

spis treści

przedmowa xvii

podziękowania xix

o książce xxi

o autorze xxv

ilustracja na okładce xxvi

I Pierwsze kroki 1

1.1. O Erlangu 1

1.1.1. Wysoka dostępność 3

1.1.2. Współbieżność Erlanga 4

Odporność na błędy 4 ■ Skalowalność 4 ■ Rozproszenie 5
■ Czas reakcji 5

1.1.3. Systemy serwerowe 6

1.1.4. Platforma programistyczna 8

1.2. O Elixirze 9

1.2.1. Uproszczenie kodu 9

1.2.2. Składanie funkcji 13

1.2.3. Szersza perspektywa 14

1.3. Wady 14

1.3.1. Prędkość 14

1.3.2. Ekosystem 15

2 Podstawy 16

2.1. Interaktywna powłoka 17

2.2. Praca ze zmiennymi 18

2.3. Organizacja kodu 20

2.3.1. Moduły 20

2.3.2. Funkcje 21

2.3.3. Arność funkcji 25

2.3.4. Widoczność funkcji 26

2.3.5. Importy i aliasy 27

2.3.6. Atrybuty modułu 28

Specyfikacje typów 29

2.3.7. Komentarze 30

2.4. Rozumienie systemu typów 30

2.4.1. Liczby 30

2.4.2. Atomy 31

Alias 32 ■ Atomy jako wartości logiczne 33 ■ Wartości nil i truthy 33

2.4.3. Krotki 34

2.4.4. Listy 35

Definicja listy rekurencyjnej 36

2.4.5. Niemutowalność 38

Modyfikowanie krotek 38 ■ Modyfikowanie list 39 ■ Korzyści 40

2.4.6. Mapy 41

Mapy dynamicznie zmieniające rozmiar 41 ■ Dane strukturalne 42

2.4.7. Ciągi binarne i ciągi bitowe 43

2.4.8. Ciągi znaków 44

Stringi binarne 44 ■ Listy znaków 46

2.4.9. Funkcja jako typy pierwszoklasowe 47

Domknięcia 48

2.4.10. Inne wbudowane typy 49

2.4.11. Typy wyższego rzędu 50

Range 50 ■ Listy asocjacyjne 50 ■ MapSet 52 ■ Czasy i daty 52

2.4.12. Listy IO 53

2.5. Operatory 54

2.6. Makra 55

2.7. Rozumienie czasu uruchomienia 57

2.7.1. Moduły i funkcje w czasie uruchomienia 57

Nazwy modułów i atomy 57 ■ Moduły Erlanga 58
■ Dynamiczne wywoływanie funkcji 59

2.7.2. Wystartowanie środowiska uruchomieniowego 59

Interaktywna powłoka 59 ■ Używanie skryptów 59 ■ Narzędzie mix 60

3 Przepływ sterowania 62

3.1. Dopasowanie wzorców 63

3.1.1. Operator dopasowujący 63

3.1.2. Dopasowanie krotek 63

3.1.3. Dopasowywanie stałych 64

3.1.4. Zmienne we wzorcach 65

3.1.5. Dopasowywanie list 66

3.1.6. Dopasowywanie map 67

3.1.7. Dopasowywanie ciągów bitowych i ciągów bajtowych 68

Dopasowywanie ciągów binarnych 69

3.1.8. Dopasowania złożone 70

3.1.9. Ogólne zachowanie 71

3.2. Dopasowanie do funkcji 71

3.2.1. Funkcje wielowarunkowe 72

3.2.2. Strażnicy 75

3.2.3. Lambdy wielowarunkowe 77

3.3. Instrukcje warunkowe 78

3.3.1. Rozgałęzianie z funkcjami wielowarunkowymi 78

3.3.2. Klasyczne konstrukcje rozgałęziania 80

If oraz unless 80 ■ Instrukcje warunkowe 81 ■ Case 81

3.3.3. Specjalna forma with 82

3.4. Pętle i iteracje 85

3.4.1. Iteracja z rekurencją 86

3.4.2. Wywołanie ogonowe 87

Rozpoznawanie wywołań ogonowych 89 ■ Ćwiczenie 90

3.4.3. Funkcje wyższego rzędu 90

Reduce 93

3.4.4. Składanie list 95

3.4.5. Strumienie 97

Ćwiczenie 100

4 Abstrakcje danych 102

4.1. Tworzenie abstrakcji za pomocą modułów 104

4.1.1. Podstawowa abstrakcja 104

4.1.2. Łączenie abstrakcji 106

4.1.3. Strukturyzacja danych za pomocą map 107

4.1.4. Abstrakcje za pomocą struktur 109

Struktury a mapy 112 ■ Rekordy 113

4.1.5. Transparentność danych 113

4.2. Praca z danymi hierarchicznymi 115

4.2.1. Generowanie ID 115

4.2.2. Aktualizacja wpisów 118

Zabawa z dopasowywaniem wzorców 119

4.2.3. Niemutowalne hierarchiczne aktualizacje 120

Dostępni pomocnicy 121 ■ Ćwiczenie: usuwanie wpisu 122

4.2.4. Aktualizacje iteracyjne 122

4.2.5. Ćwiczenie: importowanie z pliku 123

4.3. Polimorfizm z protokołami 124

4.3.1. Podstawy protokołów 125

4.3.2. Implementacja protokołu 126

4.3.3. Protokoły wbudowane 127

Collectable lista zadań 127

5 Podstawy współbieżności 129

5.1. Współbieżność w BEAM 129

- 5.2. Praca z procesami 132
 - 5.2.1. Tworzenie procesów 133
 - 5.2.2. Przekazywanie wiadomości 134
 - Algorytm receive 137 ■ Wysyłanie synchroniczne 137
 - Zbieranie wyników kwerendy 138
- 5.3. Procesy serwerów stanowych 140
 - 5.3.1. Procesy serwera 140
 - Procesy serwera są sekwencyjne 143
 - 5.3.2. Utrzymywanie stanu procesu 144
 - 5.3.3. Stan mutowalny 146
 - Refaktoryzacja pętli 148
 - 5.3.4. Stany złożone 149
 - Podjęcie współbieżne a podejście funkcyjne 152
 - 5.3.5. Procesy rejestrowane 152
- 5.4. Dyskusja na temat czasu uruchomienia 153
 - 5.4.1. Proces jest sekwencyjny 154
 - 5.4.2. Nieograniczone skrzynki pocztowe procesu 155
 - 5.4.3. Współbieżność współdzielenia niczego (*shared-nothing concurrency*) 156
 - 5.4.4. Działanie dyspozytora 157

6 Procesy serwera generycznego 160

- 6.1. Tworzenie procesu serwera generycznego 161
 - 6.1.1. Podłączanie się pod moduły 161
 - 6.1.2. Implementowanie kodu generycznego 162
 - 6.1.3. Zastosowanie magazynu danych par 163
 - 6.1.4. Wsparcie dla żądań asynchronicznych 165
 - 6.1.5. Ćwiczenie: refaktoryzacja serwera zadań 167
- 6.2. Wykorzystanie GenServer 167
 - 6.2.1. Zachowania (*behaviours*) OTP 168
 - 6.2.2. Podłączanie się do GenServer 169
 - 6.2.3. Obsługa żądań 170
 - 6.2.4. Obsługa wiadomości 171
 - 6.2.5. Pozostałe cechy GenServer 173
 - Sprawdzanie czasu kompilacji 173 ■ Rejestracja nazw 174
 - Zatrzymywanie serwera 175

- 6.2.6. Cykl życia procesu 176
- 6.2.7. Procesy zgodne z OTP 177
- 6.2.8. Ćwiczenie: serwer zadań zasilany GenServer 177

7 Tworzenie współbieżnego systemu 179

- 7.1. Praca z mix 180
- 7.2. Zarządzanie wieloma listami zadań 182
 - 7.2.1. Implementowanie pamięci podręcznej 182
 - 7.2.2. Pisanie testów 185
 - 7.2.3. Analiza zależności procesu 187
- 7.3. Utrwalanie danych 189
 - 7.3.1. Kodowanie i utrwalanie 189
 - 7.3.2. Używanie bazy danych 191
 - Przechowywanie danych 192 ■ Odczyt danych 192
 - 7.3.3. Analiza systemu 194
 - 7.3.4. Podejmowanie kroków względem wąskiego gardła 195
 - Współbieżna obsługa zadań 196 ■ Ograniczanie współbieżności poprzez pulę 197
 - 7.3.5. Ćwiczenie: pula procesów i synchronizacja 198
- 7.4. Dyskusja na temat procesów 199

8 Podstawy odporności na błędy 201

- 8.1. Błędy czasu uruchomienia 202
 - 8.1.1. Typy błędów 202
 - 8.1.2. Obsługa błędów 204
- 8.2. Błędy w systemach współbieżnych 207
 - 8.2.1. Łączenie procesów 208
 - Przechwytywanie wyjść 210
 - 8.2.2. Monitory 210
- 8.3. Nadzorcy (*supervisor*) 211
 - 8.3.1. Przygotowanie istniejącego kodu 213
 - 8.3.2. Uruchamianie procesu nadzorcy 214
 - 8.3.3. Specyfikacja dzieci 216
 - 8.3.4. Opakowanie nadzorcy 218

- 8.3.5. Korzystanie z modułu wywołania zwrotnego 219
- 8.3.6. Łączenie wszystkich procesów 220
- 8.3.7. Częstotliwość restartów 222

9 Izolowanie skutków błędów 225

9.1. Drzewa nadzorców 226

- 9.1.1. Oddzielanie luźno powiązanych elementów 226
- 9.1.2. Rozbudowane wykrywanie procesów 229
- 9.1.3. Krótka via 232
- 9.1.4. Rejestracja procesów roboczych bazy danych 233
- 9.1.5. Nadzór nad procesami roboczymi baz danych 235
- 9.1.6. Organizacja drzewa nadzorców 239
 - Procesy zgodne z OTP 240 ■ Zamykanie procesów 240
 - Unikanie restartów procesu 241 ■ Strategie restartów 242

9.2. Dynamiczne uruchamianie procesów 242

- 9.2.1. Rejestracja serwerów ToDo 242
- 9.2.2. Nadzór dynamiczny 243
- 9.2.3. Wyszukiwanie serwerów zadań 245
- 9.2.4. Stosowanie tymczasowej strategii restartu 246
- 9.2.5. Testowanie systemu 247

9.3. „Daj mu się wywalić” 248

- 9.3.1. Procesy, które nie powinny ulegać awarii 249
- 9.3.2. Obsługa oczekiwanych błędów 250
- 9.3.3. Zachowanie stanu 251

10 Poza granicami GenServer 253

10.1. Zadania (*tasks*) 254

- 10.1.1. Zadania oczekujące 254
- 10.1.2. Zadania nieoczekujące 256

10.2. Agenci (*agents*) 258

- 10.2.1. Podstawowe zastosowania 259
- 10.2.2. Agenci i współbieżność 259
- 10.2.3. Serwer zadań oparty na agencie 261
- 10.2.4. Ograniczenia agentów 262

10.3. Tabele ETS 265

10.3.1. Podstawowe operacje 267

10.3.2. Magazyn danych par klucz/wartość oparty na ETS 270

10.3.3. Pozostałe operacje ETS 273

Wzorce dopasowania 274 ■ Inne przypadki zastosowania ETS 275

■ Poza ETS 276

10.3.4. Ćwiczenie: rejestr procesów 277

11 Praca z komponentami 279**11.1. Aplikacje OTP 280**

11.1.1. Tworzenie aplikacji za pomocą narzędzia mix 280

11.1.2. Zachowanie aplikacji 282

11.1.3. Uruchomienie aplikacji 282

11.1.4. Biblioteki jako aplikacje 283

11.1.5. Tworzenie aplikacji to-do 284

11.1.6. Struktura folderu aplikacji 286

Środowiska projektów mix 286 ■ Struktura skompilowanego kodu 286

11.2. Praca z zależnościami 288

11.2.1. Dodawanie zależności 288

11.2.2. Adaptacja puli 289

11.2.3. Wizualizacja systemu 292

11.3. Tworzenie serwera WWW 293

11.3.1. Wybór zależności 293

11.3.2. Uruchamianie serwera 294

11.3.3. Obsługa żądań 295

11.3.4. Rozprawianie o systemie 298

Wydajność 300 ■ Wywołania a rzuty (*call vs cast*) 300**11.4. Konfiguracja aplikacji 302**

11.4.1. Środowisko aplikacji 302

11.4.2. Zmienna konfiguracja 303

11.4.3. Kwestie skryptu konfiguracji 305

12 Tworzenie systemu rozproszonego 307

12.1. Prymitywy rozproszenia 309

12.1.1.	Uruchomienia klastra	309
12.1.2.	Komunikacja między węzłami	311
12.1.3.	Odnajdywanie procesów	313
	Grupy procesów	315
12.1.4.	Łączy i monitory	316
12.1.5.	Inne usługi rozproszenia	317
12.2.	Tworzenie klastra odpornego na błędy	319
12.2.1.	Kompozycja klastra	320
12.2.2.	Rozproszona pamięć podręczna zadań	320
	Odnajdywanie serwerów zadań	321
	Alternatywna metoda odnajdywania procesów	324
12.2.3.	Implementacja replikowanej bazy danych	325
12.2.4.	Testowanie systemu	328
12.2.5.	Wykrywanie partycji	330
12.2.6.	Systemy wysokiej niezawodności	331
12.3.	Kwestie sieciowe	332
12.3.1.	Nazwy węzłów	332
12.3.2.	Ciasteczka	333
12.3.3.	Węzły ukryte	334
12.3.4.	Zapory sieciowe	334
	Bezpieczeństwo	335
13	Uruchamianie systemu	337
13.1.	Uruchomienie systemu przy użyciu narzędzi Elixir	338
13.1.1.	Stosowanie komend mix i elixir	338
13.1.2.	Uruchamianie skryptów	340
13.1.3.	Kompilacja na potrzeby produkcji	341
13.2.	Wydania OTP	342
13.2.1.	Tworzenie wydania za pomocą destylarni (<i>distillery</i>)	343
13.2.2.	Stosowanie wydania	344
13.2.3.	Zawartość wydania	346
	Skompilowane binaria	346
	Konfiguracje	348
	Spakowany pakiet wydania	348
	Precyzyjny montaż wydania	349

13.3. Analiza zachowań systemu	349
13.3.1. Debugowanie	350
13.3.2. Logowanie	351
13.3.3. Interakcja z systemem	352
13.3.4. Śledzenie	353

indeks	357
---------------	-----