Przemysław Szymoniak

Przetwarzanie współbieżne. Programowanie równoległe i rozproszone.

Sprawozdanie z laboratorium 8.
Celem laboratorium było nabycie umiejętności pisania programów w Javie z wykorzystaniem puli wątków.

W ramach zajęć zrealizowałem następujące kroki:

- Pobrałem i rozpakowałem wszystkie wymagane pliki.

- Kod sekwencyjny

```
Calka_callable calka = new Calka_callable(xp, xk, interv);
System.out.println("PIERWSZA CALKA"+calka.compute());
```

- Kod wykorzystujący stałą pulę wątków:

```
…..
Counter counter = new Counter();
ExecutorService executor = Executors.newFixedThreadPool(NTHREADS);

List<Future<Double>> result = new ArrayList<>();
Callable<Double> calka;
Future<Double> part;
double step = (xk-xp)/100;

for (int i = 0; i < 100; i++) {
        calka = new Calka_callable(i*step, (i+1)*step, 0.001);
          part = executor.submit(calka);
          result.add(part);
        }

double out = 0;
for(Future<Double>p_out:result){
      try {
         out+=p_out.get();
      } catch (InterruptedException e) {
         e.printStackTrace();
      } catch (ExecutionException e) {
         e.printStackTrace();
      }
}

System.out.println("Wynik po execut:"+out);
executor.shutdown();
…..
```

- Próba napisania programu obliczającego całkę przy użyciu puli wątków z powodu wielu błędów przy jego działaniu nie została załączona.

Wnioski:

- Aby wykorzystać wątki w dużej skali Java udostępnia szereg ułatwiających pracę udogodnień.
- Obiekty klasy ThreadPoolExecutor realizują zarządzanie dostarczonymi zadaniami są to m.in. uruchamianie, zatrzymywanie, sprawdzanie stanu.
- Pula wątków jest pewnego rodzaju kolejką w której przechowujemy wątki do wykonania.
- Wykonywanie zadań w ramach Executor'a zostało dostosowane do masowości wykorzystywania wątków dlatego też zostało lepiej zoptymalizowane pod kontem narzutu w porównaniu do ręcznego tworzenia pojedynczych wątków.
- Obiekty Futures służą do sprawdzenia zakończenia działania obiektów Runnable oraz do przekazywania wyników z obiektów Callable.
- Interfejs Callable pozwala na definiowanie zadań zwracających wynik
- Najprostrze zadania interfejsu Runnable mogą nie zwracać wyników.