

Java 线程

Thread II

java web下到处都是多线程

- servlet / filter
- spring controller / services 等

Thread类常用的几个方法

- Thread.currentThread()
- Thread.sleep(long ms)
- yield() 让出CPU供其他线程运行

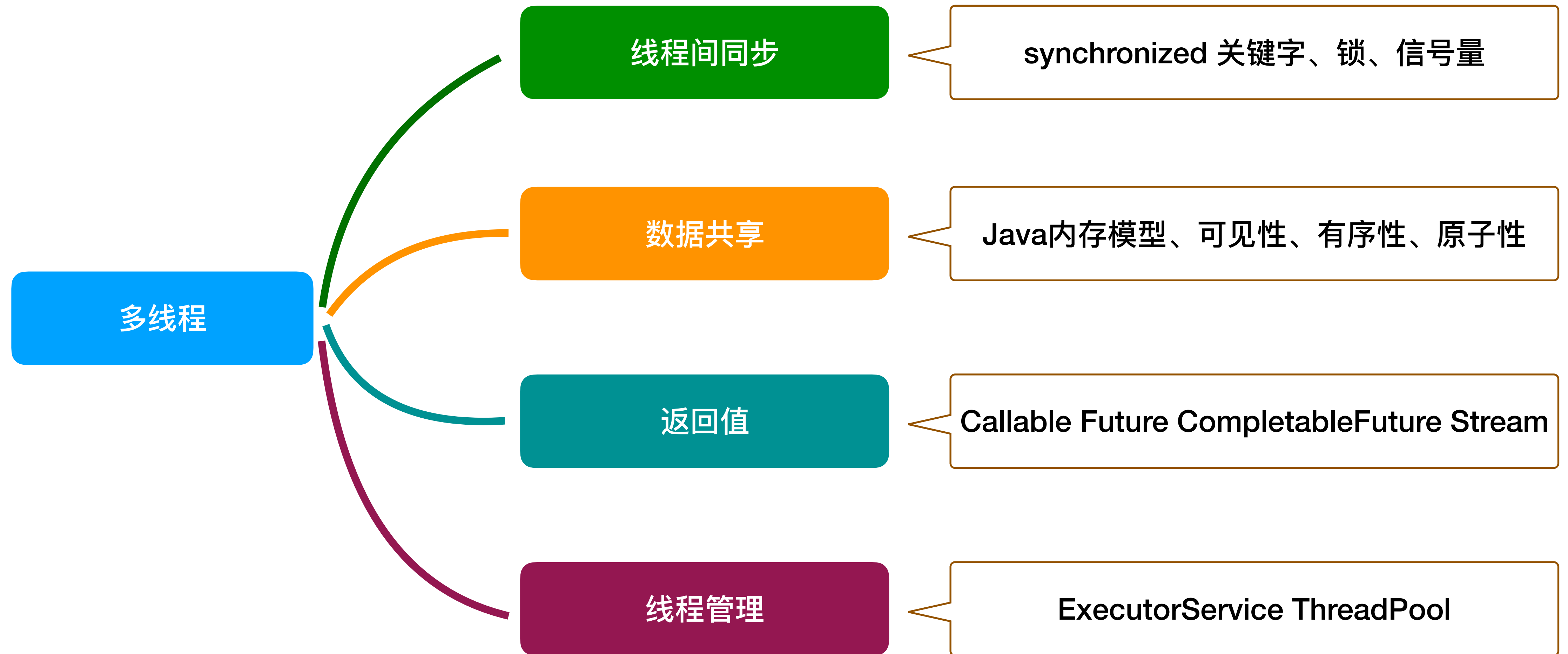
- setDaemon(boolean on) 守护线程标志

- run()
- start()
- ~~stop()~~ 不推荐使用
- interrupt() 给线程发送中断信号
- join() 等待一个线程结束

Thread State (Enum Thread.State)

- NEW
- RUNNABLE
- BLOCKED
- WAITING
- TIMED_WAITING
- TERMINATED

`thread.getState()` 查询线程的状态



Callable vs. Runnable

java.util.concurrent.Callable<V>	java.lang.Runnable
V call()	void run()
<pre>ExecutorService service = Executors.newCachedThreadPool(); Future<Object> future = service.submit(callable); // 可以执行其他代码, <i>callable</i>会自动开始计算并将结果保存到 <i>Future</i>内 Object result = future.get(); //如果<i>callable</i>的<i>call</i>方 法尚未运算完毕, 会等待运算完毕</pre>	<pre>new Thread(runnable).start()</pre>

```
FetureTask<Object> task = new FetureTask<>(runnable);  
new Thread(task).start();  
...  
Object result = task.get();
```

RunnableFuture

RunnableScheduledFuture

Runnable

Thread

ForkJoinWorkerThread

Task

FutureTask

TimerTask