Thread Executors

dasturlash.uz

- Thread Pool
- ***Executors***
- ***Executor***
- ***ExecutorService***
- ***ScheduledThreadPoolExecutor***
- ThreadPoolExecutor

dasturlash.uz

Thread Pool

- Thread pool – Thread hovuzi, Thread basseyni

Thread Pool Description

- **Java Thread pool** represents a group of worker threads that are waiting for the job and reused many times.
- In the case of a thread pool, a group of fixed-size threads is created.
- A thread from the thread pool is pulled out and assigned a job by the service provider.
- After completion of the job, the thread is contained in the thread pool again.

- Java Thread Pool bu ishlashga tayyor bo’lgan va ko‘p marta qayta ishlatiladigan Thread lar guruhini ifodalaydi.
- Thread Pool holatida bu aniq bir o’lchamga ega bo’lsan Thread lar guruxidir.
- Thread Pool dan Thread chiqariladi va xizmat ko'rsatuvchi provayder tomonidan ish tayinlanadi.
- Ish tugagandan so'ng, Thread yana Thread Pool ga joylashtiriladi.

dasturlash.uz

Thread Pool Description

- Creating a new thread for every job may create performance and memory problems. To overcome this we should go for thread Pool.
- Thread Pool is poll of already created threads ready to do our job.
- Java 1.5 version introduces thread pool framework to implement thread pools.
- Thread pool framework also known as executor framework.
- Har bir ish uchun yangi Thread yaratish samaralik jihatidan va hotira tomonidan muommo keltirib chiqarishi mumkin. Buni hal qilish uchun biz Thread Pool dan foydalanamiz.
- Thread Pool bu oldindan yaratilgan va ishni bajarishga tayyor bo’lsan Threadlar dir.
- Java 1.5 versiya birinchi marta Thread Pool Franework ni tanishtirgan.
- Thread Pool framework ki ning ikkinchi nomi executor framework dir.

dasturlash.uz

Advantage of Java Thread Pool

- **Better performance** It saves time because there is no need to create a new thread.
-

Executors, *Executor* and *ExecutorService*

dasturlash.uz

Executors

- The *Executors* helper class contains several methods for the creation of preconfigured thread pool instances. Those classes are a good place to start. We can use them if we don't need to apply any custom fine-tuning.
- Executer lar bu oldindan configuratsiya qilingan ThreadPool instances larni yaratish uchun yordamchi class lardir. Bu class lar boshlashga juda yaxshi. Hech qanday maxsus configlar/sozlashni qo'llash kerak bo'lmasa, biz ulardan foydalanishimiz mumkin.

Executor

- The *Executor* interface has a single *execute* method to submit *Runnable* instances for execution.
- Executor interface da bitta method bo’lib u Runnable interface ni ishlatish uchun kerak bo’ladi.

ExecutorService

- The *ExecutorService* interface contains a large number of methods to **control the progress of the tasks and manage the termination of the service.** Using this interface, we can submit the tasks for execution and also control their execution using the returned *Future* instance.
- ExecutorService class da fazifalarni bajarilishini nazorat qilish va ularni tugashini boshqarish uchun juda ko’p methoflarni tagdim qilgan. Bu interface dan foydalanib task larni ishga tushurishimiz mumkin va return qilingan Future orqali ularni kontrol qilishimiz mumkin.

ExecutorService Methods

- **newFixedThreadPool(int s):**
  - The method creates a thread pool of the fixed size s.
  - Bu method s ta thread hajmidagi thread pool yaratadi.
- **newCachedThreadPool():**
  - The method creates a new thread pool that creates the new threads when needed but will still use the previously created thread whenever they are available to use.
  - Bu method kerak bo’lganda yangi thread larni yaratadigan Thrad Pool yaratadi shu bilan birga oldin yaratilgan va bosh bo’lgan thread larni ham ishlatadi.
  - newCachedThreadPool method creates an executor having an expandable thread pool.
  - newCachedThreadPool metodi kengayadigan/kattalashadigan ThreadPool yaratadi.
  - Qisqasi ish ko’p bo’lsa va Thread yetmasa Yangi Thread yaratadi.
- **newSingleThreadExecutor():**
  - The method creates a new thread.
  - Bitta Thread dan tashkil topgan Yangi ThreadExecutor yaratadi. Unda faqat bitta Thread bo’ladi holos

ExecutorService Example 1

- Thread job

```
public class PrintJob implements Runnable {
   @Override
    public void run() {
        System.out.println(" ... Job started by Thread " + Thread.currentThread().getName());

        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        System.out.println(" ...Job Completed by Thread " + Thread.currentThread().getName());

    }
}
```

ExecutorService Example 2

- Main

```
public static void main(String[] args) {
        PrintJob[] jobs = {new PrintJob("Ali"),
                new PrintJob("Vali"),
                new PrintJob("Bob"),
                new PrintJob("BigMan"),
                new PrintJob("Sancho")};

        ExecutorService service = Executors.newFixedThreadPool(2);

        for (PrintJob job : jobs) {
            service.submit(job);
        }

    // System.out.println(" shutDown");

        service.shutdown();
    }
```

ThreadPoolExecutor Class

- **Executors** class provides simple implementation of **ExecutorService** using **ThreadPoolExecutor**, but ThreadPoolExecutor provides much more feature than that.
- **ThreadPoolExecutor classi ExecutorService class dan implements olgan class hisoblanadi va u juda ko'p imkoniyatlar yaratib beradi.**

ThreadPoolExecutor Methods

- public int getActiveCount()
  - method returns the approximate number of threads that are actively executing tasks.
  - Bu method hozirda active bo’lib ishlab turgan thread lar sonini return qiladi.
- public boolean isTerminated()
  - method returns true if all tasks have completed following shut down. Note that isTerminated is never true unless either shutdown or shutdownNow was called first.
  - Method true return qiladi agar barcha Thread lar tugagan bo’lsa. Eslatma isTerminated hechqachon to’g’ri true bo’lmaydi agar shutdown() yoki shutdowNow() methodlari chaqirilmasa.

dasturlash.uz

Additional Problem

- Solve file scanner problem with Thread Pool

dasturlash.uz

ScheduledThreadPoolExecutor

- The ScheduledThreadPoolExecutor extends the ThreadPoolExecutor class and also implements the ScheduledExecutorService interface with several additional methods:
- ScheduledThreadPoolExecutor classi ThreadPoolExecutor clasiidan nasil olgan va ScheduledExecutorService interface dan implement qilgan.
- ***ScheduledThreadPoolExecutor classi*** = ThreadPoolExecutor + timeDelay

- *Schedule* method allows us to run a task once after a specified delay. And other features.
- Jadval methodlari task ni malum bir vaqtdan keyin ishlatish umkonini beradi. Va boshqa imkoniyatlarni bor. O’zingiz o’qib oling.
-

Links

- ThreadPool
  - https://www.javatpoint.com/java-thread-pool
- ThreadPoolExecutor
  - https://www.javatpoint.com/java-threadpoolexecutor
  - https://www.tutorialspoint.com/java_concurrency/concurrency_threadpoolexecutor.htm
  - https://howtodoinjava.com/java/multi-threading/java-thread-pool-executor-example/
- All
  - https://www.baeldung.com/thread-pool-java-and-guava