Programowanie równoległe. Przetwarzanie równoległe i rozproszone.

Laboratorium 7

Cel:

- nabycie umiejętności pisania programów w języku Java z wykorzystaniem puli wątków

Zajęcia:

1. Utworzenie katalogu roboczego (np. *lab_Java_threadpool*).
2. Napisanie sekwencyjnego programu obliczania całki korzystając z dostarczonej klasy *Calka_callable* (jak zwykle przykładowo można przyjąć całkowanie ze znanym dokładnym wynikiem, np. całka z *sin(x)*, w przedziale *(0, Math.PI)*:
   a) *Calka_callable* oblicza całkę w zadanym przedziale, z granicami przedziału i dokładnością zadaną w konstruktorze (liczba trapezów jest obliczana wewnętrznie przez *Calka_callable* - w tym laboratorium nie należy zajmować się liczbą trapezów!)
   b) jako obiekt typu *Callable Calka_callable* ma funkcję *call*, która będzie wykonana przez wątek, np. związany z obiektem klasy Executors **(ocena)**
3. Pobranie paczki java_executor_test.tgz , rozpakowanie, uruchomienie, sprawdzenie poprawności działania.
4. Na podstawie przykładu z java_executor_test.tgz, wykorzystującego interfejs *ExecutorService* oraz klasę *Executors*, zmodyfikowanie programu obliczania całki tak, aby używać puli o stałej liczbie wątków
   a) pojedyncze zadanie dla obiektu typu *Executor* ma stanowić obliczenie całki w podprzedziale, w tym celu zadany przedział (np. (0, PI)) należy podzielić na tyle podprzedziałów, ile jest zadań
   b) liczba wątków i liczba zadań są niezależnymi parametrami - zazwyczaj liczba zadań jest większa od liczby wątków (aby umożliwić równoważenie obciążenia, liczba wątków powinna być związana z liczbą rdzeni procesora)
   c) zadania należy tworzyć i przekazywać do wykonania w jednej pętli, wyniki (korzystając z interfejsu *Future*) powinny być odbierane w kolejnej pętli - tylko tak możliwe jest działanie równoległe
5. Uruchomienie i przetestowanie poprawności działania. **(ocena)**

---------- 3.0 ----------

6. Utworzenie katalogu roboczego (np. *lab_Java_fork_join*).
7. W katalogu roboczym, na podstawie wykładu i dostarczonego szkieletu klasy do wykorzystania w sortowaniu przez scalanie (*scal_tab.java*), utworzenie programu sortowania, wykorzystującego pule wątków Javy, ale tym razem korzystając z klasy *ForkJoinPool*
8. Uruchomienie i przetestowanie działania. **(ocena)**

---------- 4.0 ----------

Dalsze kroki:

1. Zmodyfikowanie programów obliczania histogramu (laboratorium 6), tak aby korzystały z puli wątków (szczególnie dla dekompozycji tablicy znaków, szczególnie dla dekompozycji 2D).
2. W przypadku niezrealizowania w całości tematów laboratorium 5 - wykonanie punktów dodatkowych (obliczanie całki w sposób wielowątkowy z wykorzystaniem środowiska *Pthreads*, wykorzystując **wzorzec dekompozycji w dziedzinie problemu** - podobnie jak to jest przeprowadzane w obecnym laboratorium, ale z inną implementacją)

---------- 5.0 ----------

Warunki zaliczenia:

1. Obecność na zajęciach i wykonanie co najmniej kroków 1-5
2. Oddanie standardowego sprawozdania – zgodnie z regulaminem laboratoriów