**Sprawozdanie z testów P/K na AO**

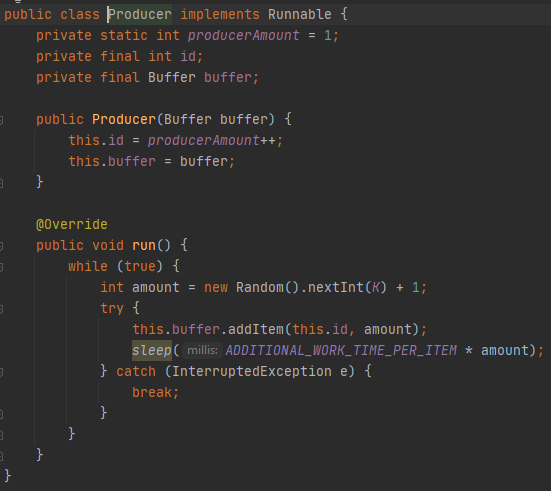
Szymon Zieliński

AGH WIEiT

Teoria Współbieżności

**1. Różnice w implementacji wątków producentów/konsumentów**

I. Rozwiązanie na monitorze:  
Producent w kółko wykonuje operację dodania produktów(metoda w monitorze) oraz swoją dodatkową pracę

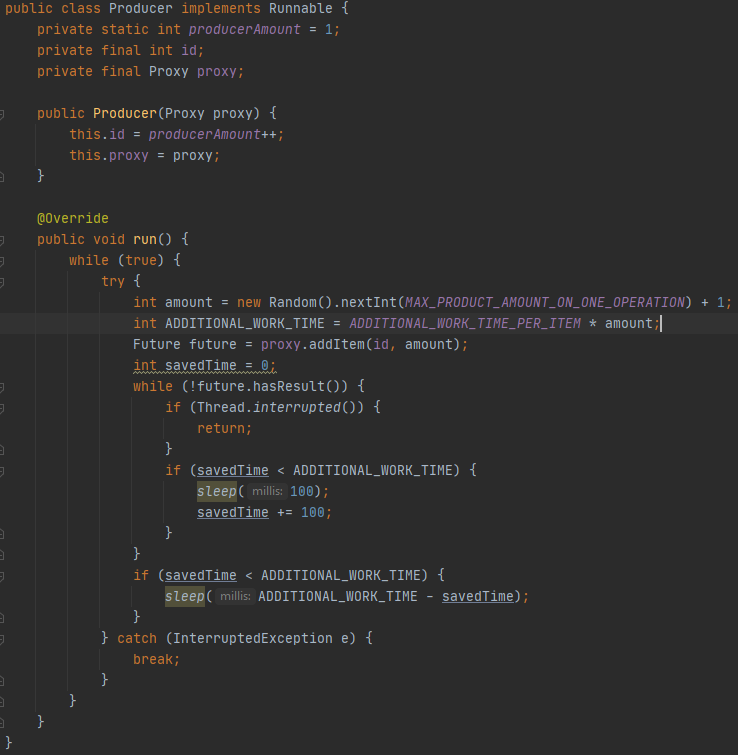


II. Rozwiązanie przy użyciu wzorca AO:

Producent w kółko wykonuje:

- zleca operację dodania produktów

- wykonuje swoja dodatkową pracę sprawdzając co jakiś czas czy operacja dodania produktów została zakończona, jeśli została, a dodatkowa praca nie została jeszcze ukończona to ją kończy



**2. Implementacja dodatkowej pracy:**

Czas dodatkowej pracy zrobiłem wprost proporcjonalny do ilości wyprodukowanych/skonsumowanych produktów, aby wyniki porównywane w dalszej części nie były zależne od wylosowanej ilości produktów. (Dlatego, żeby czas dodatkowej pracy był taki sam dla takiej samej ilości dodanych/usuniętych produktów w obydwu programach)

I. Rozwiązanie na monitorze: dodatkowa praca została zrealizowana jako pasywne czekanie przez czas odpowiadający wykonaniu pracy.

II. Rozwiązanie przy użyciu wzorca AO: dodatkowa praca jest wykonywana po kawałku podczas oczekiwania na dodanie produktów.

**3. Kolejka do zgłaszanych zadań – implementacja**

Kolejka została zaimplementowana jako zwykła linked lista, dodawanie i usuwanie z kolejki jest zrealizowane za pomocą metod zsynchronizowanych za pomocą locka.

Przypadek gdy chcemy pobrać dane z pustej kolejki jest obsługiwany przez zawieszenie się wątku, wątek zostaje wznowiony w momencie dodania elementu do kolejki.

**4. Parametryzacja zadania**

*TAKING\_ONE\_ITEM\_TIME* = 40; // milliseconds  
*ADDING\_ONE\_ITEM\_TIME* = 40; // milliseconds  
*PRODUCER\_AMOUNT* = 4;  
*CONSUMER\_AMOUNT* = 4;  
*MAX\_PRODUCT\_AMOUNT\_ON\_ONE\_OPERATION* = 50;  
*ADDITIONAL\_WORK\_TIME\_PER\_ITEM* = 300; // milliseconds  
*PROGRAM\_RUNNING\_TIME* = 10\_000; // milliseconds

TAKING/ADDING\_ONE\_ITEM\_TIME – czas ile trwa dodanie lub usunięcie jednego produktu Testowany zakres: 40ms

PRODUCER/CONSUMER\_AMOUNT – ilość wątków producentów/konsumentów. Testowany zakres: 1 - 70

MAX\_PRODUCT\_AMOUNT\_ON\_ONE\_OPERATION – maksymalna ilość produkowanych/konsumowanych produktów w pojedynczej operacji (wielkość bufora została ustalona jako podwojona ta wartość!). Testowany zakres: 50

ADDITIONAL\_WORK\_TIME\_PER\_ITEM – czas dodatkowej pracy (per produkt). Testowany zakres: 10ms – 2.5s

PROGRAM\_RUNNING\_TIME – czas działania pojedynczego testu na danych parametrach. Testowany zakres: 5min

**5. Środowisko testowe**

Wyniki były mierzone poprzez ilość produkcji/konsumpcji w danym czasie.

Testy nie były powtarzane, a wartości nie były uśredniane ze względu na względnie długi czas działania programu, co powinno zapewnić wiarygodne dane.

Zakres zmian parametrów został podany w pkt 4.

**6. Wyniki porównawcze**

Wykres 1. Ilość produkcji/konsumpcji a ilość wątków producentów/konsumentów

Z informacji na Wykresie 1 wynika, że program asynchroniczny działa lepiej dla niewielkiej ilości wątków, a zbyt duża ilość wątków jest nieopłacalna w żadnym z rozwiązań.

Wykres 2. Ilość produkcji/konsumpcji a czas dodatkowej pracy. Czas dodawania/usuwania produktu był niezmienny i wynosił 40ms.

Z informacji na Wykresie 2 wynika, że program asynchroniczny działa troszkę lepiej. Najlepsze wyniki możemy zaobserwować dla czasu dodatkowej pracy kilka razy większego od czasu produkcji/konsumpcji.

- Program asynchroniczny i synchroniczny mają podobne wyniki dla długiego i krótkiego czasu dodatkowej pracy, myślę, że wynika to w przypadku krótkiego czasu pracy z braku czasu który można by zaoszczędzić, natomiast w przypadku długiego czasu pracy z bardzo niewielkiego zaoszczędzonego czasu względem czasu dodatkowej pracy.

Wykres 3. Ilość produkcji/konsumpcji a czas konsumpcji/produkcji. Czas dodatkowej pracy był niezmienny i wynosił 100ms.

Z informacji na Wykresie 3 wynika, że program asynchroniczny działa troszkę lepiej tylko dla bardzo małego czasu produkcji/konsumpcji. Nie mam pojęcia dlaczego aż tak bardzo.

Wykres 4. Ilość produkcji/konsumpcji a czas dodatkowej pracy. Czas produkcji/konsumpcji był niezmienny i wynosił 100ms. 3/3 oznacza 3 konsumentów 3 producentów, 5/5 analogicznie

**6. Wnioski**

- Program działa najlepiej dla niewielkiej ilości wątków, domyślam się, że działa najlepiej dla ilości wątków zbliżonej do ilości wątków logicznych procesora.

- Przy bardzo dużej ilości wątków programy działają gorzej, co jest oczywiste, ale z jakiegoś powodu wyniki dla programu asynchronicznego dla dużej ilości wątków są lepsze, nie mam pomysłu dlaczego.