

Java.util.concurrent

Example1

Pokazuje w jaki sposób utworzyć kilka wątków. Można zaobserwować że kolejność wykonywania jest niedeterministyczna.

Example2

Przykład demonstruje użycie bariery.

Example3

Pokazuje w jaki sposób kontrolować kolejność uruchamiania się wątków, zademonstrowanie oczekiwania na śmierć wątków.

Example4

Przykład pokazujący jak zabić wątki w czasie ich trwania z wykorzystaniem przerwania.

Zadanie:

1. Wykonaj sprawozdanie, opisz wymienione poniżej funkcje/klasy/interfejsy paczki `java.util.concurrent` odpowiadające za zrównoleglenie, użyte w przykładach.
 - interfejs `Runnable`
 - interfejs `Callable<T>`
 - klasa `Executor` (w tym metoda `newFixedThreadPool()`)
 - klasa `ExecutorService` (szczególnie metody `shutdown()`, `shutdownNow()`, `awaitTermination()`, `isTerminated()`)
 - klasa `FutureTask<T>` i jej metoda `T`
 - metody `Thread.sleep()`, `Thread.yield()`, `<<Thread Object>>.join()`
 - funkcje `System.currentTimeMillis()`
 - czym różni się `Catch(Exception e)` od `Catch(InterruptedException e)`

Wyniki oraz program prześlij do swojego repozytorium. Umieść je w folderze o tej samej nazwie co ten PDF.