Râșteiu Roxana-Maria

Grupa 30136

Laboratorul 6

Fire de execuţie în Java SE – Pachetul java.util.concurent - Partea 2 –

- **Clasa CyclicBarrier:**
  - Implementează un mecanism de sincronizare care asigură faptul că un grup de fire de execuție se așteaptă reciproc într-un punct comun(metoda **await()**);
  - Barieră ciclică→ reutilizabilă;
  - Se poate specifica un timp limită de aşteptare la barieră, iar dacă în acest timp restul firelor nu au ajuns la barieră, se consideră că bariera a fost spartă. Toate firele care așteaptă "la barieră" vor primi de la metoda await() o excepție de tipul **BrokenBarrierException**;

- **Clasa CountDownLatch:**
  - Are rolul de a coordona execuția unui grup de fire;
  - Are un constructor cu un argument întreg care indică valoarea iniţială a contorului;
  - Nu ese reutilizabilă;
  - Fierare fir poate decrementa contorul clasei invocând metoda **countDown()**.Această metodă nu blochează firele de execuţie care o apelează, ci doar decrementează contorul;
  - Această clasă este utilă când o problemă se poate descompune în mai multe părţi, fiecare fir primind o subproblemă.

- **Clasa Exchanger:**
  - Implementează două canale de comunicaţie între două fire care cooperează;
  - Această clasă se utilizează în probleme în care un fir de execuţie umple un buffer, iar un altul consumă informaţii (de acelaşi tip) dintr-un alt buffer. Când ambele fire au terminat de umplut, respectiv de golit buffer-ele, acestea sunt interschimbate.