

Programowanie obiektowe II

Laboratorium nr 14

Stefan Sokołowski

Zadanie 1:

Proszę obejrzyć i przetestować program

iis.ans-elblag.pl/~stefan/Dydaktyka/ProgObiII/Prog14/Dom/SleepSort.java

stosujący wątki dla posortowania podanych liczb naturalnych (czyli całkowitych niemniejszych od zera) i wypowiedzieć się na temat jego zalet i wad.

Jak on działa? Czy tak można sortować?

Zadanie 2:

Przedstawione na wykładzie 14¹ programy ilustrujące skrzyżowanie² oraz system bankowy³ stosowały *busy waiting* (aktywne oczekiwanie). Proszę zmienić je tak, żeby zamiast tego stosowały mechanizm *wait-notify*.

Zadanie 3:

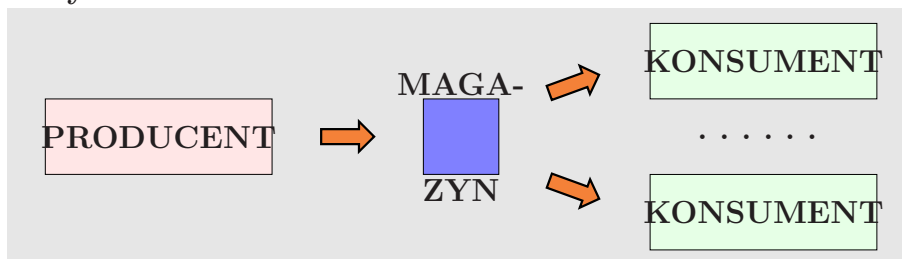
Program

iis.ans-elblag.pl/~stefan/Dydaktyka/ProgObiII/Prog14/Dom/ProdKons.java

przedstawia współpracę producenta i konsumenta z magazynem nieograniczonej wielkości i z zastosowaniem oczekiwania typu *busy waiting*. Producent produkuje 5000 napisów, umieszcza je w magazynie, a następnie ogłasza *fajrant*⁴. Jednocześnie konsument zabiera napisy z magazynu i drukuje je; może zakończyć działanie dopiero wtedy, gdy został ogłoszony fajrant i magazyn jest pusty. Program drukuje czas działania całego programu od wystartowania obu watków aż do ich zakończenia.

Proszę poeksperymentować z następującymi modyfikacjami programu:

- (a) **Usunięcie współbieżności.** Niech *najpierw* producent napełnia magazyn wyprodukowanymi napisami; a dopiero *potem* niech konsument je drukuje. Porównać czas działania z czasem działania programu oryginalnego.
- (b) **Zwiększenie liczby konsumentów.**



¹Zob. iis.ans-elblag.pl/~stefan/Dydaktyka/ProgObiII/Slajdy/14a.pdf

²Str. 2.

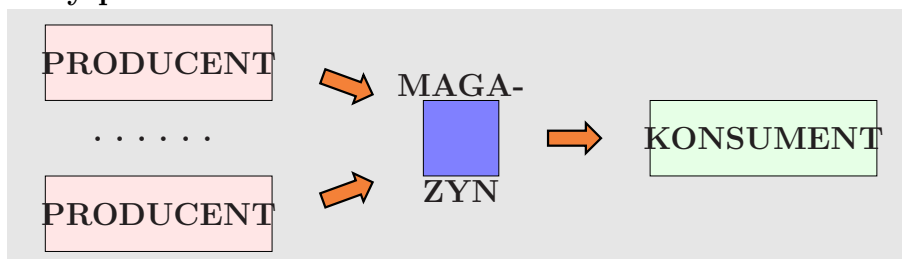
³Str. 4–5.

⁴Koniec dnia roboczego.

Konsumenci będą wtedy również między sobą konkutować o dostęp do wspólnego magazynu.

Program powinien dysponować stałą całkowitą, łatwą do szybkiej zmiany, określającą liczbę konsumentów. Proszę porównać czasy działania dla różnych liczb konsumentów; w szczególności ustalić ich liczbę optymalną, dającą minimalny czas działania.

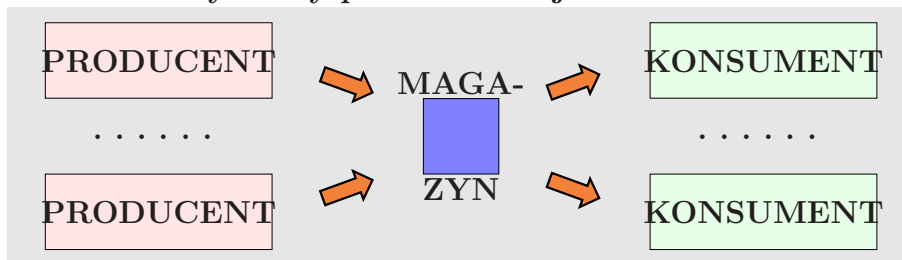
(c) **Zwiększenie liczby producentów.**



Proszę zauważyć, że teraz fajrant może zostać ogłoszony dopiero wtedy, gdy **wszyscy** producenci zakończyli produkcję.

Jak w punkcie (b), zaopatrzyć program w stałą całkowitą, łatwą do szybkiej zmiany, określającą liczbę producentów. Proszę porównać czasy działania dla różnych liczb producentów.

(d) **Zwiększenie zarówno liczby producentów jak konsumentów.**



(e) **Ograniczenie pojemności magazynu.**

To wymaga wprowadzenia *busy waiting* również po stronie producenta: zanim włoży on nowy produkt do magazynu, musi zaczekać na wolne miejsce w magazynie.

Poeksperymentować z wpływem wielkości magazynu na czas działania programu.

Zadanie 4:

W programie

iis.ans-elblag.pl/~stefan/Dydaktyka/ProgObiII/Prog13/Dom/ProdKons.java

(tym samym, co w zad. 3) proszę zastąpić mechanizm *busy waiting* przez *wait-notify*.

Następnie proszę dokonać podobnych jego modyfikacji co w zad. 3 i wyciągnąć wnioski odnośnie czasów działania.

Zadanie 5:

Proszę zamienić *busy waiting* na *wait-notify* (jak w zad. 4) w programach

(a) iis.ans-elblag.pl/~stefan/Dydaktyka/ProgObiII/Prog13/A1_PojazdyBusy.java

(b) iis.ans-elblag.pl/~stefan/Dydaktyka/ProgObiII/Prog13/A2_Bank.java
