

Paweł Pasternak



Swift

od podstaw

Praktyczny przewodnik

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.

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

Redaktor prowadzący: Małgorzata Kulik

Projekt okładki: Studio Gravite / Olsztyn
Obarek, Pokoński, Pazdrijowski, Zaprucki

Fotografia na okładce została wykorzystana za zgodą Shutterstock.com

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/swifpk>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-3497-7

Copyright © Helion 2017

Printed in Poland.

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

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

Spis treści

- Wstęp 9**
- //TODO: 1. Swift 11**
 - //MARK: 1.1. Język programowania Swift 11
 - //MARK: 1.2. Narzędzia programisty 11
 - //MARK: 1.3. Podstawowe informacje o składni języka 12
- //TODO: 2. Playground 13**
 - //MARK: 2.1. Uruchomienie środowiska playground 13
 - //MARK: 2.2. "Hello, playground" 14
 - //MARK: 2.3. Quick Look 14
 - //MARK: 2.4. Pomoc kontekstowa 15
- //TODO: 3. Wyświetlamy dane print() 17**
 - //MARK: 3.1. Wyświetlanie tekstu 17
 - //MARK: 3.2. Wyświetlamy zmienne 18
- //TODO: 4. Deklarowanie stałych i zmiennych 19**
 - //MARK: 4.1. Deklarowanie zmiennych 19
 - //MARK: 4.2. Deklarowanie stałych 20
 - //MARK: 4.3. Optional 20
- //TODO: 5. Komentarze 23**
 - //MARK: 5.1. Komentarz liniowy 23
 - //MARK: 5.2. Komentarz blokowy 23
 - //MARK: 5.3. Komentarz bardziej czytelny 23
 - //MARK: 5.4. Komentarze zagnieżdżone 24
 - //MARK: 5.5. Grupowanie kodu, znacznik MARK 24
 - //MARK: 5.6. Grupowanie kodu, znacznik TODO 24
 - //MARK: 5.7. Grupowanie kodu, znacznik FIXME 24
- //TODO: 6. Typy danych 25**
 - //MARK: 6.1. Typ danych Bool 25
 - //MARK: 6.2. Typ danych Bit 25
 - //MARK: 6.3. Typ danych Int 25
 - //MARK: 6.4. Typ danych Int8 26

//MARK: 6.5. Typ danych UInt8	26
//MARK: 6.6. Typ danych Int16	26
//MARK: 6.7. Typ danych UInt16	26
//MARK: 6.8. Typ danych Int32	27
//MARK: 6.9. Typ danych UInt32	27
//MARK: 6.10. Typ danych Int64	27
//MARK: 6.11. Typ danych UInt64	27
//MARK: 6.12. Typ danych Float	27
//MARK: 6.13. Typ danych Double	28
//MARK: 6.14. Typ danych Float80	29
//MARK: 6.15. Typ danych Character (znakowy)	29
//MARK: 6.16. Typ danych Range	29
//MARK: 6.17. Typ danych String	30
//MARK: 6.18. Typ danych Array	31
//TODO: 7. Operator	33
//MARK: 7.1. Operator przypisania (=)	33
//MARK: 7.2. Operatory jednoargumentowe (- + !)	33
//MARK: 7.3. Operator inkrementacji (++)	33
//MARK: 7.4. Operator dekrementacji (--)	34
//MARK: 7.5. Operatory arytmetyczne (+ - * / %)	34
//MARK: 7.6. Operatory porównania (== === != <= < > >=)	35
//MARK: 7.7. Operator trójargumentowy (?)	35
//TODO: 8. Kontrola przepływu działania programu	37
//MARK: 8.1. Instrukcja if	37
//MARK: 8.2. Instrukcja switch	38
//TODO: 9. Pętle	41
//MARK: 9.1. Pętla for-in	41
//MARK: 9.2. Pętla for	41
//MARK: 9.3. Pętla while	42
//MARK: 9.4. Pętla repeat-while	42
//TODO: 10. Funkcje	43
//MARK: 10.1. Funkcje bez argumentów	43
//MARK: 10.2. Funkcje z argumentami	43
//MARK: 10.3. Funkcja zwraca wartość	45
//TODO: 11. Tablice	47
//MARK: 11.1. Deklarowanie tablicy	47
//MARK: 11.2. Wyświetlenie elementów tablicy	48
//MARK: 11.3. Liczba elementów tablicy	48
//MARK: 11.4. Dodawanie elementów do tablicy	49
//MARK: 11.5. Usuwanie elementu z tablicy	49
//MARK: 11.6. Łączenie dwóch tablic	51
//MARK: 11.7. Sortowanie tablicy	51

//TODO: 12. Słowniki	53
//MARK: 12.1. Deklarowanie słowników	53
//MARK: 12.2. Wyświetlenie elementów słownika	53
//MARK: 12.3. Liczba elementów słownika	54
//MARK: 12.4. Dodawanie elementów do słownika	54
//MARK: 12.5. Edycja elementu słownika	55
//MARK: 12.6. Usuwanie elementu słownika	55
//TODO: 13. Touples, czyli krotki	57
//MARK: 13.1. Deklaracja krotki	57
//MARK: 13.2. Przypisywanie krotki do zmiennej	58
//MARK: 13.3. Zagnieżdżone krotki	59
//TODO: 14. Typ wyliczeniowy	61
//MARK: 14.1. Deklaracja typu wyliczeniowego	61
//TODO: 15. Klasy	63
//MARK: 15.1. Deklaracja klasy	63
//MARK: 15.2. Konstruktor klasy	64
//MARK: 15.3. Destruktor klasy	64
//MARK: 15.4. Metody klasy	65
//MARK: 15.5. Getter i setter	66
//MARK: 15.6. Obserwator	66
//MARK: 15.7. Dziedziczenie	67
//TODO: 16. Extensions	71
//TODO: 17. Narzędzie Xcode	73
//MARK: 17.1. Uruchamiamy Xcode	73
//MARK: 17.2. Interfejs aplikacji	76
//TODO: 18. Budowa aplikacji	81
//MARK: 18.1. Outlets	81
//MARK: 18.2. Actions	84
//MARK: 18.3. Delegate	87
//TODO: 19. Kontrolki do obsługi tekstu	91
//MARK: 19.1. UILabel, najważniejsze atrybuty	91
//MARK: 19.2. UILabel, wyświetlenie tekstu	91
//MARK: 19.3. UITextField	94
//MARK: 19.4. Pobieramy tekst z TextField i wyświetlamy w Label	94
//MARK: 19.5. TextField — wprowadzanie hasła	97
//MARK: 19.6. TextField — ukrywanie klawiatury	98
//TODO: 20. Elementy formularza	101
//MARK: 20.1. Switch	101
//MARK: 20.2. Segmented Control	104
//MARK: 20.3. Slider	106

//TODO: 21. Nawigacja	111
//MARK: 21.1. Navigation Bar	111
//MARK: 21.2. Tab Bar	117
//TODO: 22. Kontrolery	121
//MARK: 22.1. Zmiana nazwy kontrolera (podstawowy widok)	121
//MARK: 22.2. Zmiana nazwy kontrolera (kolejne widoki)	122
//MARK: 22.3. Przesyłanie danych między kontrolerami	126
//TODO: 23. Table View	131
//MARK: 23.1. Tabela	131
//MARK: 23.2. Tablica, czyli źródło danych dla TableView	134
//MARK: 23.3. Dostosowujemy wiersz tabeli	136
//TODO: 24. Mapy	143
//MARK: 24.1. Uzyskanie uprawnień dla aplikacji	143
//MARK: 24.2. Moja lokalizacja na mapie	146
//MARK: 24.3. Dodanie punktu na mapie	148
//MARK: 24.4. Utworzenie regionu na mapie	149
//TODO: 25. Core Data	157
//MARK: 25.1. Podstawy	157
//MARK: 25.2. Struktura danych	158
//MARK: 25.3. Zapis danych	159
//MARK: 25.4. Odczyt danych	162
//TODO: 26. Testy	165
//MARK: 26.1. Automatyczne testowanie kodu	165
Podsumowanie	169
Skorowidz	171

//TODO: 6. Typy danych

//MARK: 6.1. Typ danych Bool

```
var toJestPrawda:Bool = true  
var toJestFalsz:Bool = false
```

Typ danych Bool przechowuje tylko dwie wartości — true lub false (prawda lub fałsz).

//MARK: 6.2. Typ danych Bit

```
var bitZero = Bit.Zero  
var bitOne = Bit.One
```

Typ danych Bit może przechowywać tylko dwie wartości — Zero lub One.

//MARK: 6.3. Typ danych Int

```
var liczbaCalkowita: Int = 23453
```

Typ danych Int jest jednym z najpopularniejszych typów zmiennych. Może przechowywać liczby z zakresu od -9223372036854775808 do 9223372036854775807 dla maszyn 64-bitowych, z kolei dla maszyn 32-bitowych zakres ten wynosi od -2147483648 do 2147483647. Analogicznie zachowuje się typ UInt, czyli liczba całkowita bez znaku (unsigned). Przyjmuje on różne maksymalne zakresy, w zależności od tego, czy program pracuje na platformie 32-bitowej, czy 64-bitowej. W języku Swift zdefiniowano osiem typów danych służących do przechowywania liczb całkowitych; przedstawia to tabela 6.1.

Tabela 6.1. *Typy danych służących do przechowywania liczb całkowitych*

Typ	Wartość minimalna	Wartość maksymalna
Int8	-128	127
Int16	-32 768	32 767
Int32	-2 147 483 648	2 147 483 647
Int64	-9 223 372 036 854 775 808	9 223 372 036 854 775 807
UInt8	0	255
UInt16	0	65 535
UInt32	0	4 294 967 295
UInt64	0	18 446 744 073 709 551 615

//MARK: 6.4. Typ danych Int8

```
var liczbaCalkowita: Int8 = -100
```

Typ Int8 może przechowywać liczby z zakresu od -128 do 127.

//MARK: 6.5. Typ danych UInt8

```
var liczbaCalkowita: UInt8 = 255
```

Typ UInt8 może przechowywać liczby z zakresu od 0 do 255. Literka *U* na początku nazwy oznacza *Unsigned*, czyli bez znaku.

//MARK: 6.6. Typ danych Int16

```
var liczbaCalkowita: Int16 = - 30000
```

Typ Int16 może przechowywać liczby z zakresu od -32768 do 32767.

//MARK: 6.7. Typ danych UInt16

```
var liczbaCalkowita: UInt16 = 65000
```

Typ UInt16 może przechowywać liczby z zakresu od 0 do 65535.

//MARK: 6.8. Typ danych Int32

```
var liczbaCalkowita: Int32 = - 30000000
```

Typ Int32 może przechowywać liczby z zakresu od -2147483648 do 2147483647.

//MARK: 6.9. Typ danych UInt32

```
var liczbaCalkowita: UInt32 = 4000000000
```

Typ UInt32 może przechowywać liczby z zakresu od 0 do 4294967295.

//MARK: 6.10. Typ danych Int64

```
var liczbaCalkowita: Int64 = 90000000000
```

Typ Int64 pozwala przechować bardzo duże liczby, z zakresu od -9223372036854775808 do 9223372036854775807.

//MARK: 6.11. Typ danych UInt64

```
var liczbaCalkowita: UInt64 = 10000000000000
```

Typ UInt64 pozwala przechować bardzo duże liczby dodatnie, z zakresu od 0 do 18446744073709551615.

//MARK: 6.12. Typ danych Float

```
var liczbaPi: Float = 3.14
```

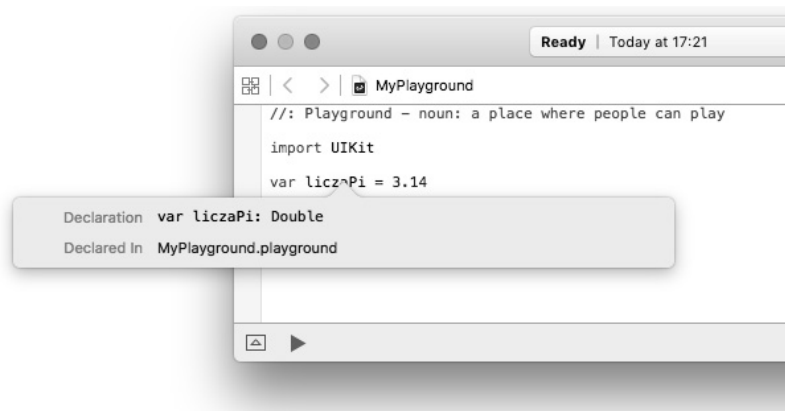
W języku Swift do reprezentacji liczb zmiennoprzecinkowych służą trzy typy danych; są to typ Float (pozwala na przechowywanie danych na 32 bitach), typ Double (<alias Float64> przechowuje dane na 64 bitach) oraz Float80 (<extended-precision> wykorzystuje aż 80 bitów).

```
var liczbaPi = 3.14
```

Deklarując zmienną w ten sposób, zadeklarujemy ją nie jako typ `Float`, lecz jako typ `Double`.

Swift domyślnie zwiększa dokładność danych (rysunek 6.1).

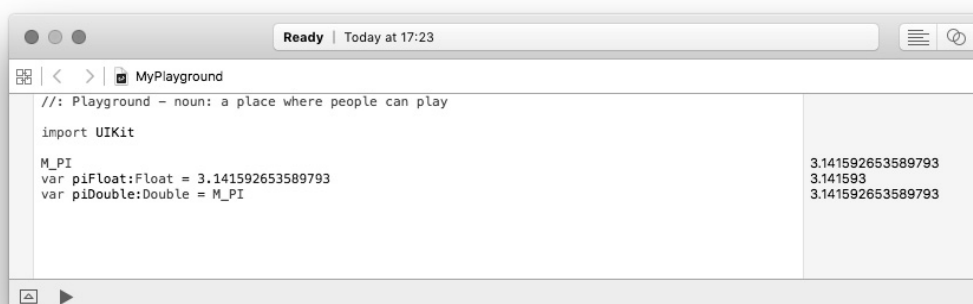
Rysunek 6.1.
Deklaracja zmiennej
typu `Float`



//MARK: 6.13. Typ danych `Double`

```
var liczbaPiWiekszaPrecyzja: Double = 3.141592
```

Typ `Double` również pozwala na przechowywanie liczb zmiennoprzecinkowych, jednak w odróżnieniu od typu `Float` ten typ umożliwia przechowywanie liczb z większą dokładnością po przecinku niż typ `Float` (rysunek 6.2).



Rysunek 6.2. Deklaracja zmiennej typu `Double`

Dla przykładu porównamy liczbę π przypisaną do zmiennej typu `Float` i `Double`. Stała `M_PI` jest liczbą π zdefiniowaną we frameworku Foundation. Kiedy podstawimy tę wartość do zmiennych typu `Float` i `Double`, widzimy różnicę na szóstym miejscu po przecinku. Przy dokładnych obliczeniach mogłoby to generować błędy.

//MARK: 6.14. Typ danych Float80

```
var liczbaZmiennoPrzecinkowa: Float80 = 3.141592
```

Typ Float80 wykorzystuje 80 bitów do przechowywania danych.

//MARK: 6.15. Typ danych Character (znakowy)

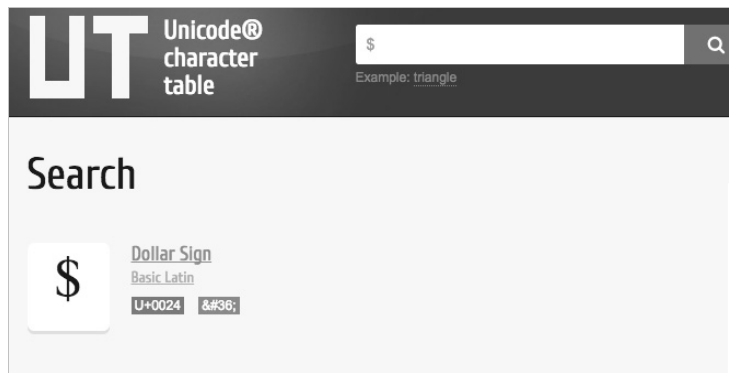
```
var litera:Character = "a"
```

Typ danych Character służy do reprezentacji jednego znaku tekstowego.

```
var symbolDolara = "\u{0024}" // wyświetlimy znak $
```

Znak dolara wyświetlamy za pomocą konstrukcji `\u{n}`, gdzie `n` jest liczbą szesnastkową reprezentującą symbol w tablicy Unicode. Listę symboli można znaleźć pod adresem <https://unicode-table.com/en/#control-character> (rysunek 6.3).

Rysunek 6.3.
Strona internetowa
z symbolami
Unicode



//MARK: 6.16. Typ danych Range

```
var dniPracujace: Range = 1..<6
```

Typ danych Range określa przedział liczb w zakresie „od do”. Taka deklaracja oznacza przedział prawostronnie otwarty. Początek przedziału to cyfra 1, a koniec przedziału to cyfra 6. Przedział w tym przypadku składa się z cyfr (1, 2, 3, 4, 5).

```
var dniPracujace: Range = 1...5
```

Taka deklaracja określa przedział prawostronnie domknięty. Początek przedziału to cyfra 1, a koniec przedziału to cyfra 6. Przedział w tym przypadku składa się z cyfr (1, 2, 3, 4, 5).

```
// Przypadek dniPracujace:Range = 1..<6
var dniPracujace:startIndex // wyświetli cyfrę 1
var dniPracujace:endIndex   // wyświetli cyfrę 5

// Przypadek dniPracujace:Range = 1...5
var dniPracujace:startIndex // wyświetli cyfrę 1
var dniPracujace:endIndex   // wyświetli cyfrę 5
```

Do określenia początku przedziału służy właściwość obiektu o nazwie `startIndex`, a do określenia końca przedziału — właściwość `endIndex`.

//MARK: 6.17. Typ danych String

```
var imie: String = "Paweł"
var nazwisko: String = "Pasternak"
```

Typ danych `String` służy do przechowywania napisów. Napisy ujmujemy w podwójny cudzysłów. Poniżej przedstawię kilka najbardziej popularnych operacji na tym typie danych.

```
imie.lowercaseString //paweł
```

Metoda `lowercaseString` zamienia duże litery na małe litery.

```
imie.uppercaseString //PAWEŁ
```

Metoda `uppercaseString` zamienia małe litery na duże litery.

```
imie + " " + nazwisko //Paweł Pasternak
```

Do łączenia dwóch napisów typu `String` służy znak `+`. Jest to duże ułatwienie w porównaniu z `Objective-C`.

```
imie.endIndex //4
```

Właściwość `endIndex` określa długość ciągu znaków.

```
var znak: Character = "1"
imie.append(znak) //Paweł1
```

Metoda `append` służy do dodawania pojedynczego znaku do zmiennej typu `String`.

//MARK: 6.18. Typ danych Array

```
var wartoscVat[Int] = [0, 9, 18, 23]
```

Tablica służy do przechowywania wartości tego samego typu. Ta sama wartość może pojawić się w tablicy wielokrotnie. Elementy tablicy umieszczamy w kwadratowym nawiasie i rozdzielamy przecinkiem. Poniżej przedstawię podstawowe operacje związane z tablicami.

```
wartoscVat.count // 4
```

Metoda count wyświetli ilość elementów tablicy.

```
wartoscVat.append(25) // [0, 9, 18, 23, 25]
```

Metoda append doda nowy element do tablicy.

```
wartoscVat.dropFirst() // [9, 18, 23, 25]
```

Metoda dropFirst usuwa pierwszy element tablicy.

```
wartoscVat.dropFirst() // [9, 18, 23, 25]
```

Metoda dropLast usuwa ostatni element tablicy.

```
wartoscVat.insert(19, atIndex:3) // [0, 9, 18, 19, 23]
```

Metoda insert wstawia element na określonej pozycji.

Skorowidz

A

- akcja, 84
- aplikacja, 81
 - uzyskanie uprawnień, 143
- atrybuty
 - nowego projektu, 75
 - pola tekstowego, 94
- automatyczne testowanie, 165

B

- biblioteka MapKit, 143
- błędy, 168
- budowa aplikacji, 81

C

- Connections Inspector, 88
- Core Data, 157
 - odczyt danych, 162
 - struktura danych, 158
 - zapis danych, 159

D

- Debug area, 79
- deklarowanie
 - klasy, 63
 - krotki, 57
 - słownika, 53
 - stałej, 20
 - tablicy, 47
 - typu Optional, 20
 - typu wyliczeniowego, 61
 - zmiennej, 19

- delegacja, 87
- destruktor klasy, 64
- dodawanie
 - elementów do słownika, 54
 - elementów do tablicy, 49
 - przycisku, 84
- dziedziczenie, 67

E

- Editor area, 78
- edycja elementu słownika, 55
- elementy formularza, 101
- etykieta Label, 91, 92
- extensions, 71

F

- formularze, 101
- framework Core Data, 157
- funkcja print(), 17
- funkcje
 - bez argumentów, 43
 - z argumentami, 43
 - zwracające wartość, 45

G

- getter, 66
- grupowanie kodu, 24

H

- hasło, 97

I

IDE, Integrated Development Environment, 11
 identyfikator TableViewCell, 139
 instrukcja
 if, 37
 switch, 38
 interfejs, 76, 82, 85

J

język Swift, 11
 Jump bar, 77

K

klasa
 Square, 165, 167
 Square, 166
 klasy, 63
 deklaracja, 63
 destruktor, 64
 dziedziczenie, 67
 konstruktor, 64
 metody, 65
 klucz CLLocationWhenInUseUsageDescription, 145
 komentarze
 blokowe, 23
 liniowe, 23
 zagnieżdżone, 24
 konstruktor klasy, 64
 kontroler, 82, 85, 121
 przesyłanie danych, 126
 zmiana nazwy, 121, 122
 kontrolka
 MKMapView, 143
 Segmented Control, 104
 Slider, 107
 Switch, 102
 TextField, 81
 kontrolki do obsługi tekstu, 91
 krotki, 57
 deklaracja, 57
 przypisywanie do zmiennej, 58
 zagnieżdżone, 59

L

Library panel, 79
 liczba elementów
 słownika, 54
 tablicy, 48
 lista, 112
 logika biznesowa, 81
 lokalizacja, 146

Ł

łączenie dwóch tablic, 51

M

mapa, 143
 dodawanie punktów, 148
 pineska, 155
 tworzenie regionów, 149
 współrzędne regionu, 156
 wyświetlanie lokalizacji, 146
 metoda
 didSet, 66
 get, 66
 set, 66
 willSet, 66
 metody
 klasy, 65
 testowe, 166
 MKMapView, 143

N

narzędzia programisty, 11
 narzędzie Xcode, 73
 Navigation Bar, 111
 kolor przycisków, 116
 przycisk przejścia, 114
 widok szczegółów, 114
 Navigator panel, 76
 nawigacja, 111
 Navigation Bar, 111
 Tab Bar, 117

O

obserwator, 66
 obsługa
 interfejsu, 82, 84
 TableViewCell, 138
 tekstu, 91
 odczyt danych, 162
 okno Xcode, 73
 opcja Quick Look, 14
 operator
 dekrementacji, 34
 inkrementacji, 33
 przypisania, 33
 trójargumentowy, 35
 operatory
 arytmetyczne, 34
 jednoargumentowe, 33
 porównania, 35
 Optional, 20
 Optional Binding, 21

P

pętla
 for, 41
 for-in, 41
 repeat-while, 42
 while, 42
 playground, 13
 pomoc kontekstowa, 15
 Quick Look, 14
 uruchomienie środowiska, 13
 plik info.plist, 145
 pole tekstowe, 94
 atributy, 94
 połączenie
 dataSource z View Controller, 132, 137
 między widokami, 128
 Outlet, 93
 projekt Map View, 143
 punkt na mapie, 148

R

region na mapie, 149
 rozszerzenia, 71

S

schemat połączeń widoków, 115
 Segmented Control, 104
 setter, 66
 składnia języka, 12
 Slider, 106
 słowniki, 53
 deklarowanie, 53
 dodawanie elementów, 54
 edycja elementu, 55
 liczba elementów, 54
 usuwanie elementu, 55
 wyświetlenie elementów, 53
 słowo kluczowe
 newValue, 66
 override, 68
 sortowanie tablicy, 51
 stałe, 20
 struktura danych, 158
 Swift, 11
 Switch, 101
 symulowana lokalizacja telefonu, 151
 symulowany ruch osoby, 151

Ś

środowisko
 playground, 13
 Xcode, 73

T

Tab Bar, 117
 łączenie widoku, 119
 tabela, 131
 dostosowanie wierszy, 136
 Table View, 131
 tablica, 134
 tablice, 47, 134
 deklarowanie, 47
 dodawanie elementów, 49
 liczba elementów, 48
 łączenie, 51
 sortowanie, 51
 usuwanie elementu, 49
 wyświetlenie elementów, 48
 tekst, 91
 testy automatyczne, 165

TextField, 94
 ukrywanie klawiatury, 98
 wprowadzanie hasła, 97
 zdarzenia, 95
 Toolbar, 77
 tworzenie
 aplikacji, 74
 Outlet, 95, 107
 połączenia, 88
 typ danych
 Array, 31
 Bit, 25
 Bool, 25
 Character, 29
 Double, 28
 Float, 27
 Float80, 29
 Int, 25
 Int16, 26
 Int32, 27
 Int64, 27
 Int8, 26
 Range, 29
 String, 30
 UInt16, 26
 UInt32, 27
 UInt64, 27
 UInt8, 26
 typy
 hierarchiczne, 111
 płaskie, 111
 sterowania treścią, 111
 wyliczeniowe, 61

U

UILabel, 91
 UITextField, 94
 ukrywanie klawiatury, 98
 uprawnienia dla aplikacji, 143
 usuwanie elementu
 słownika, 55
 z tablicy, 49
 Utilities area, 78

V

View Controller, 119

W

warstwa prezentacji danych, 81
 wartość nil, 21
 widok
 interfejsu, 82, 85, 92
 kontrolera, 82, 85, 92
 mapy, 144
 okna listy, 112
 tabeli, 134, 136, 142
 Table View, 131
 wiersz tabeli, 136
 wprowadzanie hasła, 97
 wyświetlanie
 lokalizacji, 146
 tekstu, 17
 zmiennej, 18
 elementów słownika, 53
 elementów tablicy, 48
 tekstu, 91

X

Xcode
 atrybuty nowego projektu, 75
 Debug area, 79
 Editor area, 78
 interfejs aplikacji, 76
 Jump bar, 77
 Library panel, 79
 Navigator panel, 76
 Toolbar, 77
 tworzenie aplikacji, 74
 uruchamianie, 73
 Utilities area, 78

Z

zakładka Connections Inspector, 88
 zapis danych, 159
 zintegrowane środowisko programistyczne, 11
 zmiana nazwy kontrolera, 121
 zmienne, 18
 znacznik
 FIXME, 24
 MARK, 24
 TODO, 24

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**

Apple od lat z powodzeniem i ze świetnym wyczuciem potrzeb klientów działa na rynku elektroniki użytkowej. Pracujący w tej firmie inżynierowie projektują produkty oryginalne, funkcjonalne i wyprzedzające swoją epokę. Ważną składową tego sukcesu jest wysokiej jakości oprogramowanie, które można tworzyć wyłącznie za pomocą wydajnych, wygodnych w użyciu i nowoczesnych narzędzi programistycznych. Ten przewodnik poświęcono językowi Swift, który jest podstawą oprogramowania współpracującego z macOS, iOS, watchOS oraz tvOS. Wprowadzony na rynek w 2014 roku Swift doczekał się już trzeciej wersji i powoli wypiera królujący od lat Objective-C.

Jeśli chcesz poznać najnowszą wersję języka Swift oraz środowisko programistyczne Xcode, sięgnij po tę książkę! Dzięki niej opanujesz przejrzystą składnię Swifta, typy danych oraz sposoby deklarowania stałych i zmiennych. Dowiesz się, jak tworzyć instrukcje warunkowe i pętle oraz jak korzystać z operatorów, funkcji, tablic, słowników, krotek i klas. Poznasz elementy interfejsu użytkownika, nauczysz się zarządzać danymi w swoich aplikacjach i posługiwać się komentarzami. Zrozumiesz także, jak działają narzędzia oferowane przez środowisko Xcode, i zyskasz wiedzę o podstawach pisania testów jednostkowych. Czas na naukę!

- Podstawowe informacje o języku Swift i środowisku programistycznym Xcode
- Korzystanie z Playgrounds i wyświetlanie danych na ekranie
- Proste typy danych oraz używanie zmiennych i stałych
- Korzystanie z operatorów i tworzenie funkcji
- Używanie instrukcji warunkowych i różnych rodzajów pętli
- Posługiwanie się złożonymi typami danych i możliwości obiektowe
- Korzystanie z kontrolek, obsługa akcji i delegacji
- Zapisywanie i odczytywanie danych przy użyciu Core Data
- Tworzenie komentarzy i pisanie testów jednostkowych

Wkrocz w świat programowania na platformy Apple!

Helion

księgarnia internetowa



<http://helion.pl>

zamówienia telefoniczne



0 801 339900



0 601 339900

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>

Sprawdź najnowsze promocje:
• <http://helion.pl/promocje>
Książki najchętniej czytane:
• <http://helion.pl/bestsellery>
Zamów informacje o nowościach:
• <http://helion.pl/nawosci>

sięgnij po **WIĘCEJ**



KOD KORZYŚCI

ISBN 978-83-283-3497-7



9 788328 334977

Informatyka w najlepszym wydaniu

cena: 37,00 zł