

nika oraz generować liczby losowe. Będziemy pisać własne klasy do realizowania nowych zadań, jak również wykorzystywać klasy dostępne już w języku Java do wykonywania codziennych czynności, takich jak obsługa danych wejściowych, operacje matematyczne i nie tylko.

## Zadania programistyczne

Aby utrwalić i przećwiczyć zdobyte umiejętności oraz rozszerzyć swoją wiedzę, wykonaj następujące zadania programistyczne. Jeśli napotkasz problemy, można wejść na stronę internetową tej książki o adresie <https://www.nostarch.com/learnjava/>, aby pobrać przykładowe rozwiązania lub obejrzeć wideo z instrukcjami krok po kroku w ramach kursu online dostępnego na stronie <http://www.udemy.com/java-the-easy-way/>. Rozdział 2 można wypróbować za darmo. Można również użyć kodu BOOKHALFOFF, aby otrzymać 50 procent zniżki na zakup całego kursu.

### Zadanie 1: rozszerzanie zakresu

W tym pierwszym zadaniu zmodyfikuj zgadywanę w taki sposób, aby losowała ona liczbę z szerszego zakresu. Aby użytkownik musiał odgadnąć liczbę z przedziału od  $-100$  do  $100$  zamiast od  $1$  do  $100$ .

#### Wskazówka

Pomnóż `Math.random()` przez  $200$  i odejmij  $100$  od wyniku.

Pamiętaj o zmodyfikowaniu *zarówno* instrukcji programistycznej służącej do generowania liczby losowej, *jak i* przedstawianej użytkownikowi instrukcji z informacją, w jakim przedziale mieści się odgadywana liczba.

Jeśli chcesz ułatwić grę, możesz zmniejszyć przedział od  $1$  do  $10$  i zaskoczyć znajomych umiejętnością odgadnięcia sekretnej liczby w maksymalnie czterech podejściach. Przetestuj inne zakresy, np. od  $1$  do  $1000$  lub nawet do  $1\ 000\ 000$ , bądź użyj liczb ujemnych (pamiętaj o niestosowaniu spacji ani kropek w liczbach całkowitych w języku Java). W ten sposób możesz udoskonalić nie tylko swoje umiejętności programistyczne, ale również matematyczne. Zachęcamy do zabawy z programem i modyfikowania go na różne sposoby!

### STRATEGIA ODGADYWANIA

Z reguły im dłużej gramy w zgadywanę, tym mniejszej liczby prób potrzebujemy do odgadnięcia sekretnej liczby. W ten sposób możemy odkryć, że najszybsze rozwiązanie polega na wytypowaniu liczby znajdującej się w środku zakresu w każdej nowej próbie. Ta technika jest nazywana *wyszukiwaniem binarnym*. Typowanie liczby znajdującej się w środku potencjalnego zakresu za każdym razem skraca liczbę możliwości o połowę.

Metoda ta działa w następujący sposób. Z przedziału od 1 do 100 typujemy liczbę 50. Jeśli jest ona zbyt niska, wiemy, że sekretna liczba musi leżeć w przedziale od 51 do 100, a zatem typujemy środek przedziału, czyli 75. Jeśli ta liczba również jest zbyt niska, znowu typujemy liczbę w środku przedziału od 76 do 100, czyli 87. Wyszukiwanie binarne działa bardzo optymalnie i dla liczby z przedziału od 1 do 100 pozwala zredukować liczbę prób do maksymalnie siedmiu. Spróbuj sam(a)!

Po opanowaniu procesu zgadywania liczby od 1 do 100 w maksymalnie siedmiu próbach warto spróbować odgadnąć liczbę od 1 do 1000 w jedynie 10 próbach. A prawdziwi śmiatkwowie (uzbrojeni w coś do pisania) mogą spróbować odgadnąć liczbę od 1 do 1 000 000. Choć trudno w to uwierzyć, powinno wystarczyć zaledwie 20 prób.

## Zadanie 2: zliczanie prób

Zbudowaliśmy już całkiem fajną aplikację zgadywanek, jednak warto spróbować dodać jeszcze jedną funkcję. Zadanie polega na rejestrowaniu liczby prób i wyświetlaniu informacji, ilu prób użytkownik potrzebował do odgadnięcia sekretnej liczby. Rozwiązanie mogłoby wyglądać podobnie do następującego:

---

62 to prawidłowa liczba! Wygrzasz!  
Liczba prób to jedynie: 7! Gratulacje!

---

Aby osiągnąć ten cel, trzeba utworzyć nową zmienną licznika prób (można dodać np. wiersz kodu `int numberOfTries = 0;`), a następnie zwiększać licznik prób podczas każdego wykonania pętli zgadywania. Możesz osiągnąć ten cel, zwiększając w każdym kolejnym wykonaniu pętli wartość zmiennej `numberOfTries` o jeden przy użyciu kodu `numberOfTries = numberOfTries + 1`. Pamiętaj o dodaniu tekstu informującego użytkownika o liczbie prób.

Nie zawsze od razu udaje się utworzyć kod, który realizuje odpowiednie operacje w odpowiedniej kolejności, jednak nie należy się zniechęcać, ponieważ poświęcony czas pomaga w utrwaleniu nowych umiejętności. W rozdziale 3 wbudujemy funkcję zliczania prób w inną wersję zgadywanek. A na razie mamy nadzieję, że przyjdą Ci do głowy dodatkowe pomysły na ulepszenie i zmodyfikowanie gry. Zabawa z programami, rozbieranie ich na części pierwsze i odbudowywanie to najlepszy sposób na naukę.

## Zadanie 3: gra MadLibs

Ostatnie wyzwanie w tym rozdziale będzie polegało na napisaniu całkiem nowego programu. Wiesz już, jak odczytać dane wejściowe wpisane przez użytkownika i umieścić je w zmiennej. Umiesz także drukować tekst i wartości zmiennych na ekranie. Te umiejętności umożliwiają budowanie jeszcze ciekawszych programów.