**java.util.Timer**

需要创建一个TimerTask，然后注入到Timer中

timer.schedule(TimerTask, delay, period);

灵活度低，不推荐

**ScheduledExecutorService**

基于线程池，每个调度任务都会分配到线程池的一个线程中去执行，并发执行，互不影响。

**Spring Task**

在任务类的方法上，添加注解**@Scheduled(cron=”\* \* \* \* \* ?”)**，方法体内是具体要执行的任务；在主类上使用**@EnableScheduling**注解开启对定时任务的支持。

在一个线程中串行多个定时任务，会产生影响，如果一个任务卡死，其他任务也无法执行。

**多线程配置方法**

新建一个配置类，添加注解@Configuration @EnableAsync，注册一个Executor的Bean；

在定时任务的类或者方法上添加注解@Async。

**cron表达式**

有6个或7个由空格分隔的时间元素，分别是

秒(0~59) 分(0~59) 小时(0~23) 月份中天(1~31) 月(1~12) 星期中天(1~7或SUN MON TUE WED THU FRI SAT) 年份(1970~2099)

每个元素可以是：

一个值 6

一个连续区间 1-10

一个时间间隔 8-18/4

一个列表 1,3,5

\* ?通配符

\* 代表所有可能的值

? 仅被用于天(月)和天(星期)两个子表达式，表示不指定值，两个中有一个是?

L 代表最后，用于天(月)和天(星期)，如6L表示这个月的倒数第6天

W 代表工作日，15W表示离该月15号最近的一个平日

C 计划所关联的日期，5C日历5日之后的第一天

**常用实践**

/5 \* \* \* \* ? 每5s执行一次

0 0/30 9-17 \* \* ? 朝9晚5工作时间内每半小时

0 0 12 \* \* ? 每天中午12点

0 \* 14 \* \* ? 每天下午2点到3点之间，每分钟触发

0 15 10 ? \* MON-FRI 周一到周五上午10:15

0 0 12 ? \* 6#3 每月的第三个星期五12点

**Quartz**

创建任务类，继承QuartzJobBean，实现executeInternal方法；

创建配置类，注册JobDetail，并将其注入到Trigger中去；使用Trigger绑定SimpleScheduleBuilder和JobDetail。