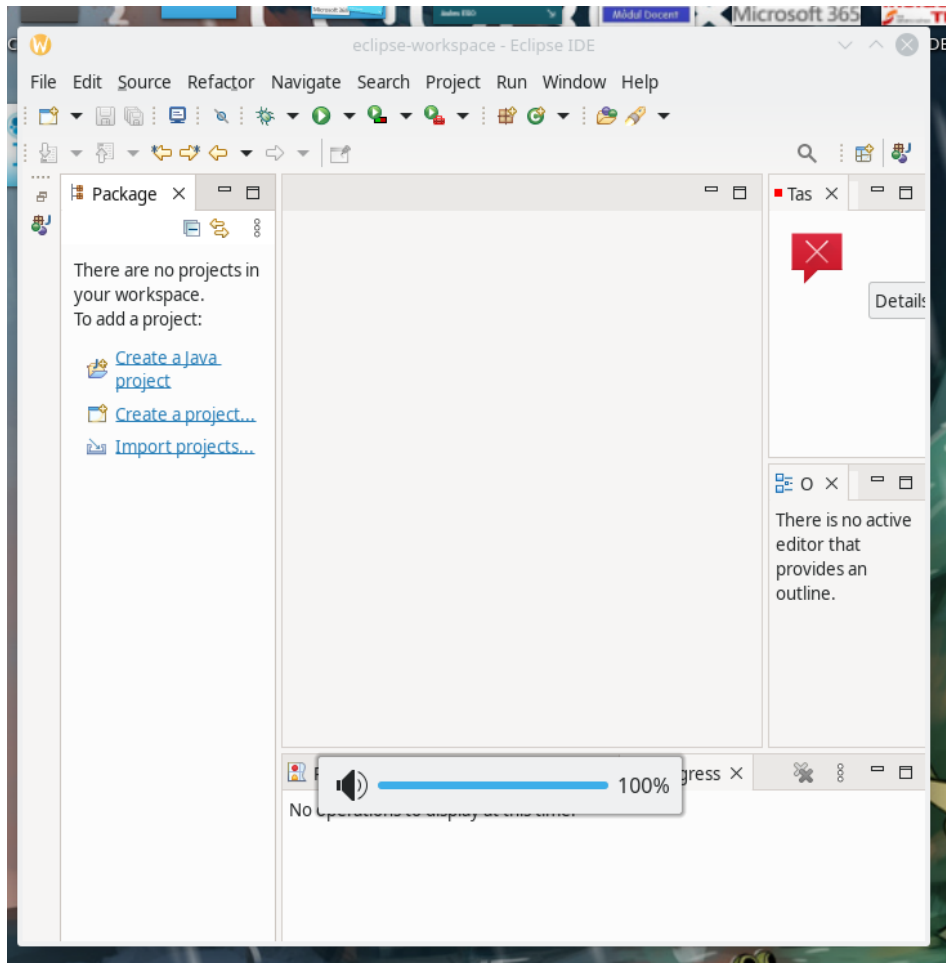
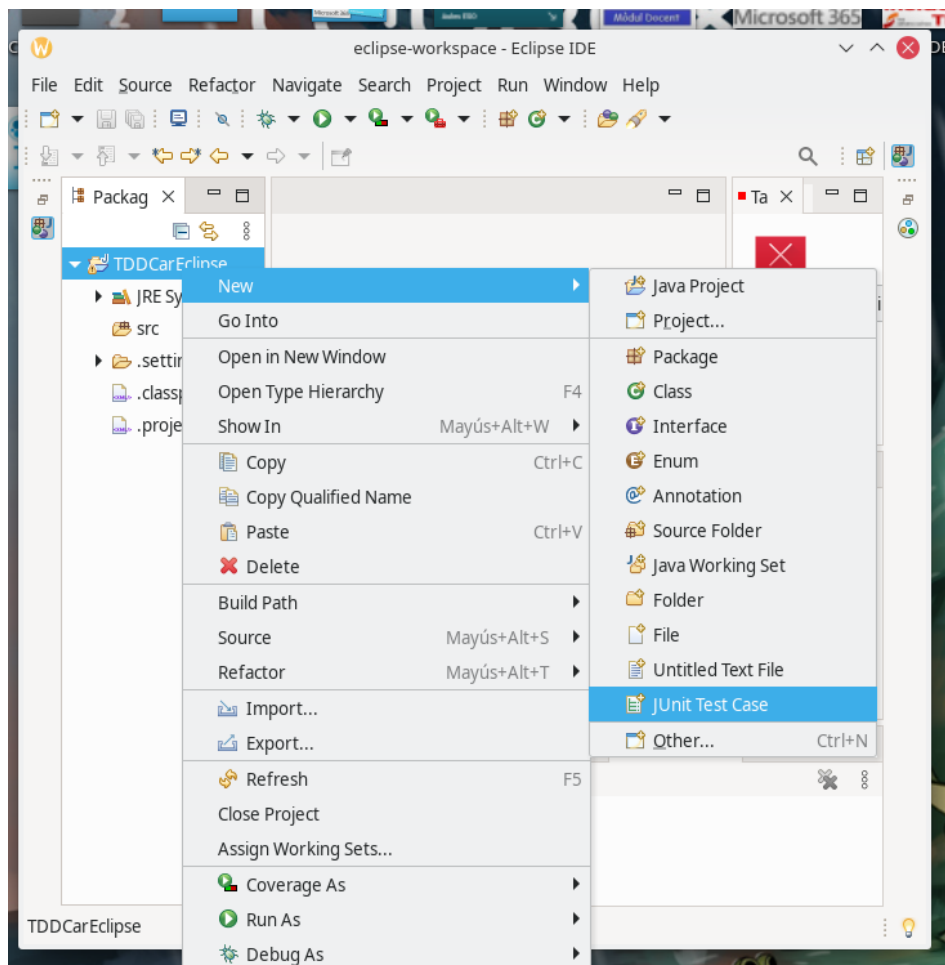


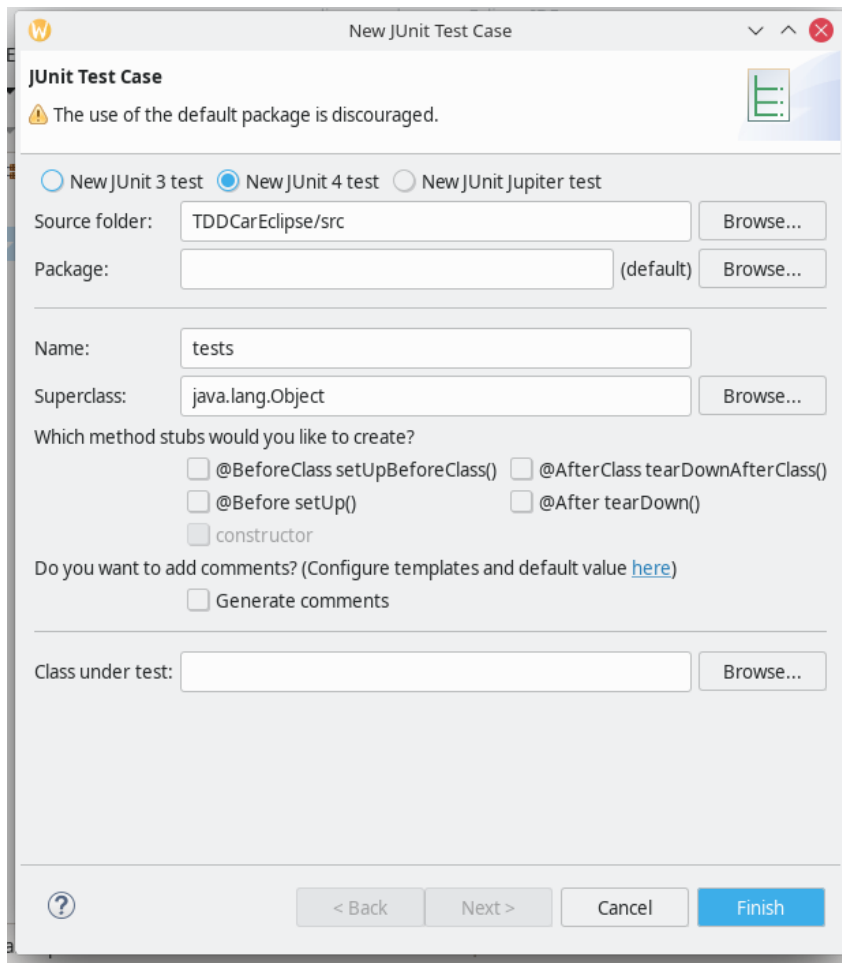
En esta práctica vamos a realizar ingeniería inversa con Eclipse. Para ello lo primero que deberemos hacer es crear un proyecto nuevo de java.



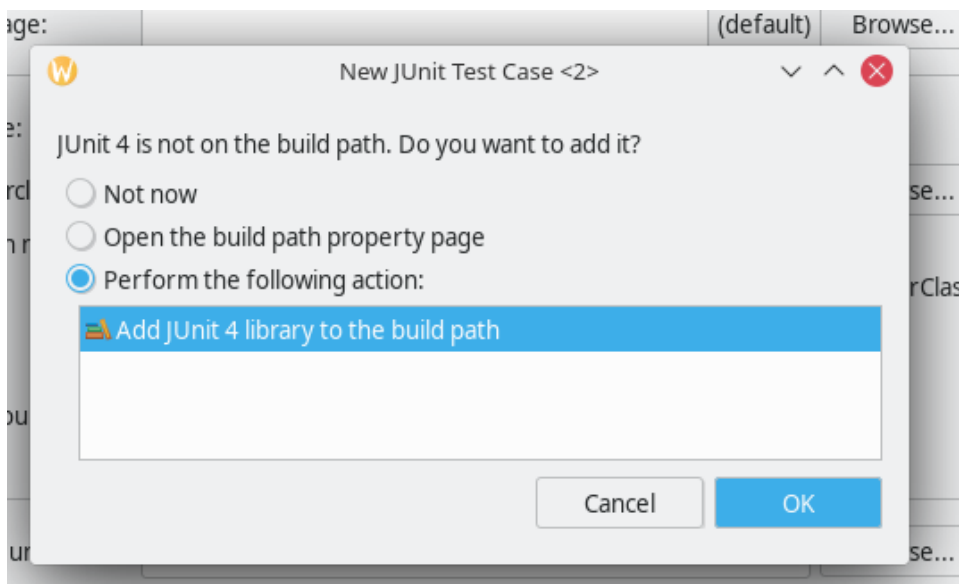
Una vez tenemos nuestro nuevo proyecto deberemos de crear un JUnit Test Case para comenzar a desarrollar nuestro proyecto de forma inversa.



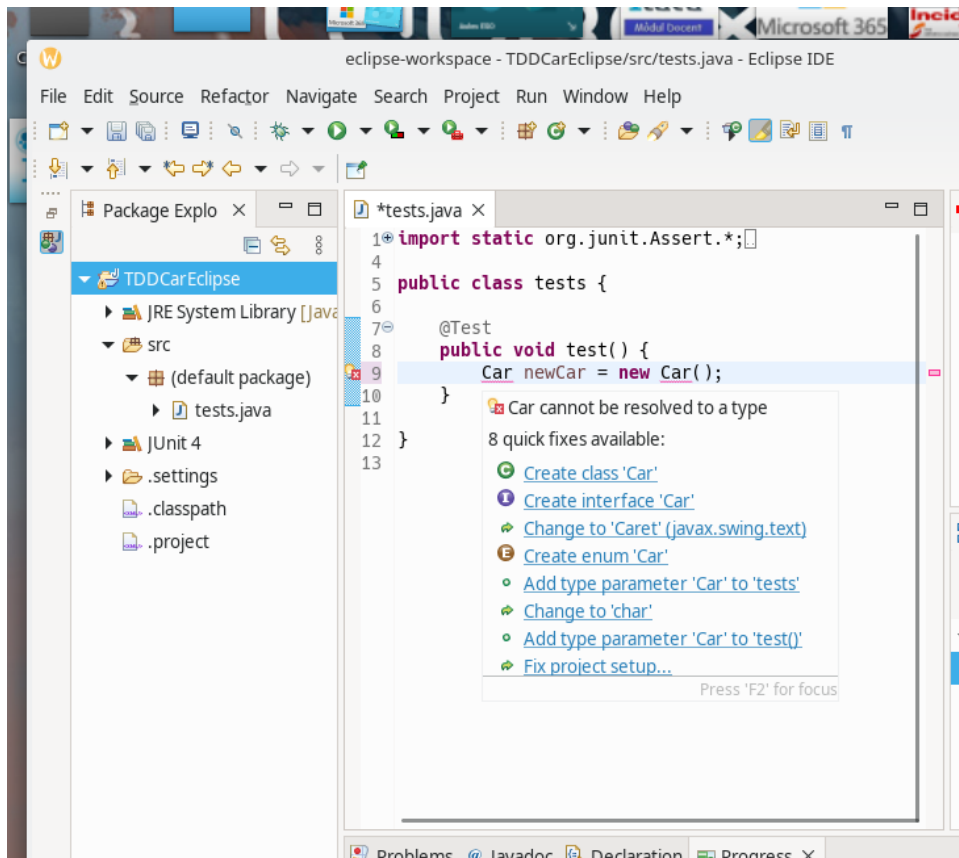
Seleccionaremos la versión de JUnit que deseamos utilizar, asignaremos un nombre para el case y finalizamos.



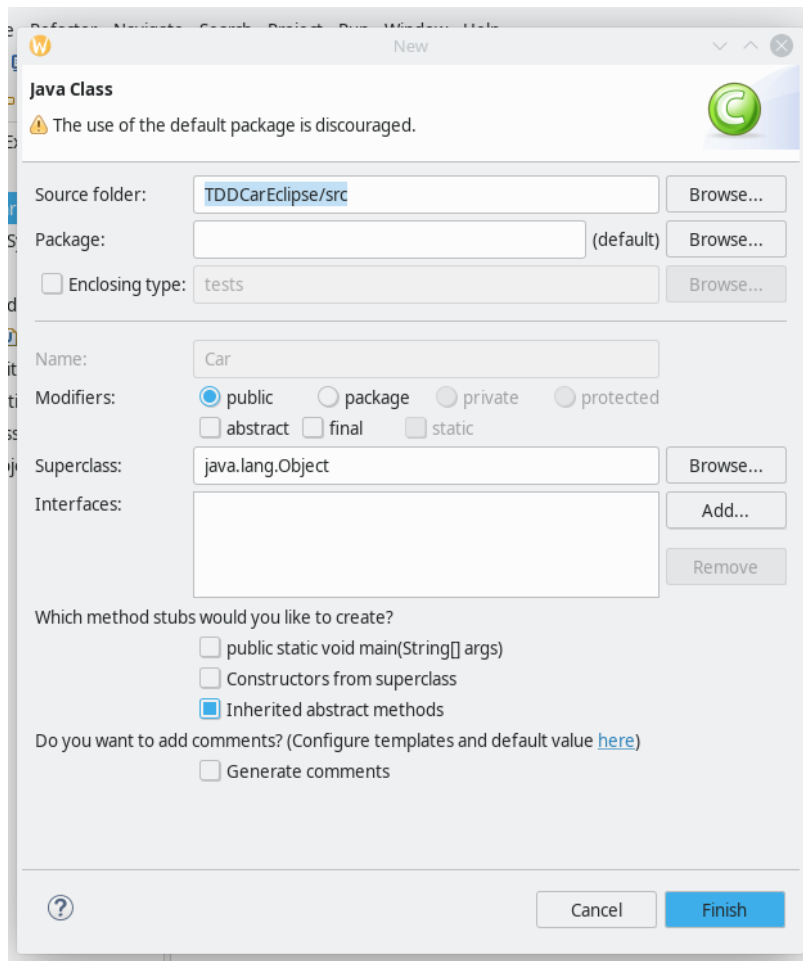
Nos solicitará que debemos añadir el JUnit a la ruta donde vamos a desarrollar nuestro proyecto.



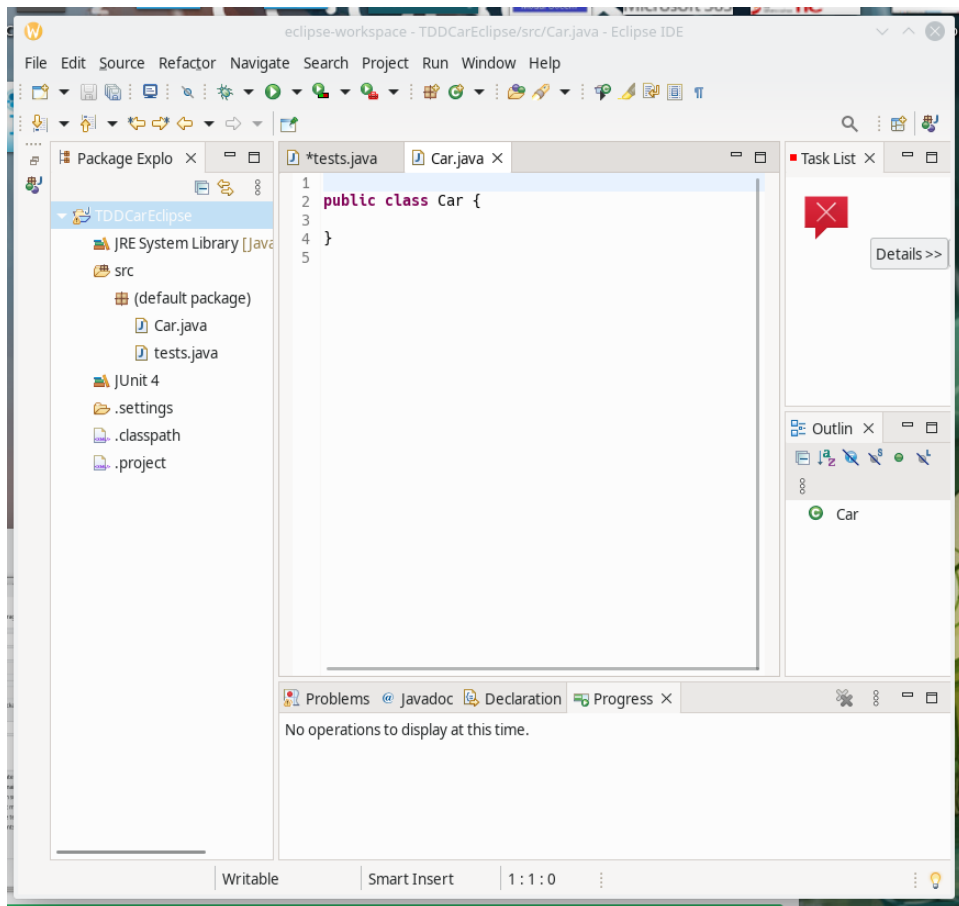
Ahora vamos a realizar la primera prueba en nuestro test. Vamos a decir que al ejecutar un método se creará un objeto de la clase Car la cual no hemos creado y nuestro test nos ayudará a crearla de forma inversa.



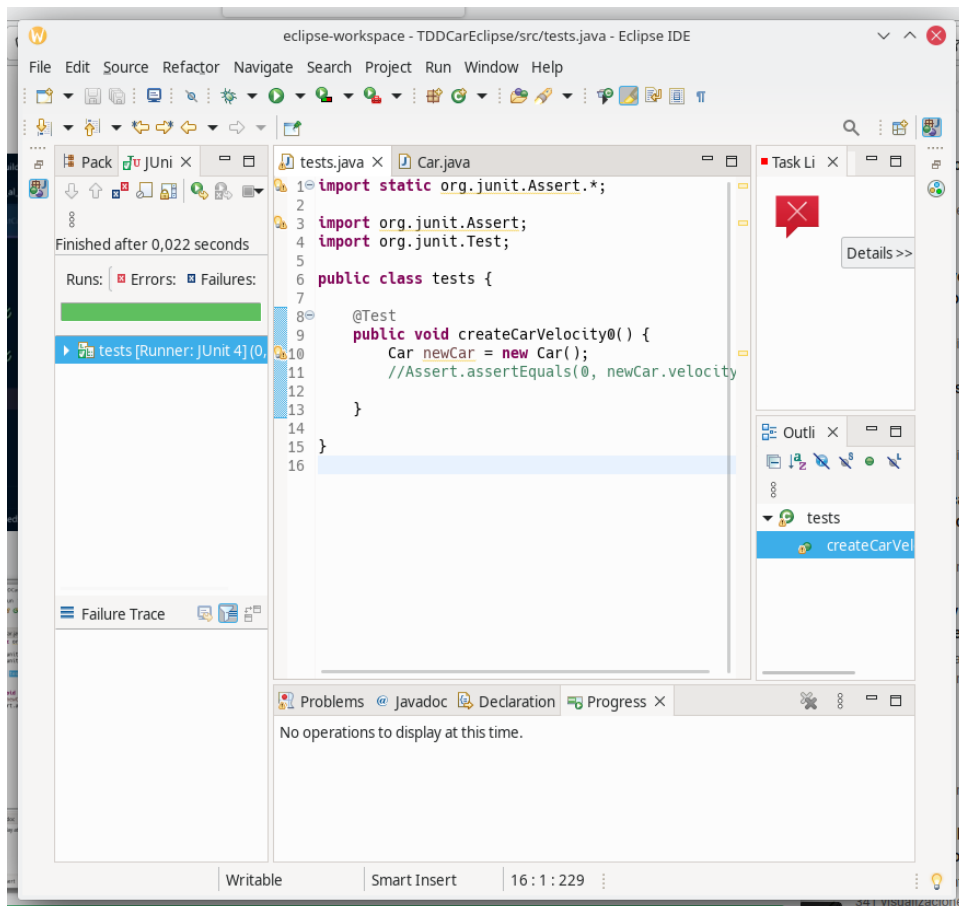
Al pulsar en la sugerencia del entorno nos abrirá una ventana para crear la nueva clase coche, en que ubicación queremos crearla y sus atributos.



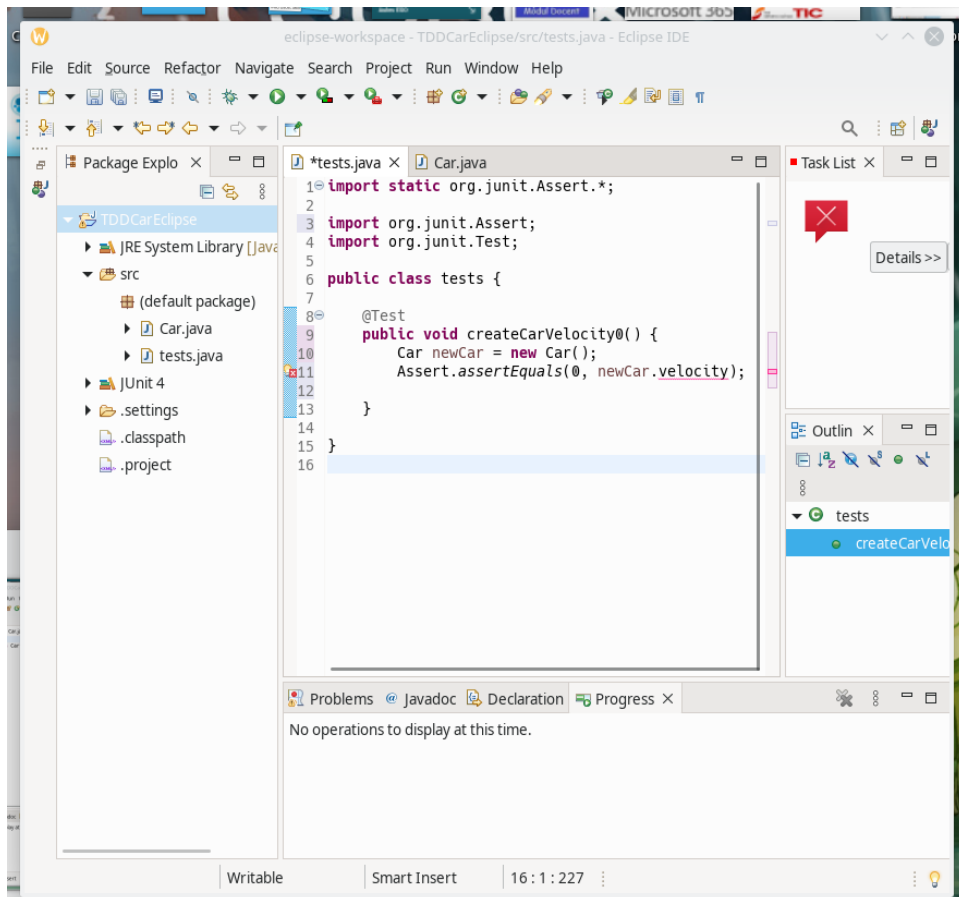
Al crearla nos aparecerá la clase de esta forma.



Si realizamos una prueba de nuestro test veremos qué ha pasado de forma correcta.

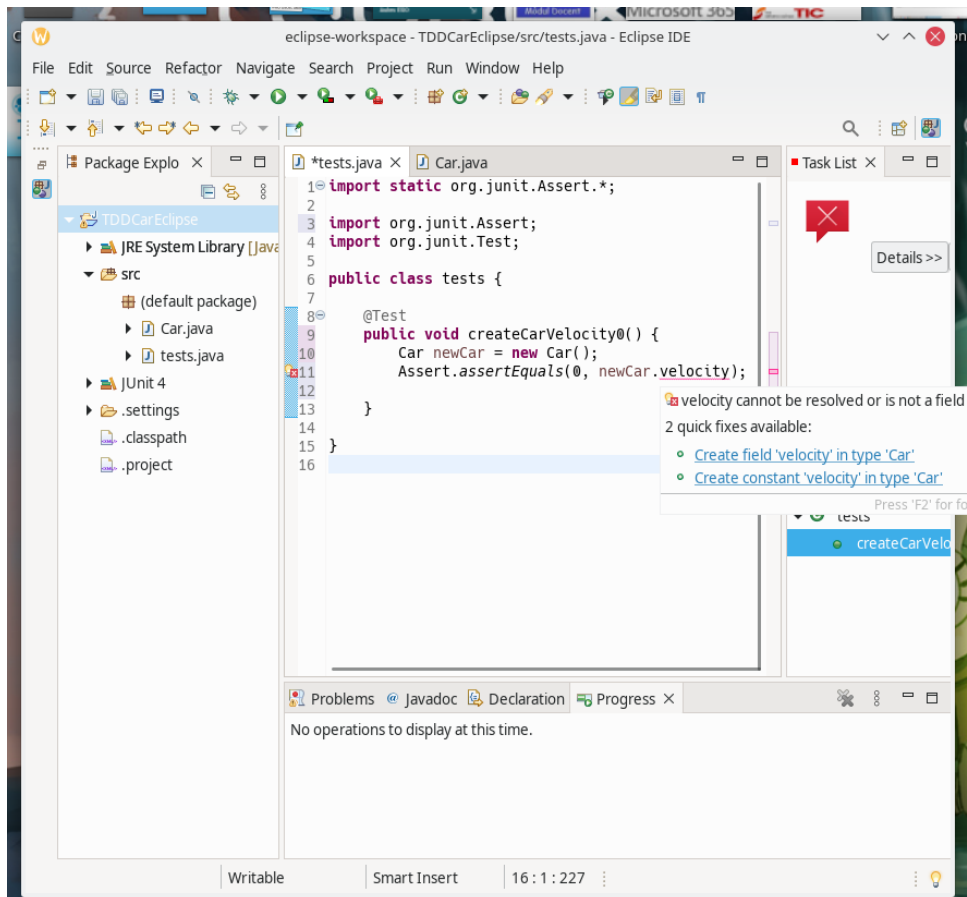


Ahora vamos a hacer que al crear nuestro vehículo vamos a establecer que su velocidad es 0, para hacer esa comprobación deberemos de importar la clase Assert y al introducir que su velocidad (que es un parámetro que no existe) nos dará error.

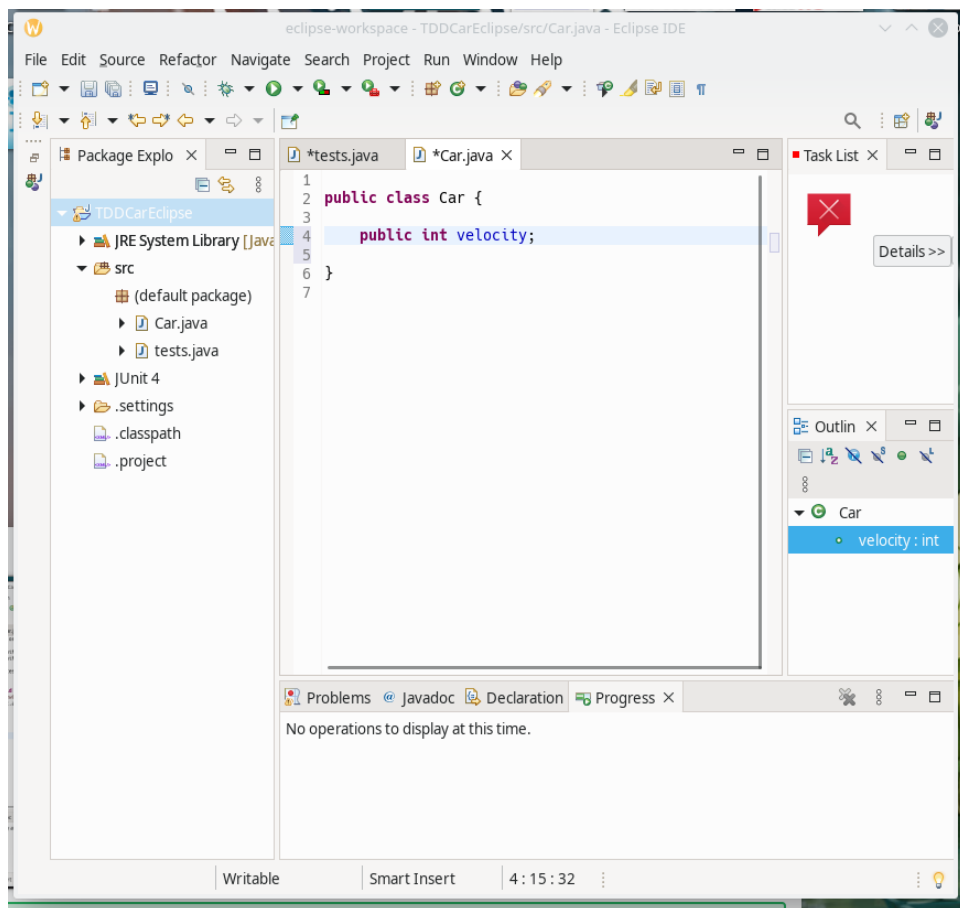


Si nos colocamos para ver la sugerencia del entorno nos indica que podemos crear el campo velocidad en Car.

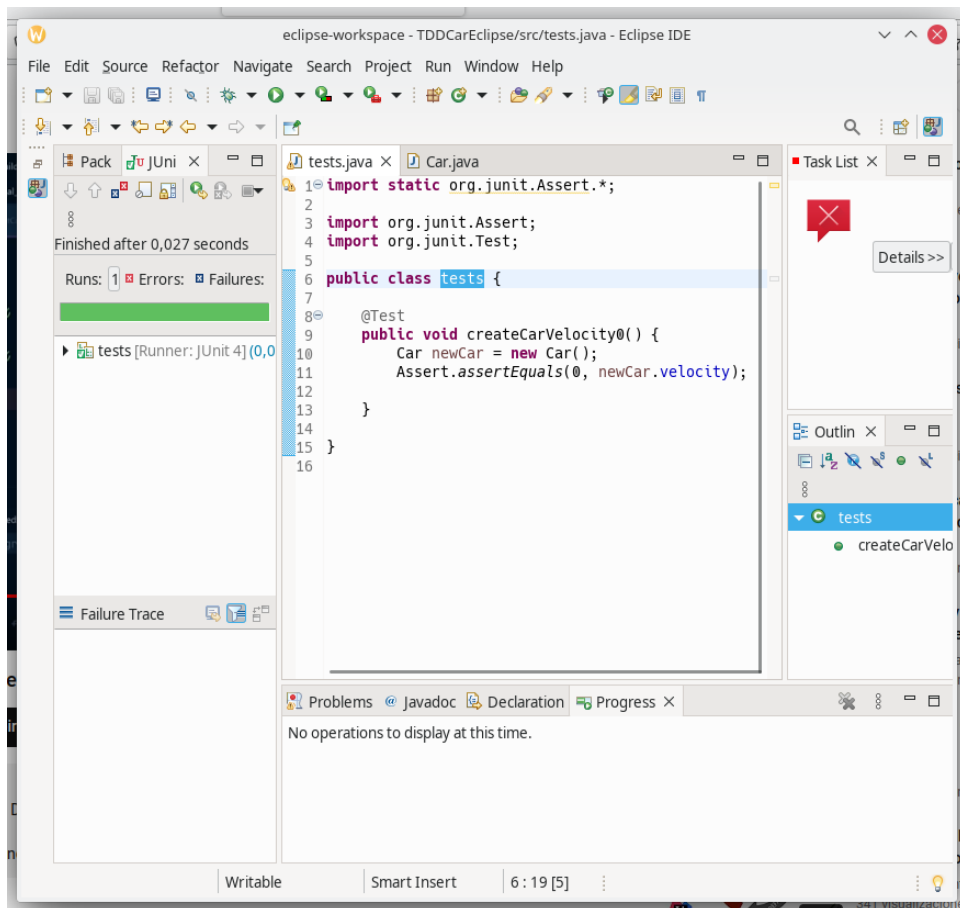




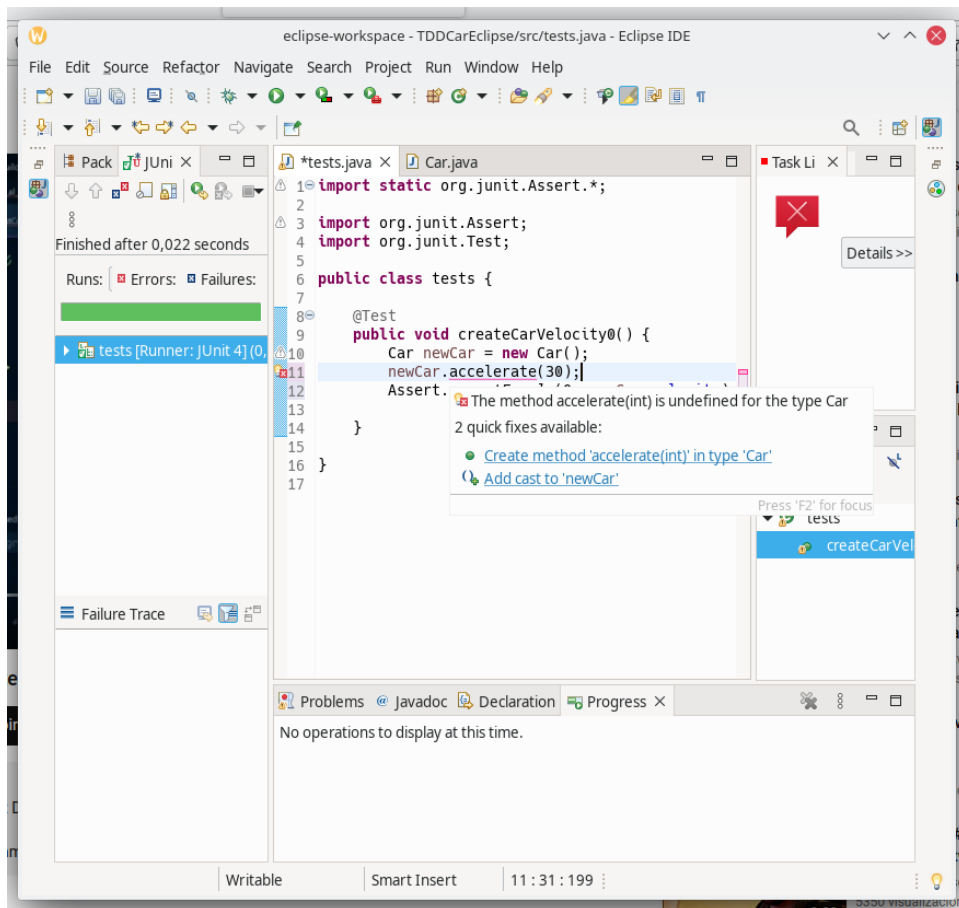
Al hacerle click nos llevará a la clase Car y veremos cómo se ha creado velocidad, pero deberemos indicar que tipo de variable es.



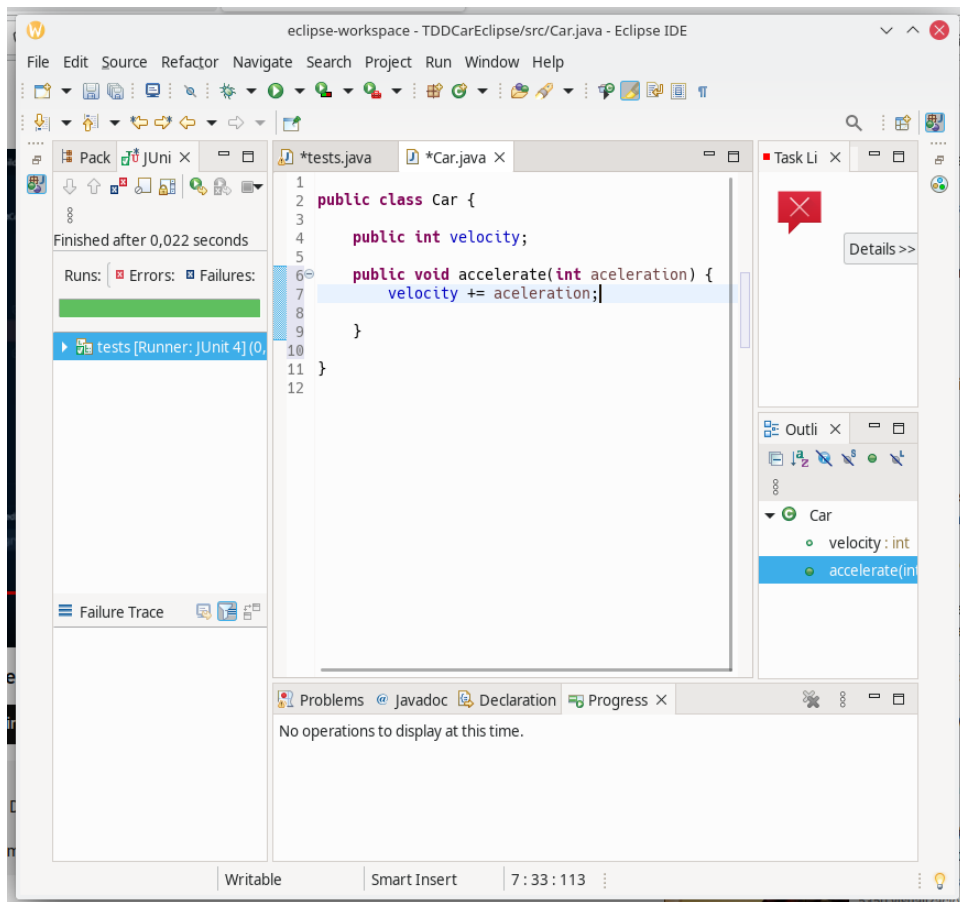
Como se ha creado correctamente el campo velocidad, al probar nuestro test será positivo de nuevo.



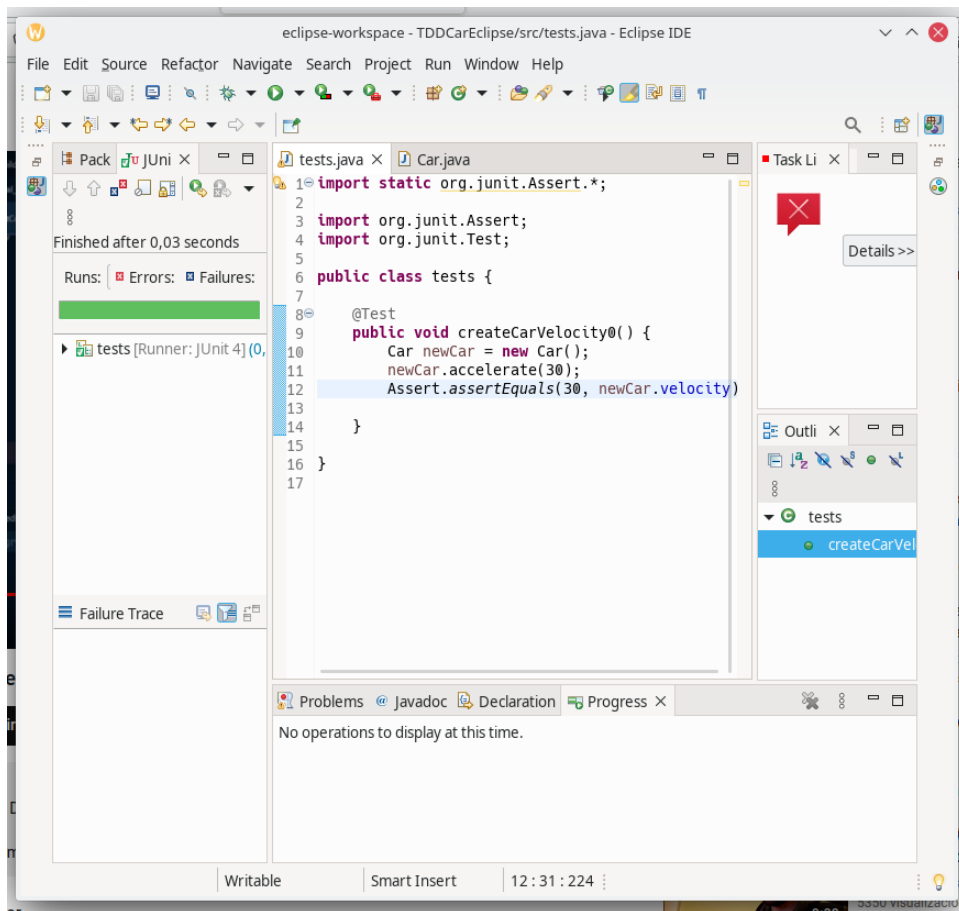
Ahora vamos a añadir métodos a nuestros test. Como no existen en nuestra clase Car nos indicara que debemos crearlo y además que será parametrizado, ya que le introducimos un valor de entrada.



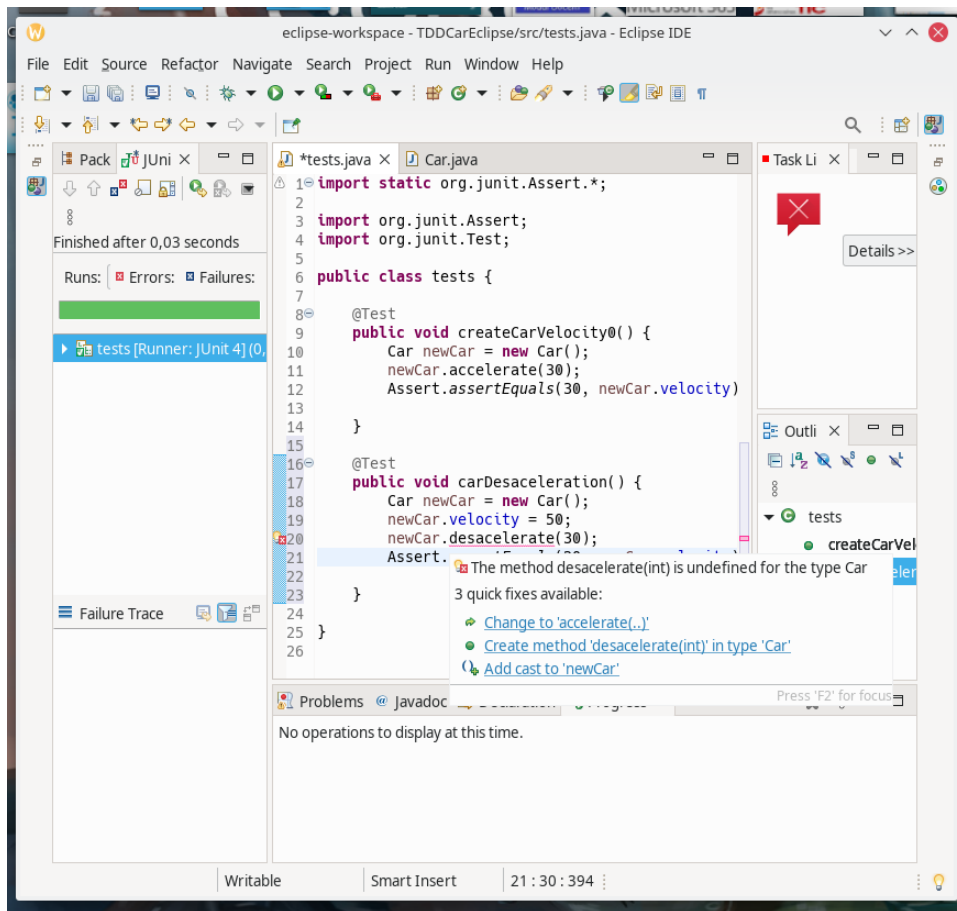
Se creará el nuevo método en la clase Car, he cambiado el nombre de la variable que entra y además he creado un procedimiento que realizará el método de acorde a lo que hemos creado.



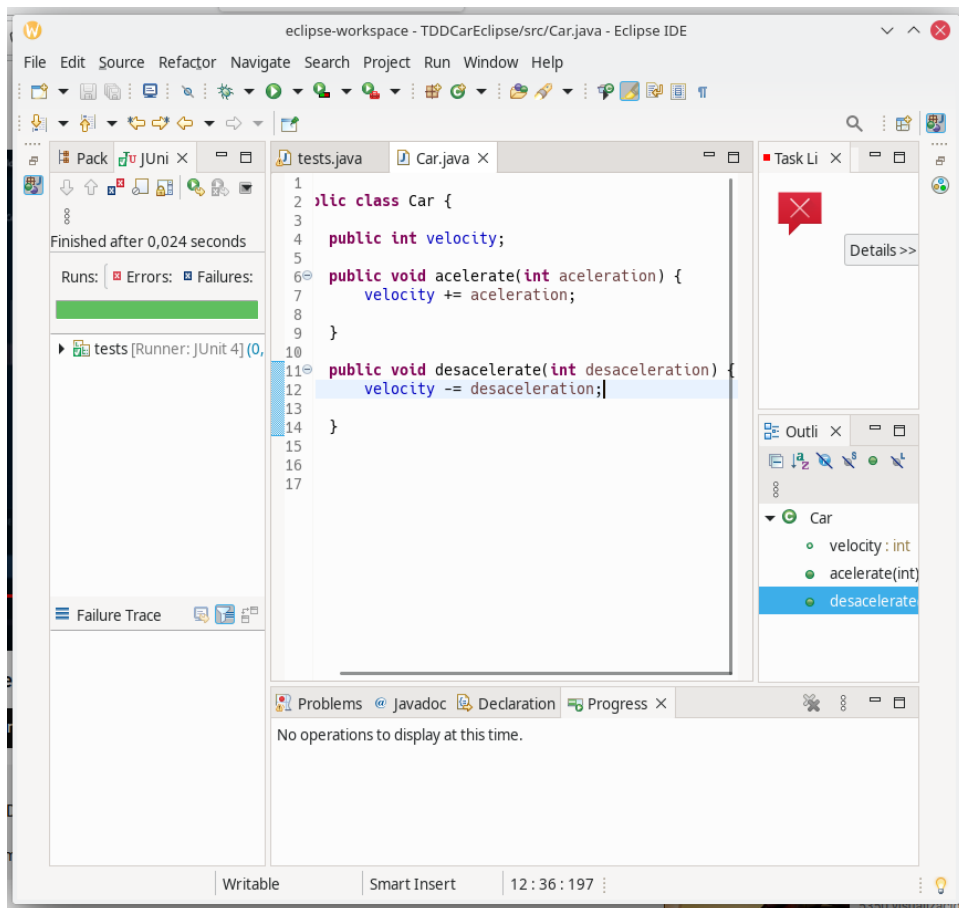
Si ahora pasamos el test, indicando los valores que tiene que comparar y con que compararlos, es decir el valor de salida debe de ser igual al del vehículo tras ejecutar el método, veremos qué ha pasado el test de forma correcta.



Ahora realizamos un método para frenar nuestro vehículo. Al igual que antes deberemos crear dicho método en la clase Car.

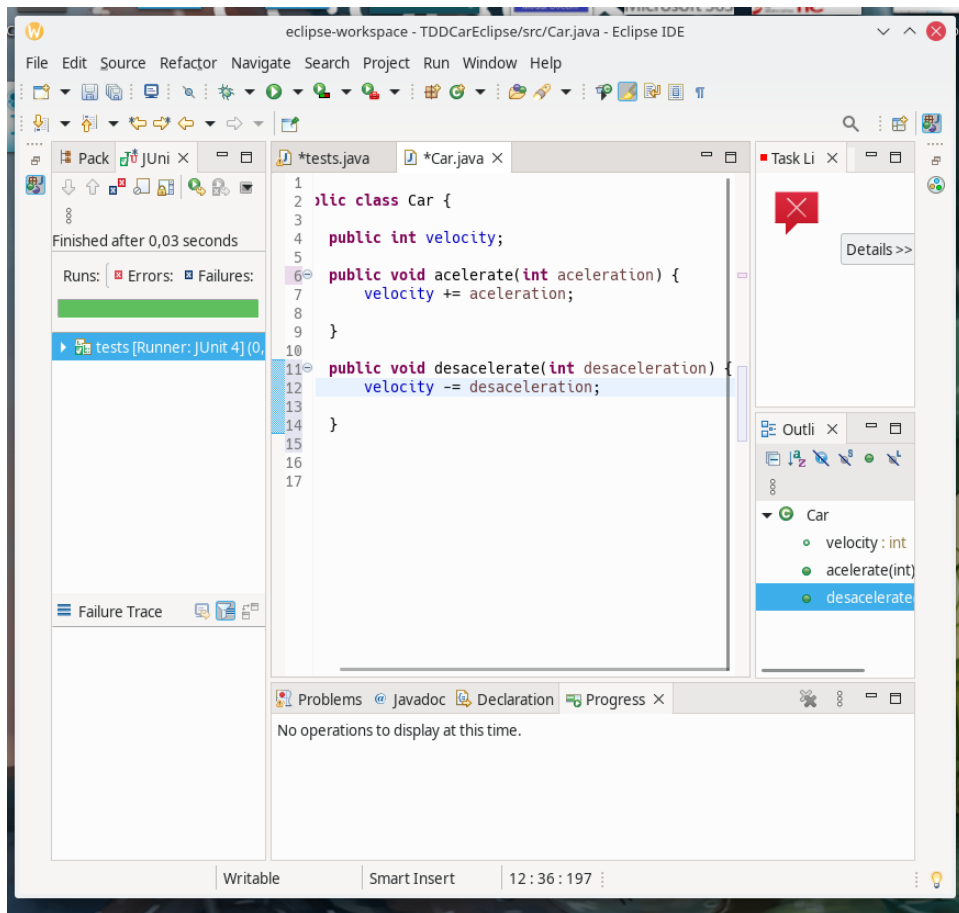


Y una vez en ella modificarlo para que se ajuste a lo que queremos.



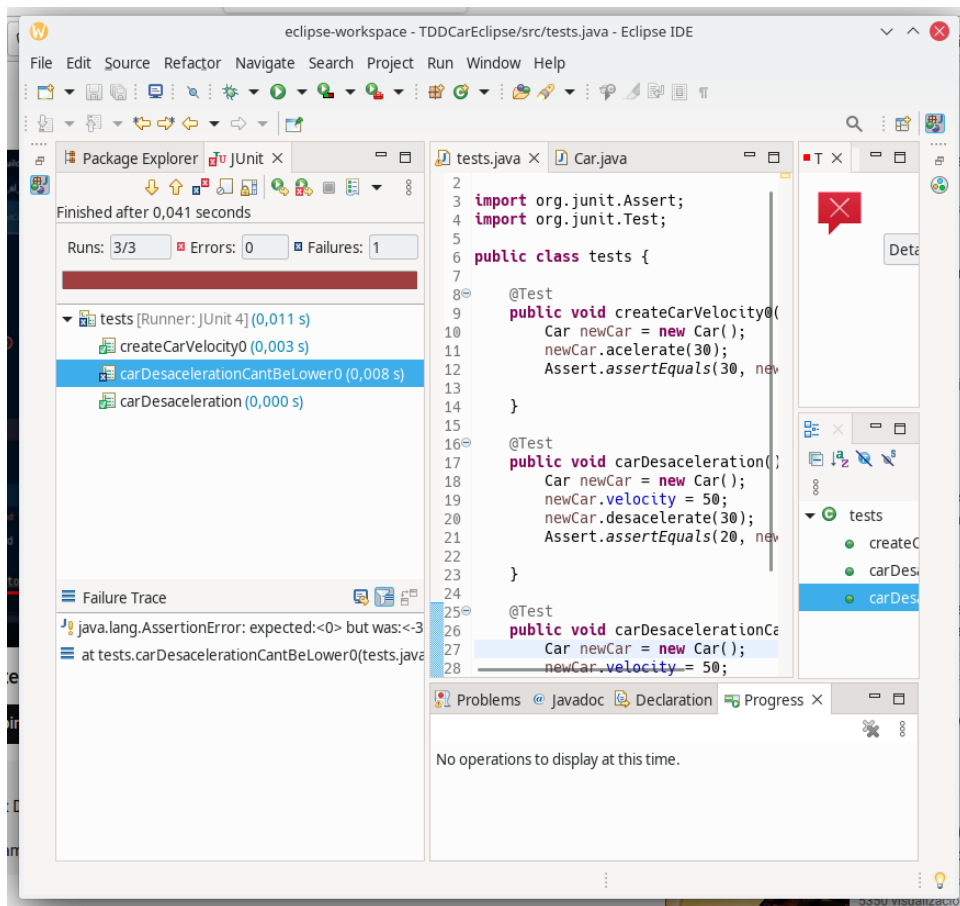
Tras ejecutarlo veremos qué ha pasado de forma correcta el test. Ya que los valores que hemos asignado para comparar coinciden con el resultado posterior de utilizar el método.



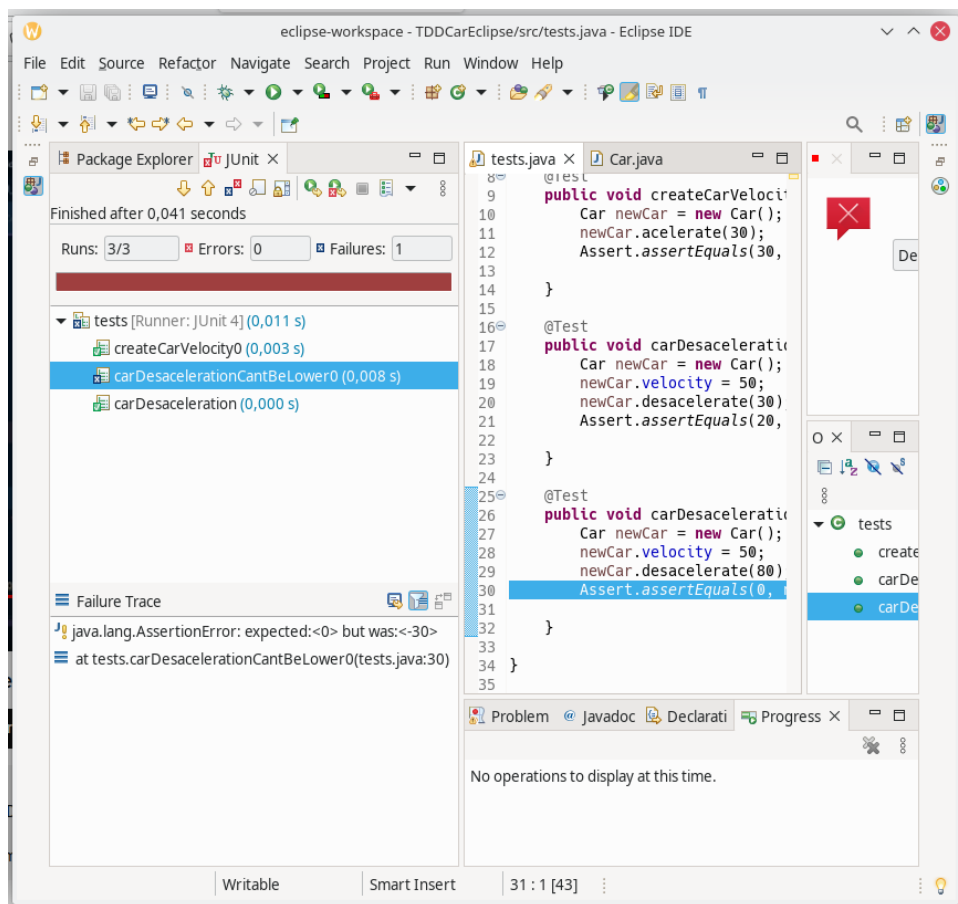


Pero qué pasaría si nosotros asignamos unos valores que no coinciden con la comprobación?

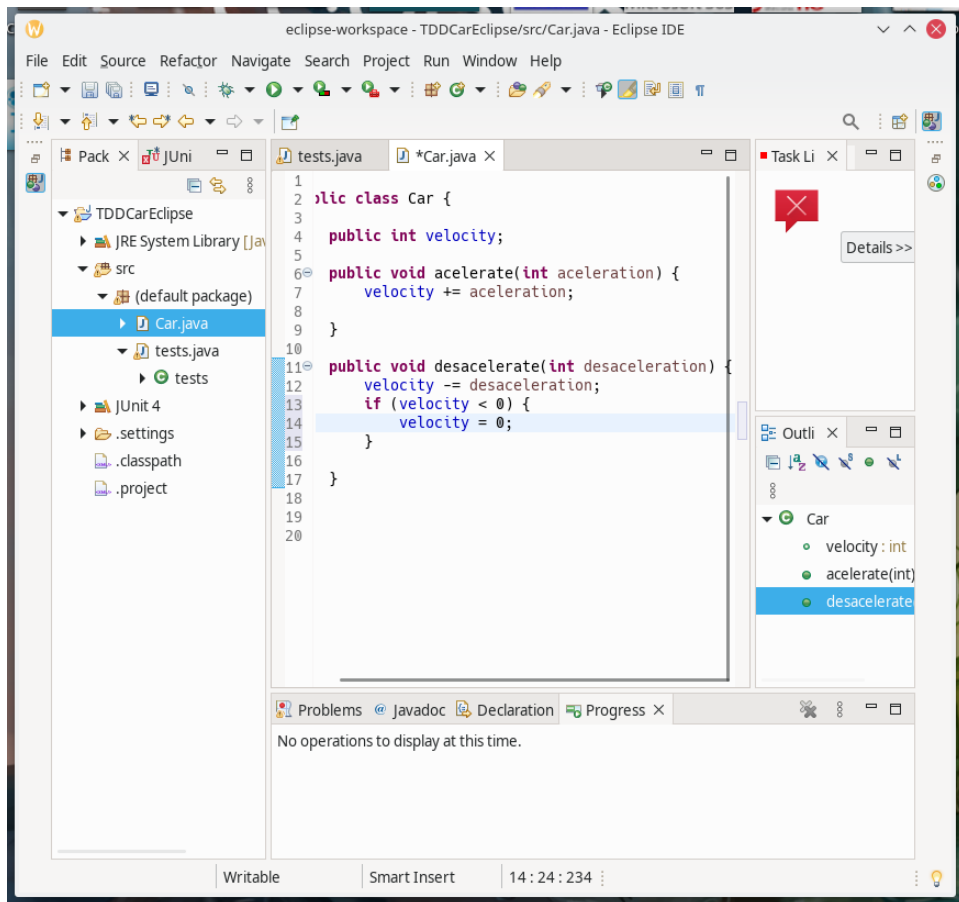
Nosotros sabemos que un vehículo que va a 50km/h y lo frenamos con una velocidad de -80km/h su velocidad final no será -30km/h sino que una vez frenado su velocidad será 0km/h. Así que vamos a probar con estos parámetros a ver qué ocurre.



Como el valor que esperábamos era 0 y nos ha devuelto -30 el test no ha pasado con éxito.



Vamos a solucionar el problema por el que el test no ha pasado. Para ello crearemos un condicional que haga que si la velocidad es menor que 0, se convierta en 0.



Al volver a probar nuestro test con los mismos parámetros que anteriormente vemos que ahora si que lo ha pasado con éxito.

