



链滴

JAVA 中线程池的使用 (1)

作者: [yhm2](#)

原文链接: <https://ld246.com/article/1577780741462>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

JAVA中线程池的使用

直接代码:

```
package com.yhm.testSpringMvc.Thread;

import java.util.Random;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

public class ThreadPoolSimpleDemo {
    public static void main(String[] args) {

        ExecutorService executorService = Executors.newFixedThreadPool(3);
        executorService.execute(new yhmThreadPoolSimpleDemo1());
        executorService.execute(new yhmThreadPoolSimpleDemo2());
        executorService.execute(new yhmThreadPoolSimpleDemo3());
        executorService.execute(new yhmThreadPoolSimpleDemo4());
        executorService.execute(new yhmThreadPoolSimpleDemo5());
        executorService.execute(new yhmThreadPoolSimpleDemo6());
        executorService.execute(new yhmThreadPoolSimpleDemo7());

        executorService.shutdown();
        System.out.println("主线程执行完毕");

    }

    static class yhmThreadPoolSimpleDemo1 implements Runnable {

        @Override
        public void run() {
            System.out.println("do somethings: yhmThreadPoolSimpleDemo1");
            try {
                Thread.sleep(new Random().nextInt(9999));
                //做其他事情
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }

    static class yhmThreadPoolSimpleDemo2 implements Runnable {

        @Override
        public void run() {
            System.out.println("do somethings: yhmThreadPoolSimpleDemo2");
            try {
                Thread.sleep(new Random().nextInt(9999));
                //做其他事情
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```

static class yhmThreadPoolSimpleDemo3 implements Runnable {

    @Override
    public void run() {
        System.out.println("do somethings: yhmThreadPoolSimpleDemo3");
        try {
            Thread.sleep(new Random().nextInt(9999));
            //做其他事情
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

static class yhmThreadPoolSimpleDemo4 implements Runnable {

    @Override
    public void run() {
        System.out.println("do somethings: yhmThreadPoolSimpleDemo4");
        try {
            Thread.sleep(new Random().nextInt(9999));
            //做其他事情
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

static class yhmThreadPoolSimpleDemo5 implements Runnable {

    @Override
    public void run() {
        System.out.println("do somethings: yhmThreadPoolSimpleDemo5");
        try {
            Thread.sleep(new Random().nextInt(9999));
            //做其他事情
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

static class yhmThreadPoolSimpleDemo6 implements Runnable {

    @Override
    public void run() {
        System.out.println("do somethings: yhmThreadPoolSimpleDemo6");
        try {
            Thread.sleep(new Random().nextInt(9999));
            //做其他事情
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

```

```

static class yhmThreadPoolSimpleDemo7 implements Runnable {

    @Override
    public void run() {
        System.out.println("do somethings: yhmThreadPoolSimpleDemo7");
        try {
            Thread.sleep(new Random().nextInt(9999));
            //做其他事情
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
}
}

```

运行结果:

```

D:\JAVA\jdk8_64\bin\java -agentlib:jdwp=transport=dt_socket,address=127.0.0.1:61843,suspend=y,server=n -Dfile.encoding=UTF-8 -classpath "D:\JAVA\jdk8_64\jre\lib\charsets.jar;D:\JAVA\jdk8_64\jre\lib\deploy.jar;D:\JAVA\jdk8_64\jre\lib\ext\access-bridge-64.jar;D:\JAVA\jdk8_64\jre\lib\ext\cldrdata.jar;D:\JAVA\jdk8_64\jre\lib\ext\dnsns.jar;D:\JAVA\jdk8_64\jre\lib\ext\jaccess.jar;D:\JAVA\jdk8_64\jre\lib\ext\jfxrt.jar;D:\JAVA\jdk8_64\jre\lib\ext\localedata.jar;D:\JAVA\jdk8_64\jre\lib\ext\nashorn.jar;D:\JAVA\jdk8_64\jre\lib\ext\sunec.jar;D:\JAVA\jdk8_64\jre\lib\ext\sunec_provider.jar;D:\JAVA\jdk8_64\jre\lib\ext\sunmscapi.jar;D:\JAVA\jdk8_64\jre\lib\ext\sunpkc11.jar;D:\JAVA\jdk8_64\jre\lib\ext\zipfs.jar;D:\JAVA\jdk8_64\jre\lib\javaws.jar;D:\JAVA\jdk8_64\jre\lib\jce.jar;D:\JAVA\jdk8_64\jre\lib\jfr.jar;D:\JAVA\jdk8_64\jre\lib\jfxswt.jar;D:\JAVA\jdk8_64\jre\lib\jsse.jar;D:\JAVA\jdk8_64\jre\lib\management-agent.jar;D:\JAVA\jdk8_64\jre\lib\plugin.jar;D:\JAVA\jdk8_64\jre\lib\resources.jar;D:\JAVA\jdk8_64\jre\lib\rt.jar;D:\ideaPros\codeGitee\AM\target\classes;D:\repo\junit\junit\4.12\junit-4.12.jar;D:\repo\org\hamcrest\hamcrest-core\1.3\hamcrest-core-1.3.jar;D:\repo\javax\servlet\servlet-api\2.5\servlet-api-2.5.jar;D:\repo\org\springframework\spring-core\4.2.5.RELEASE\spring-core-4.2.5.RELEASE.jar;D:\repo\commons-logging\commons-logging\1.2\commons-logging-1.2.jar;D:\repo\org\springframework\spring-beans\4.2.5.RELEASE\spring-beans-4.2.5.RELEASE.jar;D:\repo\org\springframework\spring-context\4.2.5.RELEASE\spring-context-4.2.5.RELEASE.jar;D:\repo\org\springframework\spring-aop\4.2.5.RELEASE\spring-aop-4.2.5.RELEASE.jar;D:\repo\aopalliance\aopalliance\1.0\aopalliance-1.0.jar;D:\repo\org\springframework\spring-expression\4.2.5.RELEASE\spring-expression-4.2.5.RELEASE.jar;D:\repo\org\springframework\spring-tx\4.2.5.RELEASE\spring-tx-4.2.5.RELEASE.jar;D:\repo\org\springframework\spring-web\4.2.5.RELEASE\spring-web-4.2.5.RELEASE.jar;D:\repo\org\springframework\spring-webmvc\4.2.5.RELEASE\spring-webmvc-4.2.5.RELEASE.jar;D:\repo\org\springframework\spring-test\4.2.5.RELEASE\spring-test-4.2.5.RELEASE.jar;D:\repo\org\slf4j\slf4j-log4j12\1.6.6\slf4j-log4j12-1.6.6.jar;D:\repo\org\slf4j\slf4j-api\1.6.6\slf4j-api-1.6.6.jar;D:\repo\log4j\log4j\1.2.17\log4j-1.2.17.jar;D:\repo\cglib\cglib-nodep\3.2.2\cglib-nodep-3.2.2.jar;D:\repo\org\aspectj\aspectjweaver\1.7.4\aspectjweaver-1.7.4.jar;D:\repo\com\alibaba\dubbo\2.5.3\dubbo-2.5.3.jar;D:\repo\org\springframework\spring\2.5.6.SEC03\spring-2.5.6.SEC03.jar;D:\repo\org\javassist\javassist\3.15.0-GA\javassist-3.15.0-GA.jar;D:\repo\org\jboss\netty\netty\3.2.5.Final\netty-3.2.5.Final.jar;D:\chengxu\intelliJIDEA\IntelliJ IDEA 2017.3.5\lib\idea_rt.jar" com.yhm.testSpringMvc.Thread.ThreadPoolSimpleDemo
Connected to the target VM, address: '127.0.0.1:61843', transport: 'socket'
do somethings: yhmThreadPoolSimpleDemo1
do somethings: yhmThreadPoolSimpleDemo2
do somethings: yhmThreadPoolSimpleDemo3
主线程执行完毕

```

```
do somethings: yhmThreadPoolSimpleDemo4
do somethings: yhmThreadPoolSimpleDemo5
do somethings: yhmThreadPoolSimpleDemo6
do somethings: yhmThreadPoolSimpleDemo7
Disconnected from the target VM, address: '127.0.0.1:61843', transport: 'socket'
```

Process finished with exit code 0

先几个线程打印出一些信息，做一些事情，然后主线程得到执行，输出：“主线程执行完毕”，接下来他线程得到执行，这样，在一个线程池中，每次都有3个线程在执行，其他线程处于等待状态（因为用了固定线程执行大小为3的类：`Executors.newFixedThreadPool(3)`）可以向线程池中加入更多线，但是同时只有3个线程在执行