

# Recursos

## Objectius

- S'ha identificat el software necessari per a treballar amb Java
- S'ha instal·lat el JDK de Java
- S'ha instal·lat un IDE de treball per a Java
- S'ha utilitzat entorns integrats de desenvolupament
- S'han creat i executat projectes Java

## Java

Java és un llenguatge de programació orientat a objectes creat per la companyia *Sun Microsystems* en 1995, hui en dia és propietat d'*Oracle*. Va ser desenvolupat per James Gosling i Patrick Naughton.

Java és un llenguatge d'alt nivell, és a dir, similar al llenguatge natural.

S'utilitza per a: aplicacions mòbils (especialment aplicacions d'Android), aplicacions d'escriptori, aplicacions web, servidors, jocs, connexió a la base de dades, etc.

## Software

Per a executar aplicacions Java necessites el **JRE** (*Java Runtime Environment*) que inclou la màquina virtual de Java i les seues llibreries.

Per a desenvolupar aplicacions en Java, necessites el **JDK** (*Java Development Kit*), que inclou el JRE que afig eines i el llenguatge (des de JDK 11 el JRE no existeix per separat).

Pots baixar JDK de <https://www.oracle.com/java/technologies/downloads/>

Elegeix el fitxer instal·lable per al sistema que utilitzes.

Per a Windows el fitxer és "**jdk-18-windows-x64\_bin.msi**" (agafa la versió més actual).

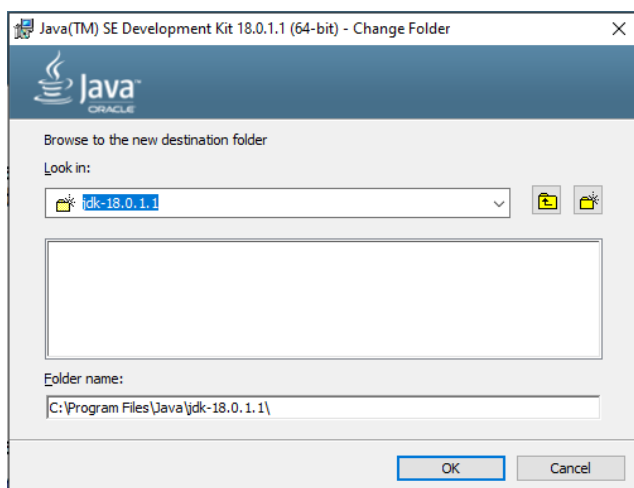
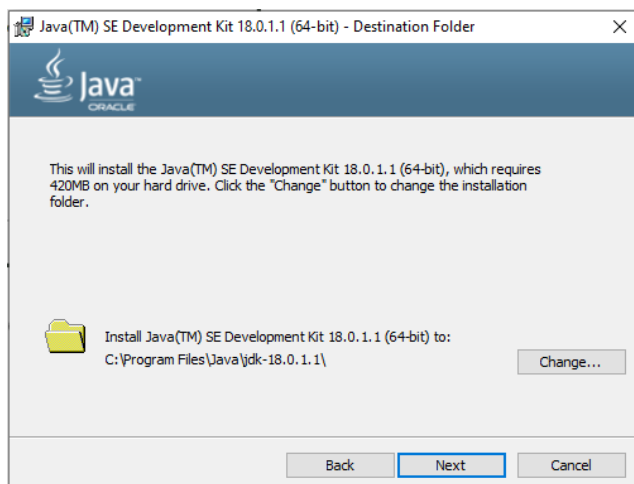
## Instal·lació del JDK

Has de tindre permisos d'administrador per a realitzar la instal·lació.



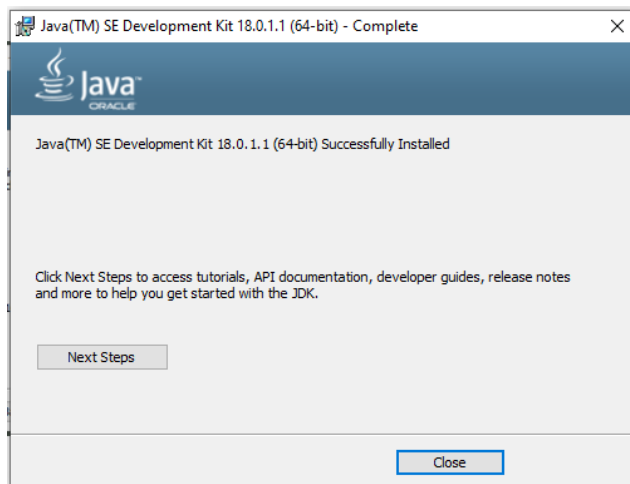


Pots canviar la carpeta on s'instal·la el JDK



Si vols, pots afegir més coses, llavors clica [Next Steps] que et duu a una pàgina web amb documentació i recursos per al JDK.





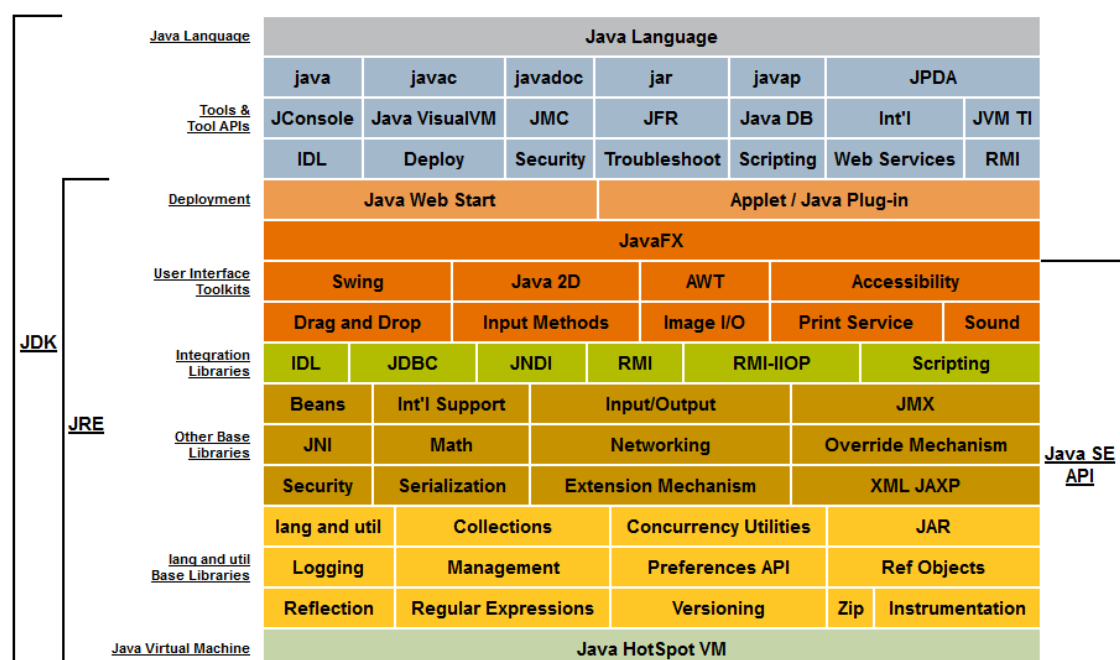
Per a comprovar si has instal·lat Java correctament, obri la consola de Windows (cmd) i executa la línia `java -version`. Ha d'aparèixer una eixida semblant a la següent

```
C:\Users\Administrador>java -version
java version "18.0.1.1" 2022-04-22
Java(TM) SE Runtime Environment (build 18.0.1.1+2-6)
Java HotSpot(TM) 64-Bit Server VM (build 18.0.1.1+2-6, mixed mode, sharing)
```

Això indica que s'ha instal·lat Java i que s'han definit els camins (PATH) als fitxers de Java correctament.

## Estructura

El diagrama conceptual de Java és

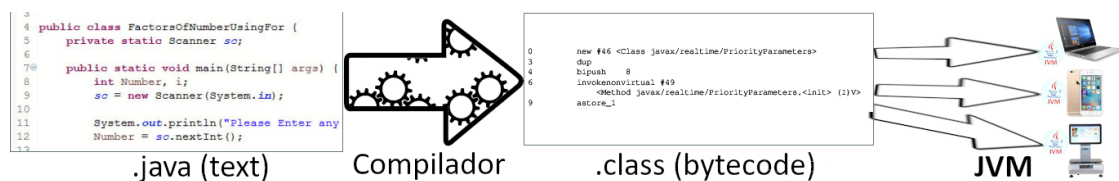


## Programa

Java és un llenguatge orientat a objectes i multiplataforma, la seua filosofia és "Escriure una vegada i executar en qualsevol dispositiu".

Les fases de desenvolupament d'un programa Java són:

1. L'escriptura del programa es realitza amb un editor de text. Els fitxers que es generen són fitxers de tipus text i tenen l'extensió **.java**
2. La compilació del programa la realitza el programa **javac**, és el compilador inclòs en el **JDK** (Java Development Kit). El compilador transforma el text d'entrada en **bytecode** (és el codi intermedi que executa la màquina virtual). El JDK conté totes les llibreries necessàries per al desenvolupament de programes en Java, a més de diverses eines, inclòs el **JRE** (Java Runtime Environment).
3. L'execució del programa la realitza la **Màquina virtual de Java** (Java Virtual Machine o **JVM**) que és part del JRE, és un programa que transforma el bytecode al codi que entén el sistema operatiu. Cada sistema operatiu té una JVM diferent, però, l'eixida que produeixen és la mateixa, per això **Java és independent de la plataforma**.



## Terminologia

Vegem termes de Java molt comuns.

**Java és un llenguatge orientat a objectes**, és a dir, els programes es plantegen com el maneig d'una col·lecció d'objectes (**tot en Java són objectes**). Un programa consistirà a definir les relacions entre els objectes.

Una **classe** és l'abstracció d'un objecte de la vida real, que es defineix mitjançant els seus atributs (camps) i funcionalitats (mètodes). Una classe és un fitxer de text amb extensió **.java**. Una classe es pot entendre com un **tipus de dada**. **Una classe és una plantilla** que ens permet crear els objectes.

Un **objecte** és una peça de programari (bytes en la memòria de l'ordinador) que reuneix les dades i la funcionalitat de qualsevol objecte del món real. Un objecte es crea amb la informació i les instruccions presents a la classe. Es poden crear molts objectes d'una mateixa classe. Els objectes es relacionen enviant-se **missatges** perquè executen algun dels seus **mètodes** (fer alguna acció o tornar informació).



classe Gos		objecte sultana	
		<div>Crear una instància</div> 	
Atributs	Mètodes	Atributs	Mètodes
color	assenta	color: negre i gris	assenta
raça	menja	raça: pastor alemany	menja
alçada	borda	alçada: 60 cm	borda
pes	corre	pes: 30 kg	corre

Un **paquet** és un contenidor que reuneix classes relacionades (pel tipus, per la comesa, per les dades que manegen, etc.), físicament es tradueix en una carpeta del sistema de fitxers. Una aplicació tindrà moltes classes, és millor ordenar tot aquest contingut, usant els paquets.

→ El nom d'un paquet s'escriu en minúscules.

Es pot repetir el nom d'una classe en paquets diferents, ja que per a accedir a una classe s'usa el camí complet que inclou el nom del paquet, per tant, encara que tinguin el mateix nom són classes diferents.

```
java.util.Date    // classe Date del paquet util que està en el paquet java
java.sql.Date     // classe Date del paquet sql que està en el paquet java
```

El nom és el mateix, ja que representen el mateix concepte, una data, però estan en paquets diferents i són classes diferents. La seua codificació és diferent, la primera és una data general i la segona una data enfocada a usar-se amb SQL.

## IDE

Qualsevol editor simple com el bloc de notes o Notepad és suficient per a escriure codi en Java encara que es recomana usar un IDE com NetBeans, Eclipse, IntelliJ, etc., ja que tenen algunes característiques que faciliten molt la programació com la revisió d'errors mentre s'escriu, l'autocompletat de noms de variables i funcions i moltes més coses (nosaltres usarem NetBeans).

## Netbeans

Podem baixar Netbeans de

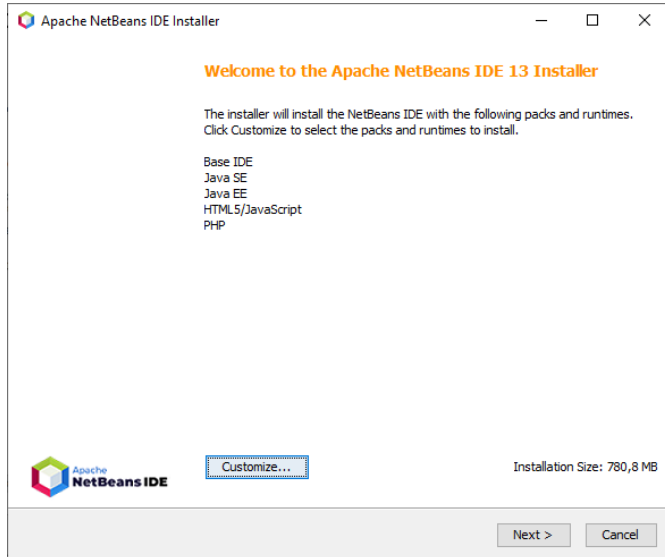
<https://netbeans.apache.org/download/index.html>



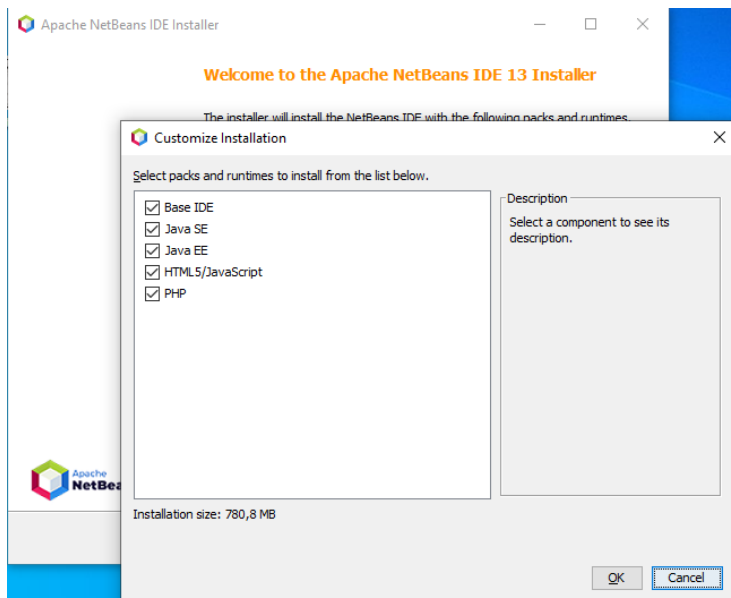
Si no és la primera instal·lació de Netbeans en la computadora, i vols una instal·lació 'neta' elimina de les carpetes de configuració de Netbeans de Windows que estan en

- C:\Users\Administrador\AppData\Local
- C:\Users\Administrador\AppData\Roaming

La instal·lació de Netbeans visualitza quins elements va instal·lar i quant espai ocuparan.

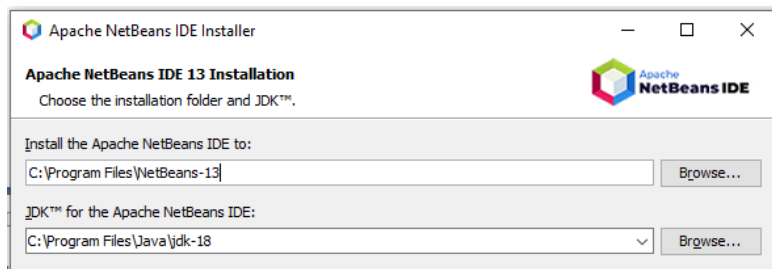


El botó **Customize...** dona la possibilitat de personalitzar la instal·lació, marcant quins elements vols.

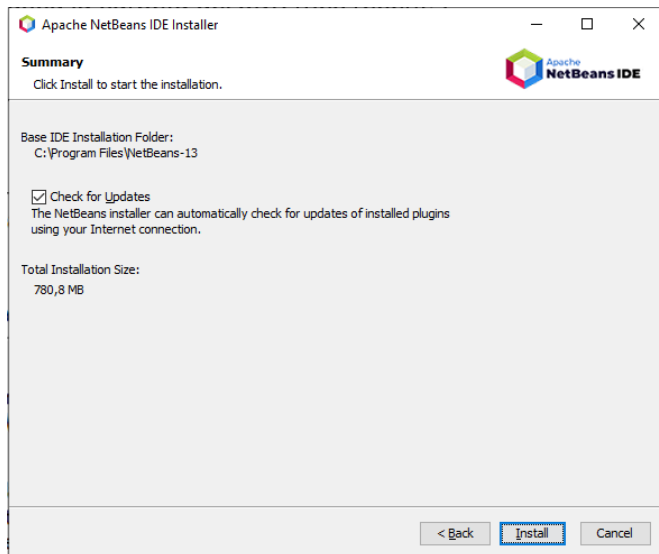


Després cal acceptar els termes de la llicència i visualitza la carpeta d'instal·lació i el JDK que utilitzarà per a l'IDE, si vols, pots canviar les carpetes.

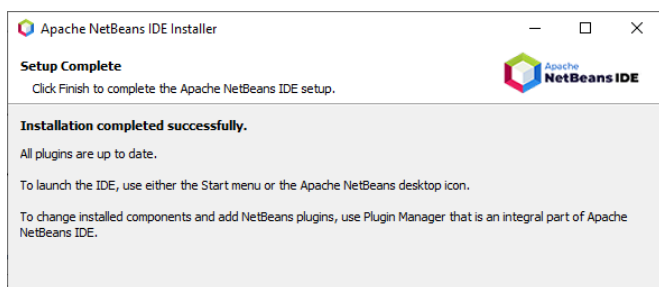




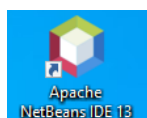
La instal·lació busca la carpeta del JDK més recent. En la computadora, pots tindre més d'un JDK instal·lat, i elegir el que vols associar a l'IDE per defecte. Pots tindre més d'un JDK associat a l'IDE, el JDK per defecte és el que elegeixes en la instal·lació.



Visualitza una finestra on s'indica que es buscaran actualitzacions de forma automàtica, prem el botó **Install**.



La icona de NetBeans que apareix en l'escriptori és

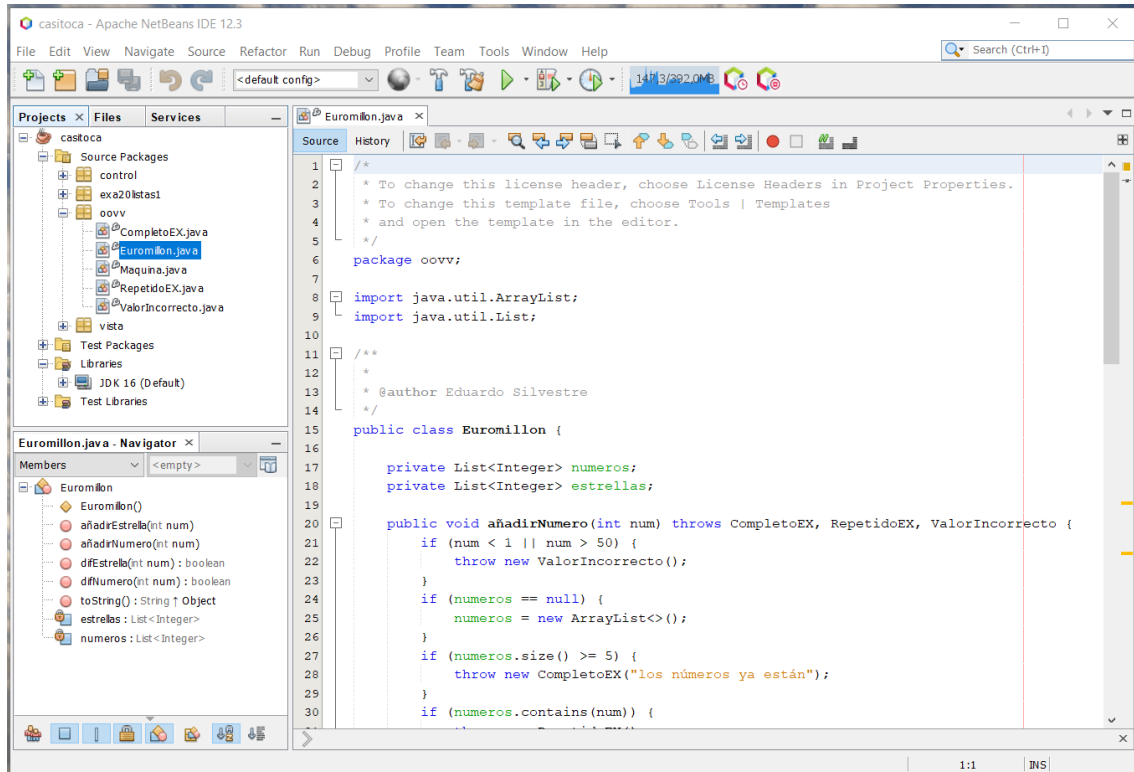


La primera vegada que fas una acció s'activa el/els mòduls necessaris. Vas a treballar amb Java SE, a l'hora de crear el primer projecte aquest s'activa. També, s'activa la llibreria [nb-javac](#) de compilació de Java i d'edició (Si no funciona correctament, hauràs de descarregar el plugin [nb-javac](#) i instal·lar-lo).



## Àrea de treball

En funció del que estàs fent, tindràs obertes finestres diferents, l'aspecte més habitual és el següent que correspon a l'edició d'un fitxer `.java`



En la imatge, les finestres obertes són: [Projects](#), [Files](#), [Services](#), [Navigator](#) i [Editor](#)

En l'opció de menú [Window](#) tens totes les finestres que ofereix l'IDE, si vols obrir una finestra, has de buscar-la allí.

La majoria de les finestres tenen associat un menú contextual amb les opcions aplicables a la finestra o al seu contingut.

La majoria de les finestres tenen associada una drecera de teclat, una combinació de tecles que obri la finestra (també, passa amb les opcions del menú).

Si l'àrea de treball s'ha 'desbaratat' molt, pots recuperar la configuració inicial amb l'opció [Window > Reset Windows](#).


## Crear un projecte

Per a crear les nostres aplicacions, sempre haurem de crear un projecte que reunisca, les classes, els recursos, les llibreries i els fitxers de configuració que relacionen tots els elements i permeten construir el projecte per a poder executar l'aplicació.

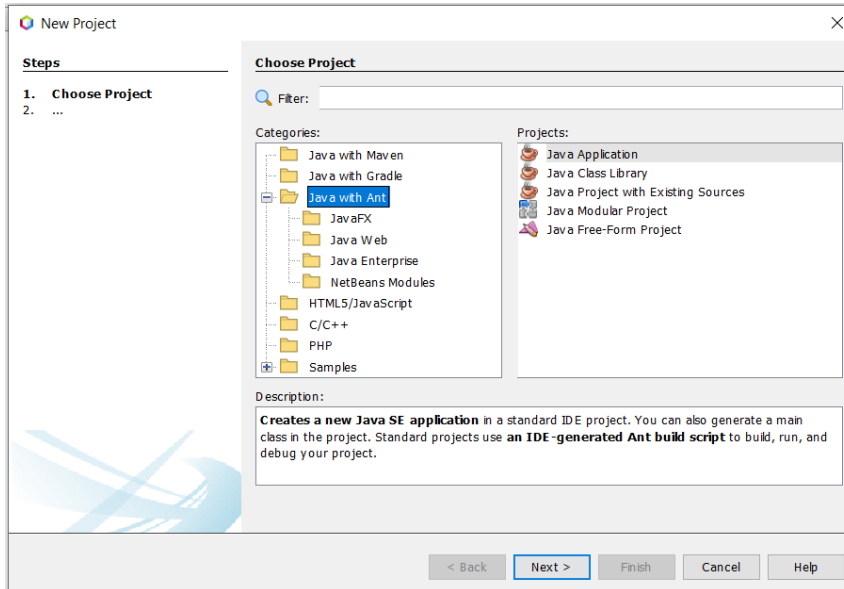




## Nou projecte

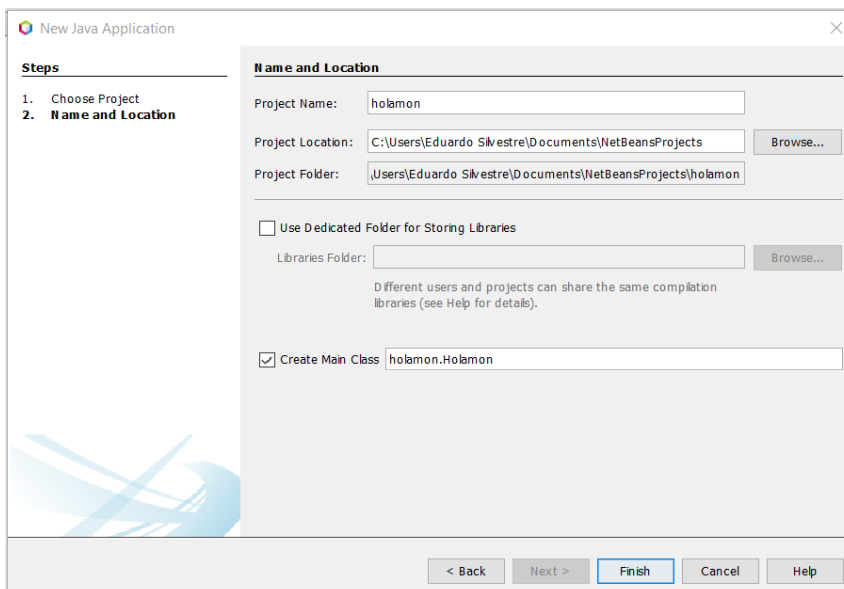
Per a crear un nou projecte, pots prémer **Ctrl+Shift+N**, fer clic sobre la icona  o obrir el menú contextual de la finestra **Projects** i triar l'opció **New Project...**

En **Categories** marca **Java with Ant** i en **Projects** marca **Java Application**



**Apache Ant** és una ferramenta usada per a la realització de tasques mecàniques i repetitives, normalment durant la fase de compilació i construcció.

Prem **Next** i s'obri la finestra que permet definir el projecte



El nom del projecte s'assigna al nom de la carpeta que guarda els fitxers del projecte en el sistema operatiu i per al paquet inicial (un paquet és un contenidor lògic).

→ El nom del projecte s'escriu en minúscules, evita els espais en blanc i els accents.



La localització del projecte, és la ruta a la carpeta del projecte. Si poses un nom de carpeta que no existeix, llavors, aquesta carpeta es crea.

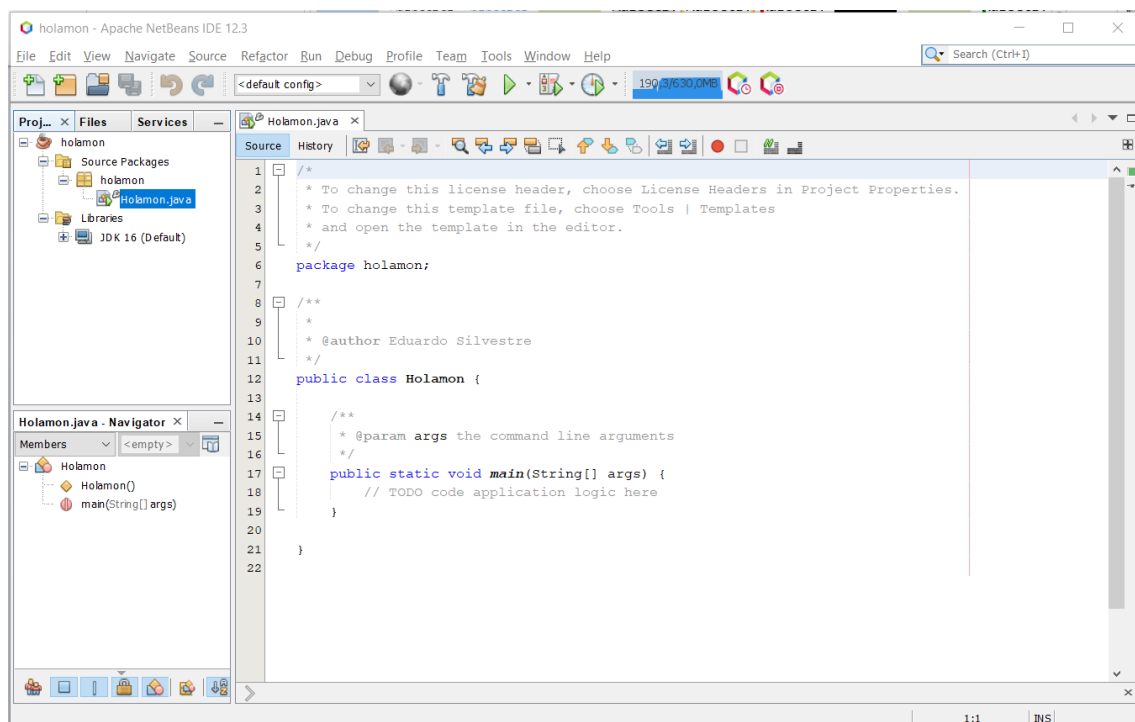
Si marques [Use Dedicated ...](#), llavors es crea una carpeta `lib` en el nostre projecte, això es fa per a compartir llibreries entre projectes (no usarem llibreries dedicades).

Si marques [Create Main Class](#), llavors es crea la classe principal amb el mètode `main()` (on s'iniciarà l'execució del projecte). El nom de la classe està format pel paquet inicial, seguit de punt (.) seguit del nom de la classe principal (la primera lletra d'una classe està en majúscules), en l'exemple de la figura tenim `holamon.Holamon`. Es pot canviar el nom del paquet o el nom de la classe.

→ Un projecte, sempre, ha de tindre una classe principal.

Prem **Finish** per a crear el projecte.

## El projecte en el IDE



En la pestanya [Projects](#), tens l'estructura lògica del projecte que has creat

- holamon (el nom del projecte)
  - Source Packages (paquets del codi font)
    - holamon (paquet inicial)
      - Holamon.java (classe principal amb el `main`)
  - Libraries (llibreries per al projecte)
    - JDK 16 (llibreria del JDK)

En la finestra d'edició tens el codi de la classe `Holamon`. En l'exemple, tens la classe principal amb el seu mètode `main`.



El projecte s'estructura a base de paquets, un **paquet** és un contenidor d'elements relacionats. Per a afegir un paquet selecciona **Source Package** o un altre paquet, i amb el menú contextual tria **New > Java Package**.

Les classes s'afigen als paquets, per a afegir una classe selecciona el paquet, i amb el menú contextual tria **New > Java Class**.

Els paquets poden contindre altres elements, enumeracions, finestres, imatges, etc.

En **Libraries** estan totes les llibreries necessàries per al projecte, per defecte està el JDK, si necessites alguna llibreria més especialitzada, llavors, cal afegir-la.

## Source Packages

És on estan les classes amb el codi font de l'aplicació, tot el codi es reparteix en classes diferents, cadascuna fa el seu treball.

Un projecte ha de tindre una classe principal amb el mètode **main**, ja que la JVM inicia l'execució de l'aplicació en el **main** de la classe principal. Pots canviar la classe principal del projecte en les propietats del projecte, apartat **Run**.

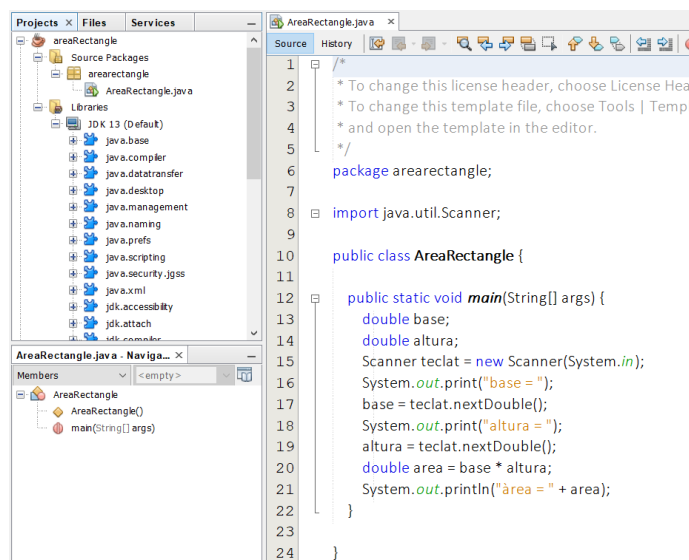
El nombre de classes pot ser molt gran, cal ser ordenat, per tant usa paquets que reuneixen classes que tenen alguna cosa en comú.

## Llibreries

Les llibreries de l'aplicació proporcionen classes i mètodes realitzats per altres persones.

Les llibreries que usa l'aplicació apareixen en **Libraries**, per defecte, es carrega la llibreria del llenguatge Java, és a dir, el JDK 13 (la versió pot variar)

En l'exemple es vol llegir de teclat, per a això s'usa la classe **Scanner** que està en el paquet **java.util**, que està en el mòdul **java.base** del JDK.



Per a usar una classe s'ha d'escriure el camí complet, paquet(s) (separats per punt) i classe (al final), en cada ús.

```
java.util.Scanner lector = new java.util.Scanner(System.in);  
// la classe Scanner està en el paquet util que està en el paquet java
```



Pots obviar l'escriptura del paquet si, prèviament, has importat la classe amb el seu nom complet. Les importacions es fan a l'inici del fitxer, l'ús de la classe en el codi.

```
import java.util.Scanner; // s'importa la classe amb el seu nom complet

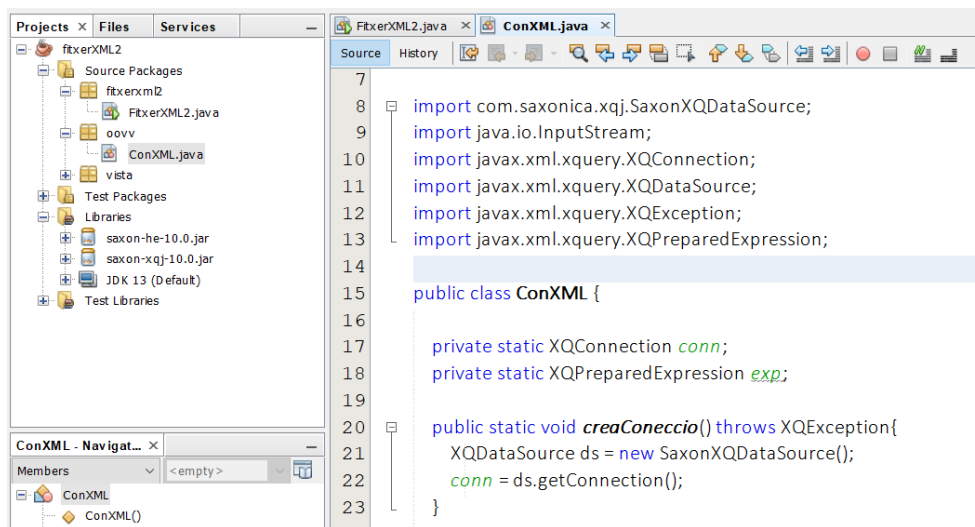
Scanner lector = new Scanner(System.in); // s'escriu, únicament, la classe
```

El paquet `java.util` té moltes més classes, pots veure-les desplegant en **Libraries** el mòdul `java.base` del JDK i després desplegant el paquet `java.util`

L'únic paquet que no cal importar és el paquet del llenguatge de Java `java.lang` (sempre està accessible), la classe `System` (el sistema) i la classe `Math` (operacions matemàtiques) pertanyen al paquet `java.lang`.

```
System.exit(0); // el mètode exit de la classe System finalitza l'execució d'una aplicació
double arrel = Math.sqrt(1234); // el mètode sqrt de la classe Math calcula l'arrel quadrada d'un valor
```

En internet hi ha moltes llibreries, i les pots afegir a l'aplicació, llavors podràs usar totes les classes que conté (**API**). Les llibreries són fitxers comprimits amb extensió `.jar`. En el tema dedicat a crear les nostres pròpies classes, veurem com crear llibreries `.jar` amb elles.



En l'exemple de la imatge s'han afegit les llibreries `saxon-he-10.0.jar` i `saxon-xqj-10.0.jar` que serveixen per a tractar arxius XML.

Hi ha moltes llibreries disponibles, per exemple `swingX`, és una llibreria que millora alguns components de Swing que s'usen en l'entorn visual. El fitxer que conté la llibreria s'anomena `swingx-all-1.6.4.jar`, es pot descarregar de

<https://java.net/downloads/swingx/releases/>

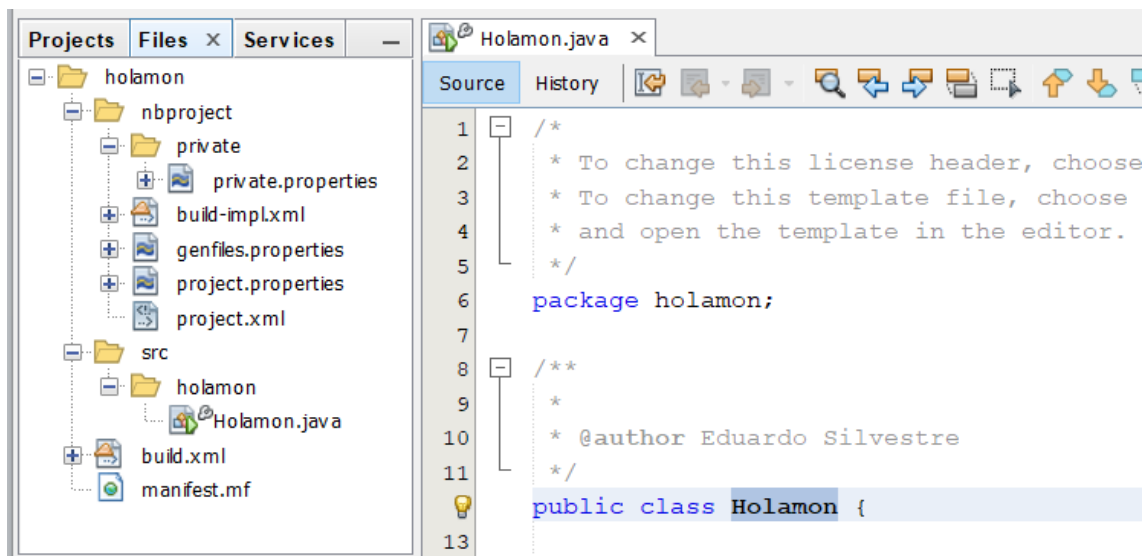
Si vols utilitzar-la, has d'afegir el fitxer `.jar` a la llibreria del projecte. Prem el botó dret sobre **Libraries** i tria l'acció **Add JAR / Folder** que afegirà un fitxer `.jar` o una carpeta a les llibreries del projecte.



## Estructura en disc

En la pestanya **Files**, està l'estructura física del projecte, és a dir, les carpetes i els fitxers que conformen el projecte en el disc dur (el que podem veure amb l'explorador de windows).

- nomProjecte (carpeta del projecte)
- build (carpeta amb les classes traduïdes)
- dist (carpeta amb el **.jar** executable del projecte)
- nbproject (carpeta amb les propietats i el fitxer ant build-impl.xml amb la implementació del build)
- src (carpeta amb els fitxers fonts de les classes)
- build.xml (fitxer ant per a la construcció del projecte)
- manifest.mf (informació de quina mena de fitxers conté el fitxer **.jar**, en un projecte com és la classe principal)



## Estructura mínima per a un projecte

En Java tot el que es maneja són objectes que es creen a partir de classes.

L'estructura mínima per a un projecte de Java és, un paquet amb una classe que té el mètode **main**. Aquesta classe és la classe principal i el mètode **main** és el mètode que busca la màquina virtual per a iniciar l'execució .

```
package paquet;

public class Principal {

    public static void main(String[] args) {
        // TODO codi de la lògica de la aplicació
    }

}
```



→ La capçalera del mètode `main` és la següent, i no s'ha de canviar.

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

`public` indica que el mètode és accessible per qualsevol altre objecte

`static` indica que només hi ha una còpia del mètode

`void` indica que el mètode no retorna res

`main` és el nom del mètode

`(String[] args)` són els arguments de la línia de comandos (una matriu de cadenes)

La capçalera d'un mètode indica com s'ha de cridar eixe mètode per a executar-lo.


En l'exemple del projecte `holamon`, el codi del `main` és

```
package holamon;

public class Holamon {

    public static void main(String[] args) {
        System.out.println("Hola mon");
    }

}
```

que visualitza el text "`Hola mon`" en la finestra `Output` (la consola de NetBeans) quan s'executa el projecte prement  o la tecla **F6**, o seleccionant l'opció de menú `Run / Run Project`.



The screenshot shows the NetBeans IDE with the `Holamon` class open. The code is as follows:

```
12 public class Holamon {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         System.out.println("Hola mon");
19     }
20
21 }
```


Below the code editor, the `Output - holamon (run)` window is visible, showing the following output:

```
Created dir: C:\Users\Eduardo Silvestre\Documents\NetBeansProjects\holamon\build\classes
Created dir: C:\Users\Eduardo Silvestre\Documents\NetBeansProjects\holamon\build\empty
Created dir: C:\Users\Eduardo Silvestre\Documents\NetBeansProjects\holamon\build\generated-sources\ap-source-output
Compiling 1 source file to C:\Users\Eduardo Silvestre\Documents\NetBeansProjects\holamon\build\classes
compile:
run:
Hola mon
BUILD SUCCESSFUL (total time: 2 seconds)
```

Pots executar, independentment de l'aplicació, qualsevol classe que tinga el mètode `main` prement **Shift+F6** o amb l'opció `Run File` del menú emergent, només s'executa el fitxer d'aquesta classe.

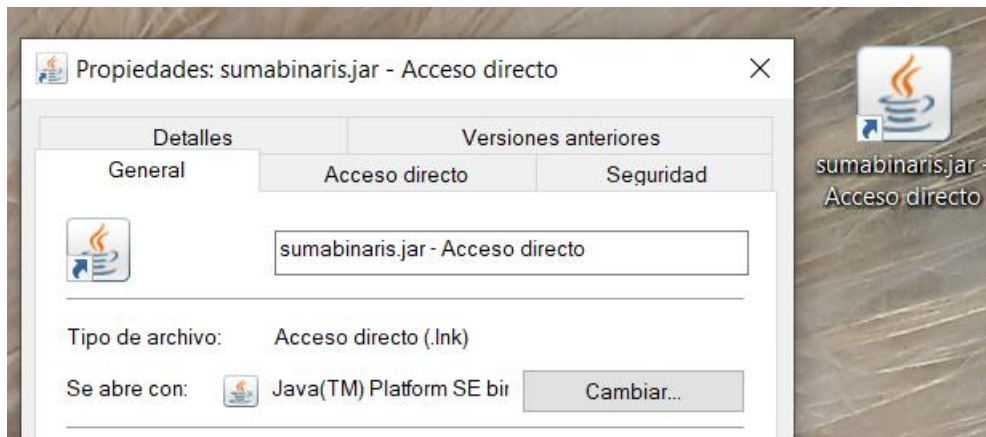


## Fitxer .jar executable

L'opció **Clean and Build Project**,  crea el fitxer .jar executable del projecte que es guarda en la carpeta **dist** del projecte.

Si el PATH de Windows està ben configurat per a accedir Java (conté la ruta **C:\Program Files\Common Files\Oracle\Java\javapath**), llavors, un doble clic sobre el nom del fitxer .jar executa el projecte (ha de ser visual).

Pots crear un accés directe al fitxer .jar i indicar que s'obri amb Java (TM) Platform SE binary (la màquina virtual de Java).



Si tens la màquina virtual en el PATH, cal posar la ruta completa a **javaw -jar** que executa un fitxer .jar en l'entorn de Windows, per exemple

```
"C:\Program Files\Common Files\Oracle\Java\javapath\javaw.exe" -jar  
"C:\Users\EduardoSilvestre\Documents\programacio\sumabinaris\dist\sumabinaris.jar"
```

## Estructura d'un fitxer font de Java

### Classe

Una classe representa una peça de programari de l'aplicació, amb ella es creen els objectes que maneja l'aplicació. La capçalera d'una classe està formada pel tipus d'accés, la paraula reservada **class** i el nom de la classe.

```
public class nomClasse {  
}
```

El codi de la classe s'escriu dins de les claus **{ }** que indiquen l'inici i el final de la classe. Una classe es crea en un paquet.

```
package holamon;           // està en el paquet holamon  
public class Holamon {     // la classe és pública i s'anomena Holamon  
}
```



## Mètodes

Els mètodes de la classe, són les accions que pot realitzar una classe. Els mètodes s'escriuen dins de les claus de la classe. El codi d'un mètode s'escriu dins de les claus {} que indiquen l'inici i el final del mètode.

Una classe ha de tindre com a mínim un mètode (si no té cap mètode, no fa res), normalment una classe té molts mètodes.

```
private tipusRetornat nomMetode(paràmetres) {  
}
```

La capçalera d'un mètode està formada pel tipus d'accés, el tipus de la dada retornada o void si no retorna res, el nom del mètode i els paràmetres

```
public static void main(String[] args) { // el mètode és públic, estàtic, no retorna res, s'anomena main i rep args  
}
```

## Sentències

Les sentències són les instruccions del llenguatge Java que permeten realitzar les accions associades al mètode.

Una sentència acaba el punt i coma o està tancada entre claus.

```
System.out.println("Hola mon"); // visualitza la cadena de text "Hola mon" en la consola
```



### Exercicis

#### 1. versioJDK

Crea un projecte de Java anomenat **versioJDK**. El codi de la classe principal és el següent

```
public class VersioJDK {  
    public static void main(String[] args){  
        System.out.println("Versió JDK: " + System.getProperty("java.version"));  
        System.out.println("Versió JRE: " + System.getProperty("java.runtime.version"));  
        System.out.println("Carpeta de Java: " + System.getProperty("java.home"));  
        System.out.println("Venedor de Java: " + System.getProperty("java.vendor"));  
        System.out.println("Ruta d'aquesta classe: " + System.getProperty("java.class.path"));  
    }  
}
```

#### 2. arearectangle

Crea un projecte de Java anomenat **arearectangle**. Còpia el codi del càlcul de l'àrea d'un rectangle i executa'l. El codi està en la imatge de l'apartat "llibreries" on apareix la pestanya d'edició amb el fitxer "AreaRectangle.java".

