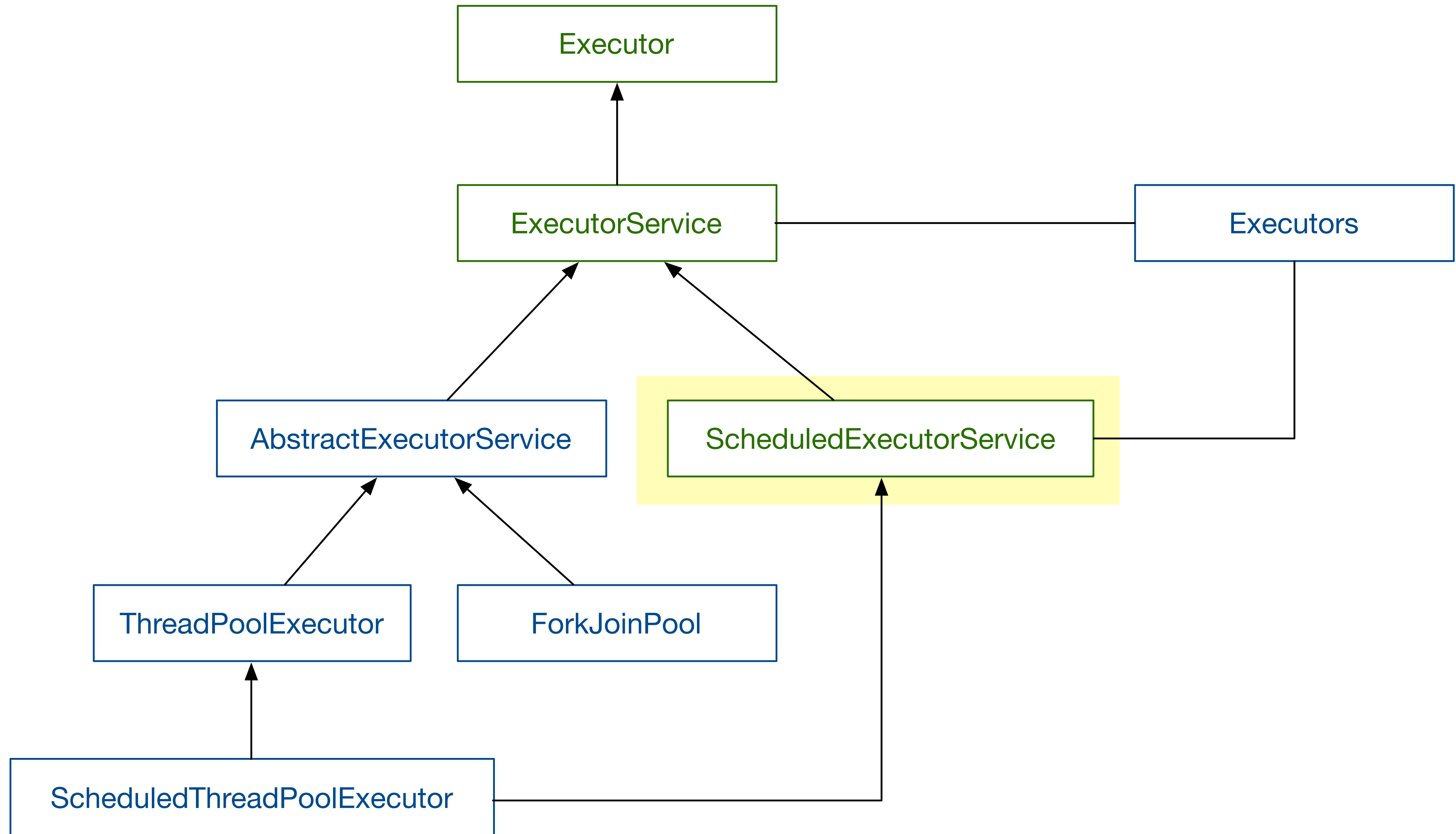


# Thread Pool

ScheduledExecutorService



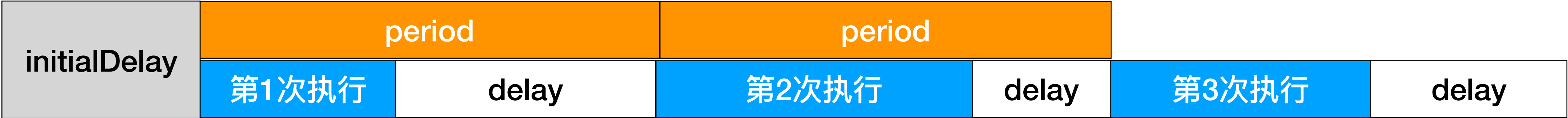
```
ScheduledExecutorService service = Executors.newScheduledThreadPool(5);

Runnable r = () -> {
    // 任务写这里
};

service.schedule(r, 60000, TimeUnit.MILLISECONDS);
```

# 提交任务

ScheduledFuture<?>	schedule(Runnable command, long delay, TimeUnit unit)
<V> ScheduledFuture<V>	schedule(Callable<V> callable, long delay, TimeUnit unit)
ScheduledFuture<?>	scheduleAtFixedRate(Runnable command, long initialDelay, long period, TimeUnit unit)
ScheduledFuture<?>	scheduleWithFixedDelay(Runnable command, long initialDelay, long delay, TimeUnit unit)



# java.util. **Timer & TimerTask**

```
Timer timer = new Timer();
TimerTask task = new TimerTask() {

    @Override
    public void run() {
        // 要执行的任务
    }
};
timer.schedule(task, 10000); // 10秒后执行
```

# Timer vs. ScheduledExecutorService

	Timer	ScheduledExecutorService
执行任务的线程数	1	可设置
执行任务的线程	创建Timer时创建	需要时创建
执行时间	依赖系统当前日期时间设定	不依赖系统日期时间
发生异常	会导致整个后续TimerTask全部不执行	不影响其他任务
返回值	无	调用schedule时返回Future，可以通过Future获得Callable的返回值
结束机制	监视线程是后台线程；任务线程可设置为后台线程	通过shutdown(), shutdownNow()等方法

# Spring Scheduling