マルチスレッドの基本

スレッド

- プログラムを実行する道筋
    - 通常はスレッドは1本（シングルスレッド）
    - マルチスレッド＝単一のプログラム内で実行箇所が2カ所以上

新規Javaプロジェクトの作成

- プロジェクト名：ThreadTest

マルチスレッド実現の2つの方法

1.Threadクラスを継承

2.Runnableインターフェイスを実装

方法1：Threadクラスを継承

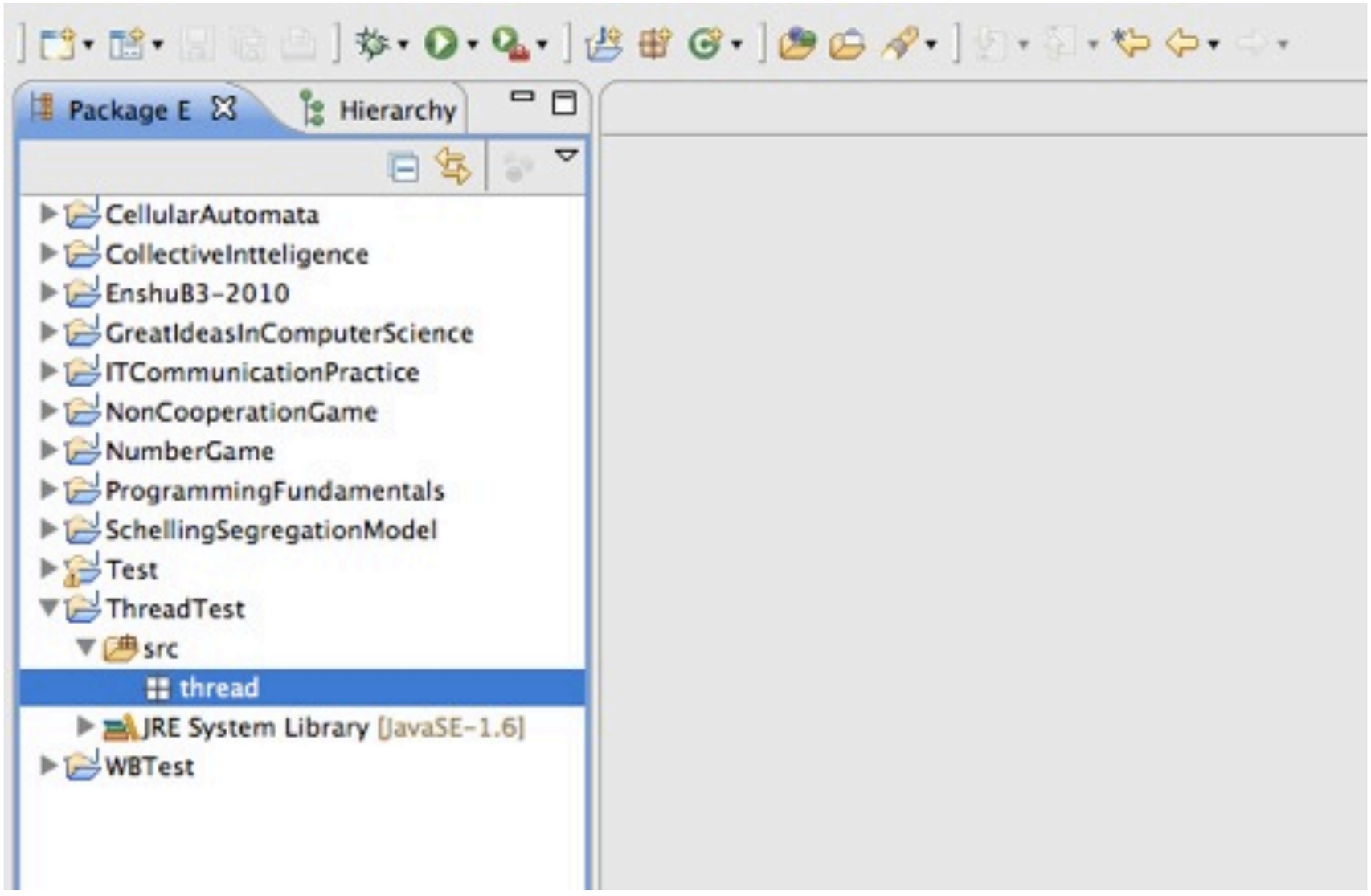
- runメソッドを実装
  → 別スレッドで実行されるメソッド
  （このメソッドを抜けるとスレッドは終了）

- startメソッドを実行
  → 別スレッドのスタート（runメソッドが呼ばれる）

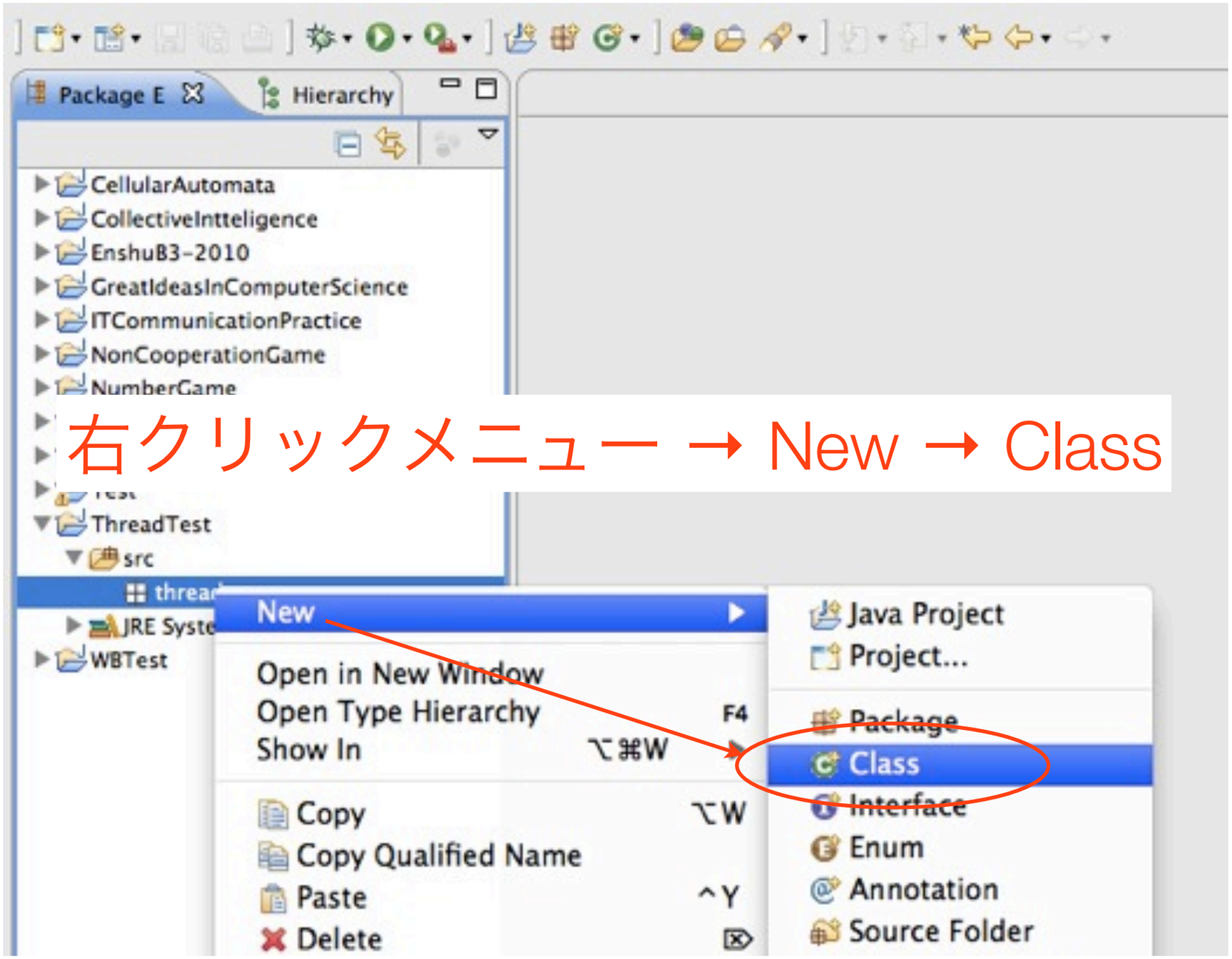
- joinメソッドを実行
  → スレッドの終了を待つ

方法1用のパッケージを作成

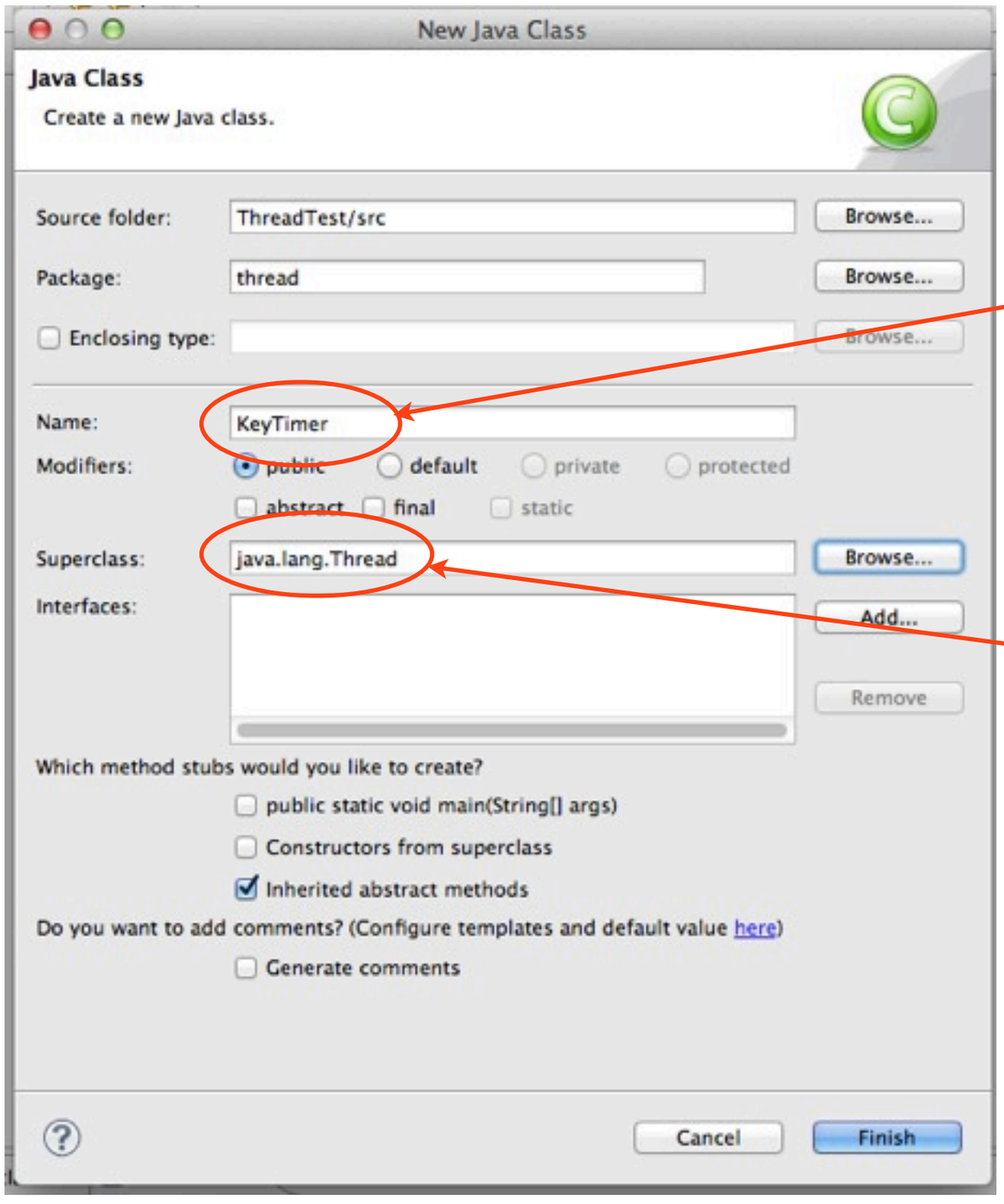
- パッケージ名：thread



新規クラスの追加


Package E
Hierarchy
CellularAutomata
CollectiveIntteligence
EnshuB3-2010
GreatIdeasInComputerScience
ITCommunicationPractice
NonCooperationGame
NumberGame
ThreadTest
src
JRE Syste
WBTest
右クリックメニュー → New → Class
New
Open in New Window
Open Type Hierarchy F4
Show In ⌥⌘W
Copy ⌥W
Copy Qualified Name
Paste ^Y
Delete
Java Project
Project...
Package
Class
Interface
Enum
Annotation
Source Folder

Threadクラスを継承



クラス名：KeyTimer

java.lang.Threadを継承

KeyTimerクラスの実装

- 配布したプリントに従って runメソッド他を実装

KeyTimerAppクラスの実装

- mainメソッドを持つクラスとして新規クラスを追加
- 配布したプリントに従って実装

KeyTimerAppクラス

```java
package thread;

import java.util.Scanner;

public class KeyTimerApp {
    KeyTimer timer;

    KeyTimerApp() {
        timer = new KeyTimer();
    }

    public void execute() {
        timer.start();

        Scanner scan = new Scanner(System.in);
        String s = scan.nextLine();
        System.out.println("[" + timer.getSeconds() + " sec.] " + s);
        timer.stopThread();
        try {
            timer.join();
        } catch(InterruptedException e) {
            e.printStackTrace();
        }
    }

    /**
     * @param args
     */
    public static void main(String[] args) {
        KeyTimerApp app = new KeyTimerApp();
        app.execute();
    }
}
```

別スレッド実行オブジェクトを生成

別スレッドをスタート

別スレッドの終了待ち

方法2：Runnableインターフェイスの実装

- runメソッドの実装 → Threadクラスを継承した場合と同じ

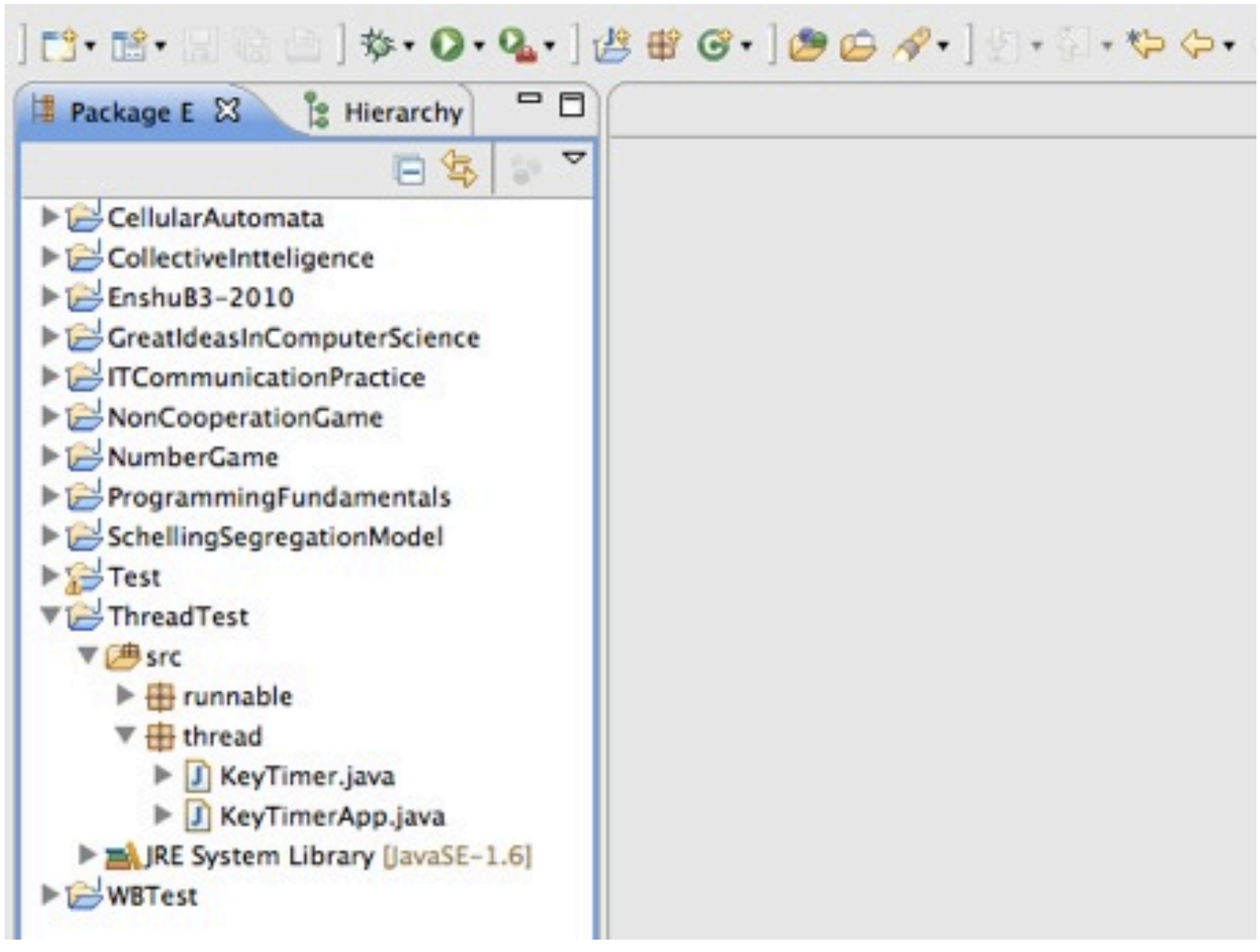
- Threadクラスのインスタンスを作成
  → Runnableインターフェイスを実装したインスタンスを渡す

- Threadクラスのインスタンスのstartメソッドを実行
  → 別スレッドのスタート

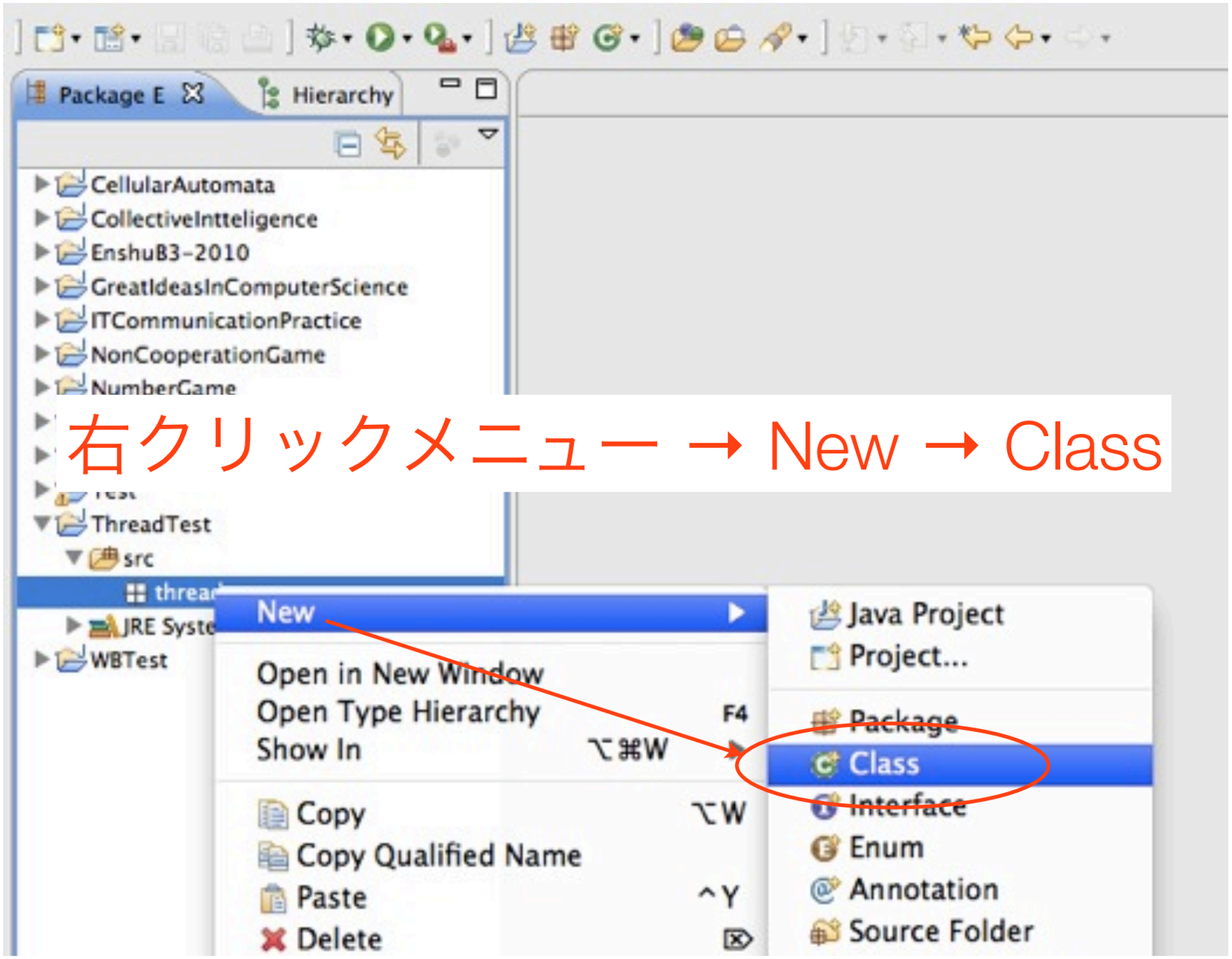
- Threadクラスのインスタンスのjoinメソッドを実行
  → 別スレッドの終了待ち

方法1用のパッケージを作成

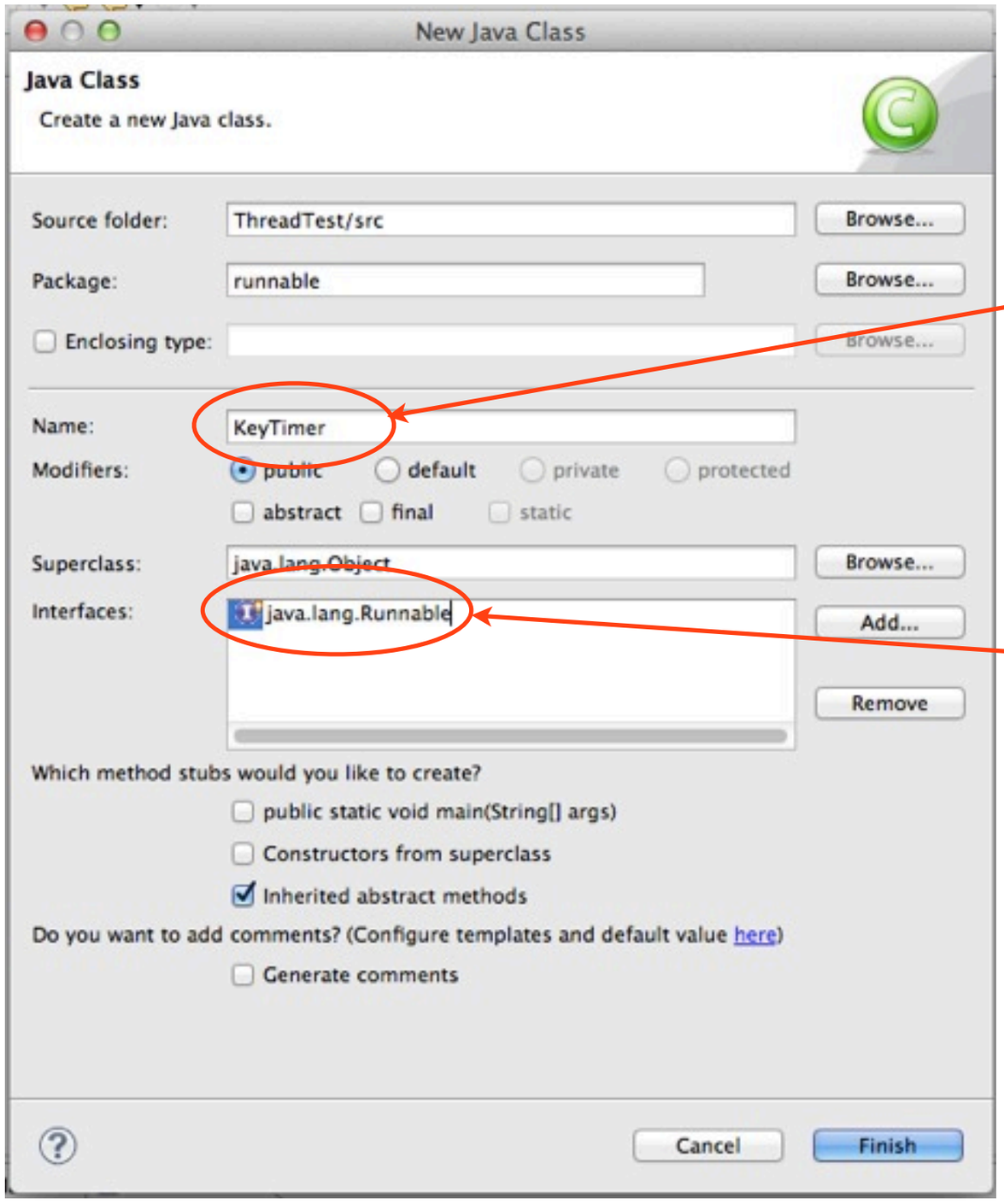
- パッケージ名：runnable



新規クラスの追加


Package E
Hierarchy
CellularAutomata
CollectiveIntteligence
EnshuB3-2010
GreatIdeasInComputerScience
ITCommunicationPractice
NonCooperationGame
NumberGame
ThreadTest
src
JRE Syste
WBTest
右クリックメニュー → New → Class
New
Open in New Window
Open Type Hierarchy F4
Show In ⌥⌘W
Copy ⌥W
Copy Qualified Name
Paste ^Y
Delete
Java Project
Project...
Package
Class
Interface
Enum
Annotation
Source Folder

Runnableインターフェイスを実装



クラス名：KeyTimer

java.lang.Runnableインターフェイス

KeyTimerクラスの実装

- 配布したプリントに従って runメソッド他を実装

KeyTimerAppクラスの実装

- mainメソッドを持つクラスとして新規クラスを追加
- 配布したプリントに従って実装

KeyTimerAppクラス

```
 1  package runnable;
 2
 3  import java.util.Scanner;
 4
 5  public class KeyTimerApp {
 6      private KeyTimer timer;
 7
 8      KeyTimerApp() {
 9          timer = new KeyTimer();
10      }
11
12      public void execute() {
13          Thread thread = new Thread(timer);
14          thread.start();
15
16          Scanner scan = new Scanner(System.in);
17          String s = scan.nextLine();
18          System.out.println("[" + timer.getSeconds() + " sec.] " + s);
19          timer.stopThread();
20          try {
21              thread.join();
22          } catch(InterruptedException e) {
23              e.printStackTrace();
24          }
25      }
26
27      /**
28       * @param args
29       */
30      public static void main(String[] args) {
31          KeyTimerApp app = new KeyTimerApp();
32          app.execute();
33      }
34  }
35
```

Runnableを実装したインスタンスを生成

Threadインスタンスを生成

別スレッドのスタート

別スレッドの終了待ち