

Podsumowanie

CO DALEJ?

To już koniec tej książki, ale jeszcze nie koniec twojej podróży. Mamy nadzieję, że dzięki niej wpadniesz na wiele pomysłów i nabierzesz chęci do dalszej nauki i do tworzenia. Dziękujemy za dołączenie do nas!

Czego nie powiedzieliśmy

Go to stosunkowo mały język i w większości już go poznałeś. Ale jest kilka spraw, których ta książka w tym wydaniu nie porusza:

- Nie opisaliśmy deklarowania stałych sekwencyjnych za pomocą przydatnego słowa kluczowego `iota`.
- Nie wspomnieliśmy o operatorach przesunięcia bitowego (`<<` i `>>`) ani o operatorach bitowego iloczynu (`&`) i sumy (`|`).
- Lekcja 3 opisuje pętle, ale pominęliśmy słowo kluczowe `continue` i słowo kluczowe `goto` oraz etykiety.
- Lekcja 4 jest poświęcona zasięgowi, ale pominęliśmy kwestię przesłaniania zmiennych.
- Lekcje od 6 do 8 wykorzystują liczby zmiennoprzecinkowe, całkowite i typy dla dużych wartości, ale pominęliśmy liczby zespolone i urojone.
- Lekcja 12 opisuje słowo kluczowe `return`, ale pominęliśmy jego użycie bez zwracanej wartości.
- Lekcja 12 wspomina o pustym interfejsie `interface{}`, ale tylko pobieżnie.
- Lekcja 13 przedstawia metody, ale nie powiedzieliśmy o wartościach metod.
- Lekcja 28 wspomina o asercjach typu, ale nie wspomina o instrukcji `switch` w wersji rozróżniającej typy.
- Lekcja 30 nie wspomina o kanałach kierunkowych.

- Nie opisaliśmy inicjowania za pomocą `init`, czyli specjalnej funkcji, podobnej do `main`.
- Nie przedstawiliśmy szczegółowych informacji o wszystkich wbudowanych funkcjach, jak `new` dla wskaźników i `copy` dla wycinków (zobacz golang.org/pkg/builtin/).
- Nie wyjaśniliśmy, jak pisać nowe pakiety, aby uporządkować kod lub umożliwić jego udostępnianie innym.

Poza placem zabaw

Jeśli dopiero zaczynasz programowanie, to zapewne podoba ci się internetowa platforma Go Playground, ale musisz wiedzieć, że ten plac zabaw ma pewne ograniczenia.

Aby uwolnić się od nich i pisać bardziej poważne programy, musisz zainstalować Go na swoim komputerze (zobacz golang.org/dl/). Otwarcie okna terminalu i korzystanie z wiersza poleceń przypomina trochę cofnięcie się w czasie o kilka dekad, więc naucz się poruszać po komputerze w ten sposób i uruchamiać programy tak, jak się to robiło w 1995 roku!

Będziesz także potrzebował edytora tekstu. My używaliśmy edytorów Sublime Text i Acme, ale dostępnych jest wiele innych edytorów z dobrym wsparciem dla Go (zobacz golang.org/doc/editors.html). Prędzej czy później będziesz potrzebować też narzędzia do kontroli wersji, takiego jak `git`, które działa trochę jak wehikuł czasu, ale na szczęście tylko w odniesieniu do kodu i innych plików.

To nie wszystko

Go to znacznie więcej niż język programowania. Wokół niego powstał bogaty ekosystem narzędzi i bibliotek, które czekają, aż je odkryjesz.

Pomogą ci one niemal we wszystkim: w zautomatyzowanych testach, w debugowaniu, w testach wydajności i w wielu innych sprawach. Biblioteka standardowa ma też bardzo dużo pakietów, które warto poznać. A jeśli poznasz już wszystko, to pamiętaj, że społeczność susłów cały czas pracuje nad wieloma niezależnymi pakietami zaspokajającymi bardzo zróżnicowane potrzeby (zobacz godoc.org).

W Internecie dostępnych jest wiele zasobów (zobacz golang.org/wiki), które pomogą ci w dalszej nauce. Dostępnych jest też wiele książek, choćby takich jak: *Go in Practice*, *Go Web Programming* oraz *Go in Action* (zobacz golang.org/wiki/Books).

Nauka nigdy się nie kończy, a więc baw się i ucz! Razem ze społecznością Go gorąco cię do tego zachęcamy!