

## Zaawansowane programowanie obiektowe

### Lab. 3

(3 pkt) Zasymuluj wyścig kolarski na czas. W wyścigu uczestniczy 12 kolarzy. Co 1 minutę (czasu symulowanego) startuje następny kolarz; jego czas, wyrażony w całkowitej liczbie sekund, jest liczbą losową o rozkładzie Gaussa ze średnią 290s i odch. standardowym 40 (ale czas nie może być krótszy niż 240s ani dłuższy niż 350s, tj. jeśli wylosuje się np. 233s, to przyjmujemy 240s, a jeśli wylosuje się 362s, to przyjmujemy 350s).

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

<http://szgrabowski.kis.p.lodz.pl/zpo19/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 4 zawodników (ich nazwiska i czasy). Na samym początku ta czołówka będzie mniej liczna (0..3 kolarzy). Dobierz odpowiednie komponenty GUI (JavaFX lub Swing).

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/en/java/javase/11/docs/api/java.base/java/util/PriorityQueue.html>