**Fix-delay**: một task **A** sẽ được thực thi sau 1 khoản thời gian thực thi của task **B** trước đó cộng thêm thời gian delay cố định cho mỗi task.

**Fix-rate:** Một task A thực thi sau khoảng thời gian cố định cho trước không quan tâm tới task **B** trước đó hay thời gian delay mà task **B** có.

**Cả 2 schedule trên** đều có 2 phương thức overload vì mỗi loại đều có thể start sau 1 khoảng delay.

**Calling cancel()** inside  **run()** implêmntation to cancel the TimerTask, Or we can cancel outside  **run() by**  using instance of **timmerTask**

@Test **public** **void** **givenUsingTimer\_whenCancelingTimerTask\_thenCorrect**() **throws** InterruptedException {

TimerTask task = **new** TimerTask() {

**public** **void** **run**() {

System.out.println("Task performed on " + **new** Date());

cancel(); **(1)**

}

};

Timer timer = **new** Timer("Timer");

timer.scheduleAtFixedRate(task, 1000L, 1000L); Thread.sleep(1000L \* 2);

// calling **task.cancle**() here instead of (1) is **ok**

}

**Chú ý: sau khi stop timerTask, có thể luồng do timmer đó tạo ra vẫn còn tồn tại. để có thể stop luồng của timmer task khi timmmer chạy xong hoặc bất kỳ lúc nào, chúng ta có thể dùng API stop của Thread bên trong phương thức Run().**

**Public void run(){**

**If(…){**

**Cancel(); // Stop the TimerTask**

**Thread.stop(); // Stop the thread**

**}**

**}**