Programowanie Java
Anna Gogolińska
Zestaw 8

Zad 1. Napisać klasę *PrintThread*, która będzie dziedziczyła z obiektu *Thread*. Obiekt klasy *PrintThread* ma posiadać własny identyfikator (int) i w metodzie *run()* wypisywać liczby od 1 do 100 wraz z tym identyfikatorem. Po wypisaniu każdej liczby wątek ma usypiać się na czas losowy od 0 do 100 milisekund. Utworzyć kilka obiektów i uruchomić wątki. Za pomocą *join()* zaczekać na zakończenie wątków w metodzie *main(-)*. Podobne zadanie wykonaj z wykorzystaniem interfejsu *Runnable* oraz za pomocą klasy anonimowej.

Zad 2. Zmodyfikuje program z poprzedniego zadania, usuń instrukcję *join()* i ustaw jednemu wątkowi maksymalny priorytet, innemu minimalny (użyj stałych z klasy *Thread*). Sprawdź działanie programu.

Zad 3. Napisz klasę *Counter*, która będzie zawierała jedno pole całkowite oraz getter, setter, metodę zwiększająca to pole o 1 i konstruktor. Zainicjuj wartość pola na dowolną wartość. Napisz następnie klasę *InceaseThread*, która będzie dziedziczyć po *Thread*. Ma on mieć pole typu Counter, którego wartość ma być ustawiana w konstruktorze. Wątek w metodzie *run()* ma zwiększać wartość z Counter 20 razy, za każdym zwiększeniem ma wypisać aktualną wartość z *Counter* wraz ze swoją nazwą. W *main()* stwórz jeden obiekt *Counter* oraz pięć wątków *InceaseThread*, którym go przekażesz. Uruchom wątki i zobacz jak działa program. Dodaj blok synchronizacji, aby poprawić działanie programu.

Zad 4. Zaimplementować problem producenta i konsumenta z buforem jednoelementowym. Dla zaobserwowania efektu, należy dodać dodatkowe instrukcje opóźniające producenta przy każdej produkcji elementu o 4 sekundy (czas produkcji), natomiast w przypadku wątku konsumującego opóźniać konsumpcję o 3 sekundy. Wątki zaimplementować wykorzystując interfejs *Runnable*.

Problem producenta i konsumenta jest jednym z klasycznych problemów programowania wielowątkowego. Chodzi o to, aby tak zsynchronizować procesy, aby proces producenta produkował tylko wtedy, jeśli ma gdzie swój produkt odłożyć, a konsument konsumował tylko wtedy, jeśli ma co skonsumować.

W prostym przypadku algorytmicznym mamy jedno miejsce na produkt (bufor jednoelementowy - niech to będzie pole typu int), producent losuje liczby i wstawia je w to pole, a klient pobiera liczby z pola i je wypisuje.

Zadania samodzielne

Zad 5. Zmodyfikuj program z zadania 4. Sparametryzuj wielkość bufora. Elementy produkowane powinny być wkładane do kolejki, z której czyta Konsument.

Zad 6. Zaimplementować dwa wątki, które będą losowały liczby od 1 do 100. Jeżeli któryś z nich wylosuje liczbę większą niż 70 ma poinformować wątek drugi o swojej wygranej, wydrukować o niej informacje i zakończyć działanie. Wątek drugi ma wydrukować informację o swojej przegranej i również zakończyć działanie.

Zad 7. Napisz program, w którym stworzone zostaną dwie tablice liczb całkowitych o takiej samej wielkości – wielkość i elementy określane w dowolny sposób (wielkość maksymalnie 1024). Program ma wyznaczyć trzecią tablicę, która będzie sumą dwóch poprzednich. Operacja ta ma być wykonana przy pomocy wątków – każdy wątek ma wyznaczać jeden element tablicy wynikowej.