

## Programowanie współbieżne

### Lista 6

1. Napisz klasę aktora dla serwera, który generuje losowo liczbę całkowitą z przedziału  $[0 .. N]$  ( $N > 1$  ma być parametrem konstruktora serwera), wyświetla odpowiednią informację i czeka na komunikaty od klientów, które próbują zgadnąć tę liczbę. Po otrzymaniu komunikatu wysyła odpowiedź z informacją, czy liczba klienta była mniejsza, większa czy równa liczbie wylosowanej.
2. Napisz klasę aktora dla klienta zgadującego liczbę, wylosowaną przez serwer. Konstruktor klienta ma mieć trzy parametry: nazwę klienta, referencję do serwera i górną granicę przedziału do zgadywania. Próby klienta powinny być optymalne (wyszukiwanie binarne). Klient ma wyświetlać informacje, umożliwiające śledzenia działania aplikacji (patrz przykład poniżej). Po odgadnięciu liczby klient ma wyświetlić odpowiedni komunikat i zakończyć działanie systemu aktorów.
3. Napisz kompletną aplikację, która utworzy instancję serwera i co najmniej dwa klienty, zgadujące liczbę wylosowaną przez serwer. W metodzie `main` wyślij do wszystkich klientów komunikat `Start`, co powinno rozpocząć zgadywanie.

**Wszystkie zadania umieść w pliku `Lista12.scala`.**

```
[info] Running Guess
Guess my number from the interval [0..100]
Client1 starting
Client2 starting
Client1 trying: 50
Client2 trying: 50
Client2. Response: too big. I'm trying: 25
Client1. Response: too big. I'm trying: 25
Client1. Response: too small. I'm trying: 37
Client2. Response: too small. I'm trying: 37
Client2. Response: too big. I'm trying: 31
Client1. Response: too big. I'm trying: 31
Client2. Response: too small. I'm trying: 34
Client1. Response: too small. I'm trying: 34
Client1: I guessed it! 34
Client2: I guessed it! 34
Goodbye! from Client2
Goodbye! from Client1
```

Gra będzie ciekawsza, kiedy każdy klient w pierwszej próbie wyśle losową liczbę z podanego przedziału, a potem będzie już zgadywał optymalnie (**i tak należy zrobić!**).