

Spis treści

Wprowadzenie

Dla kogo jest przeznaczona ta książka? (xxvi)

Wiemy, co sobie myślisz (xxvii)

Wiemy, co sobie myśli Twój mózg (xxvii)

Metapoznanie - myślenie o myśleniu (xxix)

Oto co zrobiliśmy (xxx)

Przeczytaj to (xxxii)

Podziękowania (xxxiii)

ROZDZIAŁ 1. Zaczynamy. Podstawy składni

Do biegu, gotowi, Go! (2)

Narzędzie Go Playground (3)

Co to wszystko znaczy? (4)

A jeśli coś się nie powiedzie? (5)

Wywoływanie funkcji (7)

Funkcja Println (7)

Używanie funkcji z innych pakietów (8)

Wartości zwracane przez funkcje (9)

Szablon programu w Go (11)

łańcuchy znaków (11)

Runy (12)

Wartości logiczne (12)

Liczby (13)

Operacje matematyczne i porównania (13)

Typy (14)

Deklarowanie zmiennych (16)

Wartości zerowe (17)

Krótkie deklaracje zmiennych (19)

Reguły tworzenia nazw (21)

Konwersje (22)

Instalowanie Go na komputerze (25)

Kompilowanie kodu w języku Go (26)

Narzędzia języka Go (27)

Szybkie sprawdzanie działania kodu za pomocą polecenia go run (27)

Twój przybornik do Go (28)

ROZDZIAŁ 2. Jaki kod uruchomić w następnej kolejności? Instrukcje warunkowe i pętle

Wywoływanie metod (32)

Obliczanie oceny (34)

Funkcje i metody zwracające wiele wartości (36)

Rozwiązanie 1. Zignorować wartość błędu za pomocą pustego identyfikatora (37)

Rozwiązanie 2. Obsługa błędu (38)

Instrukcje warunkowe (39)

Warunkowe rejestrowanie błędu krytycznego (42)

Unikaj zakrywania nazw (44)

Przekształcanie łańcuchów znaków na liczby (46)

Bloki (49)

Bloki i zasięg zmiennych (50)

Program do wystawiania ocen jest gotowy! (52)

Tylko jedna zmienna w krótkiej deklaracji zmiennej musi być nowa (54)

Napiszmy grę (55)

Nazwy pakietów a ścieżki importowania (56)

Generowanie liczby losowej (57)

Pobieranie liczby całkowitej z klawiatury (59)

Porównywanie wytypowanej liczby z docelową (60)

Pętle (61)

Instrukcje inicjalizacji i instrukcje wykonywane po iteracji są opcjonalne (63)

Używanie pętli w grze w zgadywanie liczb (66)

Wychodzenie z pętli zgadywania liczby (69)

Ujawnianie docelowej liczby (70)

Gratulacje, gra jest kompletna! (72)

Twój przybornik do Go (74)

ROZDZIAŁ 3. Wywołaj mnie. Funkcje

Powtarzający się kod (80)

Formatowanie danych wyjściowych z użyciem funkcji Printf i Sprintf (81)

Instrukcje formatowania (82)

Formatowanie długości wartości (83)

Formatowanie długości liczb ułamkowych (84)

Używanie funkcji Printf w kalkulatorze ilości farby (85)

Deklarowanie funkcji (86)

Deklarowanie parametrów funkcji (87)

Używanie funkcji w kalkulatorze ilości farby (88)

Funkcje i zasięg zmiennych (90)

Wartości zwracane przez funkcje (91)

Używanie zwracanej wartości w kalkulatorze ilości farby (93)

W funkcji paintNeeded potrzebna jest obsługa błędów (95)

Wartości błędów (96)

Deklarowanie wielu zwracanych wartości (97)

Używanie wielu zwracanych wartości w funkcji paintNeeded (98)

Zawsze obsługuj błędy! (99)

W parametrach funkcji zapisywane są kopie argumentów (102)

Wskaźniki (103)

Typy wskaźnikowe (104)

Pobieranie lub modyfikowanie wartości wskazywanej przez wskaźnik (105)

Używanie wskaźników w funkcjach (107)

Poprawianie funkcji double z użyciem wskaźników (108)

Twój przybornik do Go (110)

ROZDZIAŁ 4. Pakiety kodu. Pakiety

Różne programy, ta sama funkcja (114)

Współdzielenie kodu w programach z użyciem pakietów (116)

Kod pakietów jest przechowywany w obszarze roboczym języka Go (117)

Tworzenie nowego pakietu (118)

Importowanie pakietów do programu (119)

W pakietach używany jest ten sam układ plików (120)

Konwencje tworzenia nazw pakietów (123)

Kwalifikatory w postaci nazw pakietów (123)

Przenoszenie wspólnego kodu do pakietu (124)

Stałe (126)

Zagnieżdżone katalogi pakietów i ścieżki importowania (128)

Instalowanie programów wykonywalnych za pomocą instrukcji go install (130)

Modyfikowanie obszaru roboczego za pomocą zmiennej środowiskowej GOPATH (131)

Ustawianie zmiennej GOPATH (132)

Publikowanie pakietów (133)

Pobieranie i instalowanie pakietów z użyciem polecenia go get (137)

Wczytywanie dokumentacji pakietu za pomocą polecenia go doc (139)

Dokumentowanie pakietów z użyciem komentarzy narzędzia doc (141)

Wyświetlanie dokumentacji w przeglądarce internetowej (143)

Udostępnianie dokumentacji w formacie HTML na swoje potrzeby za pomocą narzędzia godoc (144)

Serwer narzędzia godoc udostępnia też TWOJE pakiety! (145)

Twój przybornik do Go (146)

ROZDZIAŁ 5. Na liście. Tablice

Tablice przechowują kolekcje wartości (150)

Wartości zerowe w tablicach (152)

Literały tablicowe (153)

Funkcje z pakietu fmt potrafią obsługiwać tablice (154)

Dostęp do elementów tablicy w pętli (155)

Sprawdzanie długości tablicy za pomocą funkcji len (156)

Bezpieczne przetwarzanie tablic w pętli za pomocą instrukcji for...range (157)

Używanie pustego identyfikatora w pętlach for...range (158)

Obliczanie sumy liczb z tablicy (159)

Pobieranie średniej liczb z tablicy (161)

Wczytywanie pliku tekstowego (163)

Wczytywanie pliku tekstowego do tablicy (166)

Modyfikowanie programu average, aby wczytywał plik tekstowy (168)

Nasz program potrafi przetwarzać tylko trzy wartości! (170)

Twój przybornik do Go (172)

ROZDZIAŁ 6. Problem dołączania elementów. Wycinki

Wycinki (176)

Literały wycinków (177)

Operator wycinka (180)

Tablice podstawowe (182)

Modyfikowanie podstawowych tablic i wycinków (183)

Dodawanie elementów do wycinka za pomocą funkcji append (184)

Wycinki i wartości zerowe (186)

Wczytywanie dodatkowych wierszy plików z użyciem wycinków i funkcji append (187)

Sprawdzanie poprawionego programu (189)

Zwracanie wycinka nil po wystąpieniu błędu (190)

Argumenty wiersza poleceń (191)

Pobieranie argumentów wiersza poleceń z wycinka os.Args (192)

Operator wycinka można stosować do innych wycinków (193)

Modyfikowanie programu, aby używał argumentów wiersza poleceń (194)

Funkcje wariadyczne (195)

Stosowanie funkcji wariadycznych (197)

Użycie funkcji wariadycznej do obliczania średnich (198)

Przekazywanie wycinków do funkcji wariadycznych (199)

Wycinki uratowały sytuację! (201)

Twój przybornik do Go (202)

ROZDZIAŁ 7. Nazywanie danych. Mapy

Zliczanie głosów (206)

Wczytywanie nazwisk z pliku (207)

Zliczanie nazwisk w trudny sposób, z użyciem wycinków (209)

Mapy (212)

Literały map (214)

Wartości zerowe w mapach (215)

Wartość zerowa zmiennej reprezentującej mapę to nil (215)

Jak odróżnić wartości zerowe od przypisanych? (216)

Usuwanie par klucz - wartość za pomocą funkcji delete (218)

Modyfikowanie programu zliczającego głosy, aby użyć map (219)

Używanie pętli for...range do map (221)

Pętla for...range przetwarza mapy w losowej kolejności! (223)

Modyfikowanie programu zliczającego głosy - zastosowanie pętli for...range (224)

Program do zliczania głosów jest kompletny! (225)

Twój przybornik do Go (227)

ROZDZIAŁ 8. Tworzenie struktur do przechowywania danych. Struktury

Wycinki i mapy przechowują wartości JEDNEGO typu (232)

Struktury są tworzone z wartości WIELU typów (233)

Dostęp do pól struktury za pomocą operatora kropki (234)

Zapisywanie danych prenumeratorów w strukturze (235)

Typy zdefiniowane i struktury (236)

Używanie typu zdefiniowanego na dane prenumeratorów magazynu (238)

Używanie typów zdefiniowanych razem z funkcjami (239)

Modyfikowanie struktury z użyciem funkcji (242)

Dostęp do pól struktur za pomocą wskaźnika (244)

Przekazywanie dużych struktur z użyciem wskaźników (246)

Przenoszenie typu struktury do innego pakietu (248)

Nazwa zdefiniowanego typu musi zaczynać się wielką literą, aby została wyeksportowana (249)

Nazwy pól struktury muszą zaczynać się wielką literą, aby zostały wyeksportowane (250)

Literały struktur (251)

Tworzenie struktury typu Employee (253)

Tworzenie struktury typu Address (254)

Dodawanie struktury jako pola w innym typie (255)

Podawanie wartości struktury w innej strukturze (255)

Anonimowe pola struktur (258)

Zagnieżdżanie struktur (259)

Typy zdefiniowane są gotowe! (260)

Twój przybornik do Go (261)

ROZDZIAŁ 9. Jesteś w moim typie. Typy zdefiniowane

Błędne typy w rzeczywistym życiu (266)

Typy zdefiniowane z prostymi typami bazowymi (267)

Typy zdefiniowane i operatory (269)

Przekształcenia między typami z użyciem funkcji (271)

Rozwiązywanie kolizji nazw funkcji z użyciem metod (274)

Definiowanie metod (275)

Parametr odbiorcy metody jest (prawie) identyczny jak inne parametry (276)

Metoda działa (prawie) jak funkcja (277)

Wskaźniki jako parametry odbiorcy metody (279)

Przeliczanie litrów i mililitrów na galony z użyciem metod (283)

Przeliczanie galonów na litry i mililitry z użyciem metod (284)

Twój przybornik do Go (285)

ROZDZIAŁ 10. Zachowaj to dla siebie. Hermetyzacja i zagnieżdżanie

Tworzenie typu strukturalnego Date (290)

Użytkownicy przypisują do pól struktury Date nieprawidłowe wartości! (291)

Settery (292)

W setterze jako odbiorcę trzeba podać wskaźnik (293)

Dodawanie pozostałych setterów (294)

Dodawanie sprawdzania poprawności danych do metod (296)

Pola nadal mogą zostać ustawione na błędne wartości! (298)

Przenoszenie typu Date do innego pakietu (299)

Jak sprawić, by pola z typu Date nie były eksportowane? (301)

Dostęp do nieeksportowanych pól za pomocą eksportowanych metod (302)

Gettery (304)

Hermetyzacja (305)

Zagnieżdżanie typu Date w typie Event (308)

Nieeksportowane pola nie są promowane (309)

Eksportowane metody są promowane w taki sam sposób jak pola (310)

Hermetyzowanie pola Title z typu Event (312)

Promowane metody są dostępne w taki sam sposób jak metody typu zewnętrznego (313)

Pakiet calendar jest gotowy! (314)

Twój przybornik do Go (316)

ROZDZIAŁ 11. Co potrafisz zrobić? Interfejsy

Dwa różne typy mające te same metody (322)

Parametr metody, który przyjmuje wartości tylko jednego typu (323)

Interfejsy (325)

Definiowanie typu implementującego interfejs (326)

Typy konkretne i typy interfejsowe (327)

Przypisz wartość dowolnego typu implementującego interfejs (328)

Wywoływać można tylko metody zdefiniowane w interfejsie (329)

Poprawianie funkcji playList z użyciem interfejsu (331)

Asercje typów (334)

Nieudane asercje typów (336)

Unikanie paniki po niepowodzeniu asercji (337)

Testowanie typów TapePlayer i TapeRecorder z użyciem asercji typów (338)

Interfejs error (340)

Interfejs Stringer (342)

Pusty interfejs (344)

Twój przybornik do Go (347)

ROZDZIAŁ 12. Znow stań na nogi. Przywracanie stanu po awarii

Jeszcze o wczytywaniu liczb z pliku (350)

Błędy powodują, że plik nie zostanie zamknięty (352)

Odraczanie wywołań funkcji (353)

Przywracanie stanu po błędach z użyciem odroczonej wywołań funkcji (354)

Używanie odroczonej wywołań do gwarantowania zamknięcia plików (355)

Wyświetlanie plików z katalogu (358)

Wyświetlanie plików z podkatalogów (bardziej skomplikowane) (359)

Wywołania funkcji rekurencyjnych (360)

Rekurencyjne wyświetlanie zawartości katalogu (362)

Obsługa błędów w funkcji rekurencyjnej (364)

Uruchamianie procedury paniki (365)

Ślad stosu (366)

Odroczone wywołania ukończone przed awarią (366)

Stosowanie funkcji panic w funkcji scanDirectory (367)

Kiedy wywoływać panikę? (368)

Funkcja recover (370)

Funkcja recover zwraca wartość z wywołania panic (371)

Przywracanie stanu po wywołaniu funkcji panic w funkcji scanDirectory (373)

Ponowne wywołanie funkcji panic (374)

Twój przybornik do Go (376)

ROZDZIAŁ 13. Udostępnianie kodu. Wątki goroutine i kanały

Pobieranie stron internetowych (380)

Wielozadaniowość (382)

Współbieżność z użyciem wątków goroutine (383)

Używanie wątków goroutine (384)

Używanie wątków goroutine w funkcji responseSize (386)

Brak bezpośredniej kontroli nad wykonywaniem wątków goroutine (388)

Instrukcji go nie można używać razem ze zwracaniem wartości (389)

Wysyłanie i przyjmowanie wartości z użyciem kanałów (391)

Synchronizowanie wątków goroutine za pomocą kanałów (392)

Obserwowanie synchronizacji wątków goroutine (393)

Zastosowanie kanałów do poprawienia programu zwracającego wielkość stron (396)

Modyfikowanie kanału, aby przysyłał strukturę (398)

Twój przybornik do Go (399)

ROZDZIAŁ 14. Kontrola jakości kodu. Testy zautomatyzowane

Testy zautomatyzowane wykrywają błędy, zanim zrobi to ktoś inny (402)

Funkcja, dla której należało przygotować testy zautomatyzowane (403)

Spowodowaliśmy błąd (405)

Pisanie testów (406)

Uruchamianie testów za pomocą polecenia go test (407)

Testowanie zwracanych wartości (408)

Tworzenie szczegółowych komunikatów o niepowodzeniu testów za pomocą funkcji Errorf (410)

Funkcje pomocnicze w testach (411)

Sprawianie, by testy kończyły się powodzeniem (412)

Programowanie sterowane testami (413)

Następny błąd do naprawienia (414)

Uruchamianie określonego zbioru testów (417)

Testy sterowane tabelami (418)

Stosowanie testów do poprawiania kodu powodującego panikę (420)

Twój przybornik do Go (422)

ROZDZIAŁ 15. Reagowanie na żądania. Aplikacje internetowe

Pisanie aplikacji internetowych w języku Go (426)

Przeglądarki, żądania, serwery i odpowiedzi (427)

Prosta aplikacja internetowa (428)

Twój komputer rozmawia sam ze sobą (429)

Omówienie prostej aplikacji internetowej (430)

Ścieżki do zasobów (432)

Reagowanie w inny sposób na różne ścieżki do zasobów (433)

Funkcje pierwszoklasowe (435)

Przekazywanie funkcji do innych funkcji (436)

Funkcje jako typy (436)

Co dalej? (440)

Twój przybornik do Go (441)

ROZDZIAŁ 16. Stosowanie szablonów. Szablony HTML

Księga gości (446)

Funkcje do obsługi żądań i sprawdzanie błędów (447)

Tworzenie katalogu projektu i wypróbowywanie aplikacji (448)

Tworzenie listy wpisów w HTML-u (449)

Zwracanie stron HTML w aplikacji (450)

Pakiet text/template (451)

Stosowanie interfejsu io.Writer razem z metodą Execute szablonu (452)

Typy ResponseWriter i os.Stdout implementują interfejs io.Writer (453)

Wstawianie danych do szablonów z użyciem akcji (454)

Tworzenie opcjonalnych części szablonu za pomocą akcji if (455)

Powtarzanie sekcji szablonu za pomocą akcji range (456)

Wstawianie pól struktury do szablonu za pomocą akcji (457)

Wczytywanie wycinka z wpisami z pliku (458)

Struktura do przechowywania wpisów i ich liczby (460)

Modyfikowanie szablonu w celu umieszczenia w nim wpisów (461)

Umożliwianie użytkownikom dodawania danych za pomocą formularzy HTML-owych (464)

Żądania przesłania formularza (466)

Ścieżki i metody HTTP do przesyłania formularzy (467)

Pobieranie z żądania wartości pól formularza (468)

Zapisywanie danych z formularza (470)

Przekierowania HTTP (472)

Kompletny kod aplikacji (474)

Twój przybornik do Go (477)

DODATEK A. Zrozumieć funkcję os.OpenFile. Otwieranie plików

Zrozumieć funkcję os.OpenFile (482)

Przekazywanie do funkcji os.OpenFile stałych reprezentujących opcje (483)

Notacja dwójkowa (485)

Operatory bitowe (485)

Bitowy operator I (486)

Bitowy operator LUB (487)

Stosowanie bitowego operatora LUB dla stałych z pakietu os (488)

Stosowanie bitowego operatora LUB do poprawienia opcji funkcji os.OpenFile (489)

Uniksowe uprawnienia do plików (490)

Reprezentowanie uprawnień za pomocą typu os.FileMode (491)

Notacja ósemkowa (492)

Przekształcanie wartości ósemkowych na wartości typu FileMode (493)

Objaśnienie wywołań funkcji os.OpenFile (494)

DODATEK B. Sześć kwestii, które nie zostały opisane. Pozostałości

Numer 1. Inicjalizacja w instrukcji if (496)

Numer 2. Instrukcja switch (498)

Numer 3. Inne typy proste (499)

Numer 4. Jeszcze o runach (499)

Numer 5. Kanały buforowane (503)

Numer 6. Dalsza lektura (506)