Teoria współbieżności

Laboratorium 8 - asynchroniczne wykonanie zadań w puli wątków przy użyciu wzorców Executor i Future

Interfejs `java.util.concurrent.Executor` i `java.util.concurrent.ExecutorService`

1. `Executor` służy do asynchronicznego wykonania zadań typu `Runnable`. Zamiast tworzyć osobne wątki:

   ```
   new Thread(new(RunnableTask())).start()
   ```

   można użyć metody execute():

   ```
   executor.execute(new RunnableTask1());
   executor.execute(new RunnableTask2());
   ...
   ```

2. Metoda `submit()` w interfejsie `ExecutorService` działa podobnie do `execute()`, ale przyjmuje zadania implementujące interfejs `Callable`, które mogą zwrócić wartość (metoda `run()` jest typu `void`). Zadanie może implementować interfejs `Future`.

   Przykład.

3. Implementacje:
   - newSingleThreadExecutor
   - newFixedThreadPool
   - newCachedThreadPool
   - newWorkStealingPool

Zadania

1. Proszę zaimplementować przy użyciu `Executor` i `Future` program wykonujący obliczanie zbioru Mandelbrota w puli wątków. Jako podstawę implementacji proszę wykorzystać kod w Javie.
2. Proszę przetestować szybkość działania programu w zależności od implementacji Executora i jego parametrów (np. liczba wątków w puli). Czas obliczeń można zwiększać manipulując parametrami problemu, np. liczbą iteracji (MAX_ITER).

Bibliografia

1. Executors Java tutorial
2. Dokumentacja pakietu Executors
3. Future
