# Zadanie 2: Sieci Liczników - wariant UDP
# SKJ (2017)

## Wstęp

Dokładny pomiar upływu czasu jest jednym z ważniejszych zadań w wielu dziedzinach techniki. Synchronizacja oddalonych od siebie liczników jest trudna, w szczególności dlatego, że uwzględnione powinny być opóźnienia przesyłu sygnałów, reakcji urządzeń, itp. W ramach zadania zbudujemy protokół synchronizacji liczników w sieci lokalnej przy użyciu komunikacji z wykorzystaniem protokołu UDP.

## Specyfikacja

### Agent

Zarządzaniem wartością liczników zajmują się **agenci**. Każdy z agentów jest osobnym węzłem (z konieczności jeden agent przypada na jeden węzeł sieci), identyfikowanym poprzez **adres IP**. Adres IP jest zależny od węzła, na którym działa agent, a numer portu UDP jest taki sam dla wszystkich agentów. Port ten służy agentowi do komunikacji ze światem.

Każdy z agentów posiada **jeden licznik**, który odmierza upływający czas w milisekundach i początkowo jest ustawiany na wartość podaną jako parametr jego uruchomienia.

Każdy agent co określony parametrem kwant czasu dokonuje synchronizacji wartości swojego licnika z wartościami liczników pozostałych agentów.

### Interakcje agentów

#### Dodawanie agentów

Wszyscy agenci działają według poniższego schematu:

1. Agent jest uruchamiany z dwoma parametrami, które znaczą kolejno:
   (a) początkową watość licznika dla tego agenta,
   (b) kwant czasu (w sekundach), co który agent przeprowadza operację synchronizacji swojego licznika z licznikami pozostałych agentów w sieci.

2. Po uruchomieniu agent cyklicznie, co określony parametrem kwant czasu, wykonuje operację synchronizacji swojego licznika z licznikami pozostałych agentów.

3. Agent może w dowolnym momencie zakończyć pracę. Jeśli to konieczne, powiadamia o tym pozostałych agentów (jeśli wymaga tego przyjęty schemat komunikacji).

### Synchronizacja liczników

Każdy z agentów cyklicznie dokonuje procedury synchronizacji licznika z pozostałymi agentami pracującymi w sieci. Może do tego celu wykorzystać rozgłaszanie:

1. Agent $i$ wysyła komunikat rozgłoszeniowy z żądaniem przesłania bieżącego stanu licznika każdego agenta, który by taki komunikat odebrał.

2. Każdy agent, który taki komunikat odbierze, odsyła do agenta źródłowego komunikat z bieżącą wartością swojego licznika.

3. Agent inicjujący zbiera odpowiedzi (należy oczekiwać nieokreślonej z góry liczby odpowiedzi, zależnej od liczby aktualnie pracujących agentów).

4. Po ich uzyskaniu ustawia nową wartość swojego licznika na średnią z pobranych wartości (wliczając w to swoją, tzn. $T_i$):

$$T_i := \frac{1}{N} \sum_{j=1}^{N} T_j.$$

5. Agent $i$ wyświetla bieżącą wartość swojego licznika $T_i$.

### Kontroler

Dodatkowa aplikacja pełni rolę kontrolera sieci liczników. Za jej pomocą można odczytać lub przestawić licznik i kwant czasu między synchronizacjami dla każdego agenta. Aplikacja jako pierwszy parametr przyjmuje adres IP agenta, na którym dokonywana jest operacja (numer portu jest ustalony i taki sam dla wszystkich agentów). Kolejne parametry zależą od wykonywanej czynności:

- Dla odczytu: 2 parametry:
  1. drugi parametr to `get`,
  2. trzeci parametr to `counter` lub `period` zależnie od odczytywanej wartości,
- Dla zmiany: 3 parametry:
  1. drugi parametr to `set`,
  2. trzeci parametr to `counter` lub `period` zależnie od zmienianej wartości,

3. czwarty parametr to nowa wartość licznika lub kwantu czasu.

Po uruchomieniu, aplikacja kontrolera wysyła odpowiedni komunikat do agenta oraz czeka na jego potwierdzenie. Po otrzymaniu odpowiedniego zlecenia, agent zapamiętuje przekazane mu wartości i potwierdza dokonanie operacji (w przypadku `set`) lub wyłącznie zwraca bieżącą wartość (w przypadku `get`).

# Wymagania i sposób oceny

1. Poprawny i pełny projekt wart jest **4 punkty**. Za zrealizowanie każdej z poniższych funkcjonalności można otrzymać punkty do podanej wartości.

   - Utworzenie sieci agentów i poprawna synchronizacja zegarów. **2 punkty**.
   - Funkcjonalność `get` w kontrolerze. **1 punkt**.
   - Funkcjonalność `set` w kontrolerze. **1 punkt**.

2. Aplikację piszemy w języku Java zgodnie ze standardem Java 8 (JDK 1.8). Do komunikacji przez sieć można wykorzystać jedynie podstawowe klasy do komunikacji z wykorzystaniem protokołu UDP.

3. Projekty powinny zostać zapisane do odpowiednich katalogów w systemie EDUX w nieprzekraczalnym terminie 22.XII.2017 (termin może zostać zmieniony przez prowadzącego grupę).

4. Spakowany plik projektu powinien obejmować:

   - Plik *Dokumentacja(nr.indeksu)Zad2.pdf*, opisujący, co zostało zrealizowane, co się nie udało, gdzie ewentualnie są błędy, których nie udało się poprawić.
   - Pliki źródłowe (dla JDK 1.8) (włącznie z wszelkimi bibliotekami nie należącymi do standardowej instalacji Javy, których autor użył) - aplikacja musi dać się bez problemu skompilować na komputerach w laboratorium w PJA.

   UWAGA: PLIK Z DOKUMENTACJĄ JEST WARUNKIEM KONIECZNYM PRZYJĘCIA PROJEKTU DO OCENY.

5. Prowadzący oceniać będą w pierwszym rzędzie poprawność działania programu i zgodność ze specyfikacją, ale na ocenę wpływać będzie także zgodność wytworzonego oprogramowania z zasadami inżynierii oprogramowania i jakość implementacji.

6. JEŚLI NIE WYSZCZEGÓLNIONO INACZEJ, WSZYSTKIE NIEJASNOŚCI NALEŻY PRZEDYSKUTOWAĆ Z PROWADZĄCYM ZAJĘCIA POD GROŹBĄ NIEZALICZENIA PROGRAMU W PRZYPADKU ICH NIEWŁAŚCIWEJ INTERPRETACJI.