

## Zaawansowane programowanie obiektowe

### Lab. 3

(3 pkt) Zasymuluj wyścig kolarski na czas. W wyścigu uczestniczy 15 kolarzy. Co 1 minutę (czasu symulowanego) startuje następny kolarz; jego czas, wyrażony w całkowitej liczbie sekund, jest liczbą losową o rozkładzie Gaussa ze średnią 300s i odch. standardowym 30 (ale czas nie może być krótszy niż 250s ani dłuższy niż 370s, tj. jeśli wylosuje się np. 246s, to przyjmujemy 250s, a jeśli wylosuje się 381s, to przyjmujemy 370s).

Nazwiska kolarzy (bez imion) mają być wzięte losowo z pliku:

<http://szgrabowski.kis.p.lodz.pl/zpo17/nazwiska.txt> (który jest odczytywany spod swojego URLa, tj. proszę go nie ściągać wcześniej na dysk), nie dopuszczamy do powtórzeń nazwisk. Plik zakodowany jest w UTF-8.

W symulacji 1 sekunda czasu rzeczywistego ma odpowiadać 25s czasu symulowanego. Na bieżąco ma być pokazywany czas zawodnika, który właśnie ukończył wyścig oraz aktualna posortowana czołówka 3 zawodników (ich nazwiska i czasy). Na samym początku ta czołówka będzie mniej liczna (0..2 kolarzy). Dobierz odpowiednie komponenty GUI (Swing lub JavaFX).

Dodatkowo każde zdarzenie symulacji ma być logowane do pliku (java.util.logging.Logger, java.util.logging.LogManager).

Wskazówki:

- 1) do losowania nazwisk bez powtórzeń możesz wykorzystać zbiór (java.util.Set),
- 2) czytanie bezpośrednio spod URLa:  
<http://docs.oracle.com/javase/tutorial/networking/urls/readingURL.html> ,
- 3) uruchamianie „zadań” w regularnych odstępach czasowych: java.util.Timer lub javax.swing.Timer,
- 3) do znajdowania aktualnej czołówki wykorzystaj kolejkę priorytetową (java.util.PriorityQueue):  
<https://docs.oracle.com/javase/8/docs/api/java/util/PriorityQueue.html>