

kurs języka Java**zgadywanie liczby wymiernej**Instytut Informatyki
Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zaprogramuj prostą grę: automat losuje liczbę wymierną $a/b \in (0, 1)$ a gracz stara się odgadnąć jej wartość. Prawidłowo wylosowana liczba wymierna a/b to taka, gdzie $0 < a < b \leq Z$, dla pewnej ustalonej i znanej graczowi wartości całkowitej $Z \geq 4$, będącej zakresem losowanych wartości. Gracz ma co najwyżej $\lceil 3 \cdot \log(Z) \rceil$ prób na odgadnięcie tej liczby (po wyczerpaniu dopuszczalnej liczby prób bez sukcesu gracz przegrywa). Po każdej próbie gracz dostaje od automatu informację o swojej propozycji: „za mało”, „za dużo” albo „zgadza się” (w tym ostatnim przypadku gra kończy się wygraną gracza).

Do zaprogramowania tej gry wykorzystaj dwie klasy: klasę reprezentującą liczbę wymierną oraz klasę reprezentującą stan gry (wspomnianego wcześniej automatu).

Część 1.

W pakiecie obliczenia zdefiniuj klasę `Wymierna` reprezentującą liczbę wymierną w postaci nieskracalnego ułamka z licznikiem i mianownikiem typu `int`.

```
public class Wymierna {
    private int licznik, mianownik = 1;
    // ...
}
```

W klasie tej zdefiniuj trzy konstruktory: bezparametrowy (tworzący liczbę $0/1$ reprezentującą zero), z jednym parametrem n (tworzący liczbę całkowitą postaci $n/1$) oraz z dwiema wartościami k i m (tworzący liczbę wymierną postaci k/m). Niech konstruktor jednoargumentowy będzie konstruktorem delegatowym korzystającym z konstruktora dwuargumentowego. W konstruktorze dwuargumentowym zadбай o to, by zgłosić wyjątek `IllegalArgumentException` w przypadku mianownika równego 0, dla liczb ujemnych przenieść znak minusa do licznika. Przekształć ułamek do postaci nieskracalnej (podziel licznik i mianownik przez NWD licznika i mianownika, korzystając z algorytmu Euklidesa zapisanego w postaci prywatnej metody rekurencyjnej); na przykład, gdy wywołamy konstruktor `Wymierna(75, -60)` to powstanie obiekt z liczbą wymierną $-5/4$ (gdyż $\text{NWD}(75, 60) = 15$ a zatem $-75/15 = -5/4$ po skróceniu).

W klasie tej powinny się znaleźć gettery udostępniające licznik i mianownik. Nadpisz metodę `toString()`, która przedstawi liczbę wymierną do postaci łańcucha znakowego oraz metodę `equals()`, która sprawdzi czy dwie liczby wymierne są takie same. Zaimplementuj także interfejs `Comparable<Wymierna>`, aby można było porównywać ze sobą liczby wymierne.

Na koniec zdefiniuj publiczne statyczne metody realizujące podstawowe operacje arytmetyczne na liczbach wymiernych: dodawanie, odejmowanie, mnożenie i dzielenie (pamiętaj aby zgłosić wyjątek `ArithmeticException` przy próbie dzielenia przez 0).

Część 2.

W pakiecie `rozgrywka` zdefiniuj klasę `Gra`, która będzie reprezentować stan gry. W skład stanu gry wchodzi wylosowana wartość liczby wymiernej, zakres wartości spośród których losowany był licznik i mianownik tej liczby, liczba wykonanych prób odgadnięcia liczby oraz informacja, czy liczbę odgadnięto w przewidzianej do tego celu liczbie prób.

```
public class Gra {
    private int zakres;
    private Wymierna liczba;
    private int maksIlośćPrób;
    private int licznikPrób;
    // ...
}
```

Każda rozgrywka rozpoczyna się od wylosowania liczby wymiernej. Proces losowania tej liczby rozpoczyna się od ustalenia zakresu losowanych wartości a następnie wylosowaniu licznika i mianownika o wartościach należących do zbioru $\{1, 2, \dots, \text{zakres}\}$:

```
public void start(int z) {
    if (z < 5 || z > 20) throw ...;
    zakres = z;
    //do {
        int licz = (int) (Math.random() * zakres) + 1;
        int mian = (int) (Math.random() * zakres) + 1;
    //while (licz >= mian);
    liczba = new Wymierna(licz, mian);
    // inicjalizacja: maksIlośćPrób, licznikPrób, ...
    assert ...; // czy 0 < liczba < 1
}
```

Funkcja `start()` ma rozpoczynać rozgrywkę. Nie zmieniaj jej implementacji, pomimo błędu logicznego – z prawdopodobieństwem > 0.5 wylosowana liczba będzie miała wartość spoza przedziału $(0, 1)$. Dopisz za to na końcu asercję, która sprawdzi czy na pewno licznik był mniejszy od mianownika – asercja ta będzie powodowała przerwanie programu w ponad połowie przypadków.

Scenariusz przebiegu rozgrywki jest następujący. Początkowo gra znajduje się w stanie *nieaktywnym* – to w tym stanie można zmieniać parametr dotyczący zakresu losowanych wartości licznika i mianownika. Następnie gra jest uruchamiana i przechodzi do stanu *aktywnego* – podczas startu następuje wylosowanie ułamka a następnie użytkownik próbuje odgadnąć wylosowaną liczbę. Rozgrywka kończy się przejściem do jednego z trzech stanów końcowych: gdy użytkownik się podda (stan *rezygnacja*) albo gdy wyczerpie wszystkie możliwe próby (stan *porażka*) albo gdy odgadnie ułamek (stan *zwycięstwo*).

Część 3.

Napisz aplikację okienkową z wykorzystaniem standardowej biblioteki graficznej AWT umieszczonej w pakiecie `java.awt` i podpakietach, która będzie umożliwiała przeprowadzenie gry przez użytkownika i będzie na bieżąco prezentowała stan gry.

W pakiecie `prezentacja` zdefiniuj główne okno aplikacji jako klasę `Okno` dziedziczącą po klasie `Frame`. W oknie umieść dwa pola tekstowe typu `TextField` przeznaczone na wpisanie ułamka (licznik i mianownik), przycisk typu `Button` do wysłania propozycji do silnika gry, suwak typu `Scrollbar` do

prezentacji ilości wykorzystanych prób, suwak typu Scrollbar do sterowania zakresem losowanych ułamków, przyciski typu Button do przerywania rozgrywki, do rozpoczęcia nowej oraz do zakończenia działania aplikacji. Zauważ, że nie wszystkie elementy GUI powinny być zawsze aktywne – powinno to być zależne od stanu rozgrywki.

W grze dodaj funkcjonalność ułatwiającą podejmowanie decyzji co do kolejnych strzałów:

- po każdym strzale użytkownik powinien otrzymać informację, czy wytypowana wartość jest mniejsza czy większa od szukanego ułamka;
- po każdym celnym strzale zmniejsza się przedział poszukiwań ułamka – przedział ten powinien być uwidoczniony i aktualizowany na bieżąco (para liczb zmiennopozycyjnych);
- po najechaniu kursorem myszy na przycisk do wysłania propozycji użytkownika do silnika gry, ale przed kliknięciem akceptującym, należy wyświetlić liczbę zmiennopozycyjną odpowiadającą zaproponowanemu ułamkowi.

Część 4.

Dodatkowo loguj wszystkie ważne wydarzenia w grze i zapisuj je w pliku rozgrywka.log w bieżącym katalogu. Ważnym wydarzeniem może być na przykład wystartowanie rozgrywki, próba trafienia wartości liczby wymiernej, informacja o stanie zakończenia rozgrywki, itp. Sposób logowania skonfiguruj we własnym pliku konfiguracyjnym logging.properties umieszczonym w bieżącym katalogu.

W aplikacji zgłaszaj wyjątki (i wyłapuj je), gdy gracz wprowadzi błędne dane: gdy wpisany ciąg znaków nie da się przekonwertować na liczbę wymierną albo gdy wpisana wartość jest spoza przedziału (0, 1) albo gdy w mianowniku wpisano wartość większą od maksymalnej dopuszczalnej. Przyczyny błędów mogą być różne, dlatego zdefiniuj własną hierarchię wyjątków na okoliczność każdego przypadku.