

Eduardo Rojo Sánchez



Servicios y Procesos

2º CFGS DAM



1. Ejercicio Hilo.



Grupo
TEIDE-HEASE
Centros de Formación

Java - Hilo1/src/hilo/SimpleHilo.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- Hilo1
 - src
 - hilo
 - SimpleHilo.java
- JRE System Library [JavaSE-1.8]

Quick Access

Java EE Java

Mylyn

to your task and ALM tools or [create](#) a local task.

SimpleHilo

```
main(String[]) : void
```

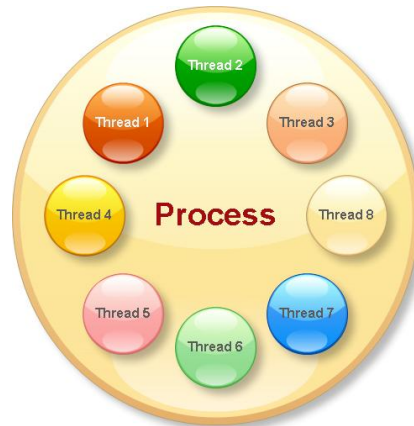
Problems

No consoles to display at this time.

Con los hilos
puedes hacer en tu
programa más de
una tarea
simultáneamente.

Objetivo

- Crear un **thread** que saque por pantalla 10 veces el nombre del *thread creado*.





Quick Access

Java EE Java

Package Explorer

Hilo1

Crea tu proyecto.

Task List

Find

Connect Mylyn

[Connect](#) to your task and ALM tools or [create](#) a local task.

Outline

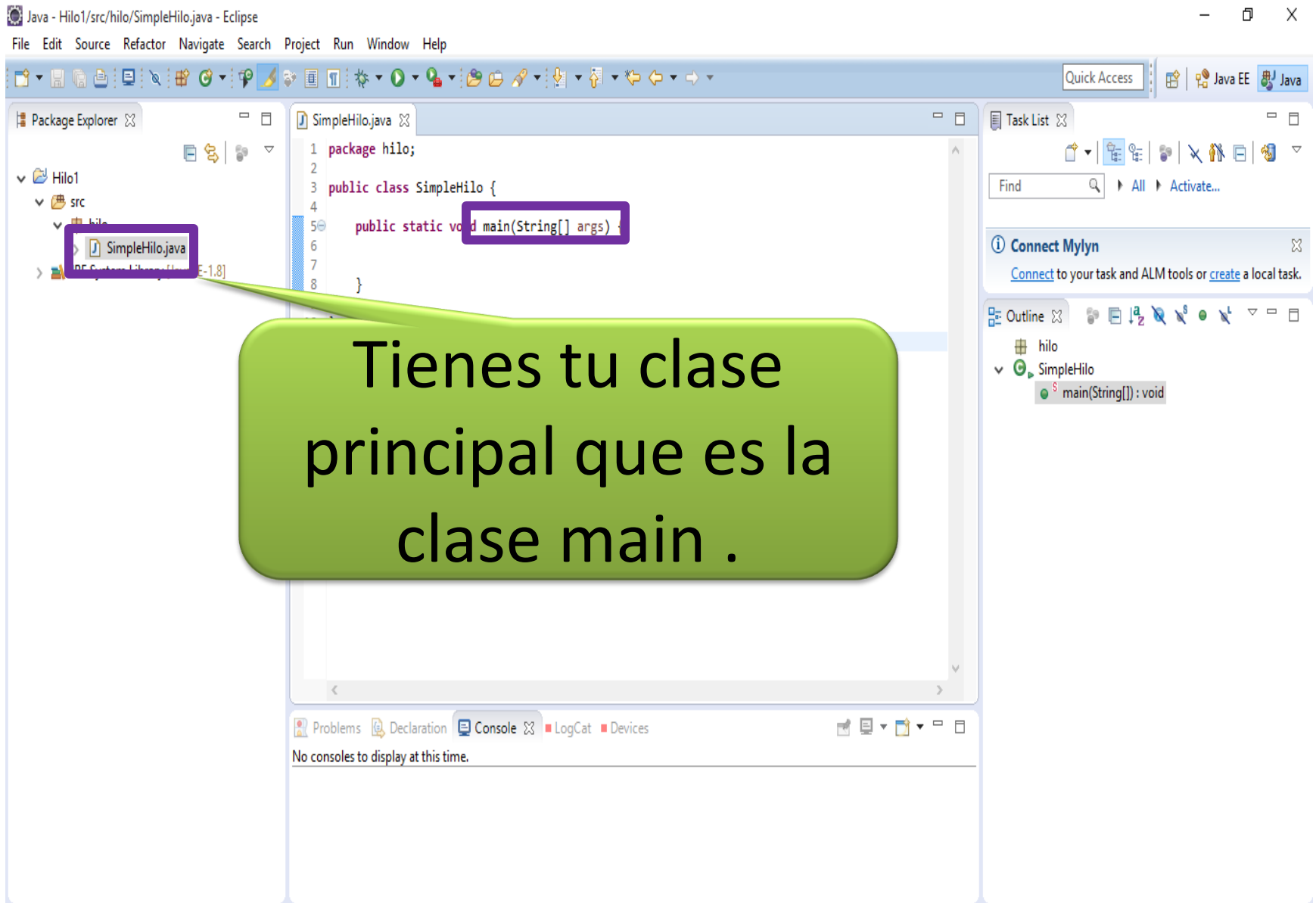
hilo

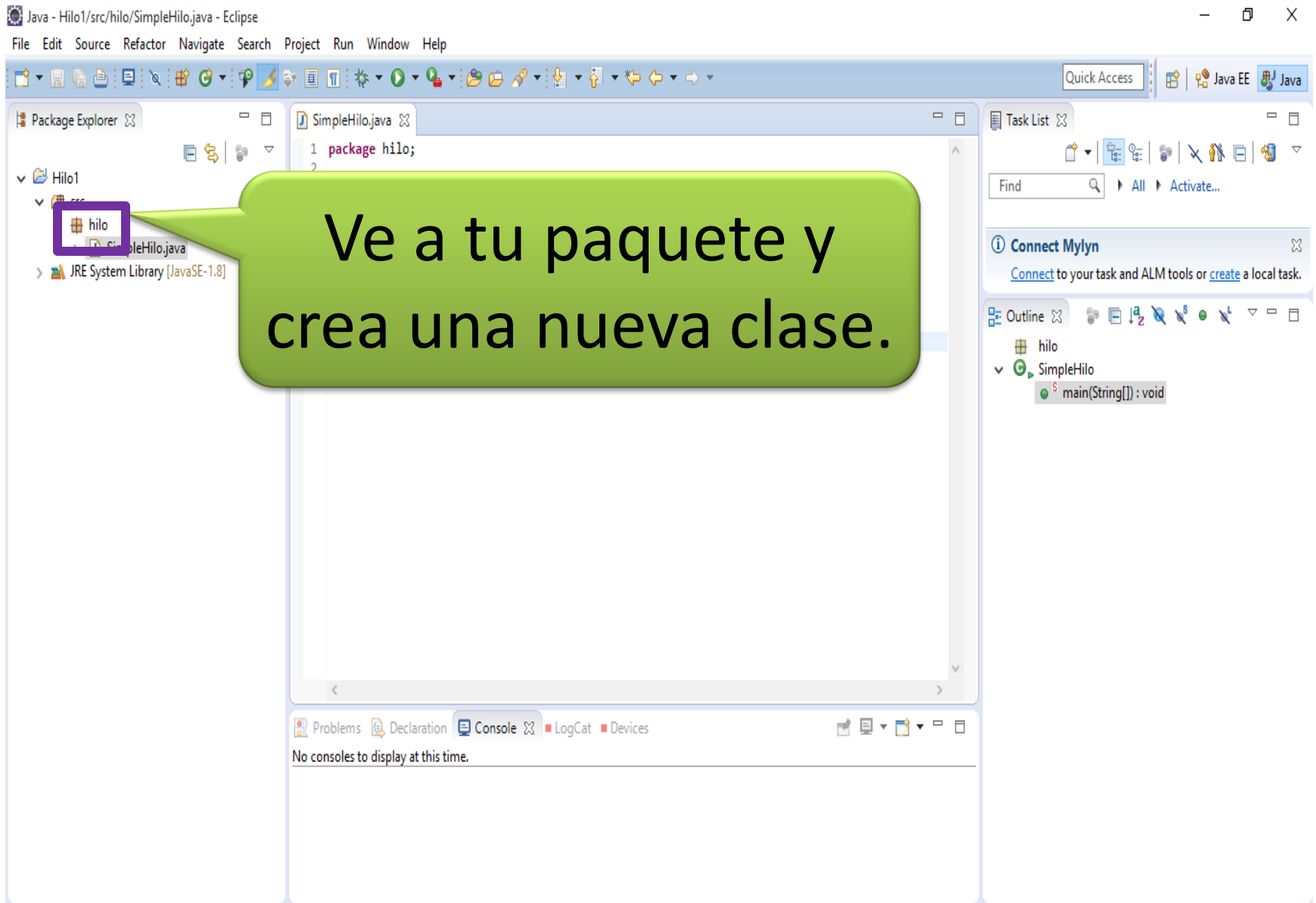
SimpleHilo

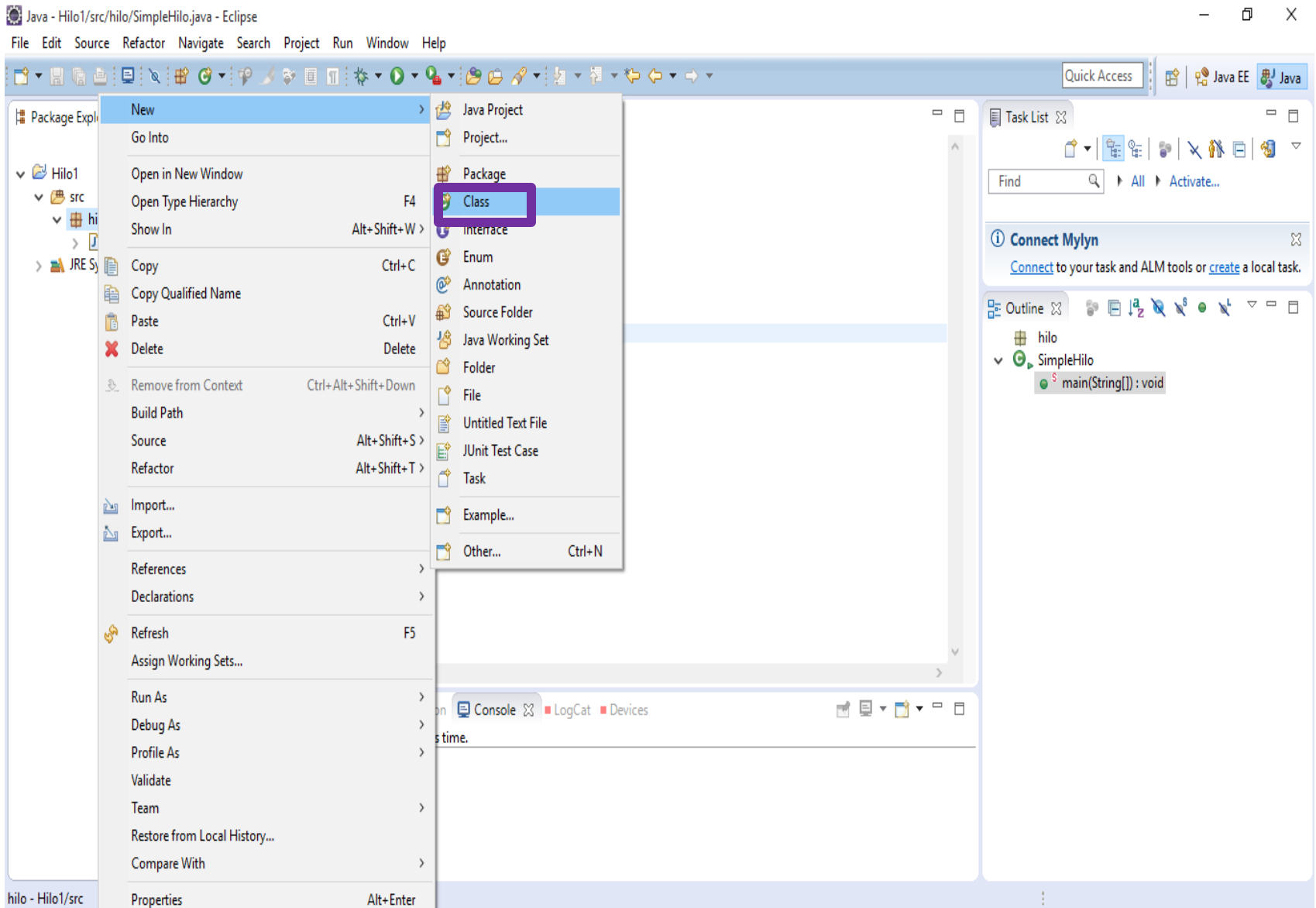
main(String[]) : void

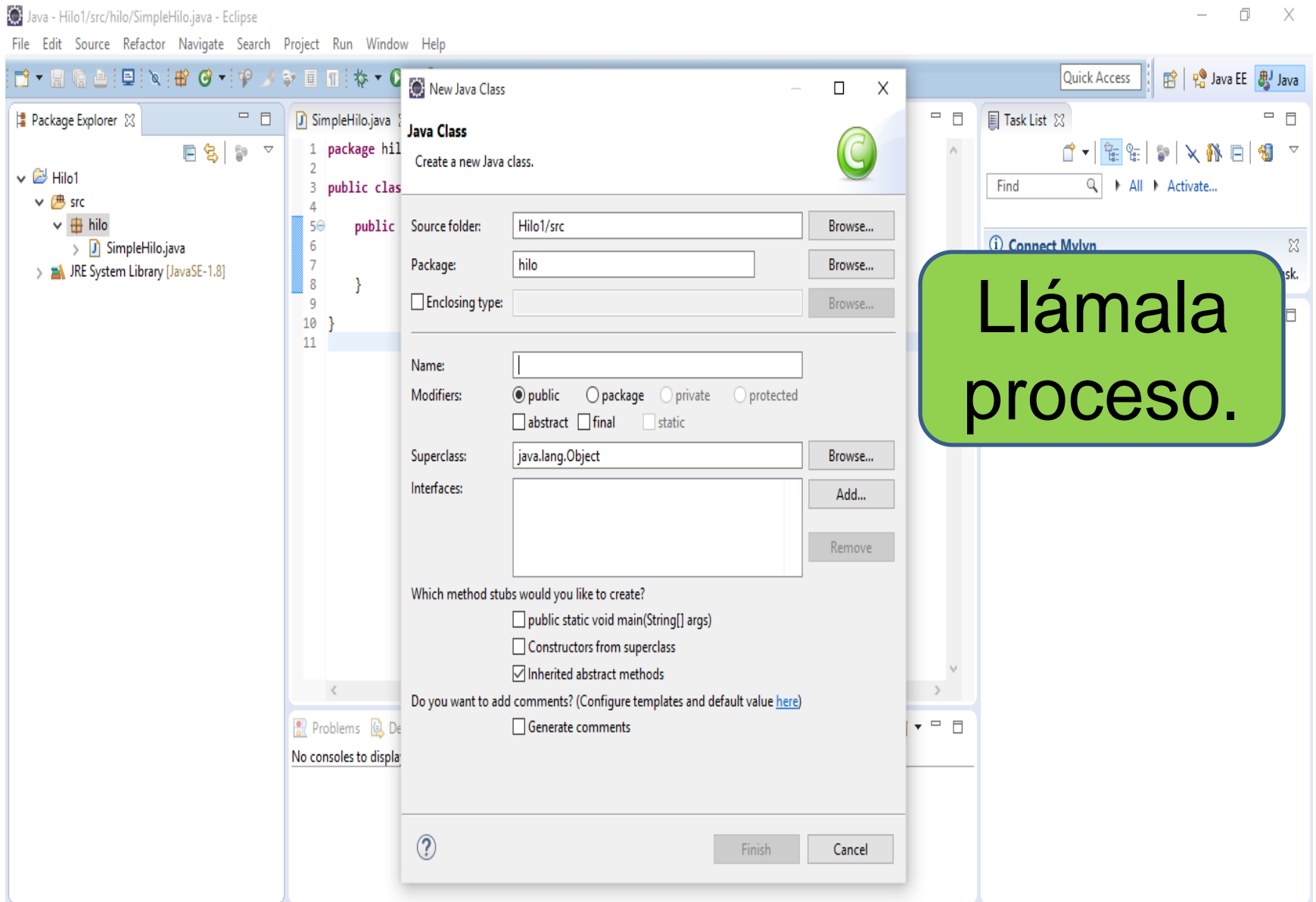
Problems Declaration Console LogCat Devices

No consoles to display at this time.









Java - Hilo1/src/hilo/SimpleHilo.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- Hilo1
 - src
 - hilo
 - SimpleHilo.java
- JRE System Library [JavaSE-1.8]

SimpleHilo.java

```
1 package hilo
2
3 public class SimpleHilo
4
5     public static void main(String[] args)
6
7
8
9
10
11
```

New Java Class

Java Class

Type name is discouraged. By convention, Java type names usually start with an uppercase letter

Source folder: Hilo1/src Browse...

Which method stubs would you like to create?

- ☐ public static void main(String[] args)
- ☐ Constructors from superclass
- ☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

- ☐ Generate comments

Finish Cancel

Task List

Find All Activate...

Connect Mylyn

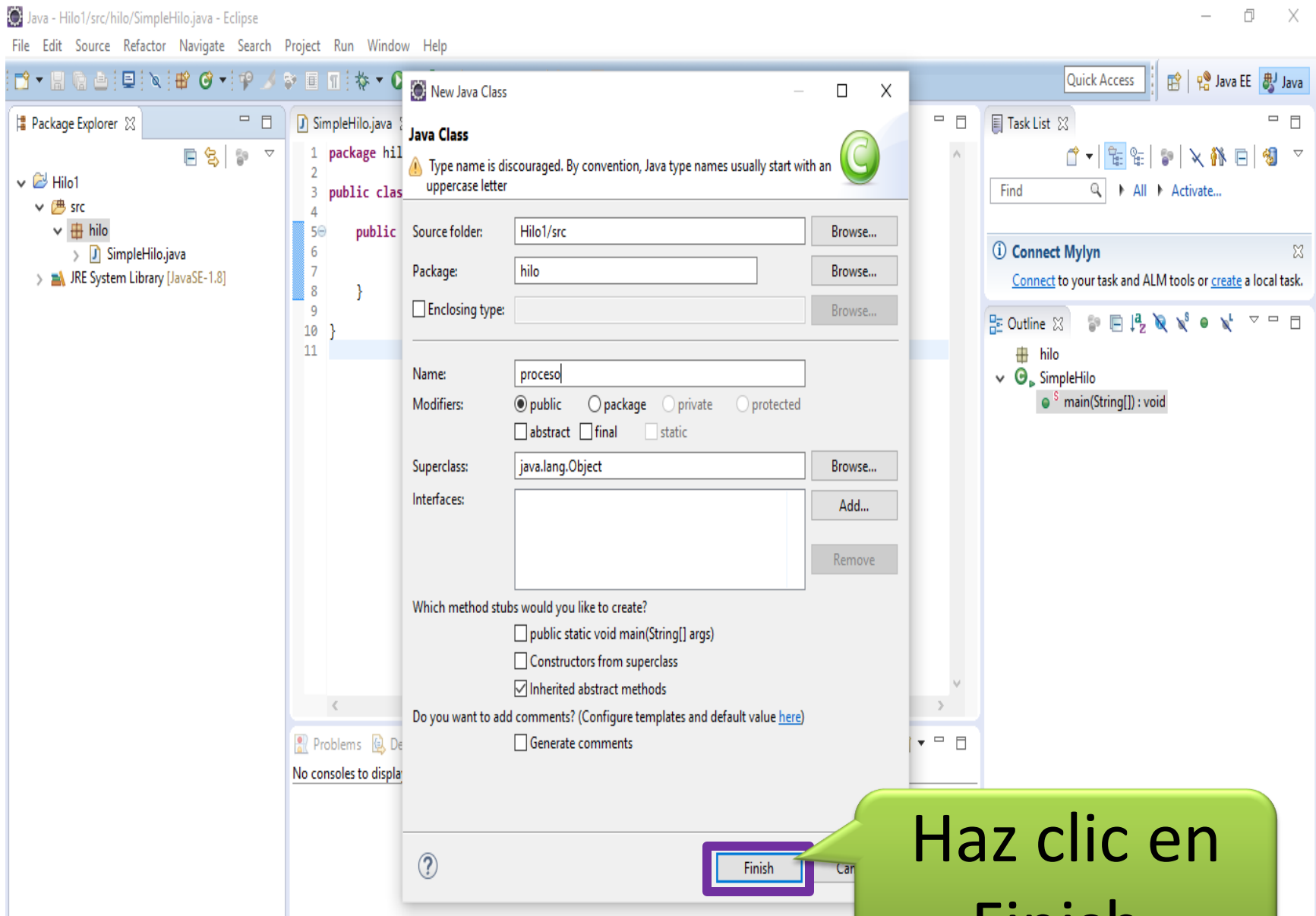
Connect to your task and ALM tools or [create](#) a local task.

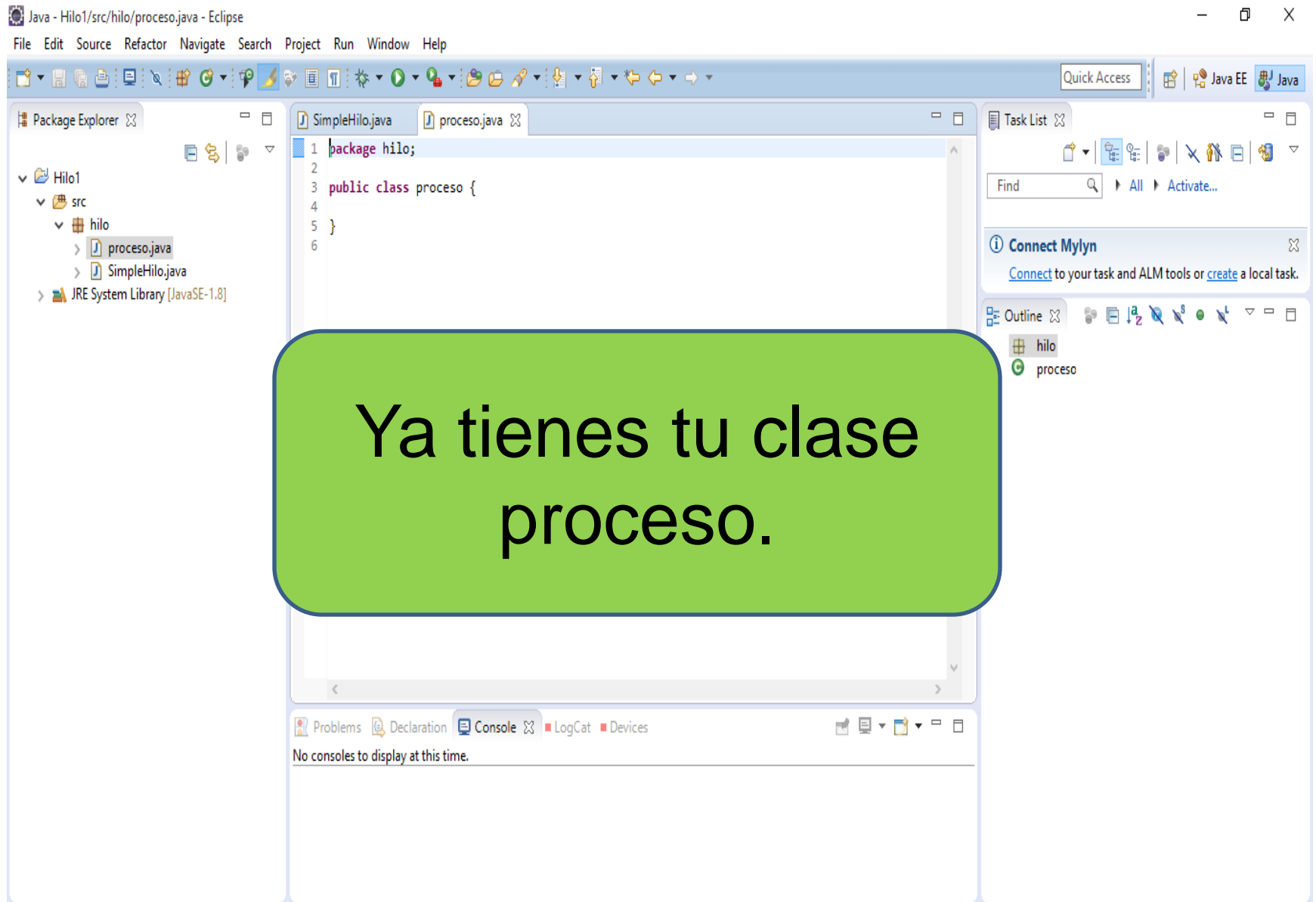
Outline

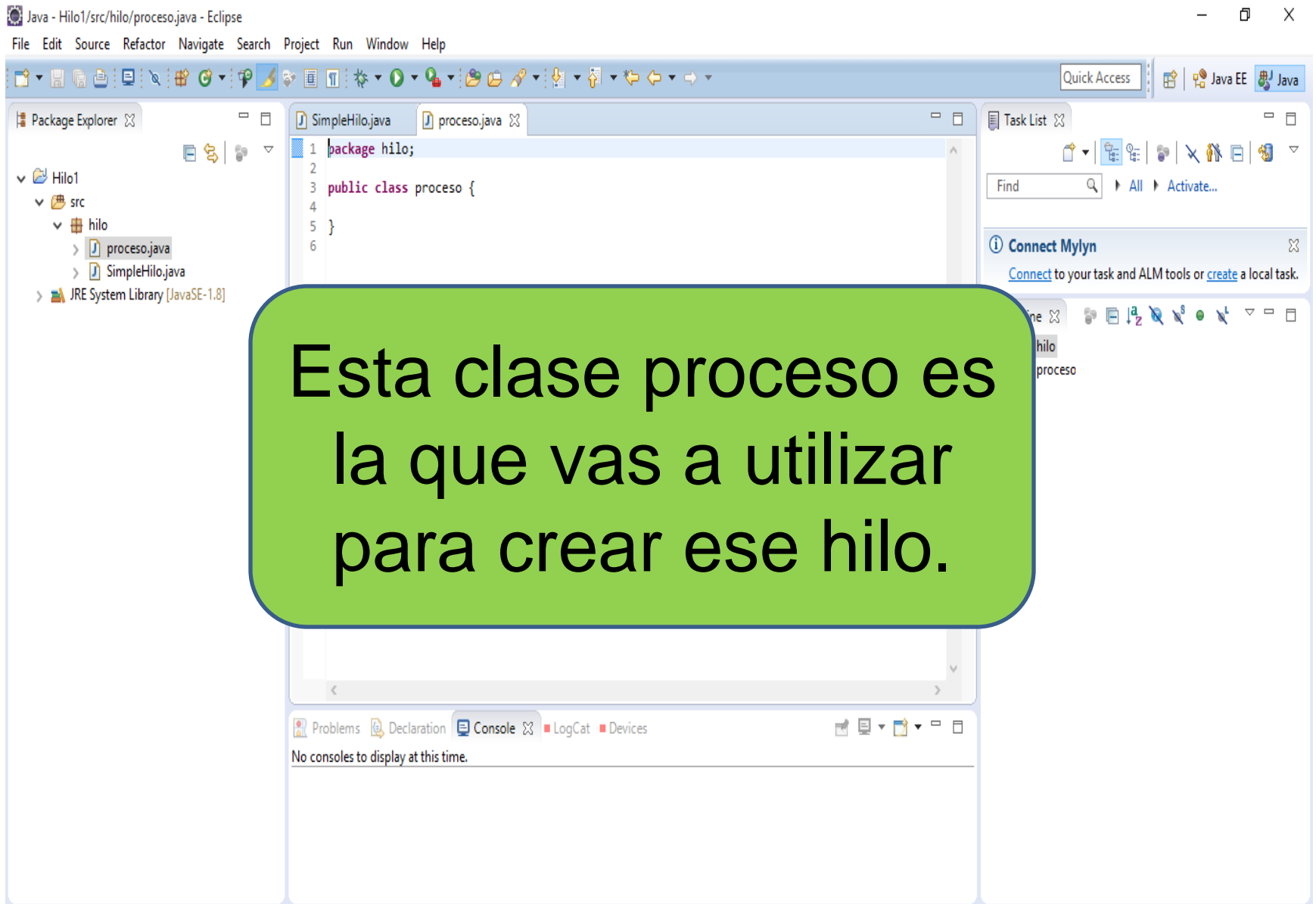
- hilo
 - SimpleHilo
 - main(String[]) : void

No consoles to display

No lleva main, ya que es una clase independiente.







SimpleHilo.java

proceso.java

```
1 package hilo;
2
3 public class proceso extends Thread{
4
5
6 }
7
8
9
```

Hereda de
Thread.

Hereden propiedades de la
clase **Thread**, que nos
permite crear esos hilos.

SimpleHilo.java

proceso.java

```
1 package hilo;
2
3 public class proceso extends Thread{
4
5     public proceso() {
6
7     }
8 }
9
```

Crea el
constructor.

SimpleHilo.java

proceso.java

```
1 package hilo;
2
3 public class proceso extends Thread{
4
5     public proceso(String str) {
6
7     }
8 }
9
10
11
```

En su constructor se utiliza un ***String*** (opcional) para poner nombre al nuevo thread creado.

SimpleHilo.java

proceso.java

```
1 package hilo;
2
3 public class proceso extends Thread{
4
5     public proceso(String
6         super(str);
7     }
8 }
9
10
11
```

Mediante ***super()*** se ***llama*** al constructor de la super-clase ***Thread.***

SimpleHilo.java

proceso.java

```
1 package hilo;
2
3 public class proceso extends Thread{
4
5     public proceso() {
6         super();
7     }
8
9
10    // redefinición del método run()
11
12 }
```

Redefinición del
método run().

SimpleHilo.java

proceso.java



```
1 package hilo;
2
3 public class proceso extends Thread{
4
5     public proceso(String str) {
6         super(str);
7     }
8
9
10    // redefinición del método run()
11
12    public void run() {
13
14    }
15
16 }
17
18
```

SimpleHilo.java

proceso.java

```
1 package hilo;
2
3 public class proceso extends Thread{
4
```

Este método es el que se va a **ejecutar** a la hora de iniciar tu proceso.

```
12 // Este método es el que se va a ejecutar a la hora de
13 // iniciar tu proceso.
```

```
14 public void run() {
15
16 }
17
18 }
19
```

SimpleHilo.java

proceso.java

```
1 package hilo;
2
3 public class proceso extends Thread{
4
5     public proceso(String str) {
6         super(str);
7     }
8
9
10
11
12
13
14
15 // Run define la actividad principal del thread.
16 public void run() {
17
18 }
19
20 }
21
22
23
```

run() define la principal actividad del thread.

SimpleHilo.java

proceso.java

```
1 package hilo;
2
3 public class proceso extends Thread{
4
5     public proceso(String str) {
6         super(str);
7     }
8
9
10    // redefinición del método
11
12    // Este método es el que se
13    // iniciar tu proceso.
14
15    // Run define la actividad principal del thread.
16    public void run() {
17
18    }
19
20 }
21
22
23
```

Esta es la estructura básica.

SimpleHilo.java proceso.java

Ve a la clase main.

```
1 pack
2
3 publ
4
5 public static void main(String[] args) {
6
7
8 }
9
10 }
11
```


Package Explorer

Hilo1

src

hilo

proceso.java

SimpleHilo.java

JRE System Library [JavaSE-1.8]

SimpleHilo.java

proceso.java

```
1 package hilo;
2
3 public class SimpleHilo {
4
5     public static void main(String[] args) {
6         // Para poner en marcha este nuevo thread se debe crear un objeto
7         // de la clase proceso que acabamos de contruir.
8
9
10    }
11
12 }
13
```

```
1 package hilo;
2
3 public class SimpleHilo {
4
5     public static void main(String[] args) {
6         // Para poner en marcha este nuevo thread se debe crear un objeto
7         // de la clase proceso que acabamos de contruir.
8
9
10
11     }
12
13 }
14
```

Para poner en marcha este nuevo thread se debe crear un objeto de la clase proceso que acabamos de construir.

*SimpleHilo.java x proceso.java

```
1 package hilo;
2
3 public class SimpleHilo {
4
5     public static void main(String[] args) {
6         // Para poner en marcha este nuevo thread se debe crear un objeto
7         // de la clase proceso que acabamos de contruir.
8
9         proceso miHilo;
10
11
12
13     }
14
15 }
16
```

Creo una variable de la clase proceso que he creado.

SimpleHilo.java x proceso.java

```
1 package hilo;
2
3 public class SimpleHilo {
4
5     public static void main(String[] args) {
6         // Para poner en marcha este nuevo thread se debe crear un objeto
7         // de la clase proceso que acabamos de contruir.
8
9         proceso miHilo;
10
11         miHilo = new proceso("Hilo1");
12
13     }
14
15 }
16 }
```

Creo un objeto de la
clase proceso.

SimpleHilo.java x proceso.java

```
1 package hilo;
2
3 public class SimpleHilo {
4
5     public static void main(String[] args) {
6         // Para poner en marcha este nuevo thread se debe crear un objeto
7         // de la clase proceso que acabamos de contruir.
8
9         proceso miHilo;
10
11         miHilo = new proceso("Hilo1");
12
13     }
14
15 }
16 }
```

Introduzco el nombre
del proceso.

```
package hilo;

public class SimpleHilo {

    public static void main(String[] args) {
        // Para poner en marcha este nuevo th
        // de la clase proceso que acabamos d

        proceso miHilo;

        miHilo = new proceso("Hilo1");

    }
}
```

```
package hilo;

public class proceso extends Thread{

    public proceso(String str) {
        super(str);
    }

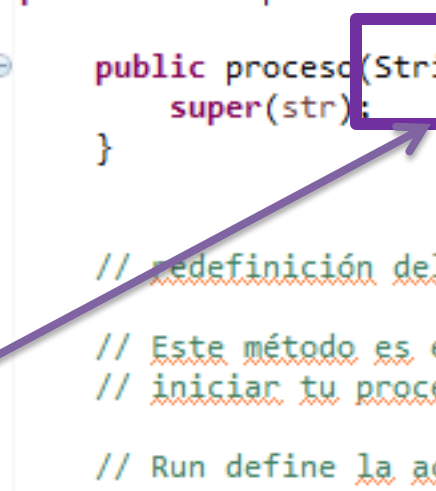
    // redefinición del método run()

    // Este método es el que se va a e
    // iniciar tu proceso.

    // Run define la actividad princip
    public void run() {

    }

}
```



```
1 package hilo;
2
3 public class SimpleHilo {
4
5     public static void main(String[] args) {
6         // Para poner en marcha este nuevo thread se debe crear un objeto
7         // de la clase proceso que acabamos de contruir.
8
9
10
11
12
13
14
15
16         // Llamo al método start(), heredado de la super clase thread,
17         // que se encarga de llamar al metodo run().
18         miHilo.start();
19     }
20
21 }
22
```

Nombre de la variable +
punto.

```
1 package hilo;
```

```
2
```

```
3 public class SimpleHilo {
```

```
4
```

```
5 pu
```

Llamar al método ***start()***,
*heredado de la super-clase
Thread, que se encarga de
llamar a run()*.

```
6 jeto
```

```
7
```

```
8
```

```
9
```

```
10
```

```
11
```

```
12
```

```
13
```

```
14
```

```
15
```

```
16
```

```
16 // Llamo al método start(), heredado de la super clase thread,
```

```
17
```

```
17 // que se encarga de llamar al metodo run().
```

```
18
```

```
18 miHilo.start();
```

```
19
```

```
19 }
```

```
20
```

```
21 }
```

```
22
```


SimpleHilo.java

proceso.java

Ve a la clase
proceso.Java.

```
1 package hilo;
2
3 public class
4
5     public
6     sup
7     }
8
9
10 // redefinición del método run()
11
12 // Este método es el que se va a ejecutar a la hora de
13 // iniciar tu proceso.
14
15 // Run define la actividad principal del thread.
16 public void run() {
17
18 }
19
20 }
21
22
```

SimpleHilo.java

proceso.java

```
1 package hilo;
2
3 public class proceso extends Thread{
4
5     public proceso(String str) {
6
7     }
8
9
10    //
11    //
12    //
13
14
15    // Run define la actividad principal del thread.
16    public void run() {
17
18        for(int i=0;i<10;i++)
19        {
20
21        }
22
23    }
24
25 }
26
27
```

Escriba 10 veces el nombre del *thread* creado.

Crea un for de 10.

SimpleHilo.java

proceso.java

Saca por pantalla el nombre del proceso. **getName** es el método que te permite mostrar el nombre de este proceso.

```
for(int i=0;i<10;i++)
{
    // Saca por pantalla el nombre del proceso.
    // getName es el metodo que te permite mostrar el nombre de este
    // proceso.
    System.out.println("Este es el thread : " + this.getName());
}
```

Ve a tu clase
SimpleHilo.

```
1 package hilo;
2
3 public class SimpleHilo {
4
5     public SimpleHilo() {
6         // thread se debe crear un objeto
7         // de la clase proceso que acabamos de contruir.
8
9         // Declaro una variable de la clase proceso.
10        proceso miHilo;
11
12        // Creo un objeto de la clase proceso.
13        // Le doy al hilo el nombre de Hilo1.
14        miHilo = new proceso("Hilo1");
15
16        // Llamo al método start(), heredado de la super clase thread,
17        // que se encarga de llamar al metodo run().
18        miHilo.start();
19    }
20
21 }
22
```

Haz clic en
ejecutar.

Package Explorer

▼ Hilo1
 ▼ src
 ▼ hilo
 > proceso.java
 > SimpleHilo.java
 > JRE System Library [JavaSE-1.8]

```
1 package hilo;  
2  
3 public class SimpleHilo {  
4  
5     public static void main(String[] args) {  
6         // Para poner en marcha este nuevo thread se debe crear un objeto  
7         // de la clase proceso que acabamos de contruir.  
8  
9         // Declaro una variable de la clase proceso.  
10        proceso miHilo;  
11  
12        // Creo un objeto de la clase proceso.  
13        // Le doy al hilo el nombre de Hilo1.  
14        miHilo = new proceso("Hilo1");  
15  
16        // Llamo al método start(), heredado de la super clase thread,  
17        // que se encarga de llamar al metodo run().  
18        miHilo.start();  
19    }  
20  
21 }  
22
```

