

```

public interface ScheduledExecutorService extends ExecutorService {

    /**
     * 提交一个待执行的任务，并在给定的延迟后执行该任务。
     *
     * @param command 待执行的任务
     * @param delay 延迟时间
     * @param unit 延迟时间的单位
     */
    public ScheduledFuture<?> schedule(Runnable command, long delay, TimeUnit unit);

    /**
     * 提交一个待执行的任务（具有返回值），并在给定的延迟后执行该任务。
     *
     * @param command 待执行的任务
     * @param delay 延迟时间
     * @param unit 延迟时间的单位
     * @param <V> 返回值类型
     */
    public <V> ScheduledFuture<V> schedule(Callable<V> callable, long delay, TimeUnit unit);

    /**
     * 该任务在 initialDelay 后开始执行，然后在 initialDelay+period 后执行，接着在
     * initialDelay + 2 * period 后执行，依此类推。
     *
     * 提交一个待执行的任务。
     * 该任务在 initialDelay 后开始执行，然后在 initialDelay+period 后执行，接着在 initialDel
     *
     * @param command 待执行的任务
     * @param initialDelay 首次执行的延迟时间
     * @param period 连续执行之间的周期
     * @param unit 延迟时间的单位
     */
    public ScheduledFuture<?> scheduleAtFixedRate(Runnable command, long initialDelay, long period, TimeUnit unit);

    /**
     * 该任务在 initialDelay 后开始执行，随后在每一次执行终止和下一次执行开始之间都存在给定的延迟。
     * 如果任务的任一执行遇到异常，就会取消后续执行。否则，只能通过执行程序的取消或终止方法来终止该任务。
     *
     * 提交一个待执行的任务。
     * 该任务在 initialDelay 后开始执行，随后在每一次执行终止和下一次执行开始之间都存在给定的延迟。
     * 如果任务的任一执行遇到异常，就会取消后续执行。否则，只能通过执行程序的取消或终止方法来终止
     *
     * @param command 待执行的任务
     * @param initialDelay 首次执行的延迟时间
     * @param delay 一次执行终止和下一次执行开始之间的延迟
     * @param unit 延迟时间的单位
     */
    public ScheduledFuture<?> scheduleWithFixedDelay(Runnable command, long initialDelay, long delay, TimeUnit unit);
}

```