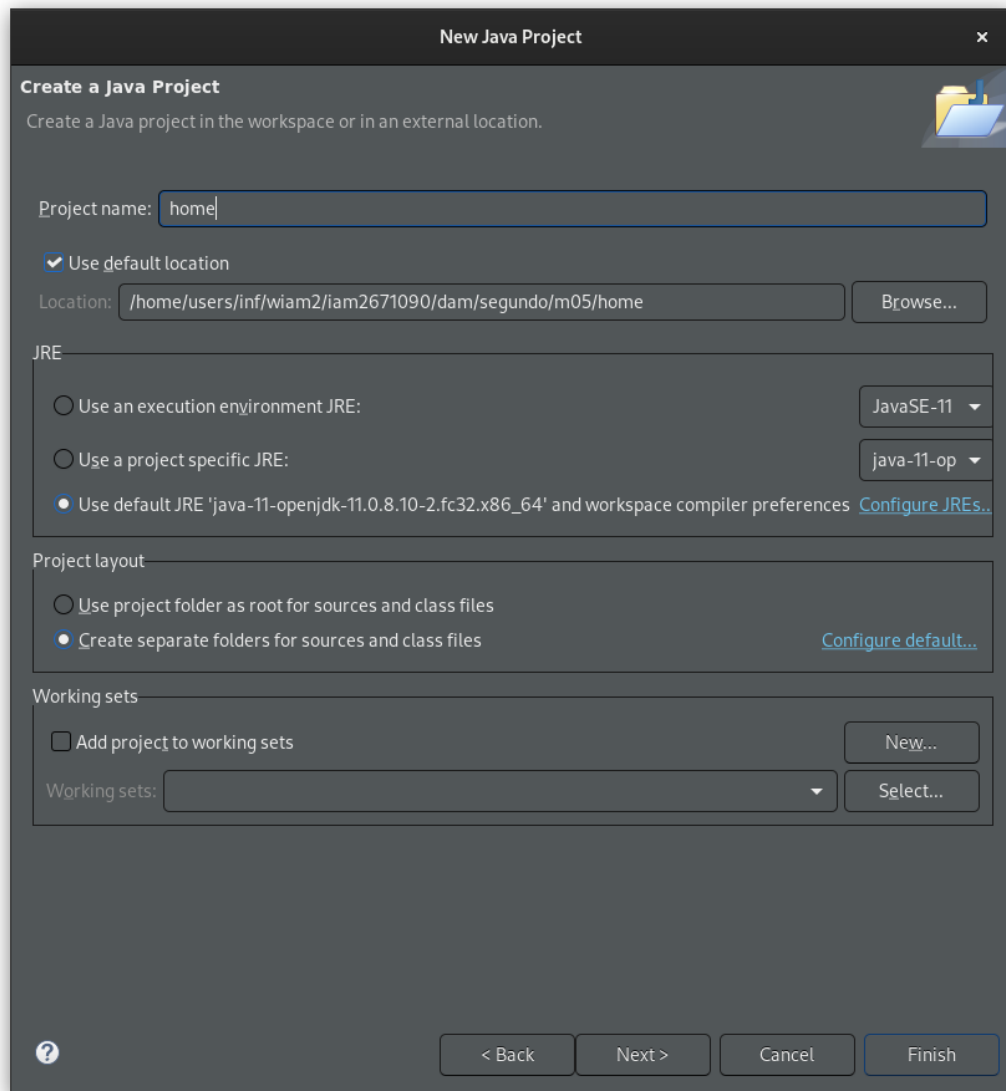


Eclipse. Exercicis 1

1.- Primer Projecte Java

Crea el projecte **hola**. El programa imprimirà "Hola Món" per pantalla. Usa les següents ajudes d'Eclipse:

- Creació del projecte
[New->Project->JAVA Project](#)



- Creació d'una classe

Click derecho en home->src y después click en New->Class

- Autocompletat

Para activarlo hay que ir a Window->Preferences->Java->Editor->Content Assistant y, en el campo Auto activations triggers for Java, añadir, después del punto, el abecedario. Se guardan los cambios haciendo click en Apply and Close.

- Provoca algun error i fes que l'Eclipse t'ajudi a resoldre'l

Se cambiará el método println por println (incorrecto)

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

```
...c void main(String[] args) {  
...ut.println("Hola");  
...  
The method println(String) is undefined for the type PrintStr  
3 quick fixes available:  
  - Change to 'print(..)'  
  - Change to 'println(..)'  
  - Add cast to 'System.out'  
Press 'F2' f
```

El error se resolvió escogiendo la opción «Change to 'println(...)'»

- Prova el Formatador de codi

Se ha hecho el program "Hola Mundo" en una sola línea

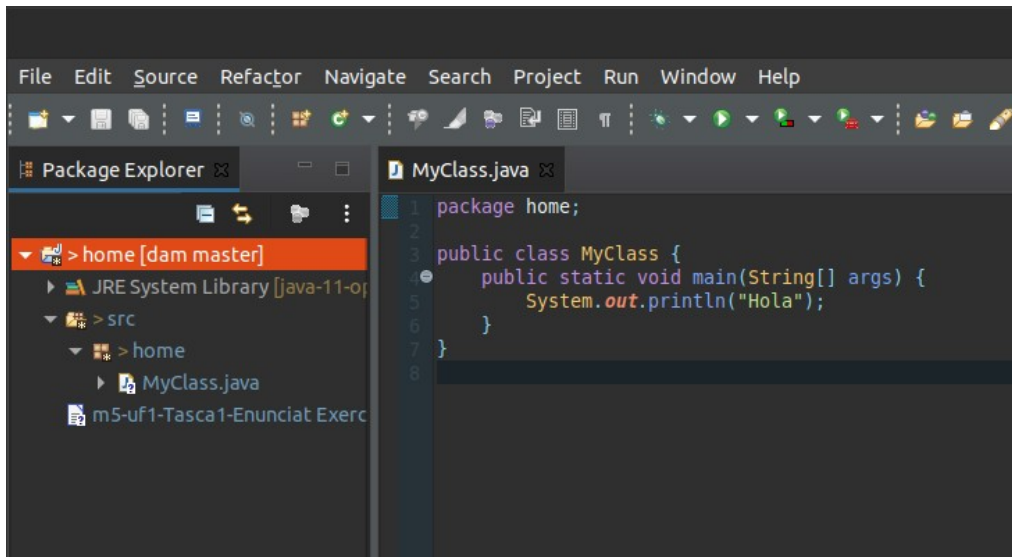
```
MyClass.java x  
1 package home;public class MyClass {public static void main(String[] args) {System.out.println  
2
```

Después de usar **Ctrl+Mayus+f** se ha puesto el formato visualmente más entendible

- On es guarda el projecte creat? Es pot indicar una ubicació diferent?

El proyecto se ha guardado dentro del workspace elegido al abrir eclipse, una de las formas de mover el proyecto es hacer click derecho sobre la carpeta «raíz» del proyecto, e ir a **Show In->System Explorer**, desde ahí se abrirá el explorador de archivos del sistema, también se puede usar **Terminal** en lugar de **System Explorer**, para usar la terminal.

Enganxa una captura de pantalla de teu IDE amb el projecte.



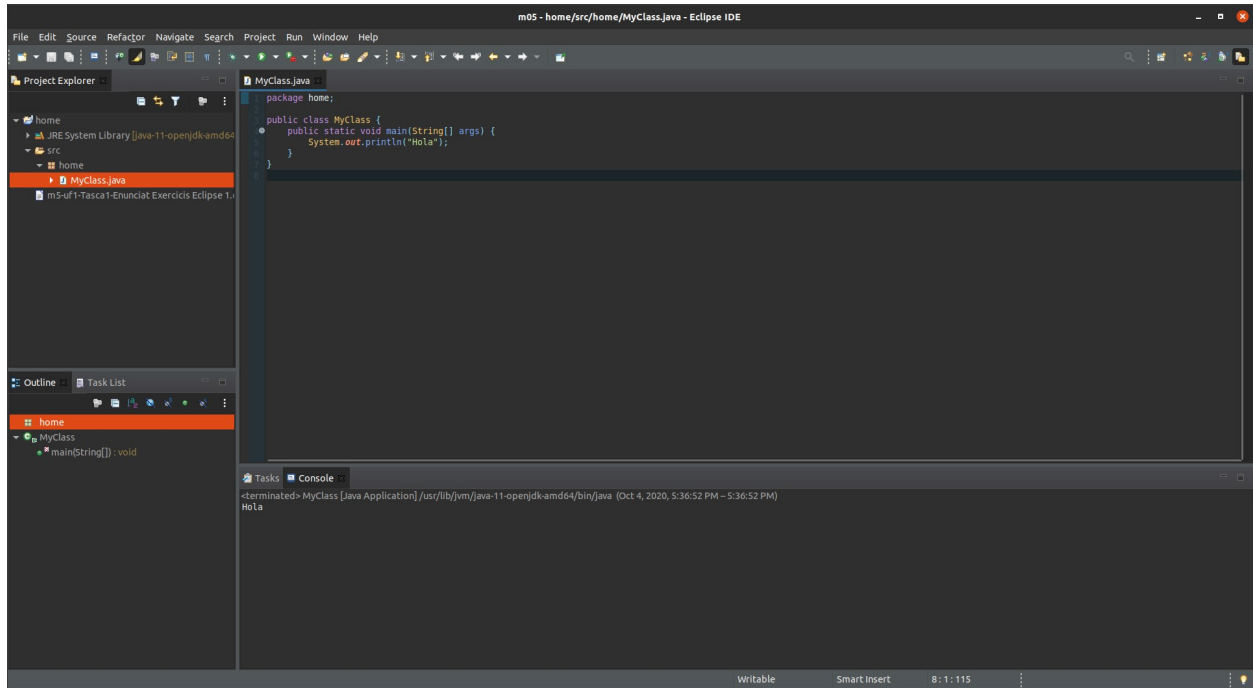
2.- Perspectives

Obre la perspectiva de “Recursos” (Resources).

Explica com ho has fet.

Enganxa una captura de pantalla de teu IDE amb aquesta perspectiva.

Click en Window->Perspective->Open Perspective->Other y seleccionar la llamada Resources.



3.- Vistes

Imagina que en el la perspectiva amb la que estas treballant no es veu la Vista “Project Explorer”. Com es pot fer per tenir-la visible?

Haciendo click en Window->Show View->Project Explorer

4.- Projecte Java

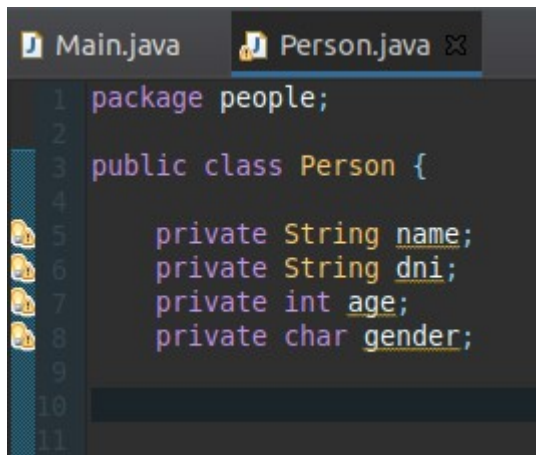
Crea el projecte **people** que contingui dues classes: la classe **Person** i la classe **Main**. Poseu a la classe Person tots aquells atributs que caracteritzen a una persona i useu la classe Main per crear una persona i mostrar els seus atributs. Useu les següents ajudes:

- Generar constructors, getters i setters.
- Generar equals & hashCode de la classe Person.
- Generar toString de la classe Person.
- Canvia el nom d'algun atribut usant refactoring. Canvia'l a tot arreu.

Nota: en tots aquests wizards mireu les diferents opcions que us donen.

Enganxa les captures de pantalla necessàries.

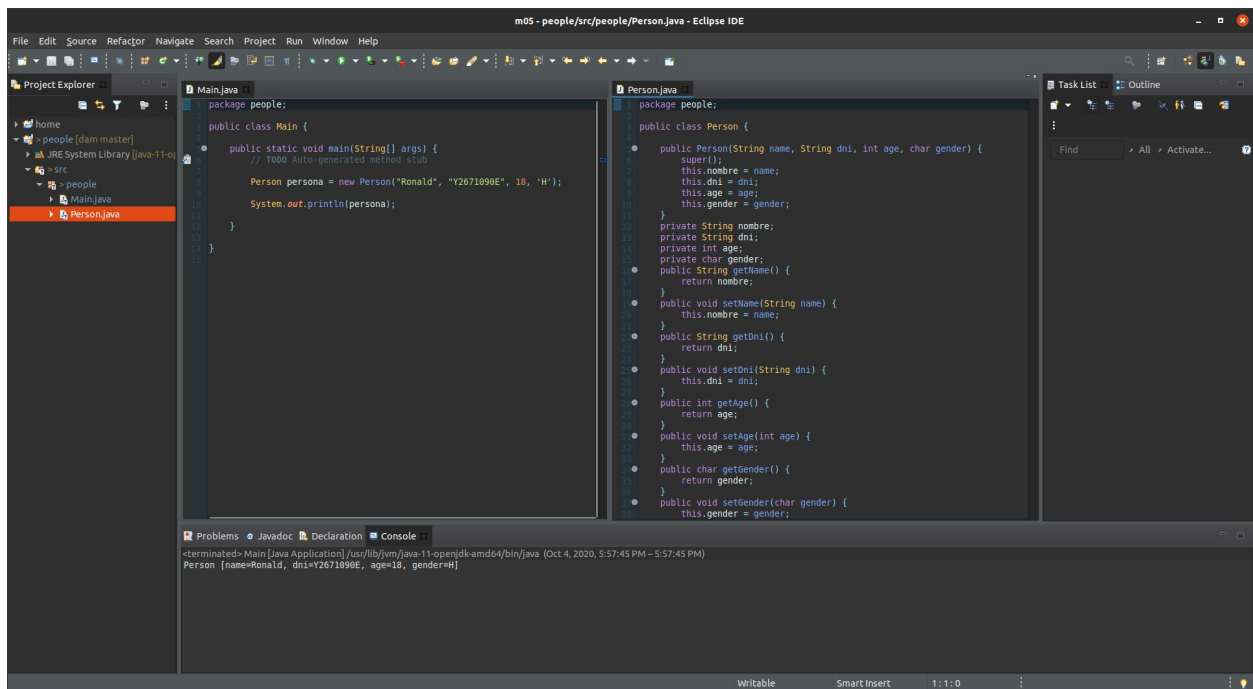
Se ha creado la clase Person y se le ha añadido los siguientes atributos:



```
1 package people;
2
3 public class Person {
4
5     private String name;
6     private String dni;
7     private int age;
8     private char gender;
9
10
11 }
```

Haciendo click derecho (Desde Project Explorer) en la clase Person, y dirigiéndonos a **Source**, se podrán generar los constructores, getters y setters, generar los metodos equals y hashCode, así como también generar el método toString. En cada una de estas opciones podremos elegir qué atributos usar y cuales no, en qué parte del código queremos insertar los métodos, o poner el acceso público, protected, etc.

Para cambiar el nombre de alguna variable usando refactoring, hay que poner el cursor sobre la variable a la cual le cambiaremos el nombre y presionar **Alt+Shift+r**, cambiamos el nombre y confirmamos con **Enter**.



```
package people;

public class Main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Person persona = new Person("Ronald", "Y2671090E", 18, 'H');
        System.out.println(persona);
    }

}

package people;

public class Person {

    public Person(String name, String dni, int age, char gender) {
        super();
        this.nombre = name;
        this.dni = dni;
        this.age = age;
        this.gender = gender;
    }

    private String nombre;
    private String dni;
    private int age;
    private char gender;

    public String getName() {
        return nombre;
    }

    public void setName(String name) {
        this.nombre = name;
    }

    public String getDni() {
        return dni;
    }

    public void setDni(String dni) {
        this.dni = dni;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public char getGender() {
        return gender;
    }

    public void setGender(char gender) {
        this.gender = gender;
    }

}
```

terminated> Main [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (Oct 4, 2020, 5:57:45 PM - 5:57:45 PM)
Person [name=Ronald, dni=Y2671090E, age=18, gender=H]

5.- User Libraries

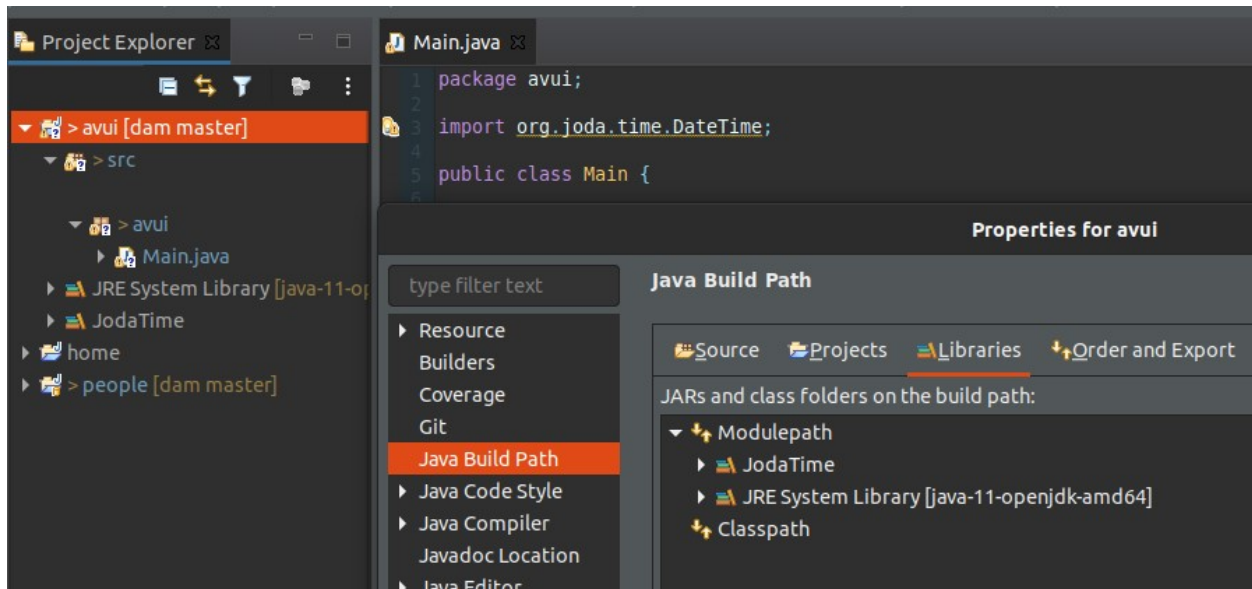
Crea el projecte **avui**. Aquest projecte imprimirà la data actual per pantalla. Instal·la la llibreria JodaTime i estableix-la com a **llibreria d'usuari** de manera que sigui molt fàcil tornar-la a usar des d'un altre projecte.

Exercici per veure les diferents maneres d'afegir una llibreria a un projecte.

Expliqueu i documenteu les diferents maneres de fer-ho que heu trobat

Hay que ir a Window->Preferences->Java->Build Path->User Libraries. Hacer click en **New** y añadimos el nombre de la librería. Cuando se haya añadido a la lista de librerías, la seleccionamos y hacemos click en **Add External JARs**, se abrirá el explorador de archivos del sistema y tendremos que elegir el archivo .jar de la librería descargada. Con eso tendremos la librería en el workspace.

Para usarla en algún proyecto, hay que hacer click derecho en la carpeta «raíz» del proyecto e ir a Properties->Java Build Path, y seleccionar la pestaña Libraries, hacemos click en Add Library, seleccionamos User Library, y por último seleccionamos la librería que queremos usar. Finalmente se hace click en finish y Apply para guardar cambios.



6.- Fixing a project:

- Importa el projecte **telematic**.
Se hace click en File->Open Projects from File System, y seleccionamos la carpeta descomprimida.
- Afegeix la llibreria **joda-time** al Build-path.

Hay que hacer click derecho en la carpeta «raíz» del proyecto e ir a Properties->Java Build Path, y seleccionar la pestaña Libraries, hacemos click en Add Library, seleccionamos User Library y por último seleccionamos la librería JodaTime.

- Canvia el nom de la classe **TelematicTest**, posa-li la classe **Main** i desa-la en un altre package de nom **appXYZ (les teves inicials!)**.

Para renombrar la clase TelematicTest se hace click derecho en el archivo y se va a Refactor->Rename, o (Shift+Alt+R), al cambiarle el nombre y dar a **next** pueden aparecer advertencias, se le da a Next si se cree conveniente (en este caso si).

Para moverlo de package, se ha creado uno nuevo haciendo click derecho en la carpeta src y dándole click a New->Package, se le pone el nombre y se le da a finish para finalizar. Por último se puede arrastar el archivo Main.java hasta la carpeta del nuevo package creado.

- Canvia el nom del package **telematic** i anomena'l **model**.

Seleccionamos el package, Shift+Alt+R, se le pone el nombre model, y por último finish.

- Crea la classe **UserWithOffer** sense escriure cap línia de codi (genera tot el codi necessari a partir de l'ajuda d'Eclipse).

Al ser una classe que extiende de User, en el moment de crear la classe, en el apartat Superclass, hacemos click en Browse para buscar la classe User. Seleccionamos la opción **Constructors from superclass** y nos aseguramos de que sea abstarcta.

New

Java Class
Create a new Java class.

Source folder:

Package:

☐ Enclosing type:

Name:

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

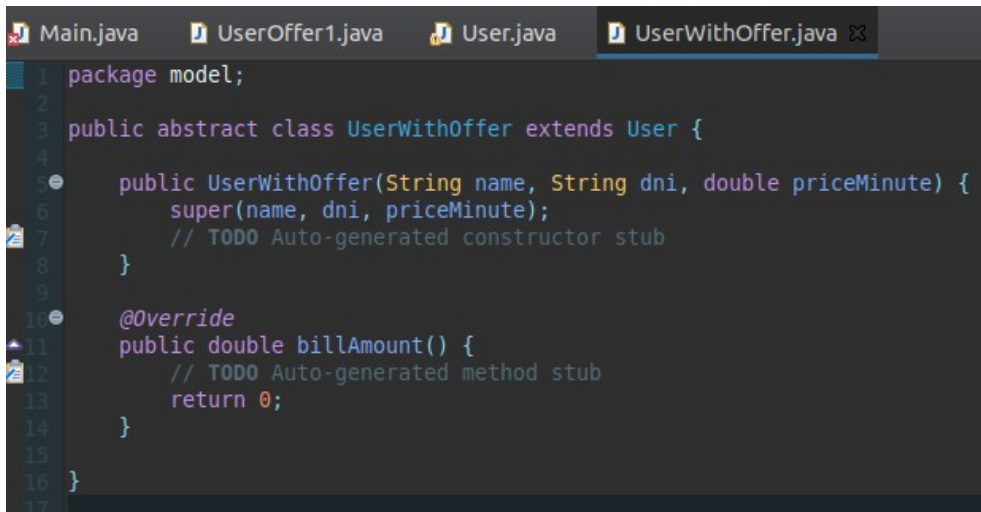
Superclass:

Interfaces:

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

Con esas opciones se habrá creado el siguiente código:



```
1 package model;
2
3 public abstract class UserWithOffer extends User {
4
5     public UserWithOffer(String name, String dni, double priceMinute) {
6         super(name, dni, priceMinute);
7         // TODO Auto-generated constructor stub
8     }
9
10    @Override
11    public double billAmount() {
12        // TODO Auto-generated method stub
13        return 0;
14    }
15
16 }
```

- A la classe **User**, genera l'equals i el hashCode i els getters deis atributs name, dni i connections.

Haciendo click derecho, después en source, y escogiendo las opciones correspondientes.

- Canviar els noms de les classes que contenen la paraula **User** per **Customer**.

Se hace click derecho en el archivo y se va a Refactor->Rename, o (Shift+Alt+R).

- Formata el codi.

Ctrl+MAYUS+F

- Exporta el projecte de dues maneres:
 - Crea un comprimido .tar.gz amb el projecte Eclipse

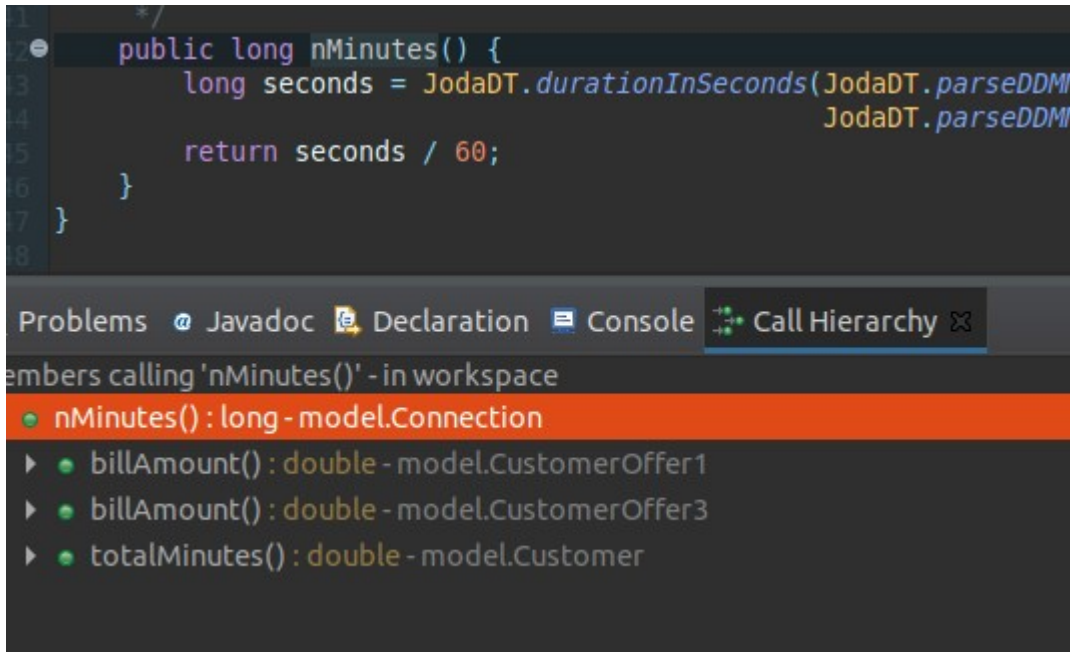
Click derecho en el proyecto e ir a Exportar. Seleccionar General->Archive File. En el apartado **to archive file**, seleccionamos a ruta a la cual exportar. Y, en el apartado **Options** seleccionamos **Save in tar format**.

- Crea un .jar executable

Esta vez al exportar seleccionamos Java->JAR file, escogemos el destino de exportación y listo.

- Des d'on es crida el mètode **Connection->nMinutes()** ?

Seleccionamos el método y presionamos Ctrl+Alt+H, y se abrirá una pestaña donde se mostrará las llamadas al método.



7.- Preguntes

Contesta les següents preguntes:

1. Amb quina tecla o combinació de tecles pots forçar l'autocompleció de codi?

Ctrl+space

2. Amb quina tecla o combinació de tecles pots forçar la formatació del codi?

Ctrl+Mayus+F

3. Amb quina tecla o combinació de tecles puc accedir a la declaració d'un mètode des de la crida a aquest?

F3

4. Quina vista fa que pugui accedir d'una manera ordenada al codi?

Project Explorer

5. Què he de fer per canviar el nom d'una classe/variable/mètode i assegurar-me que es canviara a tot arreu? I als comentaris?

[Usar Refactor](#)

El document ha d'estar ben presentat

Entregueu el document en pdf (m05_uf1_tasca1_Cognom_Nom.pdf) i el projecte telematic comprimit.

Algunes coses les hem vist a classe, altres les haureu de buscar