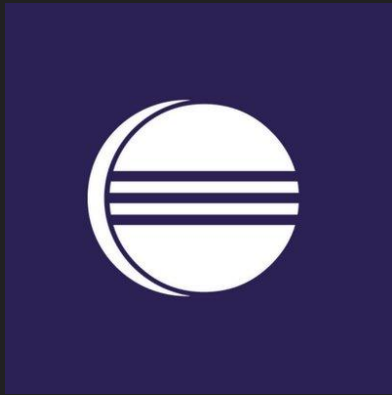


IDE Eclipse



Sergio González Guerra
Germán Alfonso Teixidó

Index

- Download and configure Eclipse
- Create Java elements
- IDE programming tools
- Eclipse views
- Ejecute and debug
- FAQ
- Bibliografía

1

DOWNLOAD AND CONFIGURE ECLIPSE

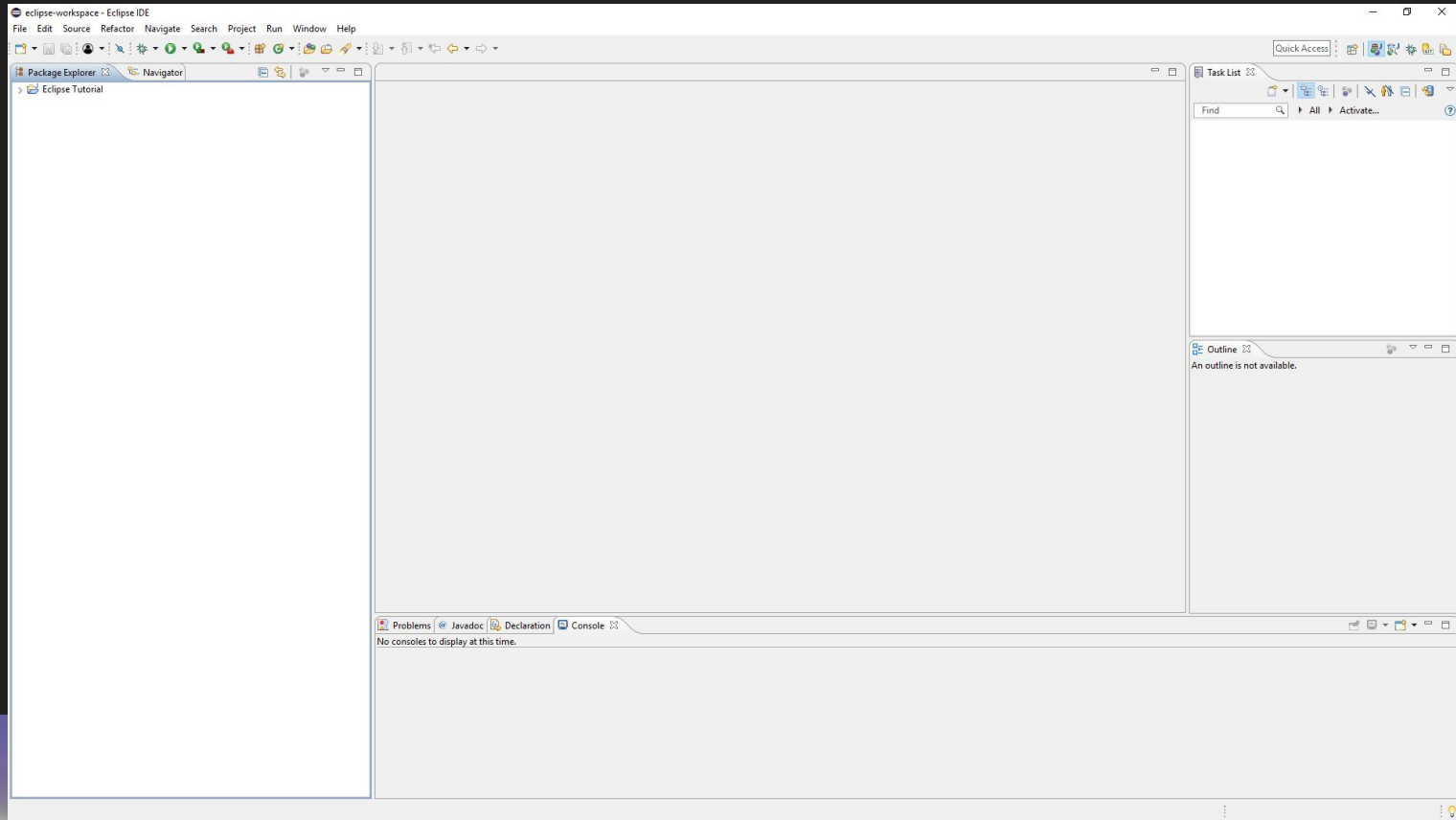
Where can we download Eclipse?



www.eclipse.org

How do we install Eclipse?

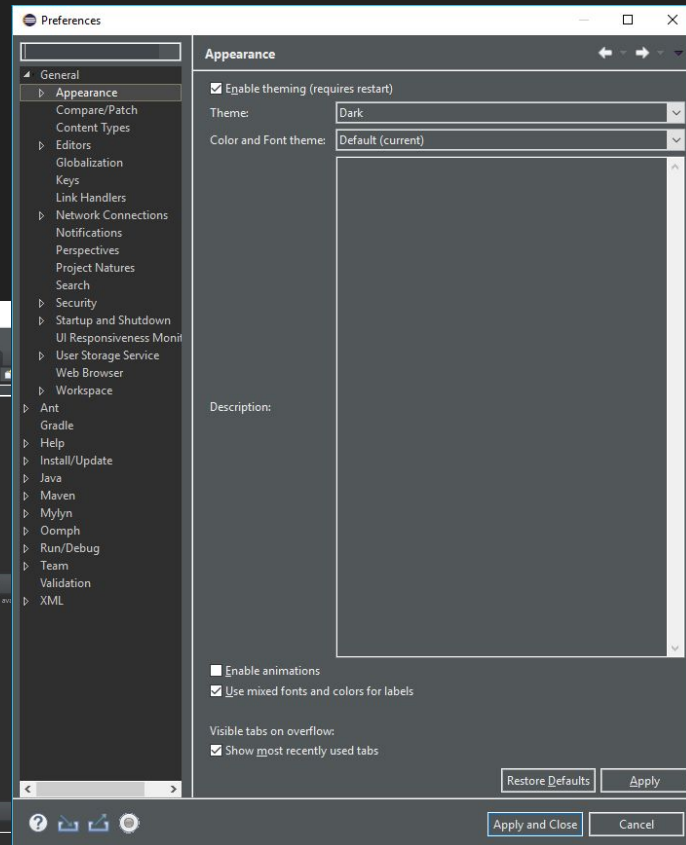
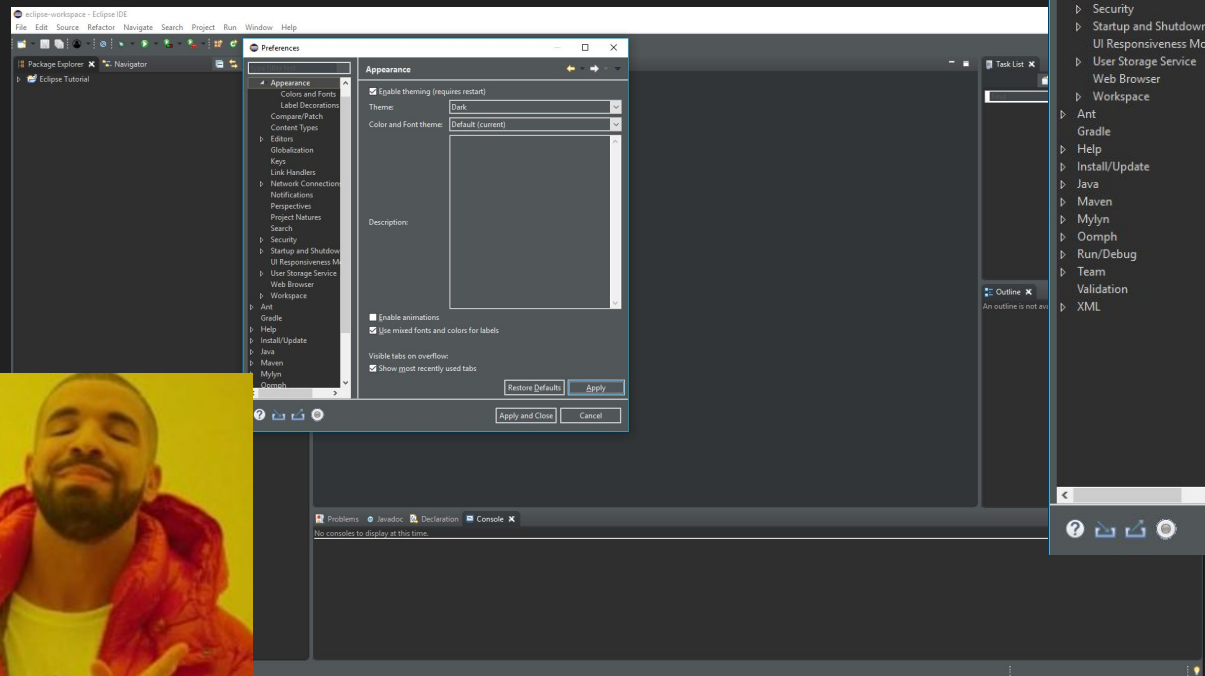
www.eclipse.org



Cambiando a modo oscuro

Window>Preferences

Appearance->Themes [Dark]



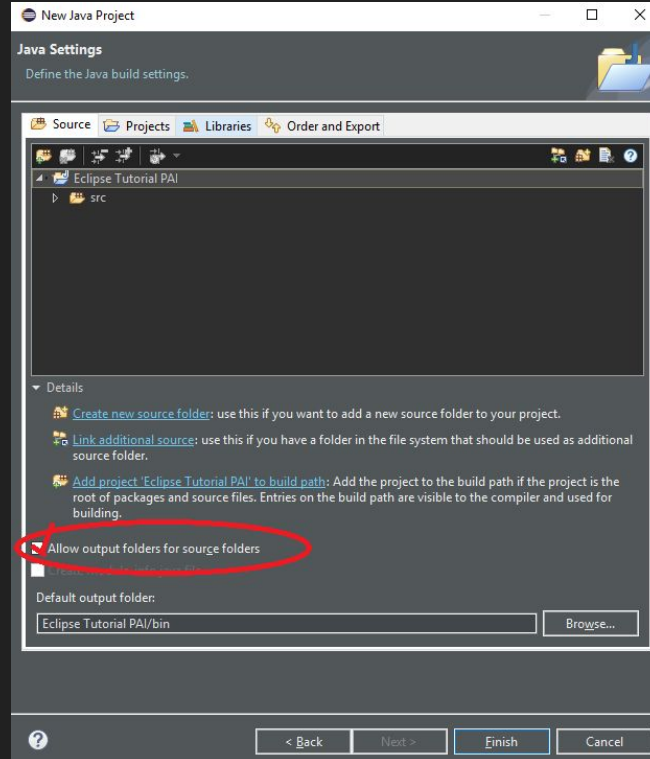
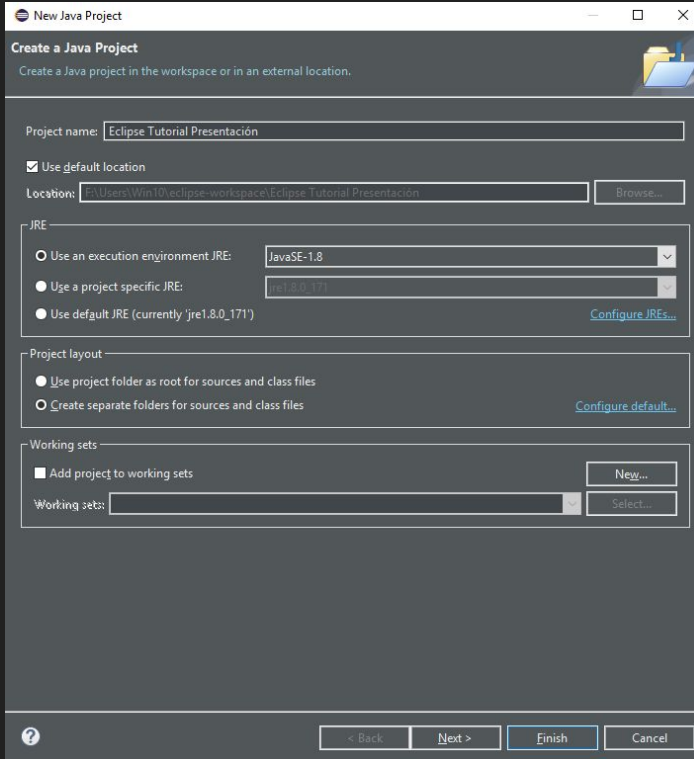
¡Recuerda aplicar
los cambios!

What kind of project can we do with Eclipse?

- Java Project
- Simple Project
- Plug-In Development Project

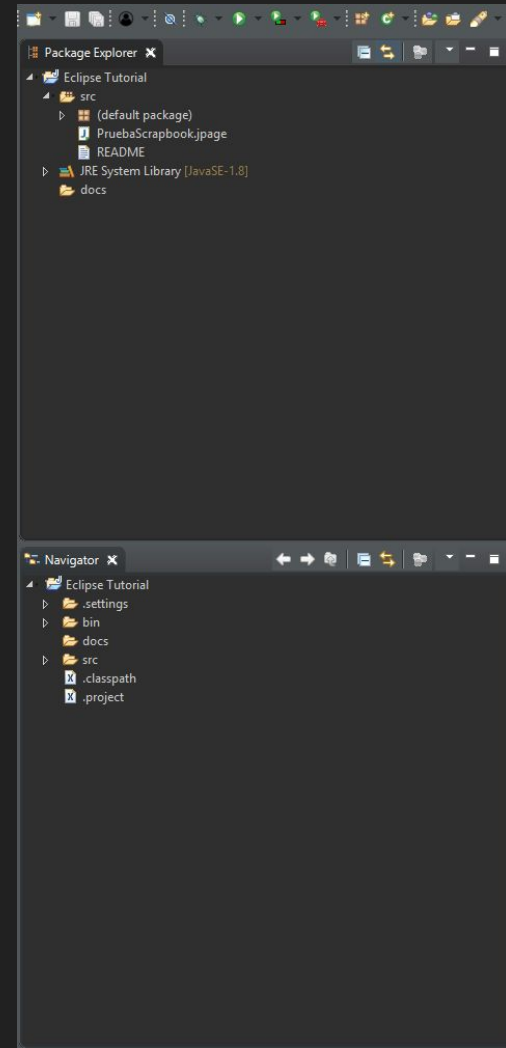
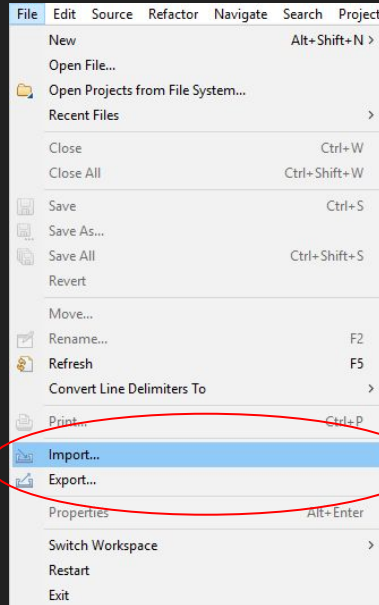
How to create a Java Project

File > New > Java Project



Import (and export) files with Eclipse?

- File > Import / Export
- Dragging files to the “Package Explorer”/”System Navigator”

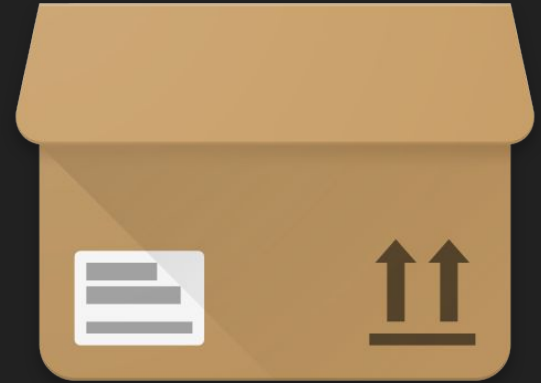


2

CREATE JAVA ELEMENTS

Add Packages

- Store and organize Java files
- Stored in /src
- Contains several parts separated by points
- Each point represent a directory
- Right click -> New -> Package



Java Package

 Discouraged package name. By convention, package names usually start with a lowercase letter



Creates folders corresponding to packages.

Source folder: HelloWorld/src

Browse...

Name: Hello_package

☐ Create package-info.java



Finish

Cancel

Java Class

- '.java' files -> Compiled on '.class' files.
- Stored in /src .
- Right click -> New -> Class, to create a new class.



New Java Class

Java Class

⚠ This package name is discouraged. By convention, package names usually start with a lowercase letter

Source folder: HelloWorld/src Browse...

Package: Hello_package Browse...

Enclosing type: Browse...

Name: HelloWorld

Modifiers: ☐ public ☒ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)

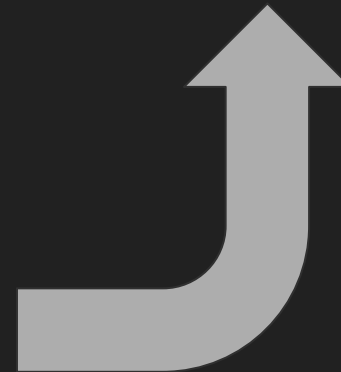
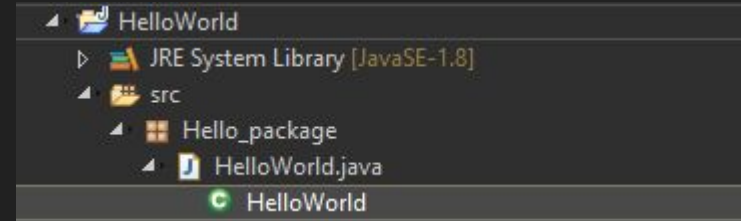
☐ Constructors from superclass

☒ Inherited abstract methods

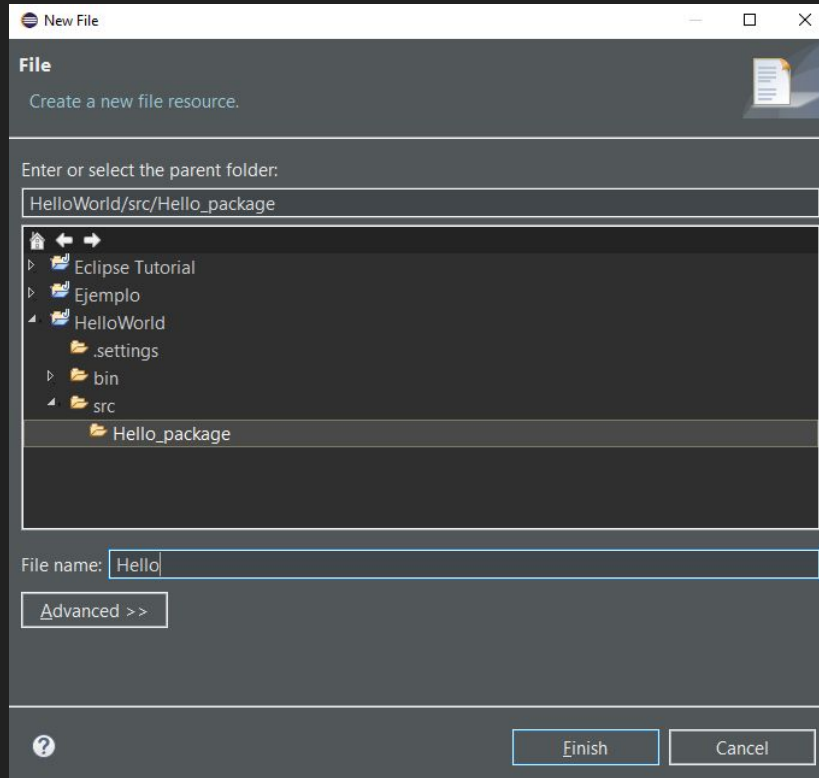
Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

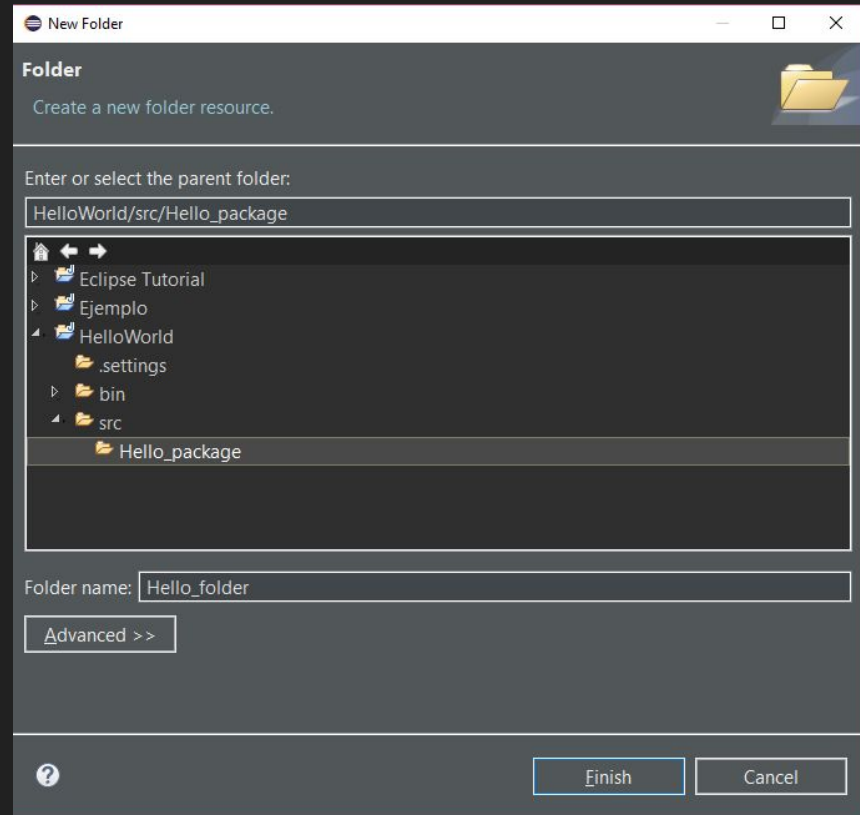
Finish Cancel



Files



Folder




Interfaces

- Lack of implementation, other classes implement them.
- Tells to the implemented class, what to do.
- Can extend to another interface by inheritance.



Java Interface



 This package name is discouraged. By convention, package names usually start with a lowercase letter

Source folder:

HelloWorld/src

Browse...

Package:

Hello_package

Browse...

☐ Enclosing type:

Browse...

Name:

Hello_interface

Modifiers:

☐ public☒ package☐ private☐ protected

Extended interfaces:

Add...

Remove

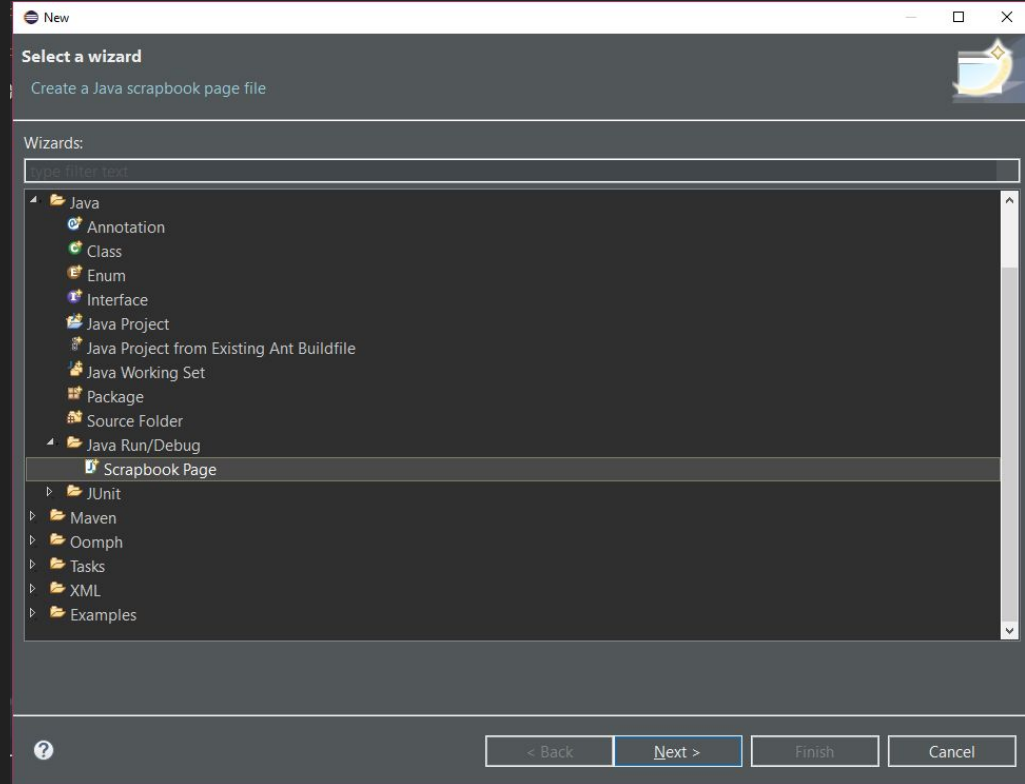
Do you want to add comments? (Configure templates and default value [here](#))☐ Generate comments

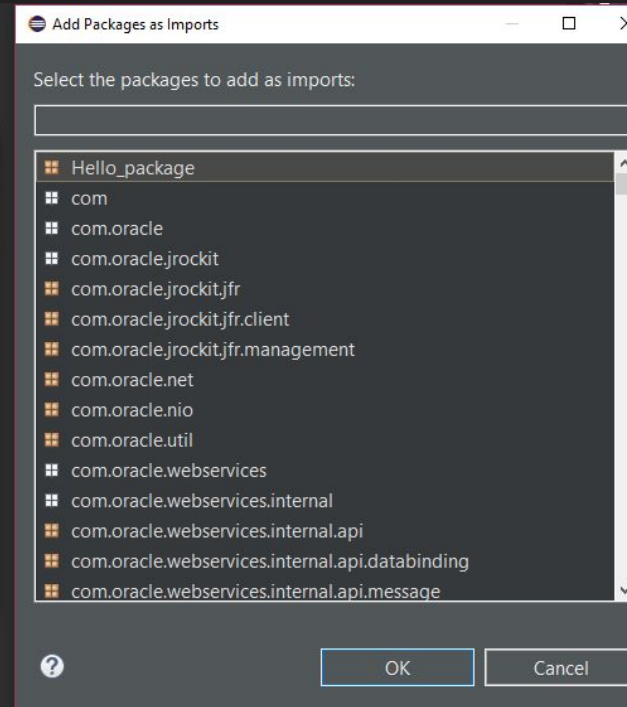
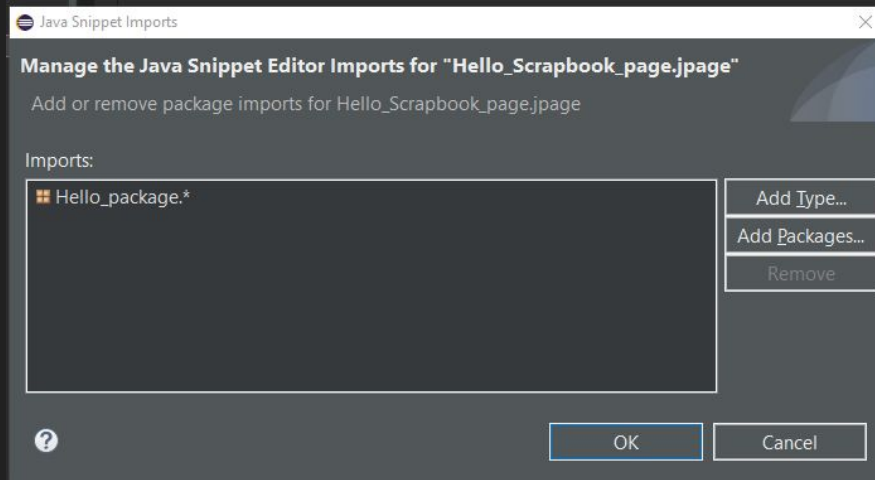
Finish

Cancel

Scrapbook Page

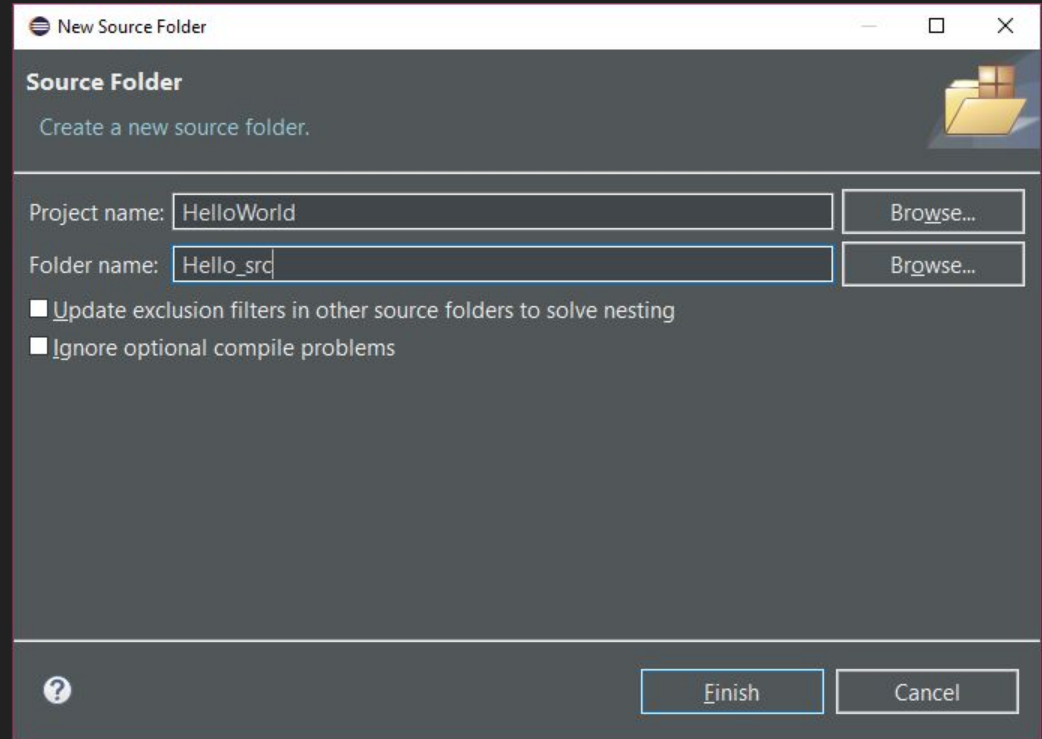
- Test code fragments.
- For creating a scrapbook page, Right click -> New -> Other..., or Ctrl+N
- For importing necessary class, Right click -> Set imports...





Source Folder

- Intended to store .java source files.
- Automatically compiled in .class



3

IDE'S PROGRAMMING TOOLS

Debugger and compiler

The screenshot shows an IDE with two tabs: `PruebaScrapbook.jpj` and `ExampleClassTest.java`. The `ExampleClassTest.java` file contains the following code:

```
1 public class ExampleClassTest {
2
3
4     int col;
5
6     public static void main(String[] args) {
7         System.out.println("¡Hola, Eclipse!");
8     }
9
10 }
11
12
13
14 /**
15  We're no strangers to love
16  You know the rules and so do I
17  A full commitment's what I'm thinking of
18  You wouldn't get this from any other guy
19  I just wanna tell you how I'm feeling
20  Gotta make you understand
21  Never gonna give you up
22  Never gonna let you down
23  Never gonna run around and desert you
24  Never gonna make you cry
25  Never gonna say goodbye
26  Never gonna tell a lie and hurt you
27  We've known each other for so long
28  Your heart's been aching but you're too shy
29  Inside we both know what's been going on
30  We know the game and we're gonna play it
31  And if you ask me how I'm feeling
32  Don't tell me you're too blind to see
33  Never gonna give you up
34  Never gonna let you down
35  Never gonna run around and desert you
36  Never gonna make you cry
37  Never gonna say goodbye
38  We're no strangers to love
39  You know the rules and so do I
40  A full commitment's what I'm thinking of
41  You wouldn't get this from any other guy
42  I just wanna tell you how I'm feeling
43  Gotta make you understand
44  Never gonna give you up
45  Never gonna let you down
46  Never gonna run around and desert you
```

The `main` method on line 6 has a red squiggly line under the `out` property access, with a tooltip that says: `out cannot be resolved or is not a field`. A white arrow points to this error.

Below the code editor, the `Problems` tab is active, showing 1 error. The error details are as follows:

Description	Resource	Path	Location	Type
✖ Errors (1 item)				

Autocomplete

Choose a lazy person to do a hard job,
because that person will find a easy way to do it.

Be clever, be lazy.

CTRL + [SPACEBAR]



Class Names

```
9
10 public static void main(String[] args) {
11
12     Obj
13
14     Syst
15 }
16
17 public s
18
19
20
21
22
23
24
25
26
27 }
28
29
30
31 /**
32  We're no strangers to love
33  You know the rules and so do I
34  A full commitment's what I'm thinking of
35  You wouldn't get this from any other guy
36  I just wanna tell you how I'm feeling
37  Gonna make you understand
38  Never gonna give you up
39  Never gonna let you down
40  Never gonna run around and desert you
41  Never gonna make you cry
42  Never gonna say goodbye
43  Never gonna tell a lie and hurt you
44  We've known each other for so long
45  Your heart's been aching but you're too shy to say it
46  Inside we both know what's been going on
```

Object - java.lang

ObjectInput - java.io

ObjDoubleConsumer - java.util.function

Object - org.omg.CORBA

ObjectAlreadyActive - org.omg.PortableServer.POAPacka

ObjectAlreadyActiveHelper - org.omg.PortableServer.PO

ObjectChangeListener - javax.naming.event

ObjectFactory - javax.naming.spi

ObjectFactoryBuilder - javax.naming.spi

ObjectHelper - org.omg.CORBA

ObjectHolder - org.omg.CORBA

Press 'Ctrl+Space' to show Java Non-Type Proposals

Class Object is the root of the class hierarchy. Every class has Object as a superclass. All objects, including arrays, implement the methods of this class.

Since: JDK1.0

See Also: [Class](#)

Press 'Tab' from proposal table or click for focus

Local variables & attributes

```
public static void main(String[] args) {
```

```
    intP
```

```
        intPrivate: int - ExampleClassTest
```

```
        intPublic: int - ExampleClassTest
```

```
    } System.out.println("IntPredicate - java.util.function
```

```
}
```

```
public static
```

```
}
```

Press 'Ctrl+Space' to show Java Non-Type Proposals

```
**
```

```
We're no strangers to love
```

```
You know the rules and so do I
```

```
A full commitment's what I'm thinking of
```

```
You wouldn't get this from any other guy
```

```
I just wanna tell you how I'm feeling
```

```
Wotta make you understand
```

```
I never gonna give you up
```

```
I never gonna let you down
```

```
I never gonna run around and desert you
```

```
I never gonna make you cry
```

```
I never gonna say goodbye
```

```
I never gonna tell a lie and hurt you
```

```
We've known each other for so long
```

```
Your heart's been aching but you're too shy to say it
```

```
Inside we both know what's been going on
```

Methods & getters and setters

The screenshot shows an IDE with two panes. The left pane displays the source code of the `ExampleClassTest` class, which contains a `main` method that creates an `Integer` object and calls `byteValue()`. The right pane shows the Java documentation for the `Integer` class, specifically the `byteValue()` method. The documentation includes the method signature, a description of its return value, and a reference to the Java Language Specification.

```

1 public class ExampleClassTest {
2
3     public static void main(String[] args) {
4
5         Integer i= new Integer(1);
6         byte b= i.byteValue();
7     }
8 }
9
10 /**
11  * We're no strangers
12  * You know the rules
13  * A full commitment'
14  * You wouldn't get this from any other guy
15  * I just wanna tell you how I'm feeling
16  * Gotta make you understand
17  * Never gonna give you up
18  * Never gonna let you down
19  * Never gonna run around and desert you
20  * Never gonna make you cry
21  * Never gonna say goodbye
22  * Never gonna tell a lie and hurt you
23  * We've known each other for so long
24  * Your heart's been aching but you're too shy to say it
25  * Inside we both know what's been going on
26  * We know the game and we're gonna play it
27  * And if you ask me how I'm feeling
28  * Don't tell me you're too blind to see
29  * Never gonna give you up
30  * Never gonna let you down
31  * Never gonna run around and desert you
32  * Never gonna make you cry
33  * Never gonna say goodbye
34  * We're no strangers to love
35  * You know the rules and so do I

```

Integer

byteValue(): byte - Integer

compareTo(Integer anotherInteger): int - Integer

doubleValue(): double - Integer

equals(Object obj): boolean - Integer

floatValue(): float - Integer

getClass(): Class<?> - Object

hashCode(): int - Integer

intValue(): int - Integer

longValue(): long - Integer

shortValue(): short - Integer

toString(): String - Integer

notify(): void - Object

byteValue()

public byte byteValue()

Returns the value of this Integer as a byte after a narrowing primitive conversion.

Overrides:

[byteValue](#) in class [Number](#)

Returns:

the numeric value represented by this object after conversion to type byte.

See The Java™ Language Specification:

5.1.3 Narrowing Primitive Conversions

Press 'Ctrl+Space' to show Java Non-Type Proposals

Press 'Tab' from proposal table or click for focus

The screenshot shows an IDE with a Java file named 'ExampleClassTest.java'. The code is as follows:

```

1
2
3 public class ExampleClassTest {
4
5     public static void main(String[] args) {
6
7         Integer i= new Integer;
8
9     }
10
11
12
13
14
15
16 }
17
18
19
20 /**
21  We're no strangers to love
22  You know the rules and so do I
23  A full commitment's what I'm
24  You wouldn't get this from any other guy
25  I just wanna tell you how I'm feeling
26  Gotta make you understand
27  Never gonna give you up
28  Never gonna let you down
29  Never gonna run around and desert you
30  Never gonna make you cry
31  Never gonna say goodbye
32  Never gonna tell a lie and hurt you
33  We've known each other for so long
34  Your heart's been aching but you're too shy to say it
35  Inside we both know what's been going on
36  We know the game and we're gonna play it
37  And if you ask me how I'm feeling
38  Don't tell me you're too blind to see
39  Never gonna give you up
40  Never gonna let you down
41  Never gonna run around and desert you
42  Never gonna make you cry
43  Never gonna say goodbye
44  We're no strangers to love
45  You know the rules and so do I
46  A full commitment's what I'm thinking of

```

A tooltip is visible over the 'Integer' type, listing several constructors:

- Integer(int value) - java.lang.Integer
- Integer(String s) - java.lang.Integer
- IntegerSyntax(int arg0) Anonymous Inner Type - javax.print
- IntegerSyntax(int arg0, int arg1, int arg2) Anonymous Inner

The right sidebar shows the 'Integer' class documentation:

Integer

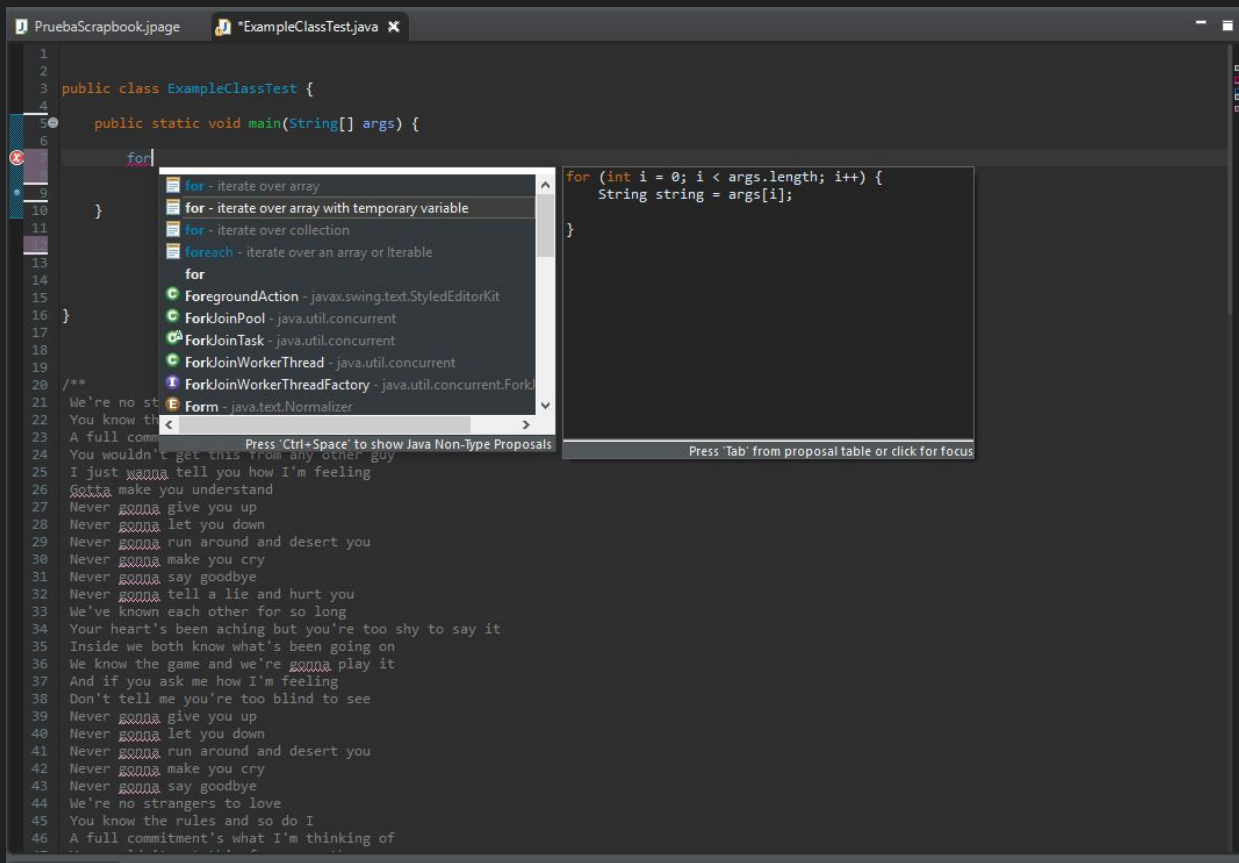
public Integer(int value)

Constructs a newly allocated Integer object that represents the specified int value.

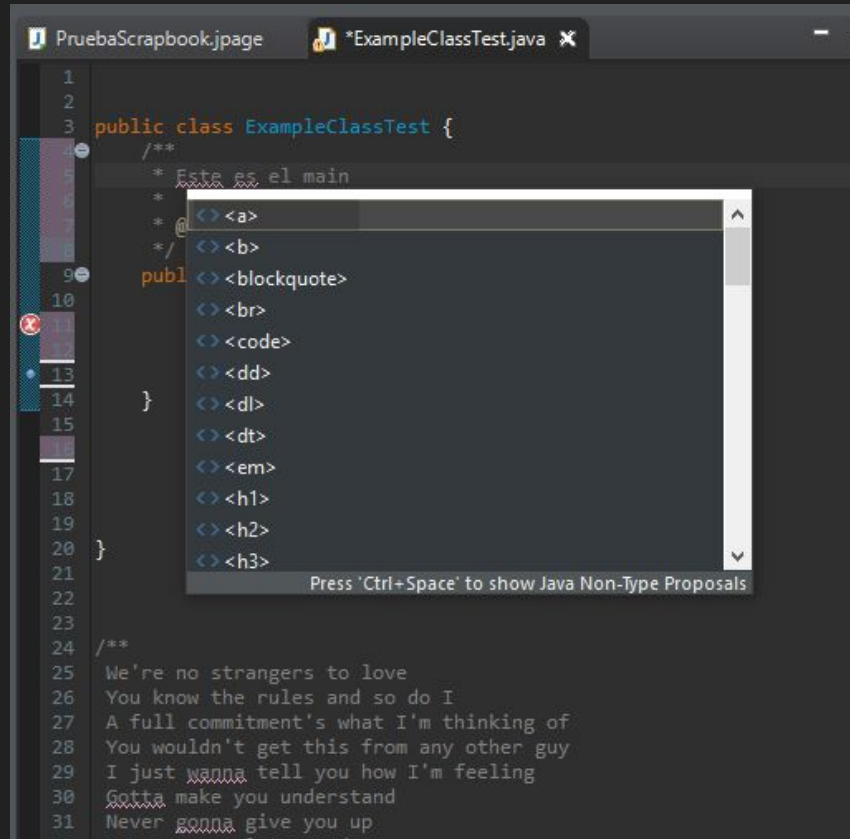
Parameters:

value - the value to be represented by the Integer object.

Loops



Javadoc tags

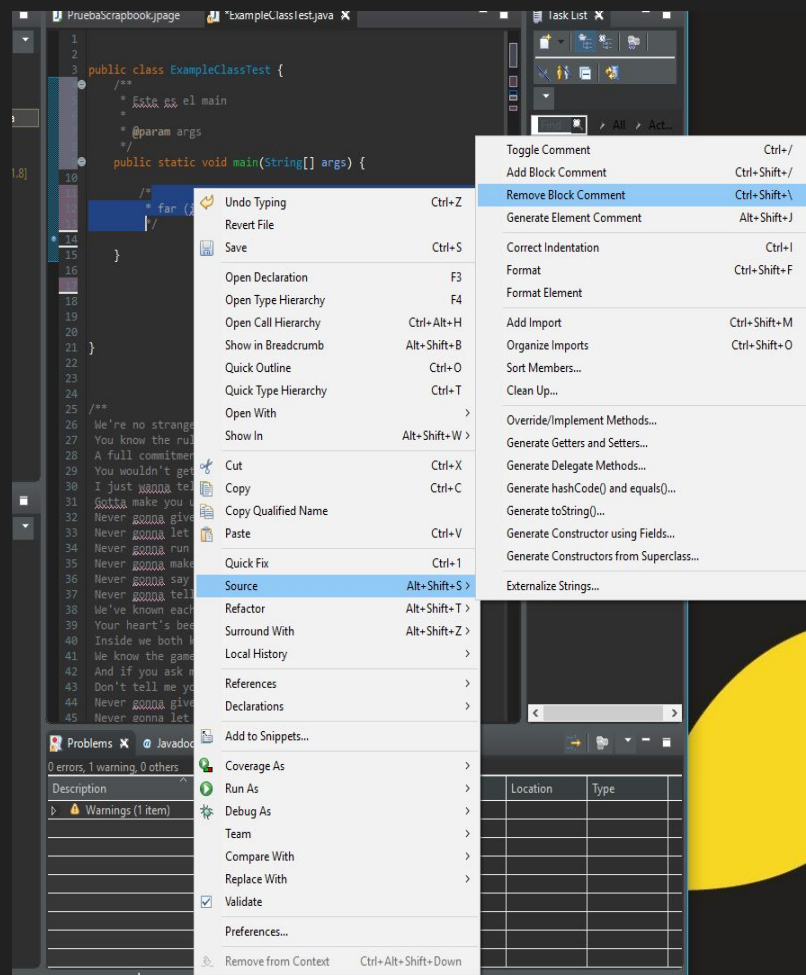


Source Menu

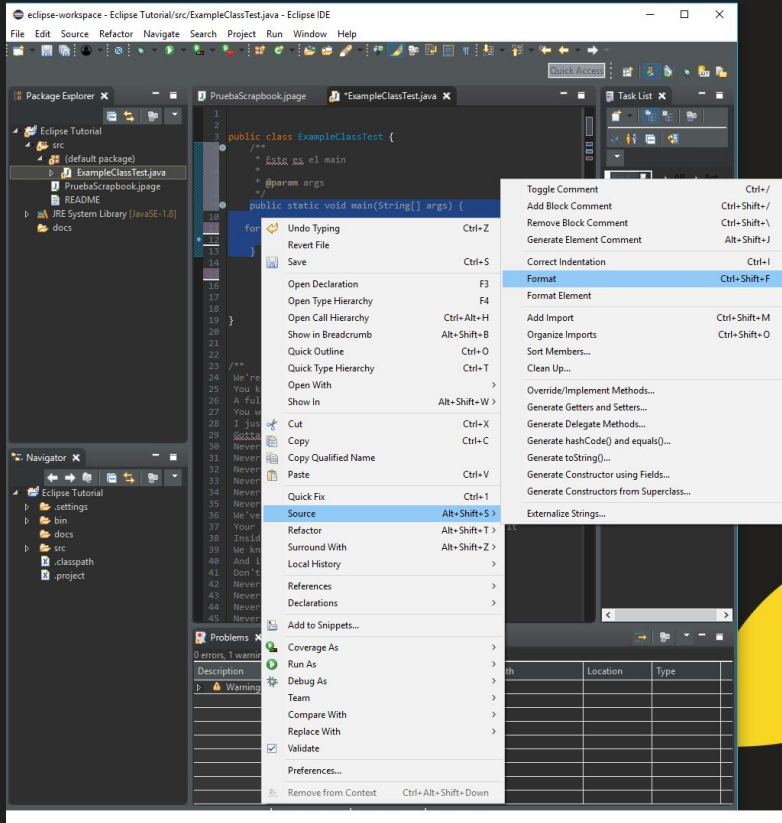
Do you have any idea how fast i am?

Comment/uncomment CTRL+/

```
//      far (int i = 0; i < args.length; i++) {  
//          //cosas  
//      }
```



Format

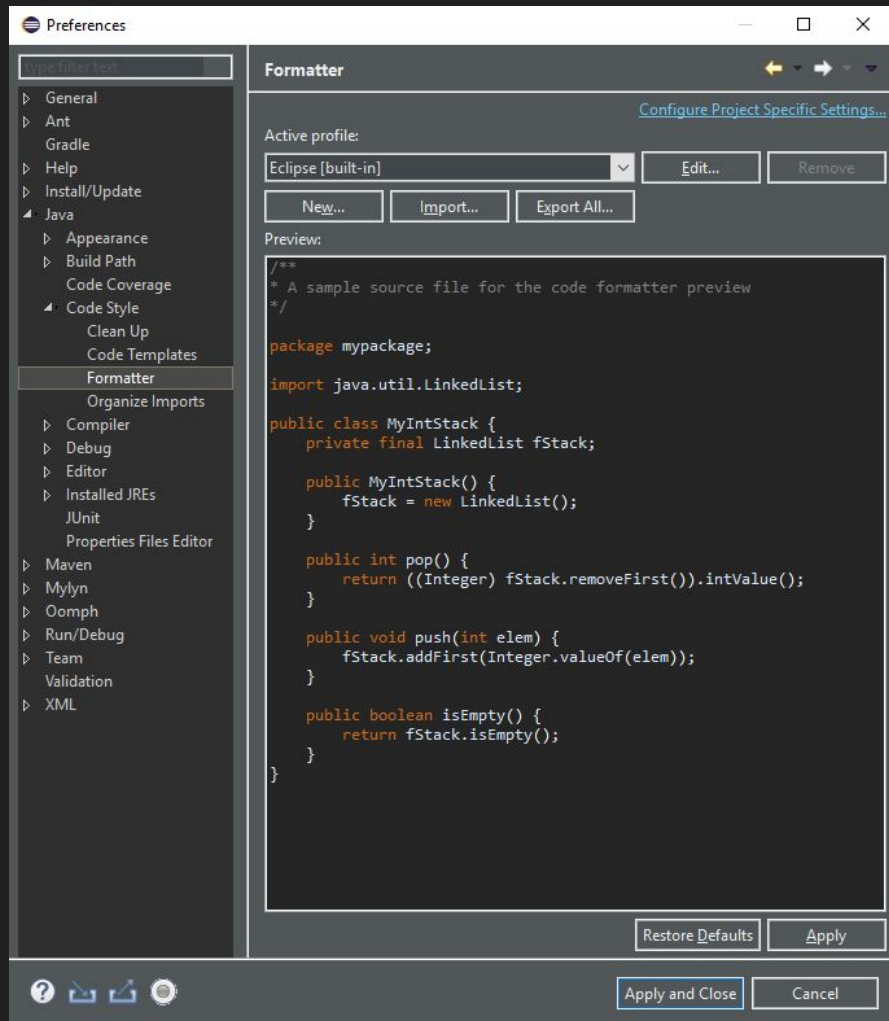


```
public static void main(String[] args) {  
  
    for (int i = 0; i < args.length; i++) {  
        /* cosas */  
    }  
  
}
```

CTRL+SHIFT+F

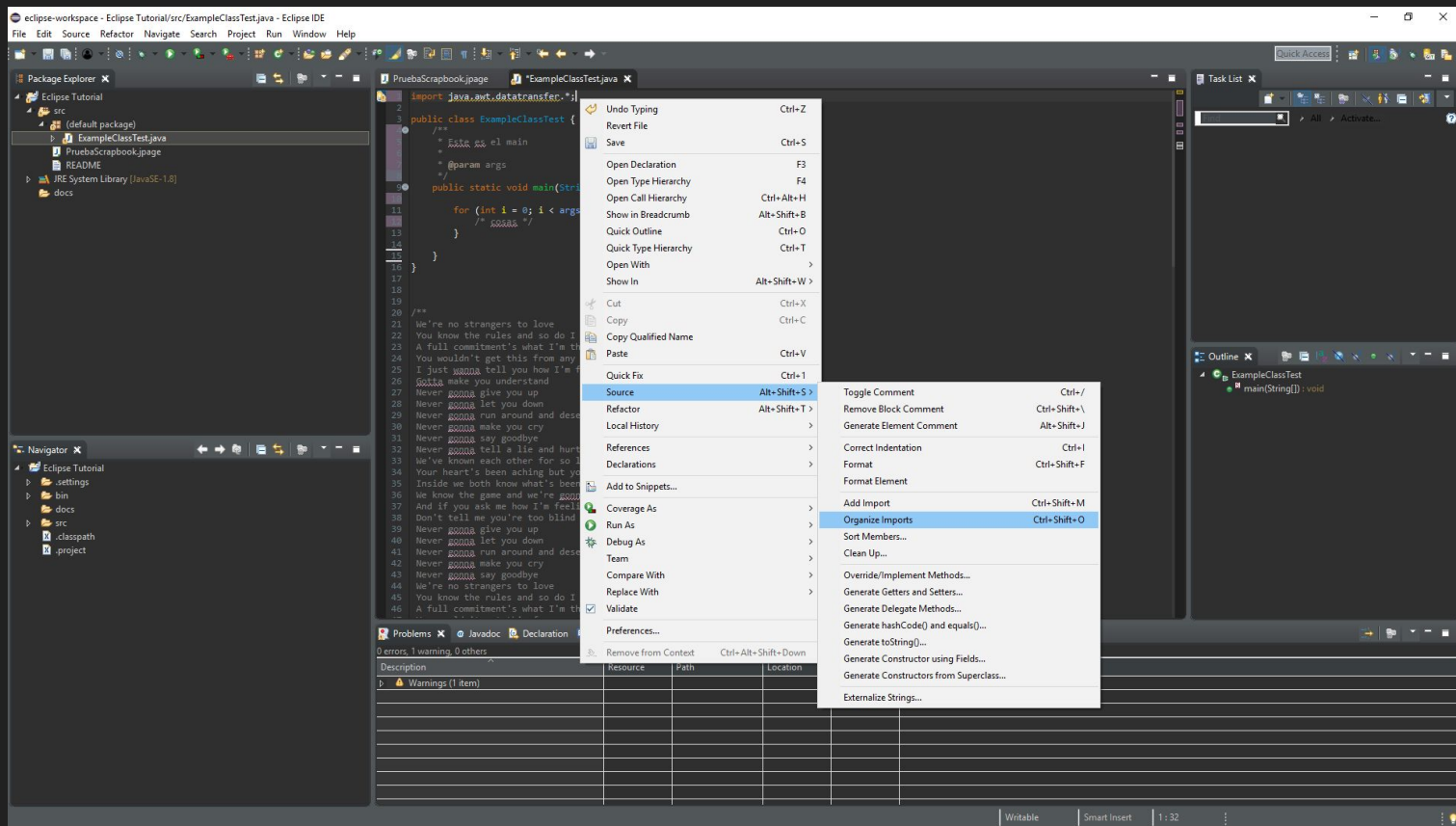
Edit format

Window> Preferences
Java> Code Style> Formatter



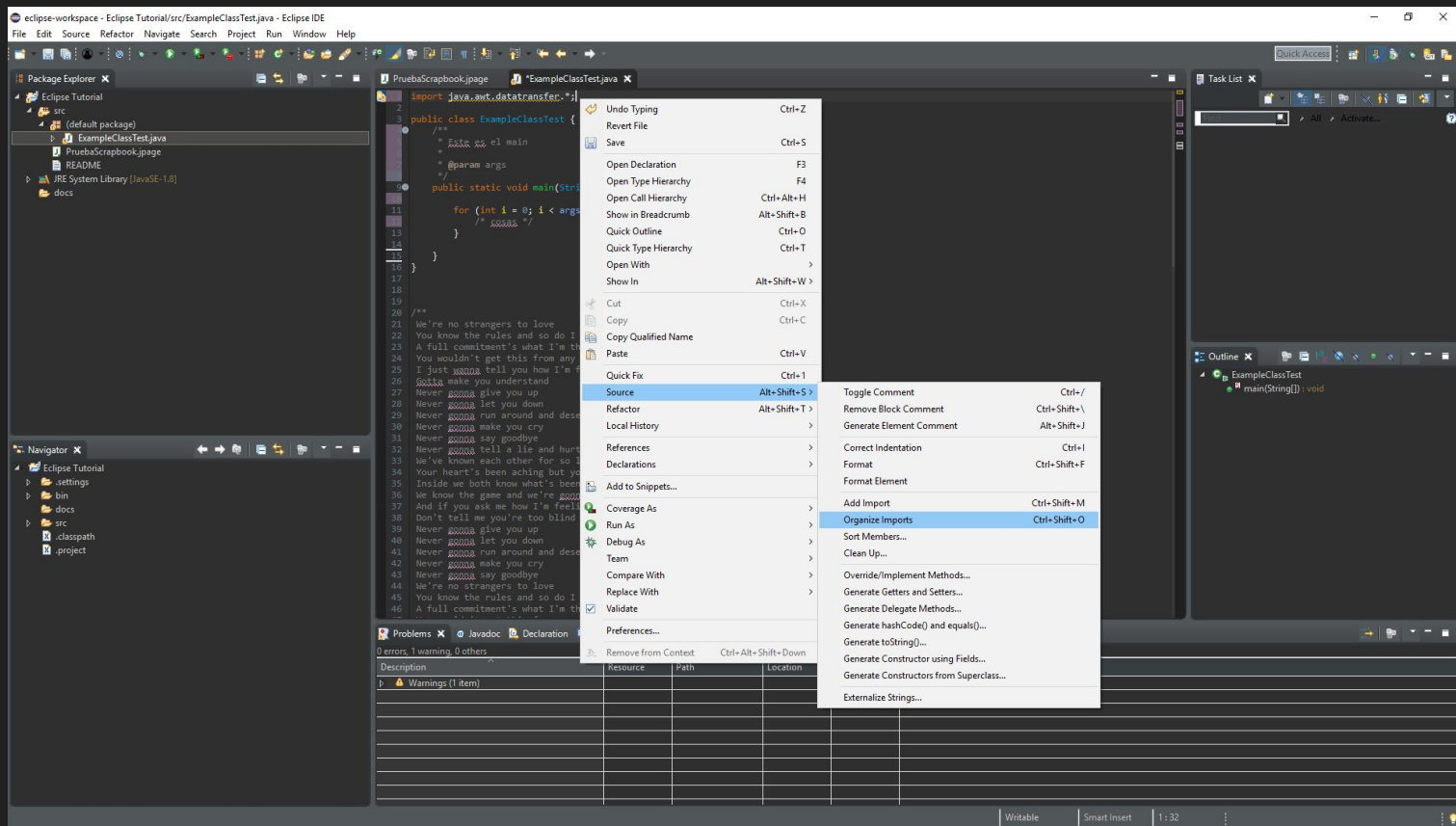
Organize imports

CTRL+SHIFT+O

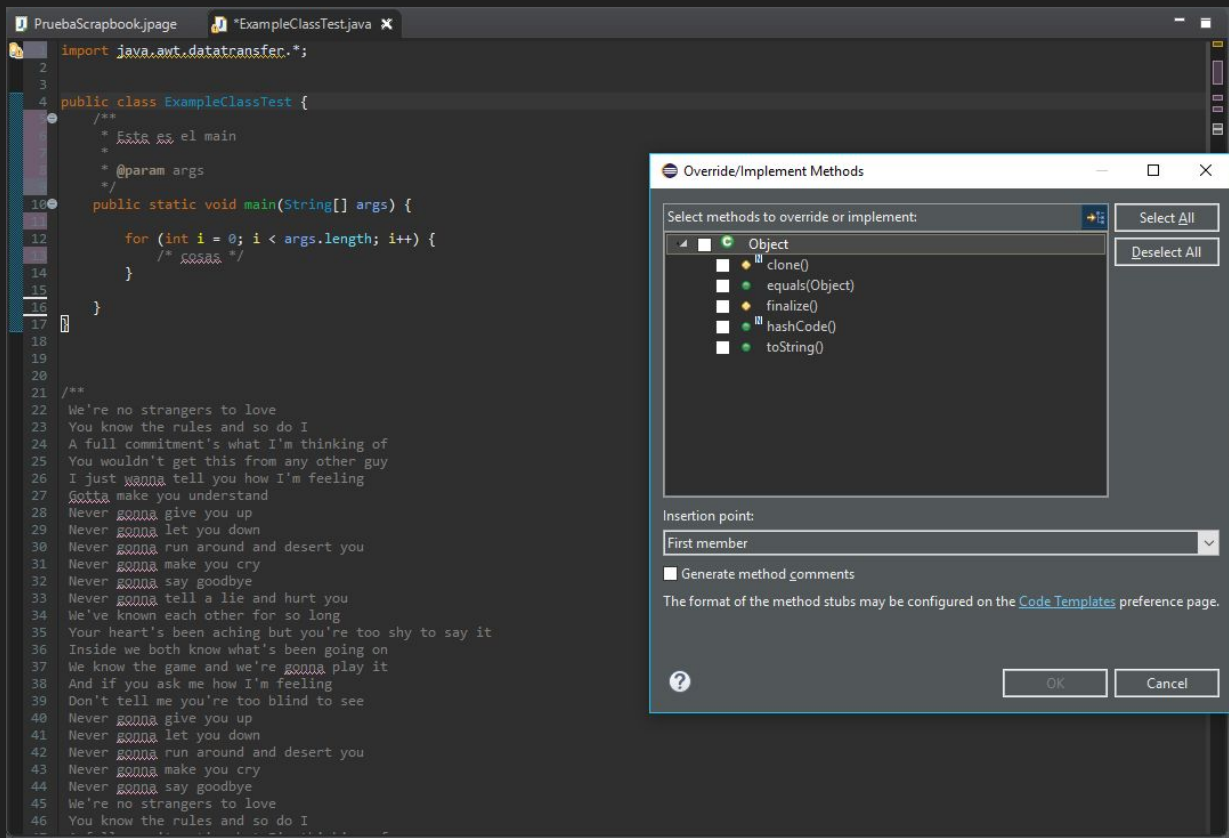


Add imports

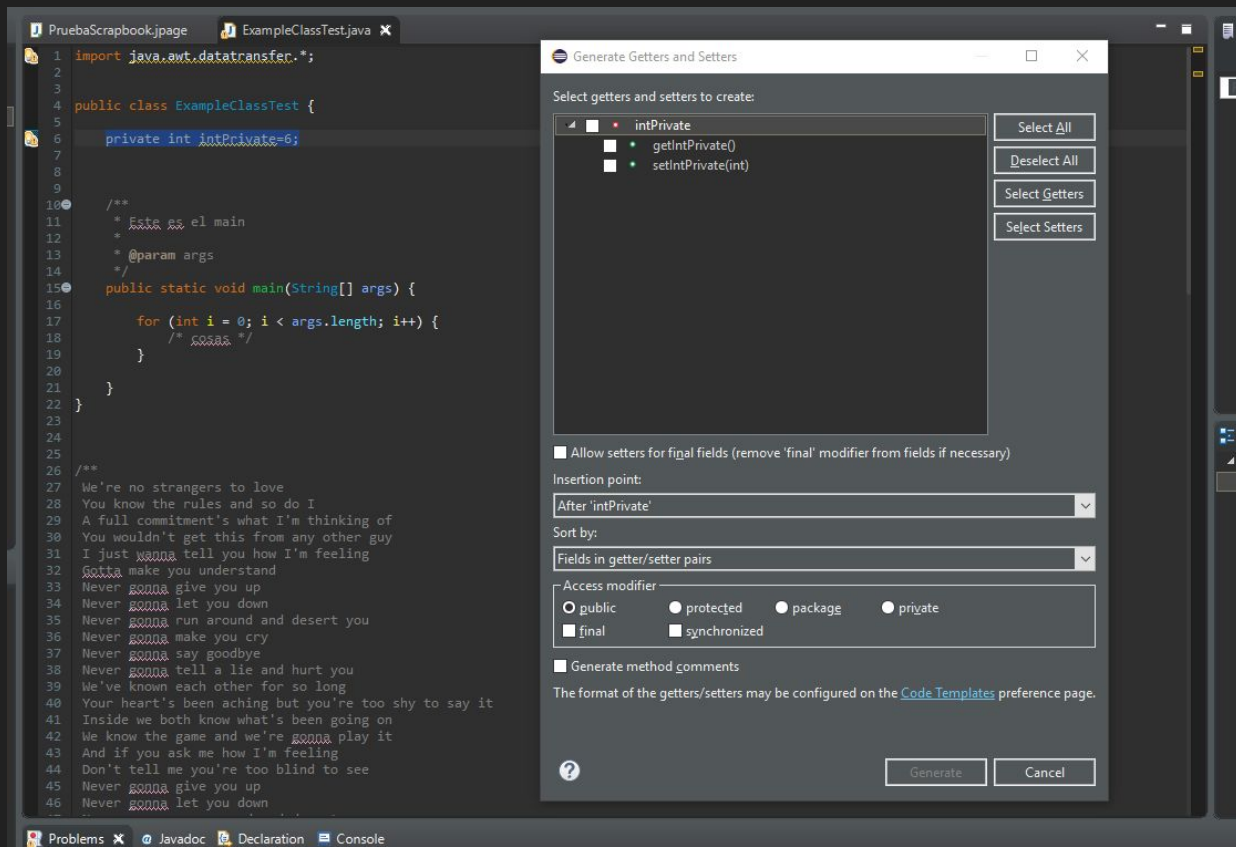
CTRL+SHIFT+M



Override/Implement Methods



Generate Getters/Setters

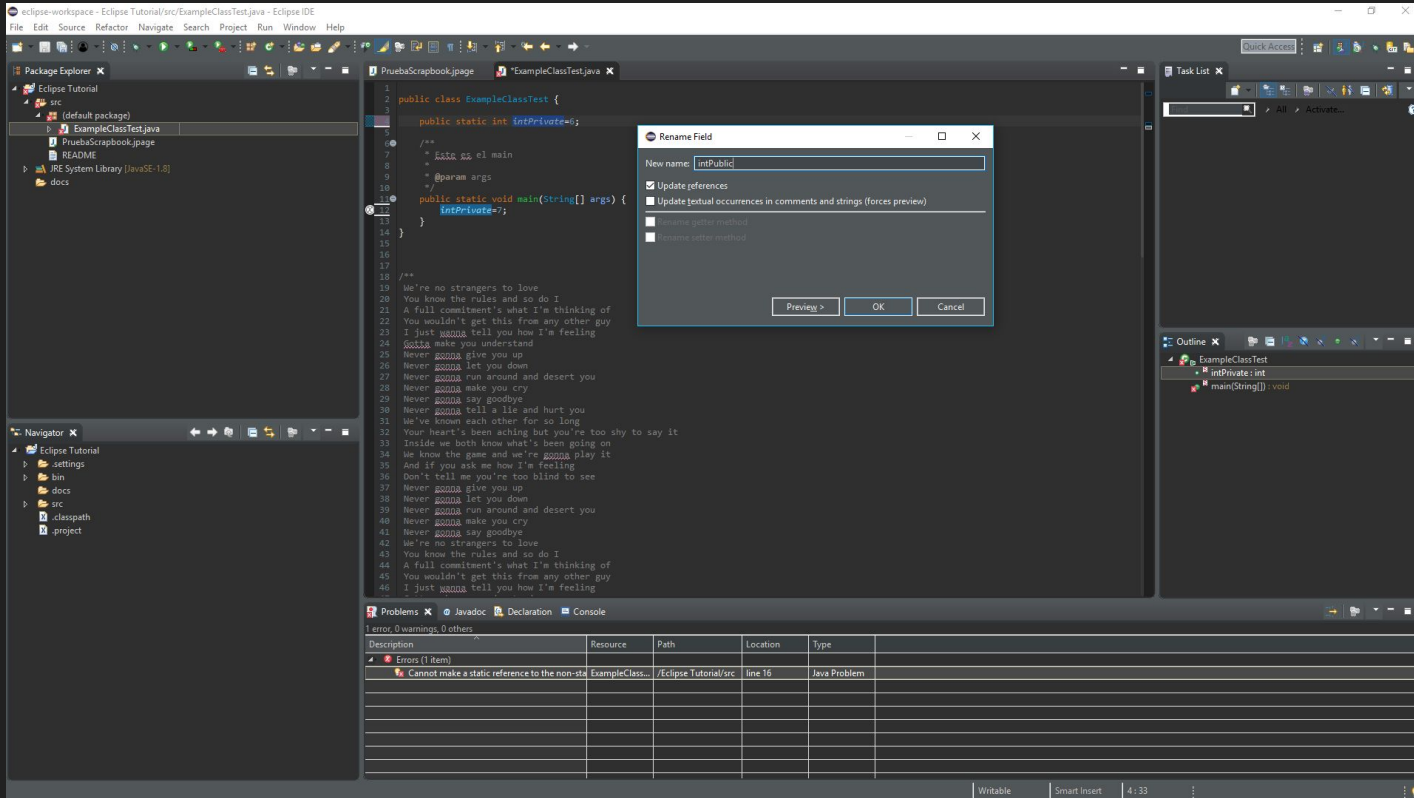


Refactor Menu

Don't use this to copy anyone's code. It won't work.

Rename

ALT+SHIFT+R



Another interesting stuff we can do with the Refactor Menu:

- Move (folders)
- Change Method Signature
- Pull Up and Push Down

Documentation

No, your code isn't self explanatory, moron.

External Javadoc doc.

- Use Mayus + F2 to check Javadoc documentation for anything.
- Javadoc has also a Eclipse view:
Window > Show View... > Java > Javadoc
- We can import JAR files on Properties > JBP -> Libraries -> +Add External JARs.

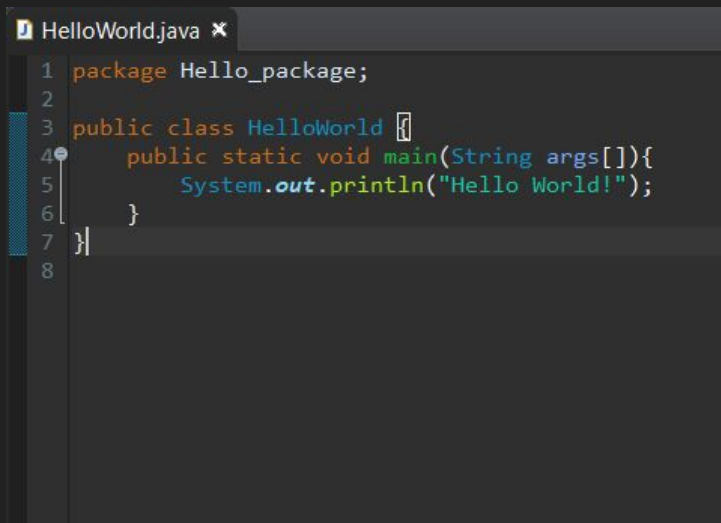
4

ECLIPSE VIEWS

Views of Eclipse

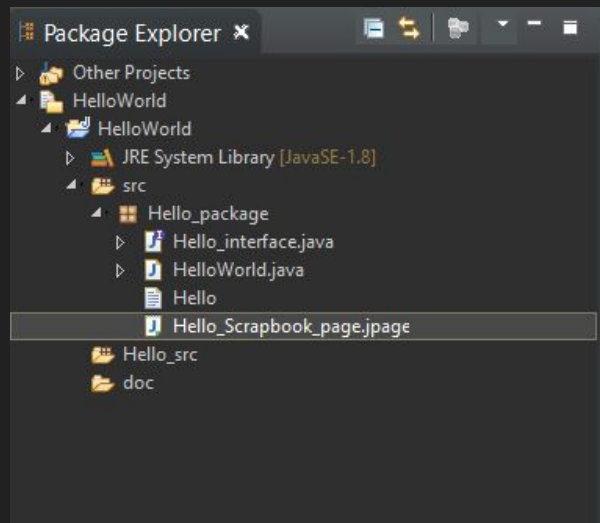
The user interface of Eclipse consists of two types:

- Editors -> Perform a complete Task
- Views -> Support functions



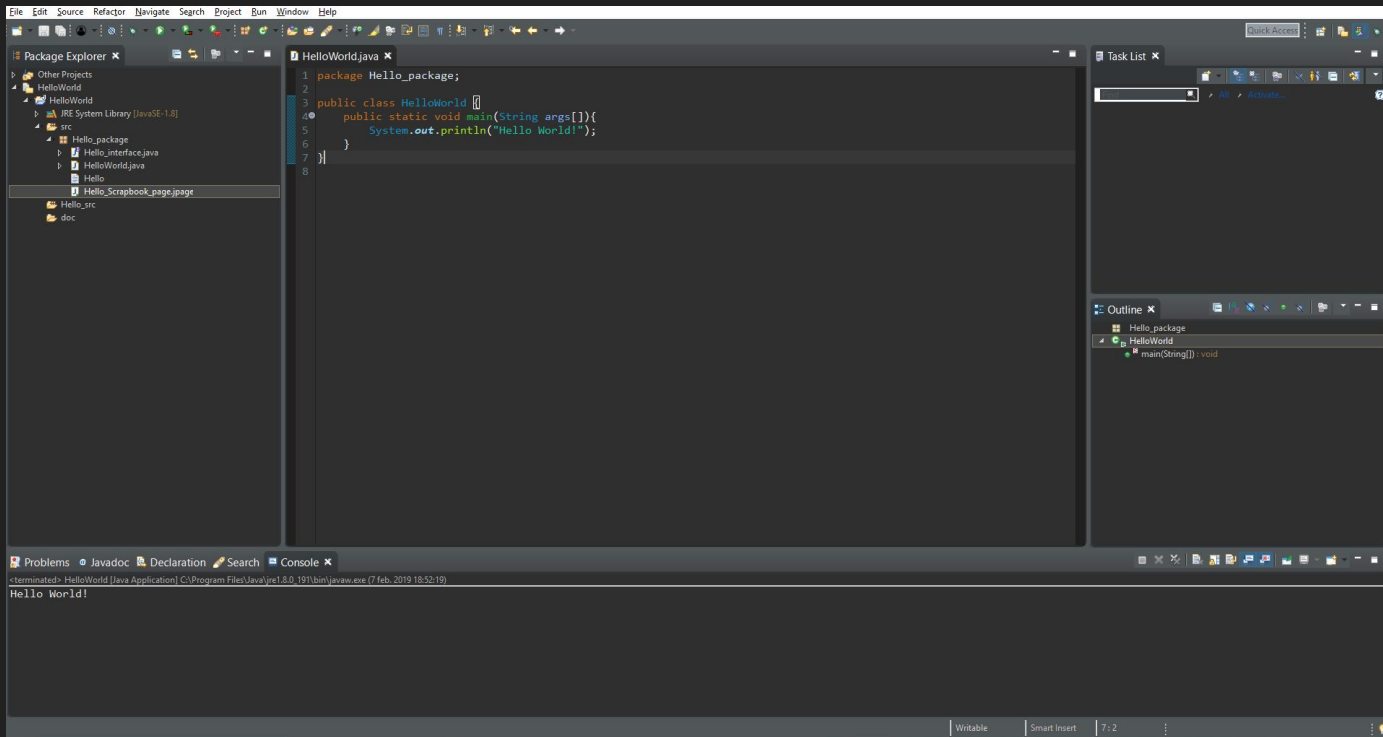
A screenshot of the Eclipse IDE's editor window. The title bar shows 'HelloWorld.java' with a close button. The editor contains the following Java code:

```
1 package Hello_package;  
2  
3 public class HelloWorld {  
4     public static void main(String args[]){  
5         System.out.println("Hello World!");  
6     }  
7 }  
8
```

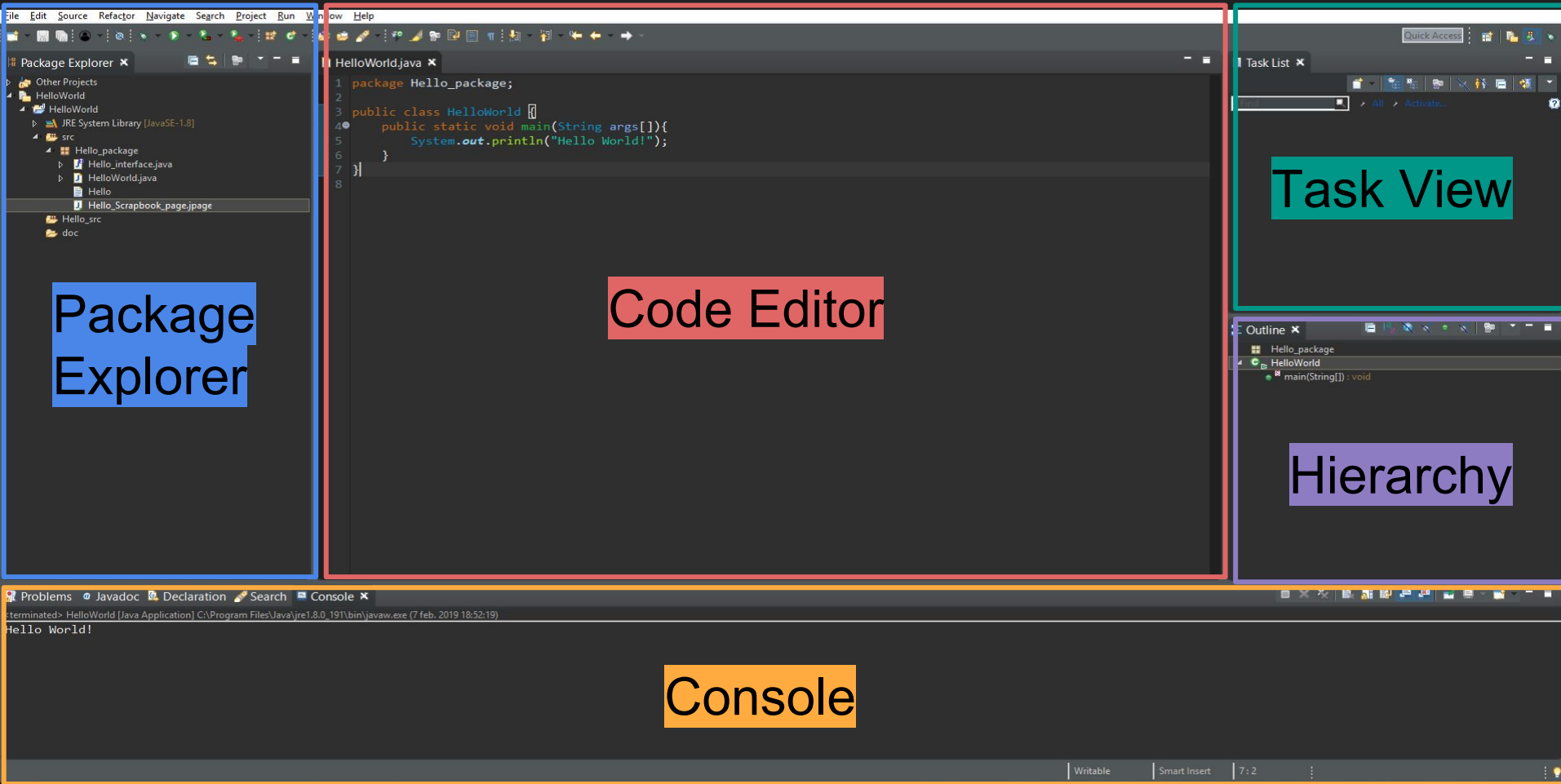


Perspectives

- Set of editors and views to support an activity



Java



Resource

Outline

Code Editor

Task View

[illegible]

Debugger

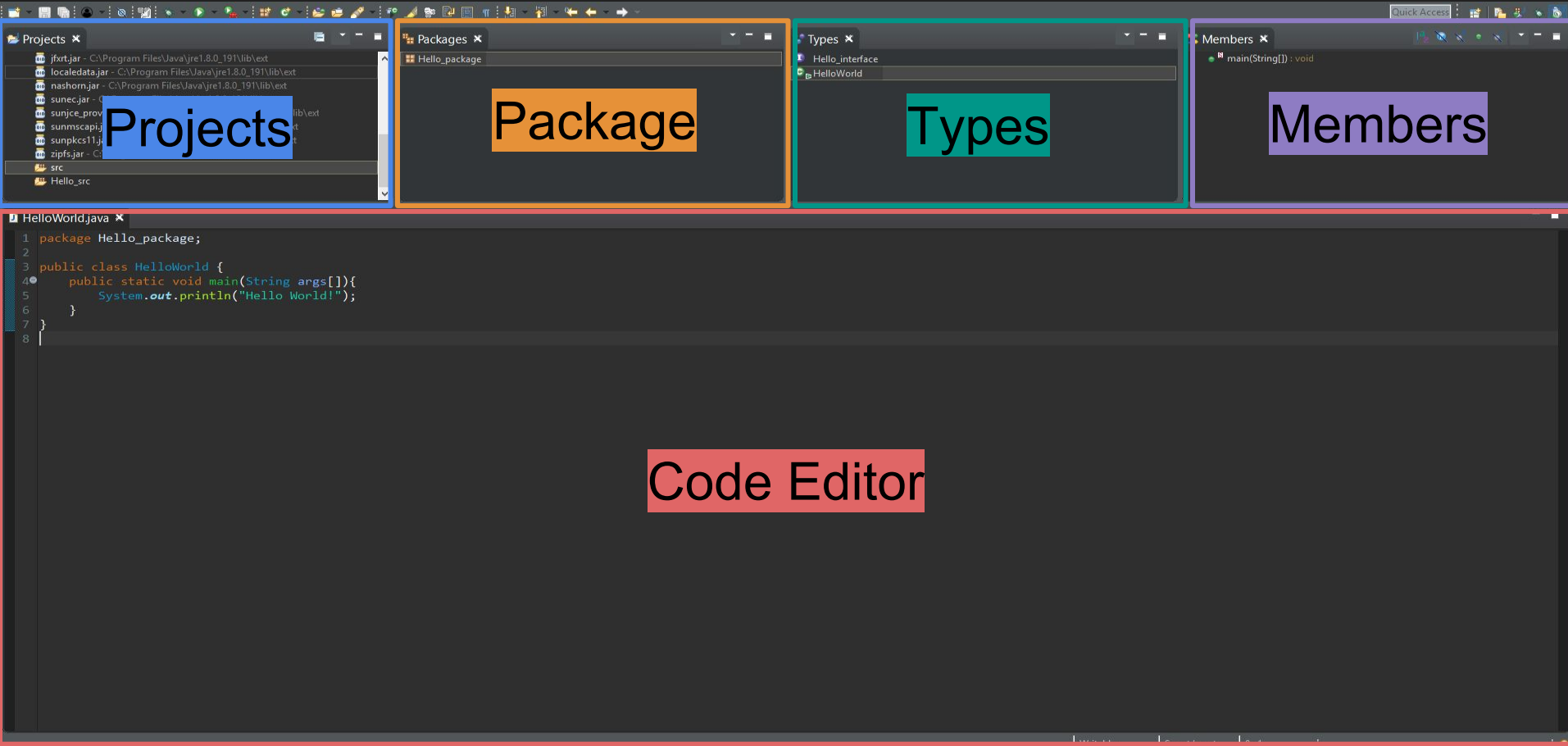
Debugger

Code Editor

Variables,
Breakpoints

Console

Java Browsing

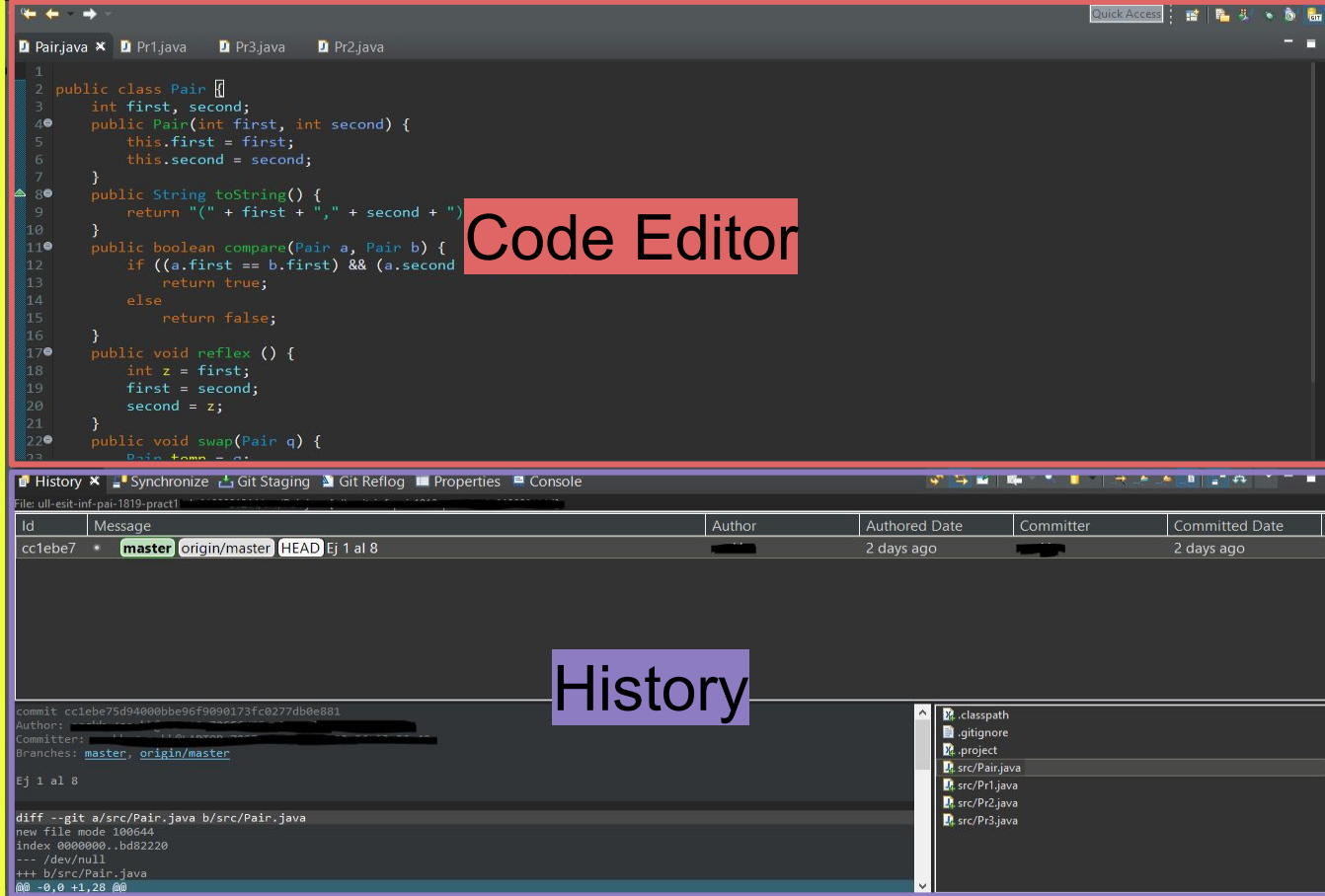


Git

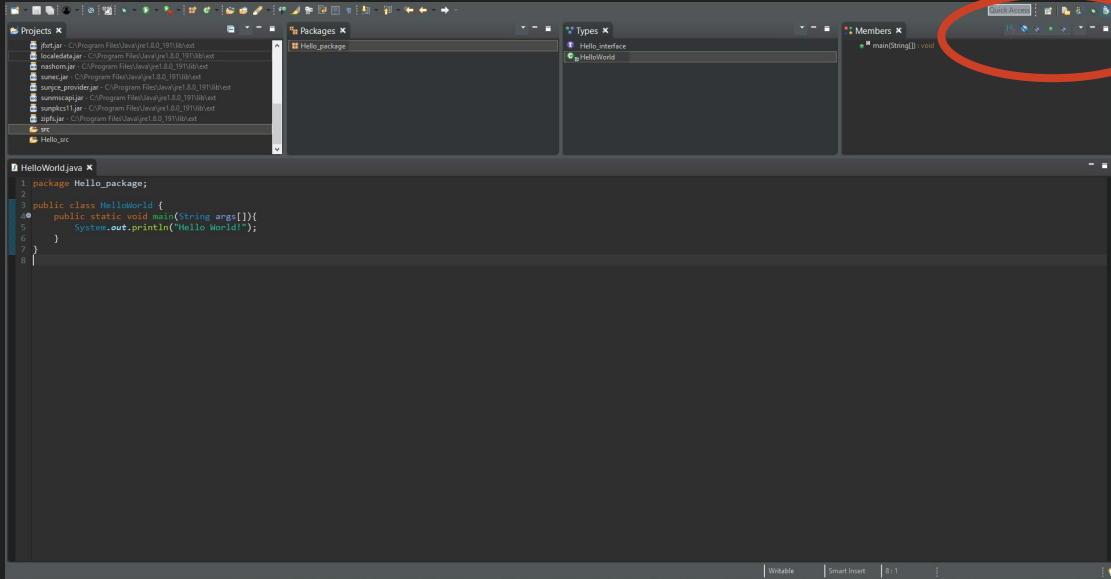
Git Repositories

Code Editor

History



Quick Access

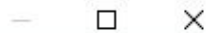


Tasks

- Allows quick management of pending tasks.
- Window -> Show View -> Tasks
- Right click -> Add Task
- Priorities can be assigned.
- Errors and warnings can appear here too.


[illegible]

 Add Task



Description:

Priority:

High 

☐ Completed

On element:

In folder:

Location:

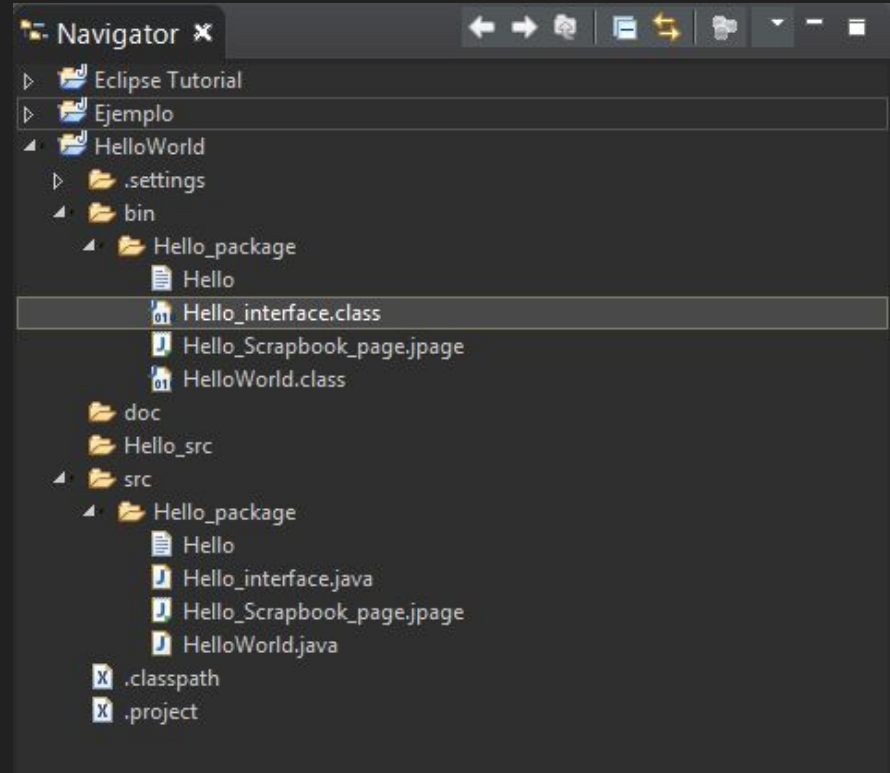


Add

Cancel

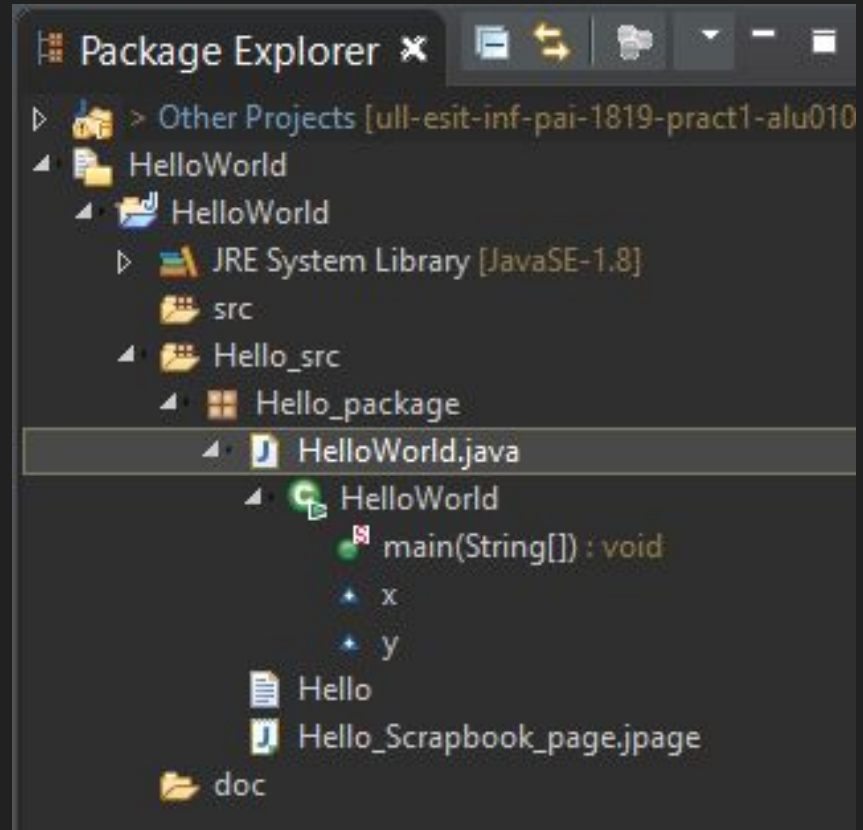
Navigator

- We can see the structure of defined files.
- The only one that shows the folder '/bin'.



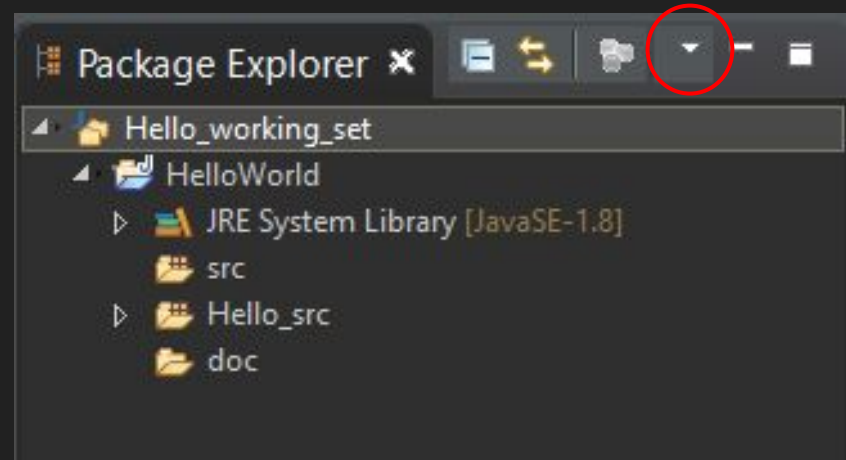
Package Explorer

- Shows the logical structure of packages and java classes.
- .java files can be expanded to show their attributes and methods



Working set

- They are used to separate the different projects that are being worked on.
- Useful when you are working on differentiated projects.
- View Menú -> Configure Working sets...



Select and sort working sets visible in Package Explorer:

- ☐ Other Projects
- ☐ HelloWorld
- ☐ Hello_working_set

Up

Down

Select All

Deselect All

☐ Sort working sets

New...

Edit...

Remove

OK

Cancel

Java Working Set

Enter a working set name and select the working set elements.

Working set name:

Workspace content:

- ▶ Eclipse Tutorial
- ▶ Ejemplo
- ▶ HelloWorld
- ▶ src [ull-esit-inf-pai-1819-pract1-alu0100901214 ma
- ▶ ull-esit-inf-pai-1819-pract1-alu010090121

Add ->

Add All ->

<- Remove

<- Remove All

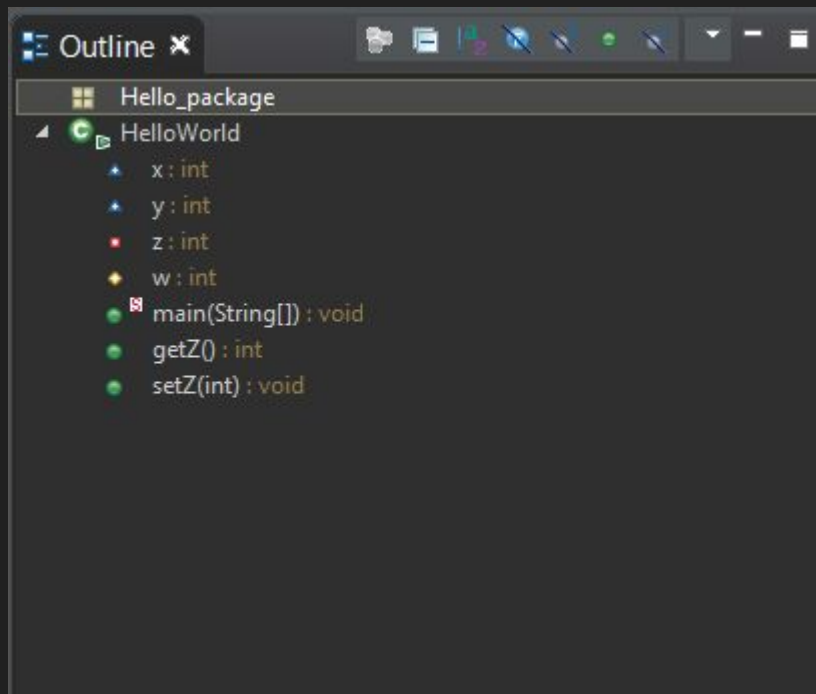
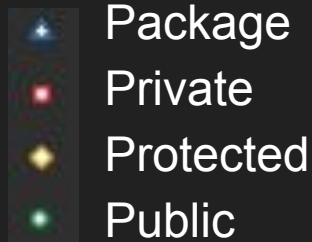
Working set content:

Finish

Cancel

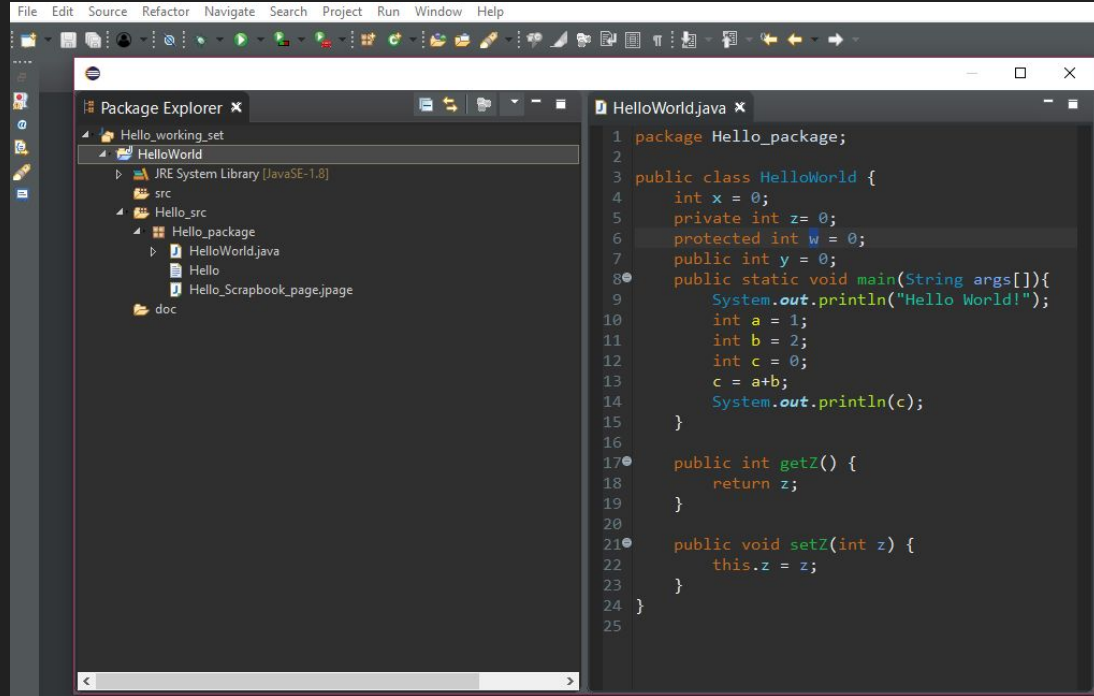
Outline View

- A quick way to see the methods and attributes of a .java class.
- Icons give additional information about visibility.



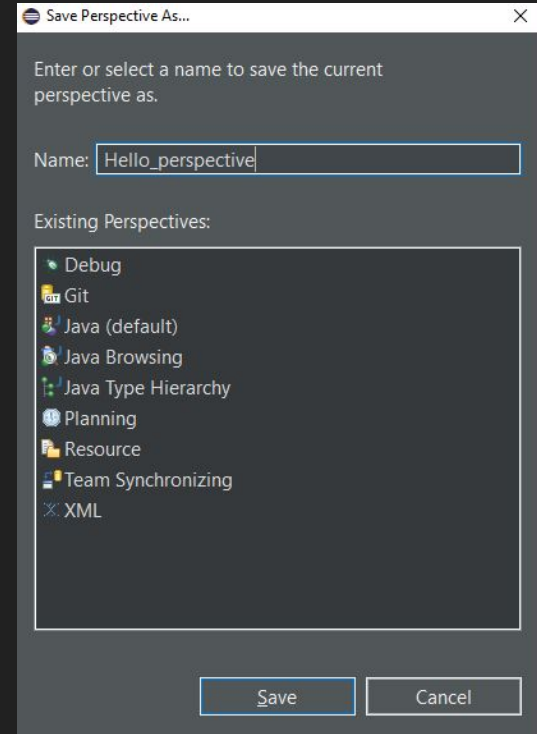
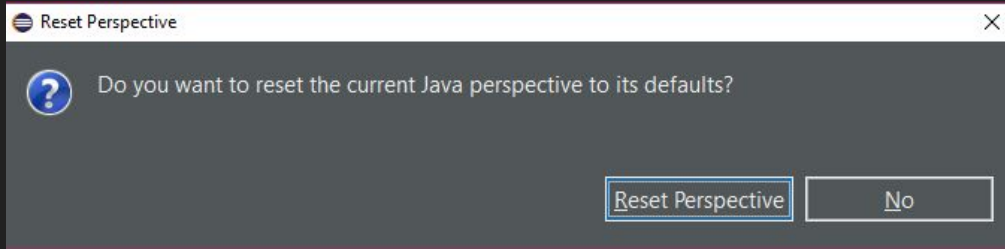
Fast views

- Used to separate the view in a new window



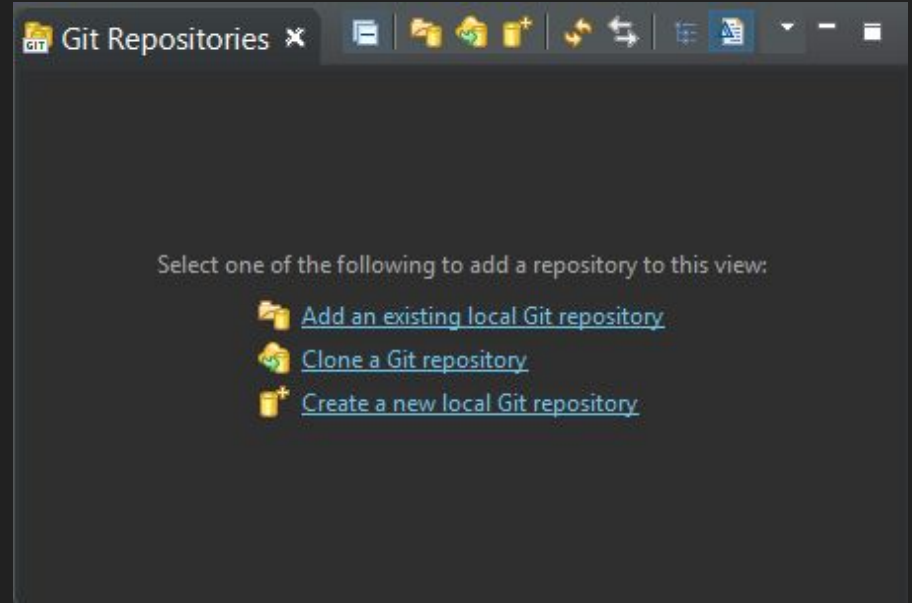
Save a perspective

- Window -> Perspective... -> Save Perspective as ...
- Window -> Perspective -> Reset Perspective



Git Repositories


- We can create a new repository.
- Add an existing one.
- Or Clone a repository



Create new file Upload files Find file Clone or download ▾

Clone with HTTPS ? Use SSH


Use Git or checkout with SVN using the web URL.

`https://github.com/ULL-ESIT-INF-PAI-1819` 

Open in Desktop Download ZIP

Clone Git Repository

Enter the location of the source repository.



Location

URI:

Host:

Repository path:

Connection

Protocol: ▾


Port:

Authentication

User:

Password:

☐ Store in Secure Store



Create a Java Project

Create a Java project in the workspace or in an external location.



Project name: ull-esit-inf-pai-1819-pract2-

☐ Use default location

Location: ull-esit-inf-pai-1819-pract2-

Browse...

JRE

- ☐ Use an execution environment JRE: javaSE-1.8
- ☐ Use a project specific JRE: jre1.8.0_201
- ☐ Use default JRE (currently 'jre1.8.0_201') [Configure JREs...](#)

Project layout

- ☐ Use project folder as root for sources and class files
- ☐ Create separate folders for sources and class files [Configure default...](#)

Working sets

☐ Add project to working sets

New...

Working sets:

Select...



The wizard will automatically configure the JRE and the project layout based on the existing source.



< Back

Next >

Finish

Cancel

Switch To >

Commit... Ctrl+#

Stashes >

Push to Upstream

Fetch from Upstream

Push Branch 'master'...

Pull

Pull...

Remote >

Merge...

Rebase...

Reset...

Import Projects...

Show In Alt+ Shift+ W >

Clean...

Collect Garbage

Remove Repository from View

Delete Repository...

Add Submodule...

Copy Path to Clipboard Ctrl+C

Paste Repository Path or URI Ctrl+V

Properties Alt+ Enter

New Branch...

master

Other...

```
public int y = 0;
public static void main(String[] args) {
    System.out.println("Hello World!");
    int a = 1;
    int b = 2;
    int c = 0;
    c = a+b;
    System.out.println("Sum: " + c);
}

public int getZ() {
    return z;
}

public void setZ(int z) {
    this.z = z;
}

Synchronize Git Stash
nf-pai-1819-pract1-alu0100901
sage
master origin/master
README.md files modified
initial version
```

5

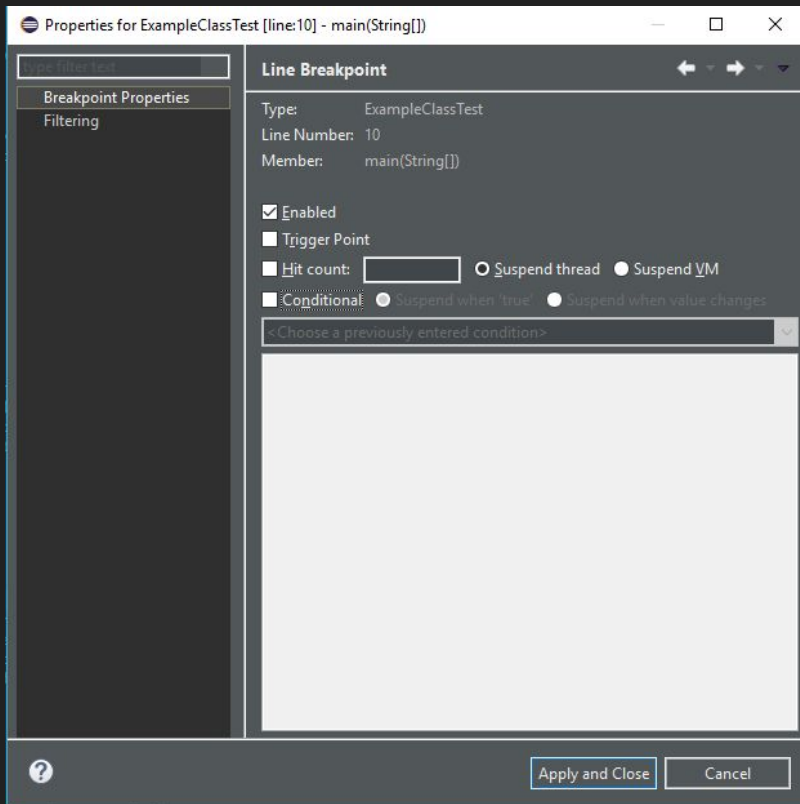
EJECUTE AND DEBUG

Shortcuts

- Run: CTRL +F11
- Debug: F11

¡Use breakpoints for debugging!

Breakpoints Properties



- Hit count
- Conditional

Debug steps



- RUN-RESUME (F8)
- SUSPEND
- TERMINATE
- STEP INTO
- STEP OVER
- STEP RETURN

Any Questions?