

Zadanie 1

Napisz program, który za pomocą klasy **ProcessBuilder** uruchomi jako proces zewnętrzny przeglądarkę Opera. Następnie zmodyfikuj go tak aby w przeglądarce wyświetlił się plik pdf z zadaniami.

Zadanie 2

1. Zaprogramuj klasę Javy będącą podklasą klasy Thread. Metoda run klasy powinna wyświetlić komunikat "Hello world!". Napisz metodę main, która tworzy wątek i go uruchamia.
2. Zaprogramuj klasę Javy implementującą interface Runnable. Metoda run klasy powinna wyświetlać komunikat "hello world". Napisz metodę main, która tworzy wątek oraz przekazuje obiekt Runnable jako parametr i uruchamia go.
3. Zmodyfikuj programy z poprzednich podpunktów tak, aby tworzonych było kilka wątków, a każdy wątek wyświetlał również swój numer.

Zadanie 3

1. Napisz w języku Java współbieżny program składający się z trzech wątków. Pierwszy wątek powinien wyświetlić (jednorazowo) liczby od 1 do 33, drugi wątek powinien wyświetlić liczby od 50 do 88, a trzeci liczby od 100 do 130. Uruchom trzy wątki.
2. Zmodyfikuj program z podpunktu 1 tak, aby drugi wątek rozpoczął wyświetlanie dopiero wtedy, gdy pierwszy zakończy swoje działanie, a trzeci dopiero wtedy, gdy drugi skończy działanie. (**thread.join()**)
3. Zmodyfikuj program z podpunktu 2 tak, aby wątki wyświetlały swoje komunikaty cyklicznie tzn. pierwszy, drugi, trzeci, pierwszy, drugi, trzeci, ... (**Thread.yield()**)
4. Zmodyfikuj podpunkt 3 tak, aby każdy wątek zasypiał na jedną sekundę po wyświetleniu każdej liczby.

Zadanie 4

1. Wykorzystując klasy prywatne napisz współbieżny program składający się z kilku różnych wątków. Niech każdy wątek wyświetla swoją nazwę (nazwa składa się ze słowa Thread i numeru wątku) z informacją, że wystartował, następnie w pętli liczby od 1 do 5, na koniec ponownie swoją nazwę i informację że zakończył.
2. W rozwiązaniu użyj klasy **ThreadGroup**. Po odczekaniu 7 sekund, główny wątek przerywa wszystkie wątki w grupie.

Zadanie 5

Napisz program, który tworzy grupę wątków o nazwie i uruchamia w niej trzy wątki. Każdy z wątków powinien wyświetlić informację o swoim uruchomieniu, poczekać przez 1 sekundę, a następnie wyświetlić informację o swoim zakończeniu. Po uruchomieniu wątków, program wyświetla liczbę aktywnych wątków w grupie oraz listę tych wątków. Po odczekaniu 3 sekund, program przerywa działanie wszystkich wątków w grupie.

Zadanie 6

Napisać programy w Javie ilustrujące mechanizm przeplotu dla dwóch wątków:

- każdy proces wyświetla swój numer,
- każdy proces wyświetla swój numer w pętli nieskończonej,
- każdy proces wyświetla swój numer w pętli skończonej.

Każdy program uruchom kilkakrotnie.

Zadanie 7

Napisz program w Javie składający się z trzech wątków. Pierwszy wątek działa w pętli nieskończonej, w każdym obrocie pętli generuje 100 liczb całkowitych i oblicza średnią arytmetyczną z nich. Drugi wątek działa w pętli nieskończonej, w każdym obrocie pętli generuje 30 liczb całkowitych i oblicza średnią geometryczną. Trzeci wątek działa w pętli nieskończonej, w każdym obrocie pętli oblicza logarytm naturalny kwadratu liczby całkowitej wylosowanej z przedziału $(-20, 20)$. Wszystkie wątki wyświetlają wynik na ekran w następującej postaci: [numer wątku] liczba.

Zadanie 8

Zdefiniuj klasę Licznik, zaimplementuj w niej metodę inc i dec odpowiednio zwiększającą zmniejszającą licznik o 1 oraz metodę get zwracającą wartość licznika. Napisz program, który uruchomi dwa wątki, z których jeden zwiększy licznik o 10000, a drugi zmniejszy go o 5000. W rozwiązaniu zaproponuj definicję jednej klasy dziedziczącej po Thread.

Zadanie 9

Napisz program, w którym wątek pierwszy losuje dowolną liczbę całkowitą, następnie wyświetla ją na ekranie i przesyła ją do wątku drugiego, który powiększa tę liczbę o dwa i wysyła ją do wątku trzeciego. Wątek trzeci wyświetla ją na ekranie.

Zadanie 10

Napisać w języku Java program współbieżny wyznaczający największy wspólny podzielnik (gcd) liczb całkowitych u_1, u_2, \dots, u_n przy wykorzystaniu własności: $\text{gcd}(u_1, u_2, \dots, u_n) = \text{gcd}(u_1, \text{gcd}(u_2, \dots, u_n))$ oraz oczywistych wniosków z niej płynących. Wymaga się, aby w programie występowały dwa typy wątków współbieżnych: jedno zadanie Master oraz dynamicznie tworzony wątek typu Slave przy czym jednocześnie mogą działać maksymalnie dwa wątki typu Slave. Pobieranie wartości n i u_1, u_2, \dots, u_n , oraz startowanie zadania Master ma być realizowane w metodzie głównej. Każde zadanie Slave ma być odpowiedzialne za wyznaczenie największego wspólnego podzielnika dwóch zadanych liczb całkowitych i przekazania wyniku do zadania Master. Zadanie Master ma realizować obliczenie $\text{gcd}(u_1, u_2, \dots, u_n)$ powołując do życia zadania typu Slave, zlecać im pracę oraz odbierać wyniki. Po obliczeniu wyniku zadanie Master ma wyświetlać otrzymaną wartość.

Zadanie 11

Napisać program składający się z następujących zadań:

- zadanie generujące liczby nieparzyste: zadanie działa w pętli nieskończonej i tworzy kolejne liczby nieparzyste rozpoczynając od 1, liczby przekazuje do zadania odbierającego oraz wyświetla komunikaty postaci "przekazałem liczbę 3"
- zadanie generujące liczby parzyste: zadanie działa w pętli nieskończonej i tworzy kolejne liczby parzyste rozpoczynając od 2, liczby przekazuje do zadania odbierającego oraz wyświetla komunikaty postaci "przekazałem liczbę 4"
- zadanie odbierające: zadanie działa w pętli nieskończonej, odbiera liczby od zadania generującego liczby nieparzyste lub parzyste, z każdych dwóch odebranych liczb wybiera większą i wyświetla komunikat postaci "większą liczbą jest 4"

Zadanie 12

Napisz program, w którym obliczysz współbieżnie sumę wyrażenia $1+1/2+1/3+1/4+\dots+1/n$. Rozmiar sumy (czyli n) powinien być wczytywany z klawiatury. Liczba zadań, która powinna współbieżnie liczyć sumę powinna być wczytywana z klawiatury. Przykładowe wyniki: dla $n=50$, suma=4.499205338329423, dla $n=100$, suma=5.187377517639621, dla $n=200$, suma=5.878030948121446. W rozwiązaniu użyj klasy **ExecutorService**.

Zadanie 13

Zgodnie z poniższym schematem zaprogramować następujące procesy:

- procesy P1, P2, P3 działają w pętli nieskończonej, tworzą dane tzn. losują liczby całkowite, przekazują procesowi P4 (po jednej liczbie), przekazywana liczba jest wyświetlana, po przekazaniu liczby należy odczekać losową liczbę sekund (od 1 do 3 sekund),
- proces P4 - odbiera po jednej liczbie od procesów P1, P2, P3, wybiera z nich największą oraz przekazuje ją procesowi P5,
- proces P5 - sortujący - odbiera liczby od procesu P4, po odebraniu n liczb sortuje je rosnąco oraz wyświetla na ekranie (wszystkie liczby w jednej linii, oddzielone przecinkami, na końcu przejście do nowej linii).

Wykorzystać potoki do wymiany danych między procesami.