WOJSKOWA AKADEMIA TECHNICZNA
im. Jarosława Dąbrowskiego
WYDZIAŁ CYBERNETYKI

PROJEKT ZALICZENIOWY

Temat: **STATKI NA RZECE**

Autor:

Marcin KOZŁOWSKI

Warszawa 2023

1. Treść zadania

Zadanie nr: **PW-7/2022**
Język implementacji: **Java**
Środowisko implementacyjne: **Eclipse, Intelij IDEA, Netbeans**
Termin wykonania: **ostatnie zajęcia**
Podstawowe wymagania:
a. liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
b. kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie" modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
c. graficzne zobrazowanie działania procesów współbieżnych,
d. odczyt domyślnych danych wejściowych ze sformatowanego, tekstowego pliku danych (xml, properties, inne),
e. możliwość modyfikacji danych wejściowych poprzez GUI.

Sprawozdanie (w formie elektronicznej) powinno zawierać następujące elementy:
1) stronę tytułową,
2) niniejszą treść zadania,
3) syntetyczny opis problemu – w tym wszystkie przyjęte założenia,
4) wykaz współdzielonych zasobów,
5) wykaz wyróżnionych punktów synchronizacji,
6) wykaz obiektów synchronizacji,
7) wykaz procesów sekwencyjnych,
8) listing programu.

Problem do rozwiązania:
Promy na rzece.
Założenia:
Przeprawę obsługuje N promów o określonej pojemności każdy.
Na przystań i z przystani prowadzi jedna droga, po której poruszają się pojazdy z różną prędkością.
Przystań ma ograniczoną pojemność. Na promy jest tylko jeden wspólny wjazd i zjazd, przez który samochody wjeżdżają/zjeżdżają pojedynczo (pierwszeństwo samochodów zjeżdżających).
Promy czekają na zapełnienie przez określony maksymalny czas, po którym:
- rozpoczynają przeprawę, gdy na pokładzie znajduje się co najmniej jeden pojazd,
- lub rozpoczynają kolejny okres oczekiwania.
Jeżeli zapełnią się wcześniej, to rozpoczynają przeprawę przed upływem tego czasu. W przypadku przybycia promu do przystani, gdy jest zajęta przez inny prom, to oczekuje w kolejce.

2. Opis problemu

Synchronizacja między samochodami a portem (przystanią)

Przy wejściu samochodów (wątków) do portu (wspólnego bufora) należało zaimplementować rozwiązanie, które rozwiąże problem ograniczonej pojemności portu, tak by tylko maksymalnie określona ilość samochodów mogła znajdować się w porcie w danym czasie. Ponadto, ponieważ (w dalszej części zostanie to szerzej omówione) samochody wyjeżdżające ze statku mają pierwszeństwo a jednocześnie przechodzą przez port a w związku z tym wliczają się do limitu portu zdecydowano, że maksymalna pojemność dla samochodów wjeżdżających do portu będzie wynosiła N-1 (gdzie N – ilość miejsc w porcie). Dzięki temu nie dojdzie do sytuacji, że port jest całkowicie zablokowany (wzajemne zakleszczenie).

Synchronizacja między samochodami a statkiem

Statek po wejściu do portu rozpoczyna wyładunek samochodów i czeka do momentu ich wyjazdu z portu. Gdy wszystkie samochody wyjeżdzające opuszczą port rozpoczyna się załadunek nowych samochodów. Statek czeka na załadunek przynajmniej jednego samochodu po czym blokuje wejście na siebie i odpływa z portu. Samochody czekają w statku do ponownego wejścia do portu przez statek po czym po odpowiednim sygnale zjeżdżają z niego i wyjeżdżają z portu.

Synchronizacja między statkami a portem

Zgodnie z treścią zadania tylko jeden statek na raz może znajdować się w porcie i wykonywać swoje czynności. Reszta statków oczekuje w kolejce do portu i może wejść do niego tylko gdy jest on niezajęty.

Pozostałe założenia:

Minimalna liczba statków, miejsc na statkach, jak i samochodów wynosi 1 (dla jednego statku). Minimalna liczba miejsc w porcie wynosi 2.

Założono, że ruch w porcie odbywa się cały czas. Również zgodnie z treścią zadania statek w porcie musi czekać do momentu wjazdu na niego przynajmniej jednego auta. Gdy nie będzie aut w porcie statek, nie będzie mógł odpłynąć a tym samym zablokuje pozostałe statki czekające na wejście do portu. Spowoduje to zakleszczenie programu. W związku z tym przyjęto założenie, że jeśli liczba statków jest większa od 1 (2,3,4…) to samochodów musi być minimum:

(Ilość_statków-1) *Ilość_miejsc_na_statkach+1

Np. dla 2 statków i 5 miejsc na nich

Minimalna liczba samochodów: 6

3. Wykaz współdzielonych zasobów

Statki i samochody (wątki) współdzielą port (wspólny ograniczony bufor) w którym wykonują określone funkcje (metody).

Samochody znajdujące się w określonym statku (wspólny ograniczony bufor wyłącznie dla samochodów) również wykonują na nim określone funkcje.

4. Wykaz wyróżnionych punktów synchronizacji

Synchronizacje:

- Wejście samochodu do portu – **synchronizacja samochód-port**
- Czekanie na statek – **synchronizacja samochód-port-statek**
- Wejście na statek – **synchronizacja samochód-statek**
- Zejście samochodu ze statku – **synchronizacja samochód-port-statek**
- Wyjście samochodu z portu – **synchronizacja samochód-port**
- Wejście statku do portu – **synchronizacja statek-port**

5. Wykaz obiektów synchronizacji

Bufory:

- Port
- Statek (dla danych samochodów)

6. Wykaz procesów sekwencyjnych

Procesy:

- Statki (Wątki),
- Samochody (Wątki),

Listing kodu programu został załączony w oddzielnych plikach z rozszerzeniem .java w związku ze znaczną długością kodu.