```java
     */
    public FutureTask( @NotNull Callable<V> callable) {
        if (callable == null)
            throw new NullPointerException();
        this.callable = callable;
        this.state = NEW;        // ensure visibility of callable
    }
```

```java
     */
    public FutureTask( @NotNull Runnable runnable, V result) {
        this.callable = Executors.callable(runnable, result);
        this.state = NEW;        // ensure visibility of callable
    }
```

```java
     */
    @NotNull
    public static <T> Callable<T> callable( @NotNull Runnable task, T result) {
        if (task == null)
            throw new NullPointerException();
        return new RunnableAdapter<T>(task, result);
    }
```

```java
     */
    static final class RunnableAdapter<T> implements Callable<T> {
        final Runnable task;
        final T result;
        RunnableAdapter(Runnable task, T result) {
            this.task = task;
            this.result = result;
        }
        public T call() {
            task.run();
            return result;
        }
    }
}
```

Runnable   callable