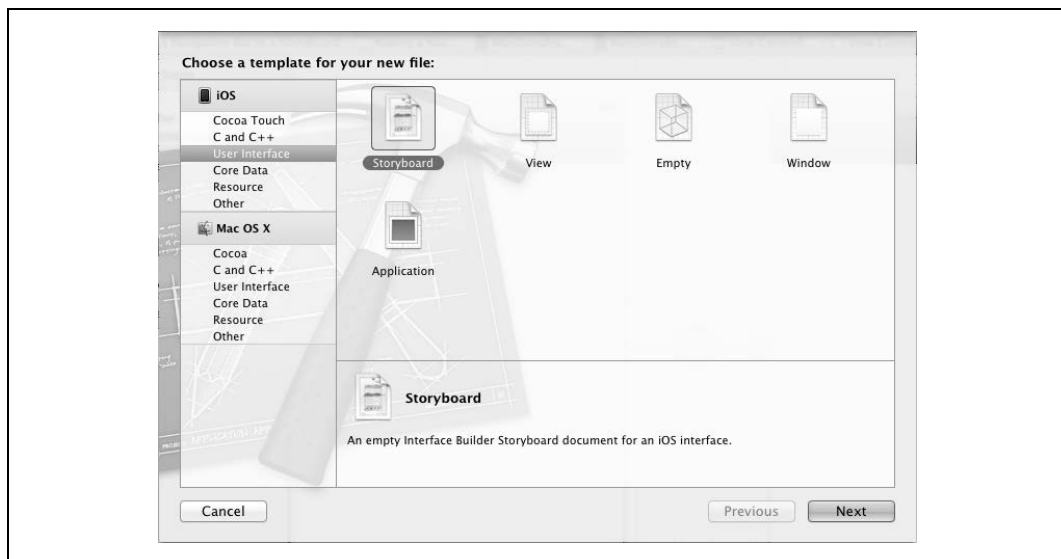
Problem

Masz przygotowaną aplikację, w której nie użyłeś funkcji Storyboard. Teraz jednak w tej aplikacji chciałbyś skorzystać z funkcji z Storyboard, zamiast ręcznie zajmować się obsługą ekranów.

Rozwiązanie

Aby użyć funkcji Storyboard w istniejącej aplikacji, wykonaj przedstawione poniżej kroki.

1. Wybierz opcję menu *File/New/New File...*
2. W wyświetlonym oknie dialogowym upewnij się, że wybrałeś kategorię *User Interface* w panelu *iOS* po lewej stronie. Następnie w prawym panelu wybierz opcję *Storyboard* (zobacz rysunek 4.24) i kliknij przycisk *Next*.



Rysunek 4.24. Dodanie pliku typu Storyboard do istniejącego projektu

3. W kolejnym oknie dialogowym określ urządzenie docelowe dla tworzonego pliku funkcji Storyboard. Jeżeli aplikacja jest przeznaczona tylko dla iPhone'a lub iPada, wybierz odpowiednią wersję. Podczas tworzenia aplikacji uniwersalnej musisz wybrać jedno urządzenie, utworzyć plik funkcji Storyboard, a następnie powrócić, wybrać drugie urządzenie i utworzyć dla niego plik funkcji Storyboard. Gdy to zrobisz, naciśnij przycisk *Next*.

4. Wybierz miejsce zapisu pliku `.storyboard`. W omawianym przykładzie mam aplikację uniwersalną bazującą na szablonie *Single View Application*. Utworzyłem więc dwa pliki `.storyboard` o nazwach `StoryboardiPhone.storyboard` i `StoryboardiPad.storyboard`.
5. W strukturze projektu odszukaj plik `Info.plist`. Zauważ, że ten plik `.plist` może być zapisany pod inną nazwą. Przykładowo omawiany tutaj projekt nosi nazwę *Adding a Storyboard to an Existing Project*, więc plik typu `.plist` ma nazwę `Adding a Storyboard to an Existing Project.storyboard`. Po kliknięciu wspomnianego pliku nastąpi automatyczne otworenie odpowiedniego edytora.
6. Jeżeli w pliku znajdują się wymienione poniżej klucze, usuń je, ponieważ są zbędne podczas używania funkcji Storyboard.
 - `NSMainNibFile` (może być wyświetlony jako *Main nib file base name*).
 - `NSMainNibFile~ipad` (może być wyświetlony jako *Main nib file base name (iPad)*).
7. Jeżeli aplikacja jest przeznaczona tylko dla iPhone'a lub iPada, utwórz nowy klucz `UIMainStoryboardFile` (dla iPhone'a) lub `UIMainStoryboardFile~ipad` (dla iPada). W aplikacji uniwersalnej musisz utworzyć obydwa wymienione klucze.
8. Jako wartości nowo utworzonych kluczy podaj nazwy plików funkcji Storyboard, które wcześniej przygotowałeś (pomiń jednak rozszerzenie `.storyboard`).
9. Zapisz plik typu `.plist`.
10. Ostatni, ale prawdopodobnie najważniejszy krok to usunięcie z metody `application:didFinishLaunchingWithOptions:` kodu odpowiedzialnego za konfigurację kontrolerów widoku. Metoda ta znajduje się w pliku implementacji delegata aplikacji. Podczas używania funkcji Storyboard w delegacie aplikacji nie trzeba ręcznie konfigurować kontrolerów widoku. Dlatego też przejrzyj tę metodę i usuń z niej zbędny kod.

Analiza

Aplikacje utworzone bez funkcji Storyboard (albo we wcześniejszych wersjach Xcode albo w bieżących) mają inną strukturę niż aplikacje używające wymienionej funkcji. W aplikacjach Storyboard nie jest dłużej stosowany główny plik nib do przechowywania okien, więc nazwę tego plik można usunąć z pliku typu `.plist`. Jak się przekonałeś w poprzednim podrozdziale, aby aplikacja mogła skorzystać z funkcji Storyboard, we wspomnianym pliku typu `.plist` trzeba dodać pewne klucze konfiguracyjne.

Po wprowadzeniu przedstawionych zmian trzeba się upewnić, że delegat aplikacji nie będzie przeszkadzał w używaniu funkcji Storyboard. Każdy projekt jest inny, więc musisz zagwarantować, że delegat aplikacji *nie przypisuje* żadnego obiektu właściwości `rootViewController` okna. W przeciwnym razie funkcja Storyboard nie zostanie użyta, a Ty poświęcisz godziny na szukanie problemu. Najłatwiejszym rozwiązaniem jest po prostu umieszczenie w komentarzu całej zawartości metody `application:didFinishLaunchingWithOptions:`, a dopiero następnie rozważenie umieszczenia w innych miejscach aplikacji własnych poleceń inicjalizacyjnych (np. inicjalizacji modelu danych). Alternatywnym rozwiązaniem jest pozostawienie wymienionej metody i umieszczenie w komentarzu jedynie tych wierszy, które powodują zmianę obiektu głównego kontrolera widoku okna.

A

adres lokalizacji, 373
aktywowane aplikacje, 530
alarm, 594
alokacja, 54, 59, 80, 89
 egzemplarza, 68
 obiektu, 58
animacja, 148, 269, 655–661
 operacji skalowania, 663
 podczas obracania, 665
aplikacja
 działająca w tle, 524
 Kalendarz, 571, 581, 590
 Kontakty, 451
 Ustawienia, 532
 typu Master-Detail Application, 198
ARC, Automatic Reference Counting, 17, 64, 322, 456
archiwizowanie aplikacji, 31
asynchroniczne
 pobieranie danych, 393
 pobieranie pliku, 529
 wykonywanie zadań, 313, 334
atrybut, 540
 __autoreleasing, 67
 strong, 64
 unsafe_unretained, 64
 weak, 64
atrybuty encji Person, 553
awaria aplikacji, 66

B

bezpieczne pole tekstowe, 112
biblioteka
 muzyczna, 444
 obiektów, 279
 zasobów, 496, 499
 zdjęć, 479

blokowanie wątku, 397
błędy, 494, 721
bufor FIFO, 293, 326

C

centrum powiadomień, 98, 101
ciąg tekstowy documentText, 710
ciągi tekstowe, 37
cienie, 638, 641
Cocoa Touch, 607
Core Animation, 608
Core Data, 537, 545
 implementacja związków, 564
 odczyt danych, 547
 sortowanie danych, 552
 tworzenie danych, 545
 usuwanie danych, 549
Core Graphics, 608
Core Location, 356, 521
Core Motion, 667
czas oczekiwania, 396
czcionki, 614
czujnik przyspieszeniomierza, 669

D

dane
 hierarchiczne, 244
 klucz-wartość, 86, 686
 przechowywane w usłudze iCloud, 687
 wyjściowe, 336, 485
 z przyspieszeniomierza, 673
 z żyroskopu, 677
data, 124
definiowanie
 harmonogramu, 345
 kontrolera widoku, 503
 obiektów delegatów, 76

- deklaracja obiektu operacji, 331
- deklarowanie metod, 54
- delegacja zadań, 71
- delegat aplikacji, 203
- deserializacja danych, 410
- deskryptory sortowania, 552
- długotrwałe zadania, 509, 511
- dodawanie
 - encji, 541
 - alarmu, 594
 - atrybutów, 541
 - cienia, 638
 - kontaktów, 462, 466
 - kontrolera, 278
 - obiektów do zbioru, 90
 - obiektu Person, 555
 - obrazu do projektu, 623
 - pliku, 49
 - przycisków, 178
 - Storyboard, 289
 - struktur, 355
 - struktury Core Motion, 668
 - struktury do aplikacji, 452
 - wydarzeń, 576, 579
 - związku, 566
- dokument RFC4791, 572
- domknięcie, closure, 294
- dostarczanie powiadomienia, 518
- dostęp do
 - biblioteki muzycznej, 444
 - danych, 553
 - filmów, 480
 - klasy, 79
 - kluczy, 487
 - metody, 77
 - pakietu, 93
 - pliku MP3, 426
 - self, 302
 - treści kalendarzy, 578
 - właściwości calendars, 574
 - właściwości FKEmployeeToManager, 568
 - właściwości obiektu, 301
 - zdjęć, 480
 - zmiennych, 305
 - zmiennych obiektu bloku, 300
 - zrzutu ekranu, 442
- dostępność, 77
- dynamiczne tworzenie kodu, 294
- dyrektywa
 - @dynamic, 545
 - @optional, 75
 - @required, 75, 234, 236
 - @synthesize, 61
- dystrybucja aplikacji, 31

E

- edycja
 - dokumentu, 706
 - filmów, 502
 - powtarzającego się wydarzenia, 591
 - właściwości wydarzenia, 604
 - wydarzenia, 605, 606
- edytor danych, 540
- edytor graficzny modelu danych, 546
- egzemplarz, 52
 - EKEventViewController, 599, 601
 - klasy ConflictViewController, 725
 - NSFetchRequest, 547, 552, 553
 - NSFileVersion, 728
 - NSSortDescriptor, 553
 - NSString, 122
 - NSURLRequest, 192
 - UISegmentedControl, 132
 - UITableViewCell, 237
- element
 - UIDatePicker, 125, 128
 - UIPickerView, 118–123
- elementy
 - klasy UISwitch, 114
 - nawigacyjne, 149
- encja, 540
 - Employee, 565
 - Manager, 566
 - Person, 541, 549
- etykieta, 162–165
 - nagłówek, 251, 254
 - UILabel, 249

F

- FIFO, 293, 326
- format
 - H.264, 439
 - XML, 21
- funkcja, 51
 - ABAddressBookCopyArrayOfAllPeople(), 456
 - ABAddressBookAddRecord(), 460
 - ABAddressBookCopyArrayOfAllGroups(), 469
 - ABAddressBookCopyArrayOfAllPeople(), 455, 471
 - ABAddressBookCopyLocalizedLabel(), 460
 - ABAddressBookCopyPeopleWithName(), 469, 473
 - ABAddressBookCreate(), 453, 460
 - ABAddressBookHasUnsavedChanges(), 454
 - ABAddressBookSave(), 461, 464
 - ABGroupAddMember(), 466

ABGroupCreate(), 464
ABMultiValueCopyLabelAtIndex(), 457, 460
ABMultiValueCopyValueAtIndex(), 458
ABMultiValueGetCount(), 457
ABPersonCopyImageData(), 473
ABPersonCreate(), 460
ABPersonHasImageData(), 473
ABPersonSetImageData(), 473
ABPersonSetImageData(), 474
ABRecordCopyValue(), 456
ABRecordSetValue(), 464
CFAbsoluteTimeGetCurrent(), 596
CFArrayGetCount(), 456
CFArrayGetValueAtIndex(), 456
CFRelease(), 460
CGAffineTransformMakeRotation(), 384, 654
CGAffineTransformMakeScale(), 392, 652
CGAffineTransformMakeTranslation(), 649
CGColorComponents(), 622
CGColorGetNumberOfComponents(), 622
CGContextAddLineToPoint(), 626
CGContextMoveToPoint(), 626
CGContextRotateCTM(), 654
CGContextSetLineJoin(), 629
CGContextSetLineWidth(), 627
CGGradientCreateWithColor(), 642
CGGradientCreateWithColorComponents(), 643
CGPathAddRect(), 635
CGPathCreateMutable(), 632
CGRectIntersection, 221, 226
CGRectMake(), 620
dispatch_after_f(), 318
dispatch_get_main_queue(), 293
dispatch_after(), 318, 319
dispatch_after() i dispatch_after_f(), 318
dispatch_after_f(), 319, 320
dispatch_async(), 307, 313
dispatch_async_f(), 307, 314
dispatch_get_global_queue(), 293
dispatch_get_global_queue(), 312
dispatch_get_main_queue(), 307
dispatch_group_async(), 323
dispatch_group_async_f(), 324
dispatch_group_create(), 323
dispatch_group_notify(), 323
dispatch_once(), 321
dispatch_queue_create(), 293, 326
dispatch_release(), 323
dispatch_sync(), 311, 312
dispatch_sync_f(), 312
displayAlertView(), 310
NSLog(), 43, 87
Storyboard, 275–277, 286–290
strlen(), 48
UIImageWriteToSavedPhotosAlbum(), 491

G

gałka suwaka, 130
GCD, Grand Central Dispatch, 291
generowanie kodu, 543, 545
geokodowanie, 373
gest, 377
 długiego naciśnięcia, 377, 386, 388
 machnięcia, 245, 377, 379
 obrotu, 377, 381
 potrząśnięcia, 674
 przesunięcia, 377, 384, 385
 stuknięcia, 377, 389
 uszczypnięcia, 377, 390, 391
główny kontroler widoku, 198, 280
gradient osiowy, 642–648
gradienty, 642
grubość linii, 627
grupowanie
 opcji, 132
 zadań, 322
grupy zasobów, 498
grzbiet kartek, 204

H

harmonogram, 345

I

iCloud, 681
iCloud dla aplikacji, 691
identyfikator
 aplikacji, 687
 Container, 693
 firmy, 687
 IntToStringConverter, 298
 metody, 54
 Team ID, 687
 zadania, 511
ikona aplikacji, 624
implementacja
 edycji, 563
 handleKeyboardWillHide, 226
 klasy MyWindow, 677
 metod, 54
 nawigacji, 142
 przycisku, 180
 związków, 564
informacje
 o stanie, 510, 524
 o wydarzeniu, 591
 z przyspieszeniomierza, 674

- inicjalizacja, 54, 59
- inicjalizacja obiektu, 58
- inspektor
 - Data Model, 542
 - pliku, 613
 - tożsamości, 614
- Interface Builder, 21
- interfejs użytkownika, 23
- iteracja elementów tablicy, 83

J

- język Objective-C, 17
- JSON, 408, 413

K

- kalendarz Google, 571
- kamera, 481, 488
- karta
 - Build Settings, 33
 - Summary, 686
- katalog, 691
 - Resources, 91
 - Resources.bundle, 95
- kategoria
 - AVAudioSessionCategoryAmbient, 435
 - AVAudioSessionCategoryAudioProcessing, 436
 - AVAudioSessionCategoryPlayAndRecord, 436
 - AVAudioSessionCategoryPlayback, 436
 - AVAudioSessionCategoryRecord, 435
 - AVAudioSessionCategorySoloAmbient, 435
- klasa, 52
 - AVAudioPlayer, 425, 428
 - AVAudioRecorder, 428, 433
 - AVAudioSession, 435
 - bazowa, 50
 - CLGeocoder, 373, 376
 - CLLocationManager, 360–362
 - CloudDocument, 713
 - CLPlacemark, 376
 - CMMotionManager, 671, 678
 - EKAlarm, 594
 - EKEventEditViewController, 604
 - EKEventStore, 576, 583
 - Father, 73
 - initWithProgressViewStyle, 214
 - isSourceTypeAvailable, 481
 - MKAnnotationView, 366, 372
 - MKMapView, 357–360, 367, 372
 - MKPinAnnotationView, 366
 - MPMediaPickerController, 444
 - MPMoviePlayerController, 438–442

- MyAnnotation, 363, 367
- MyWindow, 676
- NSArray, 78, 82
- NSEntityDescription, 545
- NSFetchedResultsController, 553
- NSFileManager, 699, 711
- NSFileVersion, 721
- NSInvocationOperation, 329
- NSJSONSerialization, 408, 411
- NSManagedObjectContext, 549
- NSMetadataQuery, 697, 706, 713
- NSMutableArray, 82, 294
- NSMutableString, 37
- NSMutableURLRequest, 399, 401
- NSNotificationCenter, 100
- NSNumber, 80
- NSObject, 17, 60, 74
- NSOperation, 295, 331, 336
- NSString, 37, 616
- NSThread, 310, 509
- NSTimer, 343, 344
- NSUbiquitousKeyValueStore, 686–688
- NSURLConnection, 396
- NSURLConnection, 394, 397, 401–406
- NSXMLParser, 418
- Person, 50, 62
- RootViewController, 203
- SecondViewController, 288
- TWTweetComposeViewController, 414
- UIAlertView, 107, 111
- UIApplication, 194, 509, 511
- UIBarButtonItem, 149–153
- UIButton, 178, 180
- UIColor, 617
- UIDatePicker, 124
- UIDevice, 508
- UIDocument, 705–708
- UIFont, 614
- UIGestureRecognizer, 378, 380
- UIImage, 370, 623
- UIImagePickerController, 479–484, 503, 505
- UIImageView, 182, 253
- UILabel, 162
- UILongPressGestureRecognizer, 386
- UILongTapGestureRecognizer, 387
- UINavigationController, 142, 148, 604
- UIPageViewController, 201
- UIPanGestureRecognizer, 384, 385
- UIPickerView, 117
- UIPinchGestureRecognizer, 391
- UIRotationGestureRecognizer, 381
- UIScrollView, 186
- UISegmentedControl, 132, 136

- UISlider, 129
- UISplitViewController, 196
- UIStoryboardSegue, 287
- UISwipeGestureRecognizer, 380
- UISwitch, 114, 126
- UITabBarController, 156
- UITableView, 231, 234
- UITableViewCell, 231, 240
- UITapGestureRecognizer, 389, 390
- UITextField, 166, 167
- UITextView, 173, 177, 713
- UIVideoEditorController, 503, 506
- UIView, 149, 187, 242, 612
- UIViewController, 138, 159, 438
- UIWebView, 191, 192
- XMLElement, 420
- klauzula else, 41
- klawiatura, 218, 223
- klawiatura podzielona, 219
- klucz
 - UIApplicationExitsOnSuspend, 534
 - UIBackgroundMode, 519
 - UIBackgroundModes, 519–522
 - UIImagePickerControllerCropRect, 488
 - UIImagePickerControllerEditedImage, 488
 - UIImagePickerControllerMediaMetadata, 488
 - UIImagePickerControllerOriginalImage, 488
 - UIMainStoryboardFile, 290
- klucze konfiguracyjne, 290
- kolejka, 308
 - główna, 293, 308
 - operacji, 295
 - szeregową, 293, 326
 - współbieżną, 293
- kolejkowanie zadań, 306
- kolor, 617
- kolor pinezki, 371
- komórki
 - tabeli, 237
 - widoku, 243
- kompilacja, 25
- kompilator, 26
- kompilator LLVM, 64
- komunikat, 108, 111
 - błędu, 26, 494
 - ostrzeżenia, 309
- konfiguracja
 - kalendarza Google, 572
 - kontrolerów widoku, 290
 - nowego identyfikatora, 684
 - opcji, 610
 - projektu, 19
 - ustawień projektu, 202
 - widoku tabeli, 724
- konfigurowanie zapytania metadanych, 697
- kontekst
 - graficzny, 645
 - obiektu zarządzanego, 539
- kontroler, 105
 - AddPersonViewController, 556, 560
 - DataViewController, 203
 - EKEventViewController, 599
 - modalny, 415
 - ModelController, 203
 - MPMediaPickerController, 445
 - nawigacyjny, 142, 207, 278–282
 - paska kart, 157, 160
 - PersonListViewController, 556, 558
 - Popover, 211
 - RootViewController, 203
 - UIImagePickerController, 486
 - UIPageViewController, 203
 - UIViewController, 157
 - widoku, 105, 115, 139, 145, 158, 275
 - widoku ConflictViewCotnroller, 728
 - widoku opcji, 143
 - widoku podzielonego, 196, 198
 - widoku szczegółowego, 198
 - widoku ustawień, 143
 - wyników pobierania danych, 554, 560
- kontrolka segmentowana, 133, 137, 155
- konwersja
 - adresu, 373
 - współrzędnych, 374
- kończenie
 - nasłuchiwania, 444
 - zadania, 509
- koordynator trwałego magazynu danych, 539
- korzystanie z usługi iCloud, 682
- kreacja dwuetapowa, 60
- kryterium porównania, 698
- książka adresowa, 452, 456
- KVC, Key-Value Coding, 296
- KVO, Key-Value Observing, 296, 331

L

- leksykalny zasięg, 303
- liczba, 80
 - pikseli, 608
 - rekordów, 121
- licznik
 - czasu, 342–345, 510
 - użycia, 64
- lista
 - kalendarzy, 574
 - uczestników, 590

Ł

łączenie

- kontrolerów, 281
- obiektów zarządzanych, 564
- przycisku z kontrolerem, 286

M

Map Kit, 356

mapa, 357, 362

- zaznaczanie położenia, 362

margines etykiet, 252

marginesy, 224, 229

menu kontekstowe, 255, 256

menu Scheme, 26

metadane, 697

metoda

- ableView:viewForHeaderInSection, 250

- addGestureRecognizer, 391

- addObject, 84

- addObjectsFromArray, 84

- addSubview, 187

- addTarget:action:forControlEvents, 117, 134, 180

- alarmWithRelativeOffset, 594

- alertView:clickedButtonAtIndex, 110

- allObjects, 90

- alloc, 58

- anyObject, 90

- application:didFinishLaunchingWithOptions, 200, 290, 329, 514, 556

- application:didReceiveLocalNotification, 515, 517

- applicationDidBecomeActive, 524, 535

- applicationDidEnterBackground, 507, 535

- applicationWillEnterForeground, 507, 525

- applicationWillResignActive, 524

- applicationWillTerminate, 535

- arrayWithObjects, 83

- assetForURL:resultBlock:failureBlock, 480

- audioPlayerBeginInterruption, 427, 428

- audioPlayerEndInterruption:withFlags, 427

- audioRecorderBeginInterruption, 434

- audioRecorderDidFinishRecording:successfully, 432

- audioRecorderEndInterruption:withFlags, 434

- audioRecordingPath, 431

- audioRecordingSettings, 431

- availableMediaTypesForSourceType, 482

- beginAnimations:context, 656, 658

- beginBackgroundTaskWithExpirationHandler, 509

- breathe, 75

- buttonTitleAtIndex, 111

- buttonWithType, 180

- callIntToString, 298

- callTrimBlock, 306

- cameraSupportsMedia:sourceType, 496

- cancelAllLocalNotification, 513

- canEditVideoAtPath, 506

- CFRelease, 454

- CGColor, 622

- colorWithRed:green:blue:alpha, 618

- controllerDidChangeContent, 562

- convertIntToString, 299

- convertIntToString:usingBlockObject, 299

- convertRect:fromView, 227

- count, 90

- createNewPerson, 560

- dataWithJSONObject:options:error, 408

- deleteObject, 549

- deleteRowsAtIndexPaths:withRowAnimation, 247, 265

- deleteSections:withRowAnimation, 265

- detachNewThreadSelector:toTarget:withObject, 509

- dismissModalViewControllerAnimated, 503

- displayMediaPickerAndPlayItem, 447

- doesCameraSupportTakingPhotos, 492

- doesGroupExistWithGroupName:inAddressBook, 471

- doTheConversion, 299

- drawAtPoint, 625

- drawAtPoint:withFont, 616

- drawInRect, 623, 625

- drawInRect:withFont, 620

- endBackgroundTask, 509, 511, 512

- enumerateKeysAndObjectsUsingBlock, 88

- enumerateObjectsUsingBlock, 84, 90, 97

- familyNames klasy UIFont, 615

- fontNamesForFamilyName, 615

- fontWithName:size, 616

- geocodeAddressString:completionHandler, 373

- handleDocumentStateChanged, 720, 726

- handleKeyboardWillHide, 226

- handleKeyboardWillShow, 224, 227

- handleLongPressGestures, 388

- handleMetadataQueryFinished, 700, 714

- handlePanGestures, 385

- handleRotations, 383

- handleSwipes, 380

- handleTaps, 390

- imageNamed, 623

- imagePickerController:didFinishPickingMediaWithInfo, 487

- imagePickerControllerDidCancel, 487

- imageWithData, 623

init, 59, 60
 initWithRecurrenceWithFrequency:interval:end, 589
 initWithBarButtonSystemItem:target:action, 152
 initWithContentsOfFile, 623
 initWithCustomView, 153
 initWithData, 623
 initWithData:encoding, 409
 initWithData:error, 426
 initWithFileURL, 710
 initWithFrame, 183
 initWithFrame:style, 232
 initWithObject, 82
 initWithObjects, 134
 initWithRootViewController, 143
 initWithTarget:selector:object, 349
 initWithURL:settings:error, 429
 insertNewObjectForEntityForName
 inManagedObjectContext, 545
 insertNewObjectForEntityForName
 inManagedObjectContext, 546
 instancesRespondToSelector, 77
 invalidate, 346
 isCameraAvailable, 492
 isCameraDeviceAvailable, 483
 isCancelled, 296
 isDeleted, 550
 isFlashAvailableForCameraDevice, 484
 isFrontCameraAvailable, 483
 isMultitaskingSupported, 508
 isOn, 117
 isSourceTypeAvailable, 481, 482
 JSONObjectWithData:options:error, 411
 loadData:MIMEType:textEncodingName:
 baseURL, 192
 loadHTMLString:baseURL, 192
 locationInView, 386, 390
 locationManager:didUpdateToLocation:from
 Location, 523
 locationOfTouch
 inView, 386
 locationServicesEnabled, 362
 mainBundle, 93
 mapView:regionDidChangeAnimated, 360
 mapView:viewForAnnotation, 360, 366
 mapView:willStartLoadingMap, 360
 mapView:willStartLocatingUser, 360
 mediaPicker:mediaPickerDidCancel, 448
 mediaPickerDidCancel, 450
 motionEnded:withEvent, 675
 moveSection:toSection, 259
 moveSection1ToSection3, 263
 moveRowAtIndexPath:toIndexPath, 263
 newGroupWithName:inAddressBook, 466
 newPersonWithFirstName:lastName:inAddress
 Book, 462, 466
 NSStringFromClass, 79
 numberOfComponentsInPickerView, 121
 numberOfSectionsInTableView, 236
 objectAtIndex, 83
 objectAtIndexPath, 554
 orientationChanged, 531
 otherVersionsOfItemAtURL, 721
 parser:didEndElement:namespaceURI:qualified,
 422
 parser:didEndElement:namespaceURI:qualified
 Name, 420, 422
 parser:didStartElement:namespaceURI:qualified
 Name:attributes, 420
 parser:foundCharacters, 420, 422
 parserDidEndDocument, 420
 parserDidStartDocument, 420
 parserDidStartDocument, 421
 pathForResource:ofType, 92
 performAddWithAlertView, 211
 performExpand, 242, 243
 performSelector:withObject:afterDelay, 146, 318
 performSelectorInBackground:withObject, 352
 pickerView:numberOfRowsInComponent, 121
 pickerView:titleForRow:forComponent, 122
 popoverViewControllerAnimated, 147
 predicateWithFormat, 698
 prepareForSegue:sender, 287, 288
 presentModalViewController:animated, 503,
 604
 presentMoviePlayerViewControllerAnimated,
 438
 presentPopoverFromBarButtonItem:permittedA
 rrow:Directions:animated, 212
 printString, 318
 pushController, 602
 pushSecondController, 146
 rangeOfString, 39
 reloadAllComponents, 124
 removeAlarm, 596
 removeDependency:, 342
 removeEvent:span:commit:error, 583
 removeObject, 84
 removeObserver, 103
 requestThumbnailImagesAtTimes:timeOption,
 442
 requestWithURL:cachePolicy:timeoutInterval,
 396
 respondsToSelector, 77, 78
 reverseGeocodeLocation:completionHandler,
 376
 saveEvent:span:error, 576

metoda

- scheduledTimerWithTimeInterval:invocation:repeats, 345
- scheduleLocalNotification, 512, 513
- scopeTest, 304
- scrollViewDidEndDecelerating, 188
- scrollViewDidEndDragging:willDecelerate, 188
- scrollViewDidScroll, 188
- scrollViewWillBeginDecelerating, 188
- selectedSegmentIndex, 135
- sendAsynchronousRequest:queue:completionHandler, 394
- sendSynchronousRequest:returningResponse:error, 397
- setAnimationDelay, 660
- setAnimationDelegate, 659
- setAnimationDidStopSelector, 659
- setAnimationDuration, 659
- setAnimationRepeatCount, 660
- setAnimationWillStartSelector, 660
- setBackgroundImage:forState, 182
- setGyroUpdateInterval, 678
- setHTTPMethod, 401
- setImage:forState, 182
- setInitialText, 414
- setNetworkActivityIndicatorVisible, 194
- setOn, 116
- setShowsSelectionIndicator, 122
- setSortDescriptors, 552
- setThumbImage:forState, 132
- setTitle:forState, 180
- setUbiquitous:itemAtURL:destinationURL:error, 699
- setViewControllers:animated, 148
- sliderValueChanged, 132
- sortUsingComparator, 85, 301
- startAccelerometerUpdates, 674
- startAccelerometerUpdatesToQueue:withHandler, 671–674
- startGyroUpdates, 679
- startPlayingVideo, 439, 441
- startSearchingForDocumentIniCloud, 714
- stopPlayingAudio, 447
- stopPlayingVideo, 441
- stopRecordingOnAudioRecorder, 432
- stopThread, 353
- switchIsChanged, 117
- systemFontSize, 616
- tableView:canPerformAction:forRowAtIndexPath:withSender, 256, 258
- tableView:cellForRowAtIndexPath, 236
- tableView:editingStyleForRowAtIndexPath, 246
- tableView:heightForFooterInSection, 251
- tableView:heightForHeaderInSection, 251
- tableView:numberOfRowsInSection, 236
- tableView:performAction:forRowAtIndexPath:withSender, 256
- tableView:shouldShowMenuForRowAtIndexPath, 256
- tableView:titleForFooterInSection, 254
- tableView:titleForHeaderInSection, 253
- tableView:viewForFooterInSection, 250, 252
- tableView:viewForHeaderInSection, 252
- textDidChange, 716
- textField:shouldChangeCharactersInRange:replacementString, 169
- textFieldDidBeginEditing, 169
- textFieldDidEndEditing, 169
- textFieldShouldBeginEditing, 169
- textFieldShouldClear, 169
- textFieldShouldEndEditing, 169
- textFieldShouldReturn, 169
- thirdCounter, 349
- titleForSegmentAtIndex, 136
- urlForDocumentsDirectoryIniCloud, 712
- URLForUbiquityContainerIdentifier, 711
- valueForProperty, 446
- view, 107
- viewDidAppear, 145
- viewDidLoad, 115, 163, 269, 386, 426, 558, 661
- viewDidUnload, 349, 440
- viewWillAppear, 178
- viewWillDisappear, 178
- webView:didFailLoadWithError, 195
- webViewDidFinishLoad, 195
- webViewDidStartLoad, 195
- writeVideoAtPathToSavedPhotosAlbum:completionBlock, 493

metody

- animacji, 656
 - delegata, 239
 - egzemplarza, 52
 - inicjalizacji, 54, 58
 - klasy, 52, 54
 - NSMutableArray, 84
 - NSNumber, 80
 - NSString, 38
 - NSUbiquitousKeyValueStore, 688
 - UIView, 658
 - obserwatora, 224
 - rysowania, 609
 - rysujące, 632
 - setter i getter, 61, 63, 301
 - sortowania, 77
- ## model
- obiekty, 543
 - obiektu zarządzanego, 539

moduł Interface Builder, 21
modyfikacja pliku nagłówkowego, 192
MVC, Model-View-Controller, 105, 138

N

nagłówek, 247
nagrywanie, 430
 audio, 428
 video, 488
narzędzie Xcode, 18, 138, 540
nasłuchiwanie
 powiadomień, 100, 175, 178, 441, 713
 powiadomień klawiatury, 216, 222, 224
nawiasy klamrowe, 43, 53
nawigacja, 142
nazwy
 metod, 53, 58
 parametrów, 58
 plików, 702
niebezpieczny atrybut, 66

O

obiekt
 ABRecordRef, 459
 ALAssetsLibrary, 480
 bloku, 293, 294, 312, 321
 bloku trimString, 306
 bloku trimWithOtherBlock, 306
 błędu, 375
 CFArrayRef, 456
 CMMotionManager, 671
 delegatów, 76
 EKCalendar, 581
 EKEventEditViewController, 604
 EKEventStore, 575
 EKParticipant, 591
 GraphicsViewControllerView, 616
 intColor, 298
 JSON, 408
 kontrolera nawigacyjnego, 279
 kontrolera widoku, 284
 modyfikowalny, 412
 nadawcy, 98
 NSDictionary, 86
 NSError, 578
 obserwatora, 101
 pierwszej klasy, 294
 powiadomień, 98
 przejścia segue, 287, 288
 przycisku, 285
 typu NSURLResponse, 394

UIColor, 619
UIDatePicker, 125, 126, 127
UIImagePickerController, 485, 488
UILocalNotification, 98, 512, 519
UIPickerView, 122, 123
UIProgressView, 215
UISwipeGestureRecognizer, 379
UITableView, 232
UITableViewCell, 243
TWTweetComposeViewController, 414
zarządzany, 539, 550
obraz, 184, 622
 kontaktu, 473
 przypisany kontaktowi, 473, 478
obrót, 654
obrót prostokąta, 655
obsługa
 błędów, 721
 edytora wideo, 503
 encji, 545
 liczb, 80
 mechanizmu delegacji, 188
 metod delegata, 600
 nasłuchiwanym powiadomieniom, 449
 połączeń sieciowych, 527
 powiadomieniom, 530, 597, 716, 725
 przejścia, 275
 usługi iCloud, 682
 wielozadaniowości, 508
 zakłóceń, 427, 434
 zmian położenia, 521
odczyt danych, 547
odmierzenie czasu, 127
odświeżanie widoku tabeli, 562
odtwarzacz audio, 520
odtworzenie, 438, 440
odtworzenie dźwięków, 425, 435, 519
odwrotne geokodowanie, 373, 375
okno
 komunikatu, 107, 110–113, 213
 konsoli, 258
 Organizer, 29, 34
 typu Popover, 206, 207, 213
określenie adresu URL, 701
opcje
 menu kontekstowego, 258
 rysowania gradientu, 646
operacje, 328, 334
 bloku, 295
 w tle, 509
 wyszukiwania, 702
 wywoływane, 295, 334
 zwykle, 295

P

- pakiet
 - główny, main bundle, 91, 92
 - kodu, 293
 - Resources.bundle, 95
- pamięć zaalokowana, 65
- panel Object Library, 23
- parametr
 - alpha, 662
 - cancelButtonTitle, 108
 - CGRect, 116
 - didCompleteWithAction, 604
 - delegate, 108
 - finalPath, 694
 - forControlEvents, 134
 - initWithNibName, 141
 - Klucz, 88
 - message, 107
 - NULL, 634
 - Obiekt, 88
 - otherButtonTitles, 108
 - paraContext, 307
 - prepareForSegue, 287
 - replacementString, 171
 - shouldChangeCharactersInRange, 171
 - title, 107
 - titleForFooterInSection, 254
 - titleForHeaderInSection, 253
 - typu *BOOL, 85
 - typu id, 84
 - typu NSInteger, 85
 - typu TWTweetComposeViewControllerResult, 417
 - typu void *, 309
 - UIProgressVisualStyle, 214
 - Wskaźnik, 88
- pasek
 - kart, 156–162
 - nawigacyjny, 144, 149, 283
 - postępu, 194, 215
 - przewijania, 189
- pętla, 44
 - for, 44, 45
 - wątku, 353
 - while, 46, 47
- piksel, 608
- pinezka na mapie, 365, 372
- plik
 - RootViewController, 198
 - ABPerson.h, 457
 - AddPersonViewController.h, 558
 - Boosting_Data_Access_in_Table_ViewsApp Delegate.h, 560
 - DetailViewController, 198
 - implementacji
 - delegata aplikacji, 335, 339
 - klasy, 49, 54, 710
 - klasy MyAnnotation, 363, 367
 - kontrolera widoku, 115, 361, 369, 713
 - operacji, 337
 - Info.plist, 290, 633
 - MyXML.xml, 419
 - nagłówkowy, 49, 54
 - delegata aplikacji, 482, 514, 525
 - kontrolera widoku, 208, 375, 382, 386, 436, 439, 475
 - Person.h, 50, 61, 559
 - Person.m, 50
 - PersonListViewController.h, 556
 - PersonProtocol.h, 75
 - RootViewController.m, 26
 - Settings.bundle, 532, 533
 - ViewController_iPhone.xib, 24
 - xcode.png, 624
- pliki
 - .bundle, 91
 - .plist, 290
 - .storyboard, 278
 - .xib, 21, 138, 141
 - audio, 425
 - nib, 23
 - plist, 519
 - wideo, 438
 - xcdatamodeld, 540
- pobieranie
 - asynchroniczne, 393
 - danych z przyśpieszeniomierza, 671
 - danych z żyroskopu, 677
 - informacji z kontaktów, 455
 - kontaktów, 458
 - listy kalendarzy, 574
 - obrazów, 473
 - pliku, 528
 - synchroniczne, 397
 - uczestników wydarzenia, 590
 - właściwości kontaktu, 456
 - wydarzeń, 591
 - zasobów, 496
 - zdjęć i filmów, 495
- pole tekstowe, 166–174, 217
- pole To-Many Relationship, 566
- polecenie
 - for, 43
 - if, 41

- if-else, 41
- SELECT, 549
- połączenie
 - asynchroniczne, 394, 407, 527
 - synchroniczne, 528
- położenie
 - grzbietu, 204
 - kształtów, 649
 - urządzenia iOS, 521
 - widoku, 655
- pomarańczowa kropka, 30
- porównywanie obiektów, 42
- powiadomienia, 98–101
 - lokalne, 512–518
 - o zmianie w bazie, 597
 - o zmianach w ustawieniach, 532
- powiadomienie
 - EKEventStoreChangedNotification, 598
 - NSCurrentLocaleDidChangeNotification, 530
 - NSUserDefaultsDidChangeNotification, 530
 - UIDeviceBatteryStateDidChangeNotification, 530
 - UIDeviceOrientationDidChangeNotification, 531
 - UIDeviceProximityStateDidChangeNotification, 530
 - UIKeyboardAnimationCurveUserInfoKey, 220
 - UIKeyboardAnimationDurationUserInfoKey, 220
 - UIKeyboardDidHideNotification, 177, 220
 - UIKeyboardDidShowNotification, 175, 220
 - UIKeyboardFrameBeginUserInfoKey, 220
 - UIKeyboardFrameEndUserInfoKey, 220
 - UIKeyboardWillHideNotification, 177, 220
 - UIKeyboardWillShowNotification, 175, 219
 - NSMetadataQueryDidFinishGatheringNotification, 697, 714
 - NSUserDefaultsDidChangeNotification, 532, 534
 - ResultOfAppendingTwoStrings, 99
 - UIApplicationDidEnterBackgroundNotification, 507
 - UIApplicationWillEnterForegroundNotification, 507
 - UIDocumentStateChangedNotification, 722
 - UIDocumentStateChangeNotification, 718, 720
- powtarzające się wydarzenie, 589
- poziom głośności, 129
- predykat typu NSPredicate, 581
- predykaty, 698
- prefiks __block, 301
- procedura, 51
 - CGContextAddPath(), 632
 - CGContextDrawLinearGradient(), 645
 - CGContextDrawPath(), 632
 - CGContextScaleCTM(), 653
 - CGContextSetShadow(), 638
 - CGContextSetShadowWithColor(), 638
 - CGPathAddLineToPoint(), 632
 - CGPathAddRect(), 632
 - CGPathMoveToPoint(), 632
 - CGPathRelease(), 632
 - sendEmailTo, 52
- procedury rozpoznawania gestów, 378, 387, 390
- profil tymczasowy, 32
- protokoły, 71
- protokół
 - AVAudioPlayerDelegate, 427, 437
 - AVAudioRecorderDelegate, 433, 434
 - CalDAV, 572
 - CloudDocumentProtocol, 710
 - MKAnnotation, 363
 - MKMapViewDelegate, 359
 - MPMediaPickerControllerDelegate, 445
 - NSXMLParserDelegate, 419
 - PersonProtocol, 71
 - UIAlertViewDelegate, 108
 - UIAlertViewDelegate., 109
 - UIApplicationDelegate, 515
 - UIImagePickerControllerDelegate, 496
 - UIImagePickerControllerDelegate, 486
 - UINavigationControllerDelegate, 486, 506
 - UIPageViewControllerDataSource, 204
 - UIPageViewControllerDelegate, 203
 - UIPickerViewDataSource, 119, 120
 - UIPickerViewDelegate, 122
 - UIScrollViewDelegate, 188
 - UITableViewDataSource, 235, 248, 253
 - UITableViewDelegate, 76, 234, 251, 257
 - UITextFieldDelegate, 169
 - UITextViewDelegate, 716
 - UIVideoEditorControllerDelegate, 506
 - UIWebViewDelegate, 193
- przechowywanie
 - filmów, 493
 - zdjęć, 491
- przeciążanie metody, 55, 56
- przejście segue, 284, 288
- przekazywanie danych, 286
- przekształcenia afiniczne, 663
- przekształcenie
 - obrotu, 654, 665
 - skalowania, 652, 664
 - translacji, 649
 - w Core Graphics, 649
- przełącznik, 116
- przełącznik na pasku, 154

przenoszenie komórek, 259
przepełnienie bufora, 47
przesuwanie widoków, 655
przetwarzanie danych, 523
przetwarzanie danych XML, 418
przewracanie kartek, 201
przezroczystość, 188
przycisk
 do usuwania komórki, 247
 detail disclosure, 241
 disclosure indicator, 241
 Dodaj, 152
 nawigacyjny, 151, 271
 Utwórz nowe konto, 179
przyciski
 na pasku, 150
 pomocnicze, 240
przypisanie
 cienia prostokątowi, 640
 delegata widokowi, 233
 profilu, 686
przyspieszeniometer, 667, 672
pula autorelease, 350
punkt, 608
 końcowy, 645
 początkowy, 609, 645
 przecięcia linii, 628

Q

Quartz 2D, 608

R

robienie zdjęć, 479, 485
rodzaje
 kolejek, 293
 przejść, 275
 przycisków, 241
 złączeń, 630
rozpoznawanie gestów, 377, 378
rozszerzanie gradientu, 646
rozszerzenie, *Patrz* pliki
rysowanie
 gradientów, 642
 gradientu osiowego, 645
 kształtów, 626, 631
 rysowanie linii, 626, 628
 na ekranie, 608
 prostokątów, 635–639
 ścieżek, 634
rzutowanie, 68

S

SDK bazowe, 77
SDK docelowe, 77
segue, 275, 284
sekcja
 Attributes, 541
 Foo, 532
 Twitter, 415
selektor, 77
 delegata, 662
 deleteOddNumbersSection, 269
 displayMediaPicker, 445
 tableView:commitEditingStyle:forRowAtIndexPath, 245
 tableView:didSelectRowAtIndexPath, 239
 tableView:editingStyleForRowAtIndexPath, 245
serializacja danych, 409
serwis Twitter, 413
sesja audio, 521
silnik Quartz 2D, 608
Singleton, 322
skalowanie, 392, 652
skalowanie prostokąta, 653
sklep iTunes App Store, 25, 433
słownik, 86, 408
 didFinishPickingMediaWithInfo, 487, 490
 NSDictionary, 87
 NSMutableDictionary, 87
 userInfo, 98, 100, 225
 źródła danych, 267
słowo and, 58
słowo kluczowe
 @interface, 51
 @property, 60
 @required, 120
 dispatch_, 291
sortowanie
 danych, 552
 tablicy, 85
specyfikator
 __bridge, 69
 __bridge_retained, 69
 __bridge_transfer, 69
 formatu %, 38
 unsigned long, 39
 rzutowania, 68
sprawdzanie
 dostępności, 77
 obsługi wielozadaniowości, 508
 dostępności czujnika, 670
stała
 ALAssetsGroupAll, 498
 DBL_MAX, 509
stan aplikacji, 524, 525

stany
 procedury rozpoznawania gestów, 378, 392
 przypisywane obiektom, 379

stopka, 247

Storyboard, 275

stronicowanie, 201

struktura
 AddressBook.framework, 451
 Assets Library, 480, 494, 497
 AV Foundation, 425, 519
 CGRect, 620
 Core Animation, 608
 Core Audio, 428
 Core Data, 537, 538
 Core Graphics, 608
 Core Location, 355
 Core Motion, 667
 Event Kit, 571
 Event Kit UI, 571
 Map Kit, 355
 Media Player, 425
 Mobile CoreServices.framework, 479
 Twitter, 393
 UIKit, 607, 618, 623

styl
 UIAlertViewStyleTextInput, 113
 UIAlertViewStyleSecureTextInput, 113
 UITableViewStyleGrouped, 232
 UITableViewStylePlain, 232
 UISegmentedControlStyleBezeled, 136

style okna komunikatu, 108

suwak, 129, 133

sygnatura typedef, 298

symulator, 27

symulator iPhone, 22

synchroniczne
 pobieranie danych, 397
 pobranie pliku, 529
 wykonanie obiektu bloku, 314
 wykonywanie zadań, 311, 328

synchronizacja słowników, 686

system iOS 5.0, 508

szablon
 Empty Application, 150
 Page-Based Application, 138, 202, 205
 Single View Application, 105, 115, 281, 610, 699

szybki enumerator, 84

Ś

ścieżka
 dostępu do obiektu, 554
 dostępu do obrazu, 93
 dostępu do pakietu, 95

ścieżki, 626, 631

T

tabela z sekcjami, 262

tablica, 82, 408
 kontrolerów widoku, 147
 modyfikowalna, 77
 obiektów, 148
 sections, 554
 subElements, 421
 typu NSArray, 725

tablice modyfikowalne, 82, 83

target, 126, 134

technologia
 ARC, 17, 64
 Automatic Reference Counting, 64
 GCD, 291–294, 306, 317, 323, 426, 511
 iCloud, 681
 Push Notifications, 512

tekst, 37, 616

tekst sformatowany, 193

tekstowe dane wejściowe, 166

token, 321

translacja, 649

tryb widoku, 173

Twitter, 413–417

tworzenie
 animacji, 656
 aplikacji, 196
 aplikacji iOS, 25
 ciągu tekstowego, 37
 danych, 545
 dokumentu, 705, 714
 egzemplarza widoku tabeli, 261
 egzemplarzy właściwości, 558
 elementu UIPickerView, 119
 etykiety, 163, 170
 grup, 323
 kalendarza CalDAV, 573
 katalogów, 691
 klasy, 48
 klasy w Xcode, 611
 komunikatu, 512
 kontrolerów widoku, 138, 722
 licznika czasu, 345
 modelu Core Data, 540
 modelu danych, 545
 nagłówka, 247
 obiektu bloku, 296
 obiektu UIImage, 500
 odwróconego związku, 567
 podklasy, 676, 708
 pola tekstowego, 167, 170
 powiadomienia, 515

- powtarzających się wydarzeń, 587
- profilu dystrybucji, 685
- profilu programisty, 685
- projektu w Xcode, 18, 610
- projektu wykorzystującego Core Data, 537
- projektu z Storyboard, 276
- protokołu, 72
- przejścia segue, 286
- przełączników, 114
- przewijanej treści, 186
- przycisku, 152, 181, 209
- przycisku nawigacyjnego, 150
- słownika, 408
- suwaka, 129
- systemowych przycisków, 151
- ścieżek, 631
- tzw. pakietu, 91
- wątku, 292, 347
- wątku w tle, 352
- widoku mapy, 357
- widoku tabeli, 231
- widoku tekstu, 174
- własnej kolejki, 326
- własnych przycisków, 242
- właściwości, 115
- wydarzeń, 576, 577, 583
- zgrupowanego widoku tabeli, 249
- związku, 569

typ

- CGLineJoin, 629
- IntToStringConverter, 298
- NSData, 409
- NSInteger, 297
- NSMutableArray, 77
- NSString, 297
- wyliczeniowy, 39
 - UIBarButtonItemSystemItem, 152
 - UIDatePickerMode, 125
 - UISegmentedControlStyle, 136
 - UITableViewCellAccessoryType, 241
 - UIViewContentMode, 185

typy

- animacji, 220
- archiwum, 34
- danych, 36
- operacji, 295
- zmiennych, 36
- związków, 564

U

- uchwyt do kolejki, 308
- UIKit, 607

- ukrywanie
 - klawiatury, 227
 - paska stanu, 633
- umieszczanie
 - danych w iCloud, 705
 - przycisku w pasku, 558
 - obrazu w pakiecie, 96
- uruchamianie aplikacji, 28, 30
- urządzenia iOS, 28, 217
- usługa iCloud, 682, 691
 - adresy URL, 692
 - dane klucz-wartość, 686
 - dokumenty, 705
 - identyfikator Container, 693
 - katalogi, 691
 - pliki, 704
 - rozwiązywanie konfliktów, 721
 - umieszczanie zasobów, 694
 - wykrywanie konfliktów, 718
 - wyszukiwanie plików, 697
- ustalenie daty dostarczenia powiadomienia, 513
- ustawienia aplikacji, user defaults, 534
- ustawienia nagrywania, 429
- usuwanie
- alarmu, 596
 - danych, 549
 - komórek, 245, 265, 271
 - obiektów Person, 555
 - obrazów, 632
 - obrazów z pamięci, 661
 - rekordu, 246
 - sekcji, 265, 270
 - wydarzeń, 582
 - zadania, 511
 - zależności, 342
- uwierzytelnianie, 29

V

- void, 51, 298

W

- wartość
 - boolowska, 42
 - location, 521
 - NAN, 386
 - nil, 66, 83, 454
 - NSComparisonResult, 85
 - UIControlEventValueChanged, 134
 - UIImagePickerControllerSourceTypePhotoLibrary, 495
 - zwrotna, 60
 - zwrotna void, 298

- warunek, 41
- wątek
 - blokowanie, 397
 - główny, 296, 330
 - pętla, 353
 - przerwanie działania, 353
 - uruchomienie, 353
 - wstrzymanie działania, 509
- wcięcia komórek, 244
- wczytywanie treści, 193
- wczytywanie stron internetowych, 191
- wiadomość release, 350
- widok
- kontrolera widoku, 141
 - mapy, 358
 - nagłówka, 248
 - obrazu, 184
 - przewijania, 189
 - sieciowy, 192
 - tabeli, 221, 231, 256, 553, 557
 - tabeli z sekcjami, 266
 - tekstu, 176
 - UIProgressView, 214
 - WebView, 192
- wielkość liter, 39
- wielkość treści, 187
- wielozadaniowość, 507, 524, 534
- właściwości, 60
 - egzemplarza EKEventViewController, 601
 - klasy UITextField, 167
 - obiektu uczestnika, 594
- właściwość
 - accessoryType, 240, 241
 - accessoryView, 242
 - adjustsFontSizeToFitWidth, 165
 - alertAction, 519
 - alertVisualStyle, 108
 - allowsCalendarPreview, 602
 - allowsContentModifications, 578
 - allowsEditing, 601
 - allowsPickingMultipleItems, 445
 - attendees, 591
 - audioRecorder, 431
 - backgroundTimeRemaining, 509, 521
 - clipsToBounds, 185
 - contentInset, 177
 - countDownDuration, 127
 - currentElementPointer, 421
 - dataSource, 235, 236
 - datePickerMode, 125
 - delegate, 168, 193
 - detailViewController, 200
 - editViewDelegate, 604
 - event, 602
 - firstName, 61
 - FKManagerToEmployees, 568
 - indicatorStyle, 189
 - kABPersonEmailProperty, 457
 - kABPersonFirstNameProperty, 457
 - kABPersonLastNameProperty, 457
 - kABPersonMiddleNameProperty, 457
 - myLocationManager, 361
 - mySegmentedControl, 134
 - mySlider, 130
 - mySwitch, 115
 - myTableView, 234
 - navigationItem, 156
 - nonatomic, 61
 - numberOfLines, 165
 - numberOfTapsRequired, 390
 - pagingEnabled, 189
 - paintingTimer, 342
 - placeholder, 171
 - popoverController, 209
 - progress, 215, 216
 - rootViewController, 290
 - rotation, 382
 - segmentedControlStyle, 136
 - showsSelectionIndicator, 122
 - sourceType, 496
 - superview, 440
 - tableViewNumbers, 267
 - textAlignment, 168
 - title, 149
 - titleLabel, 148, 149
 - typu
 - CLGeocoder, 373
 - NSArray, 554
 - NSFetchRequest, 555
 - NSXMLParser, 419
 - UIDatePicker, 124
 - UIDocumentState, 718
 - UIPopoverController, 208
 - UIViewContentMode, 185
 - videoMaximumDuration, 490
 - videoQuality, 490
 - viewController, 147
 - window, 65
- współbieżność, 291, 346
- współrzędne geograficzne, 360, 373
- wstrzymanie działania wątku, 509
- wybór
 - daty, 124
 - filmu, 504
 - godziny, 124
 - opcji, 135

- pliku xib, 613
- utworu, 450
- wartości, 129
- wartości z listy, 117
- wersji dokumentu, 730
- wyciek pamięci, 350
- wydarzenia powtarzające się, 587
- wydzielenie wątku, 296
- wykonywanie zadań w GCD, 318
- wykrywanie
 - gestu potrząśnięcia, 674
 - konfliktów, 718
 - przyśpieszoniomierza, 668
 - zmian położenia, 531
 - żyroskopu, 670
- wypełnianie
 - tablicy, 260
 - widoku tabeli, 268, 727
- wyrównywanie etykiet, 255
- wyszukiwanie
 - ciągu tekstowego, 39
 - filmu, 500
 - kontaktów, 469
 - plików, 697
- wyświetlanie
 - ciągu tekstowego, 621
 - danych hierarchicznych, 244
 - daty, 125
 - etykiet, 163
 - klawiatury, 217
 - komunikatu, 106, 515
 - kontrolerów, 161
 - kontrolerów widoku, 156
 - listy czcionek, 614
 - mapy, 357
 - menu kontekstowego, 255
 - obiektów zarządzanych, 562
 - obrazów, 148, 182, 622
 - obrazu, 316, 625
 - okna typu Popover, 205
 - paska postępu, 214
 - pasków przewijania, 190
 - pinezek, 365, 370
 - powiadomienia, 517
 - przycisku edycji, 557
 - suwaka, 131
 - tekstu, 175, 616
 - tekstu statycznego, 162
 - wierszy tekstu, 173
 - właściwości wydarzenia, 599
 - zdjęcia, 499
- wywołanie
 - funkcji, 52
 - procedury, 52
 - obiektu bloku, 305

X

Xcode, 18, 540

Z

- zablokowanie przycisku, 271
- zadania wykonywane
 - asynchronicznie, 313, 334
 - synchronicznie, 311, 328
- zakleszczenie, 295
- zakłócenia w urządzeniu, 427, 434
- zakres dat, 127
- zakres suwaka, 130
- zależności pomiędzy operacjami, 340
- zapis
 - aplikacji, 198
 - klasy, 709
 - klasy Father, 74
 - projektu, 611
 - projektu na dysku, 277
 - stanu aplikacji, 525
 - tablicy, 317
- zapytanie metadanych, 697, 702
- zarządzanie
 - dokumentami, 706
 - grupą zadań, 325
 - stanem dokumentów, 718
- zasieg zmiennej, 303
- zasoby, 496, 498
- zawartość encji, 547
- zawijanie wierszy, 620
- zaznaczanie na mapie, 362
- zbiory, 89
- zdarzenia
 - w pliku implementacji, 335
 - widoku mapy, 359
 - widoku tabeli, 238
- zdjęcia, 485
- złączenie, 629
- zmiana
 - nazwy związku, 567
 - położenia, 655
 - ustawień, 532
- zmienna
 - dictionary, 409
 - doYouWantToSaveChanges, 454
 - executingInBackground, 522
 - self, 54
- zmiennie, 35, 300
 - egzemplarza, 62
 - lokalne, 300

znak

@, 38

^, 85

równości, 42

równości podwójny, 42

zrzut ekranu, 442, 443

związek

jeden do jednego, 564

jeden do wielu, 565

odwrócony jeden do wielu, 564

odwrócony wiele do wielu, 564

wiele do wielu, 568

Ż

źródło danych, 495

źródło danych dla widoku tabeli, 267

Ż

żądanie HTTP

DELETE, 404

GET, 400

POST, 402

PUT, 406

żądanie synchroniczne, 394

żądanie typu NSURLRequest, 394

żądanie URL, 399

czas oczekiwania, 400

modyfikowalne, 400

niemodyfikowalne, 400

żyroskop, 667, 670, 678

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



- 1. ZAREJESTRUJ SIĘ**
- 2. PREZENTUJ KSIĄŻKI**
- 3. ZBIERAJ PROWIZJĘ**

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA WYDAWNICZA

 **Helion SA**

iOS 5. Programowanie. Receptury



Szeroki wachlarz urządzeń mobilnych Apple sprawia, że łatwo jest dzięki nim dotrzeć do wielkiej liczby użytkowników. Z drugiej strony stawia przed programistami nowe wyzwania – różne rozdzielczości ekranu i różne wyposażenie urządzeń to tylko niektóre z nich. Jak sobie poradzić z codziennymi problemami w trakcie tworzenia aplikacji dla iOS-a?

Jeżeli znasz iOS SDK, dzięki tej książce nauczysz się stosować setki technik z zakresu tworzenia oprogramowania na platformie iOS oraz pokonasz wiele uciążliwych problemów, które pojawiają się podczas tworzenia aplikacji dla urządzeń iPhone, iPad i iPod touch. Dowiesz się, jak zaimplementować procedurę rozpoznawania gestów, odtwarzania plików audio i wideo oraz wybierania konkretnych kontaktów i grup z książki adresowej. Ponadto znajdziesz gotowe do wdrożenia przykłady z wykorzystaniem wielozadaniowości, geolokalizacji i aparatu fotograficznego. Książka ta jest idealną pozycją dla wszystkich osób tworzących oprogramowanie dla platformy iOS. Zawarte w niej przykłady zaoszczędzą Ci mnóstwo czasu. Ta inwestycja się opłaca!

- Używanie różnych podejść podczas tworzenia interfejsu użytkownika
- Tworzenie aplikacji wykorzystujących informacje o bieżącym położeniu urządzenia iOS
- Implementacje procedur rozpoznawania gestów
- Odtwarzanie plików audio i wideo oraz uzyskiwanie dostępu do biblioteki wbudowanej aplikacji Muzyka
- Pobieranie poszczególnych kontaktów i grup kontaktów z książki adresowej
- Określanie dostępności kamery w urządzeniu iOS oraz uzyskiwanie dostępu do biblioteki wbudowanej aplikacji Zdjęcia
- Tworzenie aplikacji wykorzystujących możliwość wielozadaniowości systemu iOS
- Stosowanie i obsługa trwałego magazynu danych w aplikacjach
- Używanie struktury Event Kit do zarządzania kalendarzami i wydarzeniami
- Wykorzystanie możliwości oferowanych przez strukturę Core Graphics
- Uzyskiwanie dostępu do informacji pochodzących z przyspieszoniomierza i żyroskopu
- Wykorzystanie możliwości oferowanych przez usługę iCloud

Zaoszczędź czas dzięki pomocnym przykładom!

helion.pl
księgarnia
internetowa

Nr katalogowy: 11210



Księgarnia internetowa:
<http://helion.pl>



Zamówienia telefoniczne:
0 801 339900



0 601 339900



Helion

Sprawdź najnowsze promocje:

🔗 <http://helion.pl/promocje>

Książki najchętniej czytane:

🔗 <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

🔗 <http://helion.pl/nowości>

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel.: 32 230 98 63

e-mail: helion@helion.pl

<http://helion.pl>

sklepnij po WIECEJ



KOD KORZYŚCI

ISBN 978-83-246-4994-5



Cena 119,00 zł