

Activitat 1. IDE Eclipse

En aquesta activitat aprendrem a fer servir l'IDE Eclipse. Com a documentació d'ajuda pots fer servir aquests documents, una introducció a Eclipse:

[intro_eclipse_espanol.pdf](#)

<http://dis.um.es/~bmoros/privado/bibliografia/tutorial%20eclipse%20para%20novatos%20java%20%28Pollino%29.pdf>

També pots fer servir l'ajuda del propi Eclipse i altres materials que trobis per Internet.

L'activitat combina descripcions teòriques de com fer servir diferents utilitats de l'IDE amb exercicis per aprendre el que s'acaba d'explicar. Llegeix amb atenció les explicacions teòriques i prova-ho tot sobre el teu Eclipse encara que no hi hagi un exercici específic. És la millor manera d'aprendre a fer servir un IDE i treure el màxim profit de les eines que proporciona.

Com que el document és llarg i no faràs tot el mateix dia, a continuació tens un petit índex amb enllaços a cada apartat:

[Parts de l'entorn Eclipse](#)

[Creació d'un projecte](#)

[Vista Package Explorer vs Vista Navigator](#)

[Creació d'elements Java](#)

[Java Class](#)

[File](#)

[Folder](#)

[Funcions útils de programació](#)

[Compilar i detectar errades](#)

[Autocorregir](#)

[CTRL + Espai = Autocompletar](#)

[Noms de classes](#)

[Altres funcions d'autocompletar](#)

[Atributs i variables locals](#)

[Bucles](#)

[Menú "Source"](#)

[Toggle Comment - Block comment](#)

[Format](#)

[Consultar la documentació](#)

[Navegar per les vistes i editors](#)

[Gestió de canvis](#)

[Vistes d'Eclipse](#)

Parts de l'entorn Eclipse

Exercici 1

La següent imatge correspon a una execució inicial del programa Eclipse. **identifica cadascuna de les zones que estan numerades:**

- 1 – Explorador de projectes: Mostra la estructura d'arxius i carpetes del projecte.
- 2 – Editor de codi: És on se escriu i edita el codi font dels programes.
- 3 – Barra de menú i d'eines: On trobem totes les opcions per gestionar projectes, editar codi i executar diverses funcions.
- 4 – Task List i Outline: Mostra una llista de tasques que estan per fer.
- 5 – Vistes: Son panels que proporcionen informació o eines específiques com pot ser la vista de consola, visa de variables, etc.

Creació d'un projecte

En el entorn de desenvolupament Eclipse tots els arxius s'emmagatzemen dins d'un projecte. Qualsevol document, carpeta, arxiu de codi font (ex. .java) i codi compilat (ex .class) han d'estar dins d'un projecte.

El primer que farem serà crear un projecte

L'assistent d'eclipse ens ofereix diferents opcions de projecte:

General Project - Únicament haurien de ser creats per emmagatzemar documentació i altres arxius. Però no codi de programes que s'hauran de compilar.

Java Project: Per editar i executar programes Java. Aquí també podrem emmagatzemar tota la informació del propi projecte.

Maven Project - Per crear projectes que integrin i facin servir llibreries de Maven

Gradle Project - Per crear projectes que integrin i facin servir llibreries de Gradle.

Hi ha més tipus de projectes que es poden crear des d'Eclipse. Caldria configurar-lo perquè mostri altre tipus de menús.

Ens centrarem ara en la utilització d'Eclipse per desenvolupar programes en Java.

Exercici 2

a> Crea el Projecte Java

Omple les dades que et demana.

Recorda !, el nom del projecte sempre en minúscules.

Observa que ofereix la possibilitat de canviar de JRE, per si tinguessin més d'un instal.lat.

Els fitxers del projecte es guardaran dins d'una carpeta amb nom el nom que posem al projecte. Aquesta carpeta es crea dins del WORKSPACE que hagin definit quan obrim Eclipse.

També és possible especificar un directori diferent en el qual guardar aquests continguts. Fitxa't amb l'opció "*Create separate folders for source amb class files*" Per defecte apareix marcada. Això farà que al crear la carpeta amb el nom del projecte, crearà també les carpetes:

src, pels fitxers font (.java)

bin, pels fitxers compilats a bytecode (.class)

Ara podríem clicar directament a finish i crearà el projecte..

NOTA: Si vols editar des d'Eclipse arxius que ja tenies fets abans és necessari que aquests arxius estiguin referenciats en el workspace amb què treballa Eclipse. Per fer-ho, cal importar a l'entorn els fitxers amb els quals volem treballar. Amb la versió actual d'Eclipse, quan fas l'importació d'un fitxer o un projecte no es copia el fitxer o projecte al workspace, sinó que únicament crea una referència.

Igualment, quan obres un projecte que no es troba al Workspace, automàticament Eclipse fa el mateix procés d'importació explicat abans.

Anem a provar-ho!

A les màquines de l'aula el workspace estarà a: C:\Users\lies\eclipse

Exercici 3

Al següent enllaç teniu un projecte creat amb Eclipse

<https://drive.google.com/file/d/0B-aT8EhHLcbyVTZiejN5U2VZYXM/view?usp=sharing>
HYPERLINK "https://drive.google.com/file/d/0B-aT8EhHLcbyVTZiejN5U2VZYXM/view?usp=sharing&resourcekey=0-pBs-M7-JzugctWN-yMALDQ"& HYPERLINK
"https://drive.google.com/file/d/0B-aT8EhHLcbyVTZiejN5U2VZYXM/view?usp=sharing&resourcekey=0-pBs-M7-JzugctWN-yMALDQ"resourcekey=0-pBs-M7-JzugctWN-yMALDQ

a> Descarregueu-lo i descomprimiu-lo. A continuació, importa'l al teu workspace d'Eclipse.
file/Import/General/Projects from Folder or Archive

b> Executa'l i comprova que funciona correctament. Afegeix una captura de pantalla on es mostri l'estructura de carpetes del projecte, el codi font del programa i el resultat de l'execució.








c> Ara obre l'explorador d'arxius i obre la carpeta corresponent al workspace d'Eclipse. Es troba a dins el projecte que acabes d'importar? On es troba?

El projecte que acabo d'importar no es troba dins de la carpeta d'Eclipse workspace si no que l'arxiu o arxius son a la carpeta on els he importat. Al workspace es guarda només una referència del projecte importat.

d> Obre amb l'explorador d'arxius la carpeta de **EIMeuPrimerProjecte** i identifica cadascun dels fitxers i carpetes que conté.

Inclou una captura de pantalla d'aquest contingut i explica perquè serveix cadascun.

NOTA: Tots els fitxers els pots obrir amb l'editor Notepad++, per veure el seu contingut.

	.settings	13/09/2024 17:48	Carpeta de archivos	
	bin	16/09/2024 10:59	Carpeta de archivos	
	src	13/09/2024 17:48	Carpeta de archivos	
	.classpath	13/09/2024 17:49	Archivo CLASSPATH	1 KB
	.project	13/09/2024 17:49	Archivo PROJECT	1 KB
	_classpath.xml	13/09/2024 17:48	Archivo XML	1 KB
	_project.xml	13/09/2024 17:48	Archivo XML	1 KB

- 1- **.settings**: Conté configuracions específiques del projecte que Eclipse fa servir per mantenir les preferències de configuració.
- 2- **Bin**: En aquesta carpeta es guarden els projectes compilats (arxius .class) generats a partir del codi font que em programat.
- 3- **Src**: És la carpeta on es guarda el codi font del projecte. Aquí es guarden els arxius .java.
- 4- **.classpath**: Conté la configuració que defineix les rutes de classe que calen per compilar i executar el projecte (referències externes a biblioteques, arxius JAR, etc).
- 5- **.project**: Aquest arxiu conté la informació bàsica sobre el projecte com el nom i el tipus de projecte. Eclipse el fa servir per identificar i manejar el projecte dins del IDE.
- 6- **_classpath.xml i project.xml**: similar als dos anteriors. Son arxius generats per algun plugin de Eclipse.

Creació d'elements Java

Eclipse és un entorn de desenvolupament fet amb Java però no únicament destinat a desenvolupar programes en Java. És per això que disposa de moltíssimes opcions que

seran necessàries o no segons el que estiguem desenvolupant. Quan programem en Java és important tenir a l'abast les parts que més farem servir. Per aconseguir-ho:

La perspectiva de Java conté les vistes i editors més útils a l'hora de crear nous programes en Java.

Java Class

Les classes de Java són els arxius ".java" que contenen el codi font i que seran posteriorment compilats en arxius ".class". Aquests arxius Java han de ser emmagatzemats dins de la carpeta font recentment creada pel projecte (per exemple, "src").

Premi "New> Class" per obrir la finestra de creació de classes.

També s'accedeix des de la barra d'eines a la icona:

La carpeta font especificada hauria de ser la carpeta recentment definida (src). Si no s'especifica cap paquet per contenir les classes Java, es guardaran dins d'un paquet per defecte.

L'últim camp obligatori que hauria de ser emplenat abans de procedir a la creació de la classe Java és el propi nom de la classe.

Les convencions de Java diuen que el nom d'una classe hauria de començar amb majúscula, si el nom està format per dos paraules, cada paraula comença amb majúscula (notació Camel).

Per exemple: ExampleClass.

Si es desitja que la nova classe contingui un mètode "main" (és a dir, el punt inicial de execució del programa), pot afegir-se aquest mètode automàticament només amb marcar la casella amb l'opció apropiada.

Fitxa't que hi ha altres modificadors que poden ser fàcilment afegits a una classe des del mateix moment de la seva creació per facilitar al programador la tasca d'edició. Però, aquests elements també podrien ser afegits manualment en altres fases més avançades del procés de desenvolupament i que s'estudien en un nivell més avançat de programació.

Exercici 4

Crea un projecte Java que es digui **prova** i a dins, crea una classe que contingui el main. Ha de crear-se similar a això:

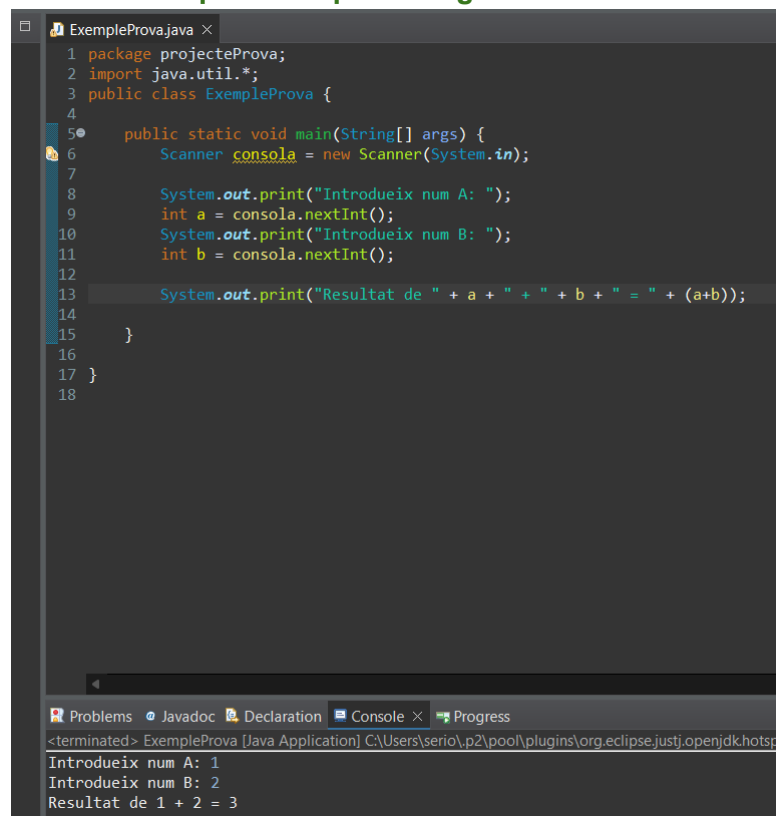
El package és el nom del nostre projecte.

La classe apareix creada amb el mètode main generat automàticament.

Això també ho podem escriure nosaltres.

Fent servir aquesta classe crea un programa que llegeixi dos números enters i ens mostra la seva suma. Compila'l i comprova que funciona correctament.

Inclou una captura en què es vegi el codi font i el resultat d'una execució.



```
1 package projecteProva;
2 import java.util.*;
3 public class ExempleProva {
4
5     public static void main(String[] args) {
6         Scanner consola = new Scanner(System.in);
7
8         System.out.print("Introdueix num A: ");
9         int a = consola.nextInt();
10        System.out.print("Introdueix num B: ");
11        int b = consola.nextInt();
12
13        System.out.print("Resultat de " + a + " + " + b + " = " + (a+b));
14    }
15 }
16 }
17 }
18 }
```

Problems Javadoc Declaration Console × Progress

<terminated> ExempleProva [Java Application] C:\Users\serio\p2\pool\plugins\org.eclipse.justj.openjdk.hotsp

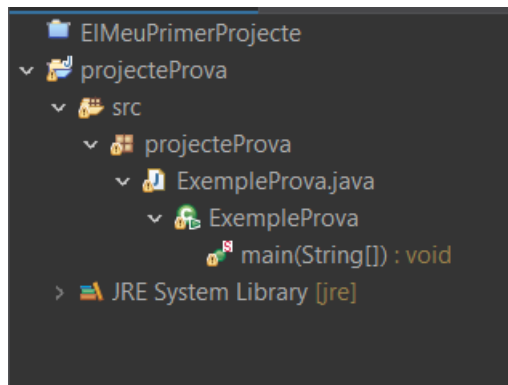
Introdueix num A: 1
Introdueix num B: 2
Resultat de 1 + 2 = 3

Vista Package Explorer vs Vista Navigator

Exercici 5

a> Accedeix a la vista Package Explorer que tens a l'esquerra de l'editor i desplega el contingut del teu projecte.

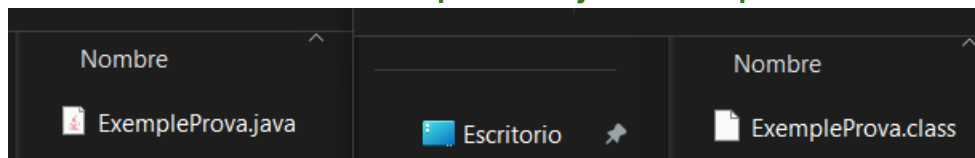
Fes una captura de pantalla del contingut.



b> **Veus el fitxer compilat .class?**

Nombre	Fecha de modificacion	tipo	tamaño
ExempleProva.class	16/09/2024 11:50	Archivo CLASS	2 KB

c> Obre la vista Navigator (la versió 2021-09 d'Eclipse ja no inclou aquesta vista, no ho facis excepte si tens versions anteriors que sí la tinguin) Si treballes amb aquestes noves versions d'Eclipse localitza la carpeta del teu projecte **prova** al teu workspace i **inclou captures on es vegin els directoris i els arxius ExempleProvar.java i ExempleProvar.class**



Desplega tot el teu projecte. Què et mostra?

Quina diferència hi ha respecte a la vista Package Explorer?

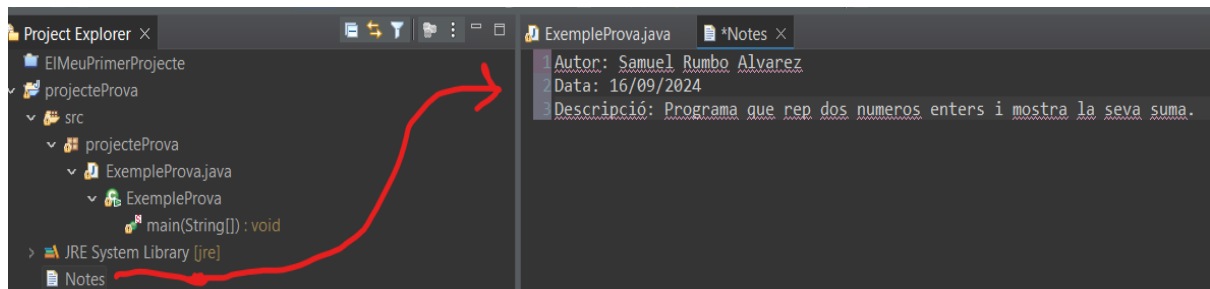
File

Els arxius que es creen a través d'aquest menú solen emmagatzemar notes i informació general. Altres arxius com ara arxius de configuració, i similars, també poden crear-se d'aquesta manera.

Crear un nou arxiu és tan senzill com prémer "New> File", seleccionar el projecte i carpeta adequats en què es vol crear l'arxiu, donar nom al nou arxiu i prémer el botó "Finish".

Exercici 6

Crea un fitxer que es digui **Notes** dins del teu projecte **provar** i fes constar allà el nom de l'autor d'aquest projecte, la data de creació i una petita descripció de què fa el teu programa. **Inclou una captura on es vegi.**



Folder

Les carpetes s'utilitzen per emmagatzemar i organitzar fitxers. Com ja s'ha esmentat anteriorment, una pràctica recomanada de programació és tenir creada una estructura de carpetes per emmagatzemar la informació del nostre projecte:

carpeta arrel del projecte

carpeta per guardar els fitxers font - (src)

carpeta per guardar els fitxers compilats - (bin)

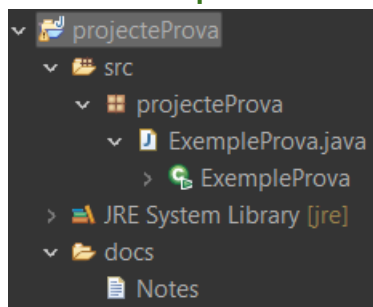
carpeta per guardar tota la documentació relacionada - (docs)

Exercici 7

En el nostre projecte ***provar*** tenim creades per Eclipse les carpetes src i bin. Ens faltaria crear la carpeta docs

Crea la carpeta **docs** al teu projecte i fica dins el fitxer Notes creat anteriorment. Fes-ho mitjançant la vista Navigator (si la tens) i fent servir el menú contextual sobre el projecte.

Inclou una captura on es vegi.



Funcions útils de programació

En Eclipse tenim diferents colors en el text quan estem editant codi font; les paraules reservades en granat, els comentaris en blau clar o verd i la resta del text en negre. Això facilita moltíssim la feina a l'hora de programar. **Cal recordar que Java distingeix entre majúscules i minúscules.**

Compilar i detectar errades

En Eclipse els errors de compilació es mostren en temps real subratllant el fragment de codi adequat amb una línia vermella. L'entorn automàticament compila els arxius salvats. Així doncs, no serà necessari passar pel tediós i lent procés de *compilar - observar els errors - corregir els errors*.

Els errors poden trobar-se fàcilment perquè es mostren a més amb marques vermelles en el marge esquerre de l'editor de codi Java. Fent clic en qualsevol dels dos tipus de marcadors d'error portarà automàticament fins a la línia en què l'error és present. Els advertiments (warnings) es mostren de la mateixa manera, però amb marques grogues.

Codi amb dos errors: marques vermelles Codi correcte amb dos advertiments (warning)

Si poses el ratolí a sobre de les marques vermelles i/o grogues, l'editor explica breument l'errada trobada en cada cas.

Autocorregir

Eclipse habitualment permet autocorregir els possibles errors fent clic a la icona d'error present en el marge esquerre de l'editor de codi. Així doncs, apareixerà una finestra mostrant totes les opcions. Seleccionem una opció mitjançant els cursors del teclat o deixar el punt del ratolí sobre aquesta opció s'obrirà una nova finestra mostrant detalladament les modificacions de codi que l'autocorrecció efectuarà. Només cal prémer l'opció seleccionada (o prémer ENTER) per fer que Eclipse dugui a terme la correcció automatitzada.

Cal tenir molta cura en fer servir aquesta opció, perquè Eclipse t'ofereix moltes possibilitats i cal escollir l'adequada. Si no ho feu, l'error desapareix però el programa no farà el que voleu.

CTRL + Espai = Autocompletar

Noms de classes

Crear referències a altres classes dins de la classe actual és una tasca de programació habitual. No obstant això, algunes classes de Java tenen noms molt llargs que són difícils de recordar. A més, cal afegir declaracions d'importació per poder resoldre aquestes referències a classes a l'hora de compilar.

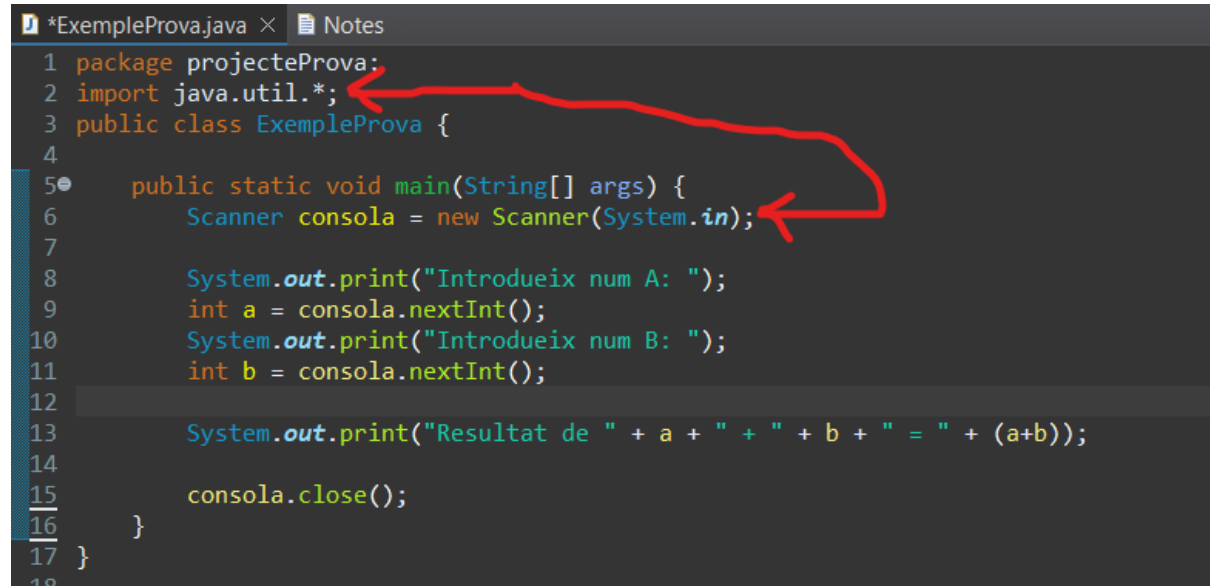
Per exemple, la classe `Scanner` que ens permet fer l'entrada de dades al programa:

Al començar a escriure la paraula `Scanner`, si fem CTRL + Espai apareix la llista de possibles classes i el paquet on es troba.

Selecció amb el ratolí la classe que volem, Eclipse incorpora el nom de la classe al codi i genera l'import corresponent del paquet on es troba perquè després no hi hagi errors de compilació.

Fes la prova començant a escriure la sentència: `Scanner reader = new Scanner(System.in);`

Inclou una captura on es vegi l'import que ha creat automàticament. No cal que ho gravis, treu-la després.



```
1 package projecteProva;
2 import java.util.*;
3 public class ExempleProva {
4
5     public static void main(String[] args) {
6         Scanner consola = new Scanner(System.in);
7
8         System.out.print("Introdueix num A: ");
9         int a = consola.nextInt();
10        System.out.print("Introdueix num B: ");
11        int b = consola.nextInt();
12
13        System.out.print("Resultat de " + a + " + " + b + " = " + (a+b));
14
15        consola.close();
16    }
17 }
18 }
```

Exercici 7: Corregir errors sintàctics

Descarrega i importa el següent projecte al teu workspace d'Eclipse. Edita el programa java.

<https://drive.google.com/file/d/0B-aT8EhHLcbycEpLX3RGdUdyQ0k/view?usp=sharing>
HYPERLINK "https://drive.google.com/file/d/0B-aT8EhHLcbycEpLX3RGdUdyQ0k/view?"

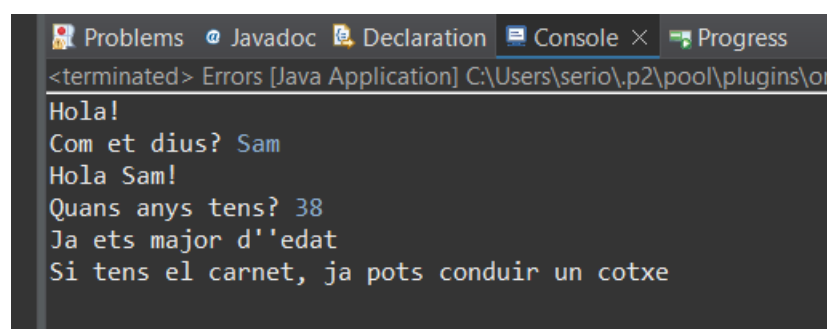
[usp=sharing&resourcekey=0-2NY6X9FxDgNoMEsAcUC1w"&](https://drive.google.com/file/d/0B-aT8EhHLcbycEpLX3RGdUdyQ0k/view?usp=sharing&resourcekey=0-2NY6X9FxDgNoMEsAcUC1w) [HYPERLINK](https://drive.google.com/file/d/0B-aT8EhHLcbycEpLX3RGdUdyQ0k/view?usp=sharing&resourcekey=0-2NY6X9FxDgNoMEsAcUC1w)
"https://drive.google.com/file/d/0B-aT8EhHLcbycEpLX3RGdUdyQ0k/view?
usp=sharing&resourcekey=0-2NY6X9FxDgNoMEsAcUC1w"resourcekey=0-
2NY6X9FxDgNoMEsAcUC1w

Corregeix totes les errades fent servir la detecció d'errades i l'autoajuda d'Eclipse. Explica quines errades hi ha al codi i com les has corregit. Afegeix comentaris explicant què has arreglat.

Quan hagi acabat executa el programa i comprova que funciona correctament.

Inclou una captura on es vegi el codi font arreglat amb els comentaris i el resultat de l'execució.

```
2
3 import java.util.Scanner;
4
5 public class Errors {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9
10        Scanner reader = new Scanner(System.in); /* FALTA EL IMPORT DE JAVA UTILS */
11        String nom;
12
13
14        System.out.print("Hola! \nCom et dius? "); /* FALTA UNA 'T' AL PRINT*/
15        nom = reader.next();
16        System.out.println("Hola " + nom.toString() + "!"); /* FALTA UN PUNT I COMA */
17
18        System.out.print("Quans anys tens? ");
19        int edat = reader.nextInt(); /* FALTA DECLARAR TIPUS DE VARIABLE */
20
21        if (edat >= 18) { /* FALTA OBRIR LA CLAU '{' */
22            System.out.println("Ja ets major d'edat");
23            System.out.println("Si tens el carnet, ja pots conduir un cotxe");
24        }
25        else {
26            System.out.println("Encara ets petit ");
27        }
28
29        reader.close();
30
31    }
32 } /* FALTA LA CLAU PER TANCAR LA CLASSE ERRORS */
```



```
<terminated> Errors [Java Application] C:\Users\serio\p2\pool\plugins\or
Hola!
Com et dius? Sam
Hola Sam!
Quans anys tens? 38
Ja ets major d'edat
Si tens el carnet, ja pots conduir un cotxe
```

Altres funcions d'autocompletar

Atributs i variables locals

Després d'escriure els primers caràcters de l'atribut o de la variable local, prémer "CTRL + Espai" mostrarà les possibles alternatives. Aquest procés és molt similar al de autocompletar el nom de les classes. Les variables locals es marquen amb la icona d'una "L" gris, mentre que els atributs es marquen amb una icona que pot variar segons la visibilitat de l'atribut.

(El concepte d'atribut el veurem en la UF-4 de POO)

Bucles

Eclipse proporciona algunes funcions d'autocompletat que poden accelerar considerablement el procés d'edició d'un programa. Només cal escriure "do", "While" o "for" i prémer "CTRL + Espai" per mostrar les possibles opcions.

Menú "Source"

Mitjançant un clic dret a l'editor de codi es mostrarà un menú de context. les funcions més importants del submenú "Source" són les següents:

Tougle Comment - Block comment

Selecciónant aquestes opcions es pot comentar o treure la marca de comentari del codi.

Això és especialment útil usat amb blocs de codi, ja que permet no executar temporalment parts del codi sense arribar a eliminar-les.

Per marcar una línia de codi amb comentari amb les tecles ràpides associades són "CTRL + /" per afegir i treure els comentaris del bloc de codi seleccionat.

Format

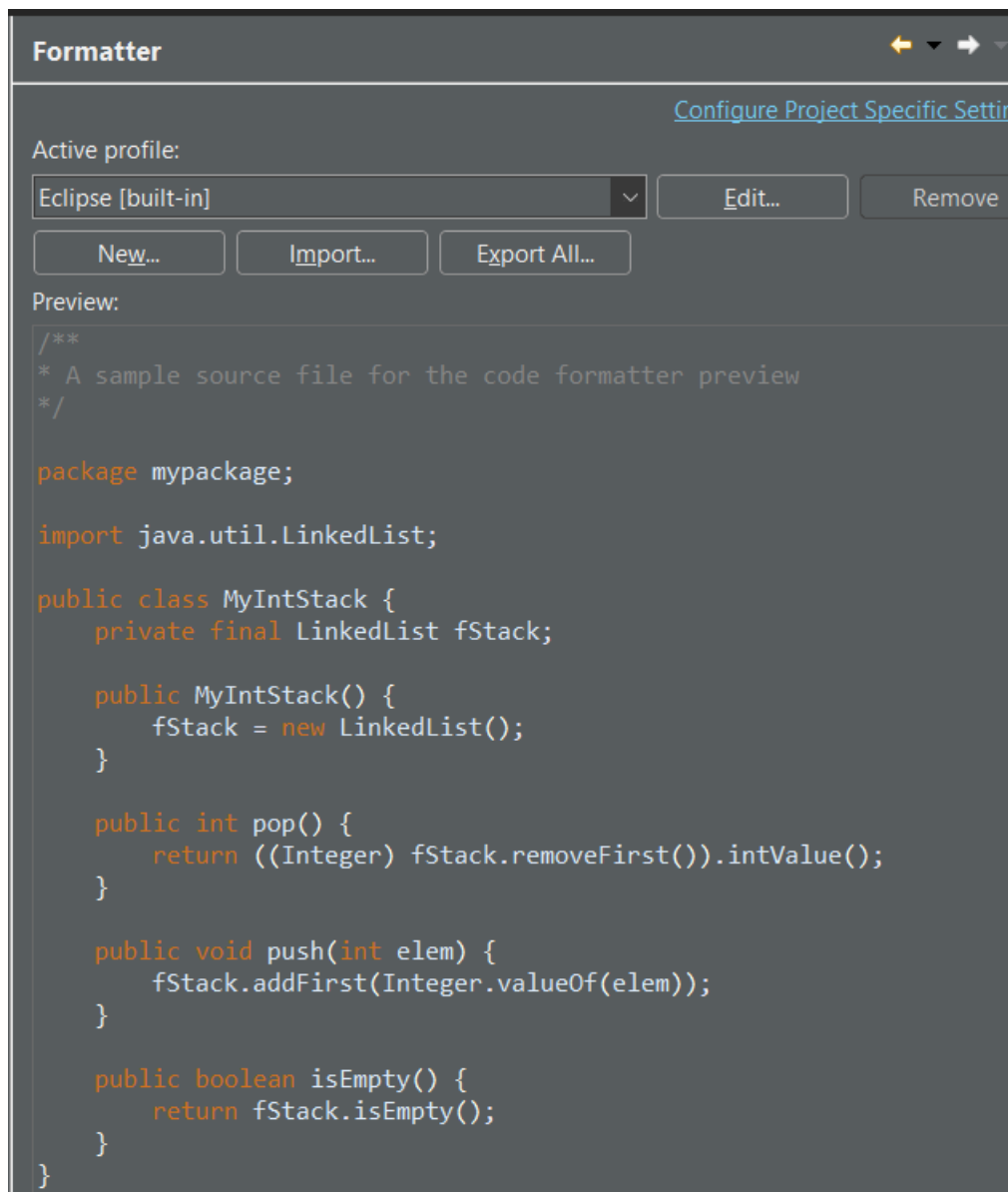
Eclipse disposa d'una funció de formateador de codi. El que fa és indentar el codi que estem editant de forma automàtica. És una forma ràpida d'aconseguir tenir un codi ordenat i comprensible.

Les opcions del formatador de codi es poden personalitzar a les preferències personals del programador accedint al menú **Windows > Preferences > Java > Code Style** i allí accedir al formatejador.

Cal tenir present que aquestes preferències ja venen especificades per Eclipse per complir les convencions de programació en Java. De moment mira les opcions però no toquis res.

Exercici 8

a> Accedeix a la finestra de preferències i observeu com té eclipse definit el format de codi. Inclou una captura de pantalla de la configuració actual.



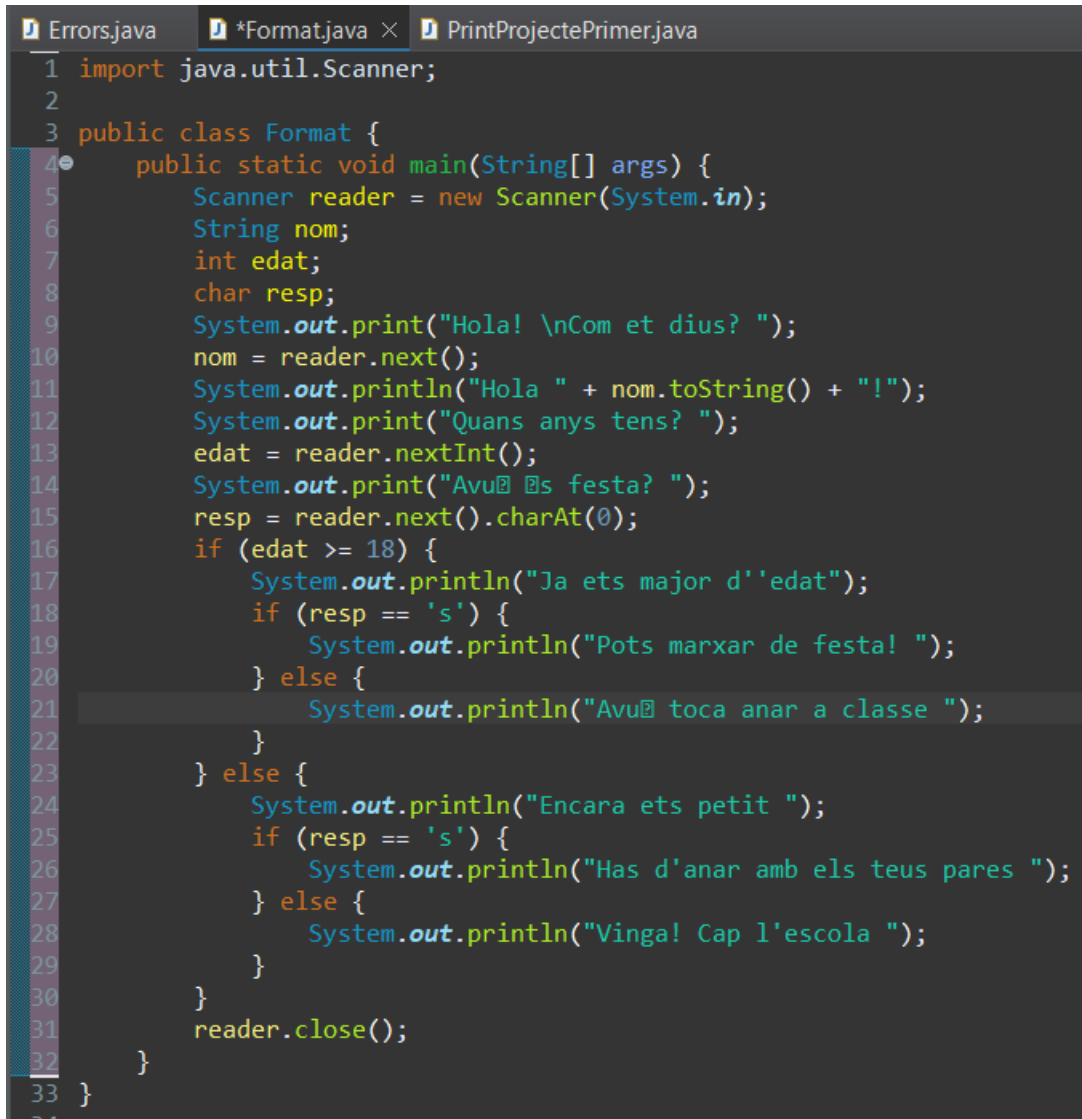
b> Obre el següent fitxer java des d'Eclipse. Per fer-ho hauràs d'incorporar-lo a un projecte.

<https://drive.google.com/file/d/0B-aT8EhHLcbycDV3VGxOREINcUU/view?usp=sharing> [HYPERLINK "https://drive.google.com/file/d/0B-aT8EhHLcbycDV3VGxOREINcUU/view?usp=sharing&resourcekey=0-O63I55OkAd4h18V64DwEUQ"&](https://drive.google.com/file/d/0B-aT8EhHLcbycDV3VGxOREINcUU/view?usp=sharing&resourcekey=0-O63I55OkAd4h18V64DwEUQ) [HYPERLINK "https://drive.google.com/file/d/0B-aT8EhHLcbycDV3VGxOREINcUU/view?usp=sharing&resourcekey=0-O63I55OkAd4h18V64DwEUQ"](https://drive.google.com/file/d/0B-aT8EhHLcbycDV3VGxOREINcUU/view?usp=sharing&resourcekey=0-O63I55OkAd4h18V64DwEUQ)

c> Entens fàcilment què fa aquest codi?

No. Es difícil d'entendre sense un format adequat.

d> Aplica'l el formateador (Source/format) i explica ara que fa aquest programa.
Incorpora una captura de pantalla del codi formatat.



```
1 import java.util.Scanner;
2
3 public class Format {
4     public static void main(String[] args) {
5         Scanner reader = new Scanner(System.in);
6         String nom;
7         int edat;
8         char resp;
9         System.out.print("Hola! \nCom et dius? ");
10        nom = reader.next();
11        System.out.println("Hola " + nom.toString() + "!");
12        System.out.print("Quans anys tens? ");
13        edat = reader.nextInt();
14        System.out.print("Avu  ets festa? ");
15        resp = reader.next().charAt(0);
16        if (edat >= 18) {
17            System.out.println("Ja ets major d'edat");
18            if (resp == 's') {
19                System.out.println("Pots marxar de festa! ");
20            } else {
21                System.out.println("Avu toca anar a classe ");
22            }
23        } else {
24            System.out.println("Encara ets petit ");
25            if (resp == 's') {
26                System.out.println("Has d'anar amb els teus pares ");
27            } else {
28                System.out.println("Vinga! Cap l'escola ");
29            }
30        }
31        reader.close();
32    }
33 }
```

Nota: Les tecles ràpides associades per formatar el codi són " **CTRL + Majúscules + F**".

El codi pregunta pel nom i edat de l'usuari i després pregunta si es dia de festa. En cas de que la persona sigui major d'edat el codi et diu que pots marxar, en cas contrari t'envia amb els teus pares.

Organise and Add Imports

Tot i que les sentències d'importació de classes adequades es mostren sempre quan s'usen les funcions d'autocompletar codi per completar el nom d'una classe Java, podem afegir noves sentències d'importació en qualsevol moment usant la funció **"Add Import"**.

Prémer "**Organise Imports**" eliminarà automàticament totes les declaracions d'importació no utilitzades, incrementant l'eficiència del codi.

El mètode abreujat del teclat per fer l'import de les classes necessàries és "CTRL + Majúscules + O".

Consultar la documentación

La documentació **Javadoc** sempre està disponible quan s'està editant un fitxer java i pot ser consultada en temps real simplement col·locant el cursor o el punter del ratolí sobre l'element triat.

També podem consultar la documentació de l'element on estigui situat el cursor obrint la vista javadoc.

(estudiarem javadoc més endavant)

Navegar per les vistes i editors

Exercici 9

Existeixen altres funcionalitats de l'IDE que faciliten la tasca dels programadors.

Prova les següents funcionalitats des del teu Eclipse

a> **Maximizar la finestra d'una vista o editor** - Doble clic al títol de la finestra. Per tornar a la situació original. tornar a fer doble click al títol.

b> **Situar-se en l'últim canvi fet** - La icona de la barra d'eines, realitza aquesta funció. CTRL + Q fa el mateix.

Si hem estat treballant amb diferents documents i volem saber amb quins fitxers, dels que tenim oberts a l'editor hem accedit cap endarrere o cap endavant del moment actual fent servir . Amb les tecles fent servir "ALT + F6"

c> Per navegar d'una notació/error en l'edició a la següent o a l'anterior fent servir les icones:

Gestió de canvis

No és estrany canviar codi del programa i adonar-se després que la nova versió funciona fins i tot encara pitjor que la primera versió. Aquest és el motiu que els programadors hem de guardar diferents versions del programa cada vegada que introduïm un nou canvi. No obstant això, aquesta pràctica inevitable és normalment tediosa i consumeix molt de temps. Afortunadament, Eclipse proporciona un potent sistema de gestió de canvis i de control de versions.

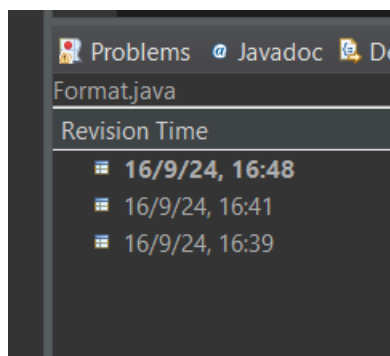
Fent clic dret en un arxiu Java dins el Package Explorer i Restore from Local History "permetrà reemplaçar la versió actual per una versió prèviament desada.

L'hora i data de modificació es mostren juntament amb dues finestres que destaquen les diferències existents entre les dues versions.

Des de la finestra de windows - Preferències podrem configurar fins a quants dies volem guardar les diferents versions.

Inclou una captura on es vegi la “Local History” d’algun dels teus programes.

(Si no tens cap versió antiga prova a fer algun canvi , guardar-lo i després recuperar la versió antiga.)



Vistes d'Eclipse

La interfície d'usuari d'Eclipse consta de dos tipus d'elements: vistes i editors. Mentre que els editors normalment permeten realitzar una tasca completa, les vistes proporcionen funcions de suport.

Perspectives

Una perspectiva d'Eclipse és una agrupació de vistes i editors de manera que donin suport a una activitat completa del procés de desenvolupament de programari. No obstant això, és possible crear perspectives pròpies afegint noves vistes i canviant la seva distribució a la pantalla.

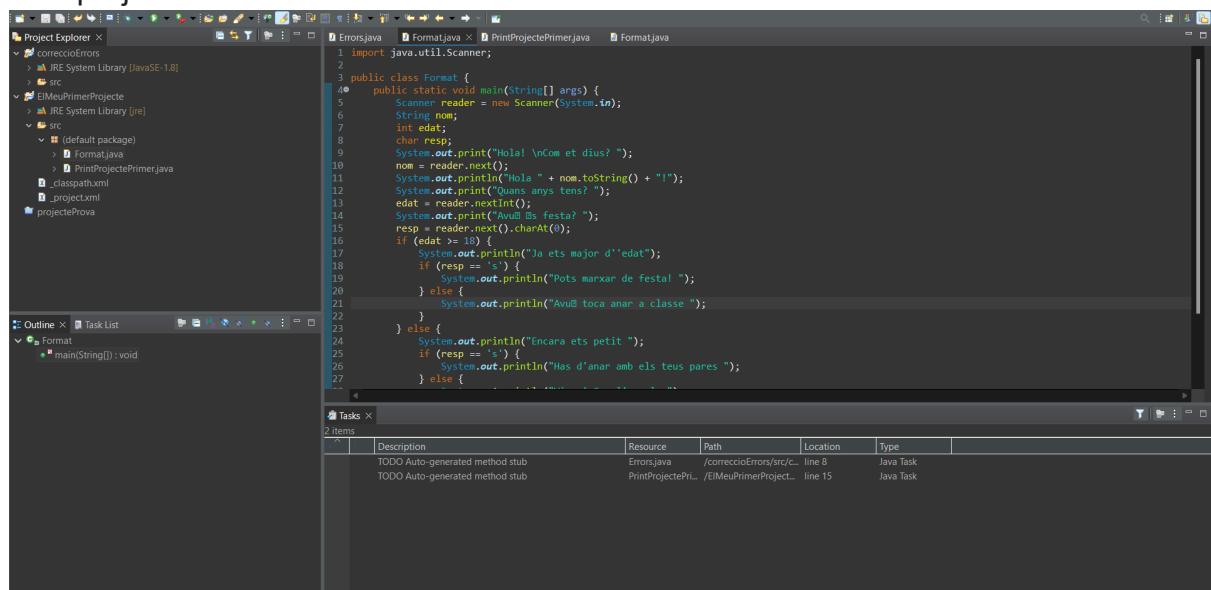
Les perspectives es poden seleccionar accedint al menú "**Window> Open Perspective**"

També estan disponibles a les icones a la dreta de la barra d'eines.

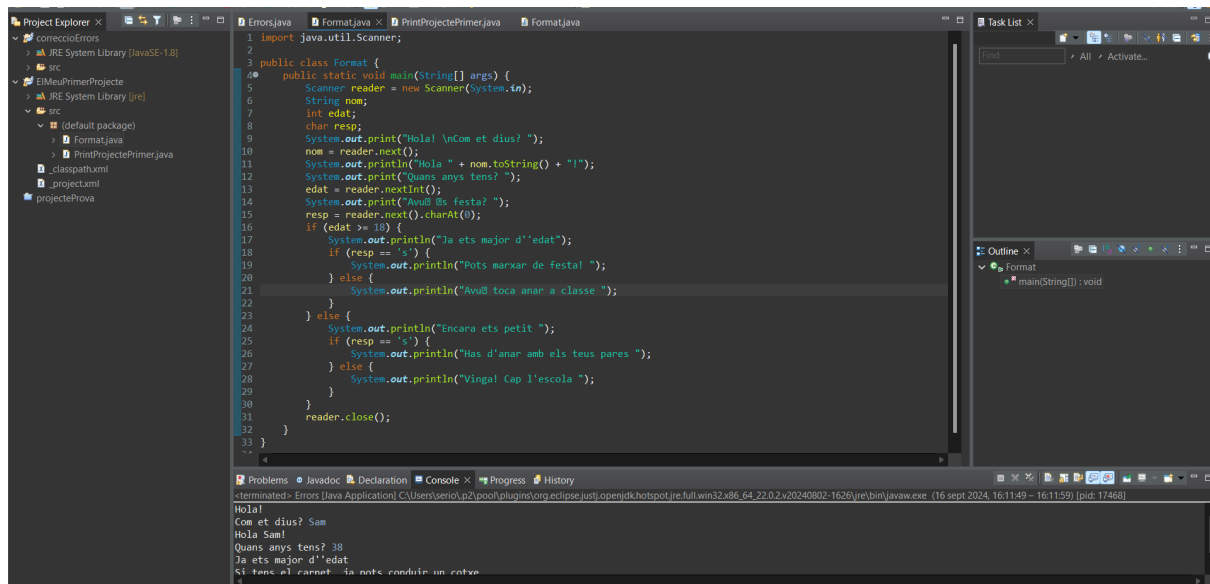
Algunes perspectives importants:

Inclou una captura de pantalla de cada perspectiva i indica per a cadascuna quines vistes mostra.

Resource: aquesta perspectiva està estretament relacionada amb el sistema d'arxius ja que representa la localització física dels recursos emmagatzemats dins dels projectes.

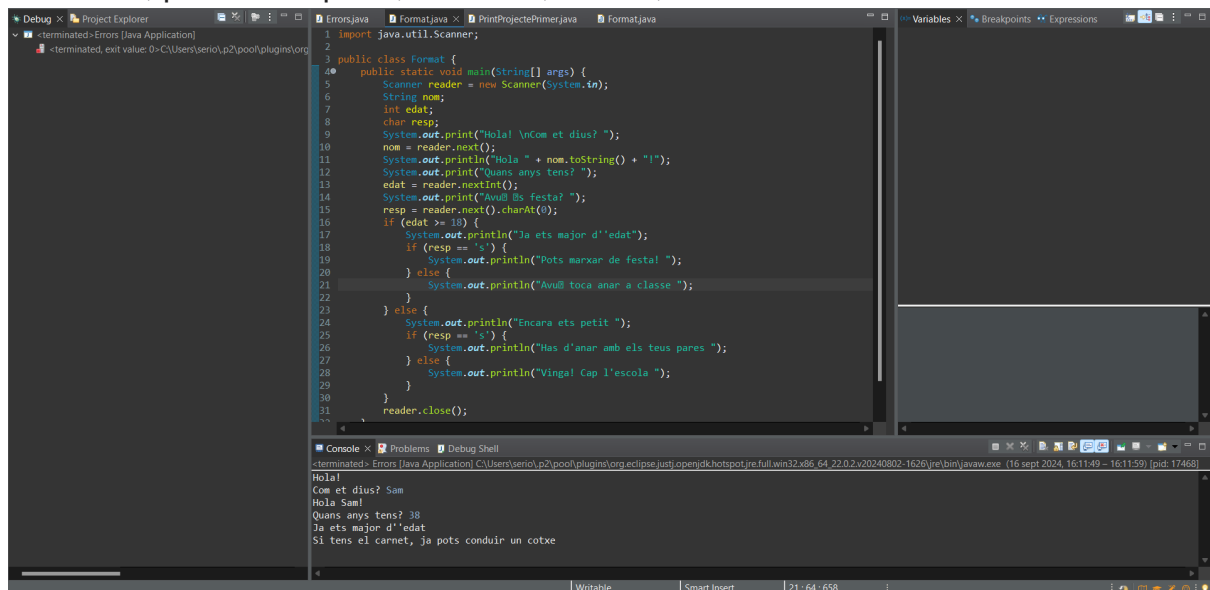


Java: aquesta perspectiva se centra en tasques de programació, mostrant paquets, classes, mètodes i atributs en les seves vistes associades.

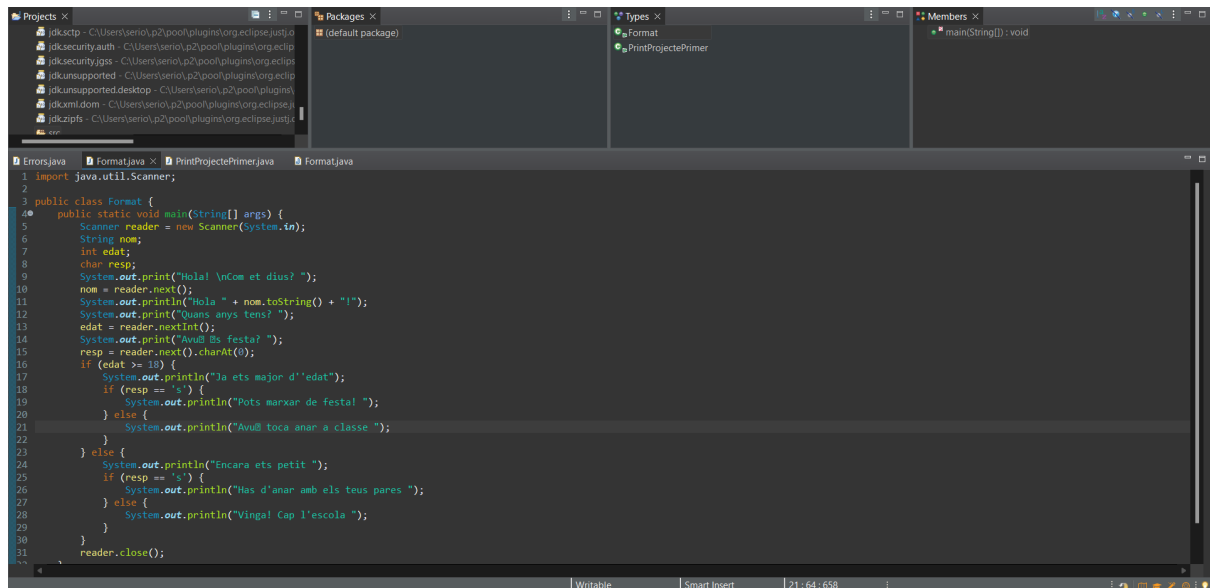


Plug-in development: la perspectiva de desenvolupament de plug-in permet als desenvolupadors afegir nous mòduls d'Eclipse.

Debug: relacionada amb la tasca de depuració. Se centra en els processos executats, punts de ruptura, variables, sortida, etc.



Java Browsing: aquesta perspectiva permet visualitzar ràpidament codi, projectes, paquets i jerarquies.



Exercici 10

Vistes

Des de qualsevol perspectiva podrem obrir sempre qualsevol vista accedint al menú de **Windows -> Show View**

Task View

Permet gestionar les tasques pendents.

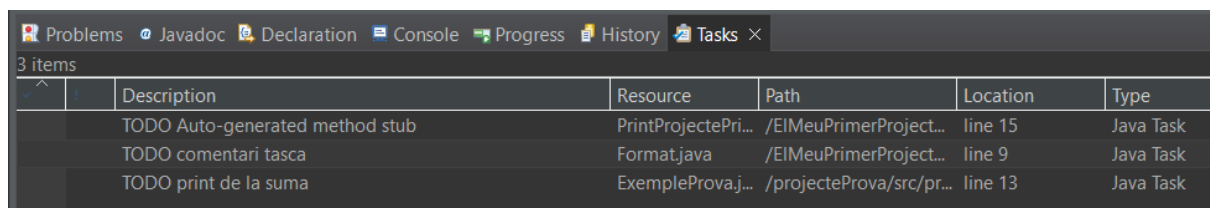
Si afegim "TODO [descripció]" a un arxiu Java afegirà una nova tasca "per fer" a la vista de tasques. Fent clic sobre la seva descripció conduirà fins al punt exacte en què l'etiqueta "TODO" es va afegir dins de l'arxiu corresponent.

Nota: Dins de l'editor de codi Java les etiquetes de "TODO" poden trobar-se ràpidament, ja que es mostren com petites marques blaves en el marge dret. (aquestes marques únicament es mostren si no hi ha errors de compilació al codi font).

Cal notar que alguns generadors de codi automatitzats insereixen aquestes etiquetes "TODO" per assegurar-se que el codi autogenerat és revisat i comentat. Més endavant ho veureu amb molta freqüència quan esteu desenvolupant codi java.

Exercici 11

a> Obre amb l'Eclipse diferents fitxers amb codi font Java i incorpora en cadascun d'ells un comentari `//TODO`



The screenshot shows the Eclipse IDE's 'Tasks' window. The window title is 'Tasks' with a close button. It displays a table with 3 items. The table has columns: Description, Resource, Path, Location, and Type. The items are:

Description	Resource	Path	Location	Type
TODO Auto-generated method stub	PrintProjectePri...	/ElMeuPrimerProject...	line 15	Java Task
TODO comentari tasca	Format.java	/ElMeuPrimerProject...	line 9	Java Task
TODO print de la suma	ExempleProva.j...	/projecteProva/src/pr...	line 13	Java Task

b> Obre la finestra Task (al costat de console) i comprova que s'han afegit com a tasques pendents cadascun dels comentaris TODO Fes clic sobre aquestes tasques. Fa el què hem explicat abans? (si no la veus fes que aparegui amb `Windows/show views/tasks`)

c> Torna a obrir el projecte Java original d'errors que has fet servir abans per corregir els errors:

<https://drive.google.com/file/d/0B-aT8EhHLcbycEpLX3RGdUdyQ0k/view?usp=sharing> HYPERLINK "https://drive.google.com/file/d/0B-aT8EhHLcbycEpLX3RGdUdyQ0k/view?usp=sharing&resourcekey=0-2NY6X9FxDgNoMEsAcUC1w"& HYPERLINK "https://drive.google.com/file/d/0B-aT8EhHLcbycEpLX3RGdUdyQ0k/view?usp=sharing&resourcekey=0-2NY6X9FxDgNoMEsAcUC1w"resourcekey=0-2NY6X9FxDgNoMEsAcUC1w

```

70 public static void main(String[] args) {
8    // TODO Auto-generated method stub
9
10   Scanner reader = new Scanner(System.in); // TODO falta import
11   String nom;
12
13
14   System.out.print("Hola! \nCom et dius? "); // TODO falta una T al print
15   nom = reader.next();
16   System.out.println("Hola " + nom.toString() + "!"); // TODO falta un punt i coma
17
18   System.out.print("Quans anys tens? ");
19   int edat = reader.nextInt(); // TODO falta declarar tipo de variable
20
21   if (edat >= 18) { // TODO falta abrir llave del if
22       System.out.println("Ja ets major d'edat");
23       System.out.println("Si tens el carnet, ja pots conduir un cotxe");
24   }
25   else {
26       System.out.println("Encara ets petit ");
27   }
28
29   reader.close();
30
31 }
32 } // TODO falta cerrar llave
33
34

```

7 items

Description	Resource	Path	Location	Type
TODO Auto-generated method stub	Errors.java	/correccioErrors/src/c...	line 8	Java Task
TODO falta import	Errors.java	/correccioErrors/src/c...	line 10	Java Task
TODO falta una T al print	Errors.java	/correccioErrors/src/c...	line 14	Java Task
TODO falta un punt i coma	Errors.java	/correccioErrors/src/c...	line 16	Java Task
TODO falta declarar tipo de variable	Errors.java	/correccioErrors/src/c...	line 19	Java Task
TODO falta abrir llave del if	Errors.java	/correccioErrors/src/c...	line 21	Java Task
TODO falta cerrar llave	Errors.java	/correccioErrors/src/c...	line 32	Java Task

d> Obre ara la vista de Problems. Què et mostra? Què passa si fas clic en alguna de les línies?

e> A continuació, obre la vista Outline. Què ens mostra aquesta vista? Quina utilitat creus que té?

f> Investiga quines altres vistes et poden ser útils per facilitar-te la feina de programador Java. Indica dues vistes i explica perquè serveixen.

Inclou captures de pantalla on es vegin la finestra del Problems i la d'Outline i la resposta a les preguntes anteriors (d> i e>) i les vistes addicionals que hakis mirat (f>).

