



## Práctica 5

### Objetivos

- Implementar un modelo de dominio expresado en UML con lenguaje Java
- Añadir funcionalidad al modelo
- Verificar la funcionalidad con test unitarios

### Ejercicios

#### *Implementar el modelo*

Siguiendo las indicaciones vistas en la sesión de teoría (transparencias en el campus virtual) implementa el modelo de dominio de la aplicación CarWorkshop. Sigue este orden.

- Implementa todas las entidades, añade:
  - \* atributos naturales
  - \* constructor(es)
  - \* *getters*, solo los *setters* que procedan
  - \* *hashCode()* y *equals()* sobre los atributos que definan la identidad
  - \* *toString()*
- Implementa los tipos valor
- Implementar las asociaciones y sus métodos de mantenimiento
  - \* Añade los atributos de referencia en la entidades
  - \* Añade los *getters*, *\_getters* y *\_setters* según proceda
  - \* Completa las clases internas de Associations.\* con los métodos *link()* y *unlink()* que procedan.

#### *Añadir funcionalidad*

Escribe el código java que se necesitaría para:

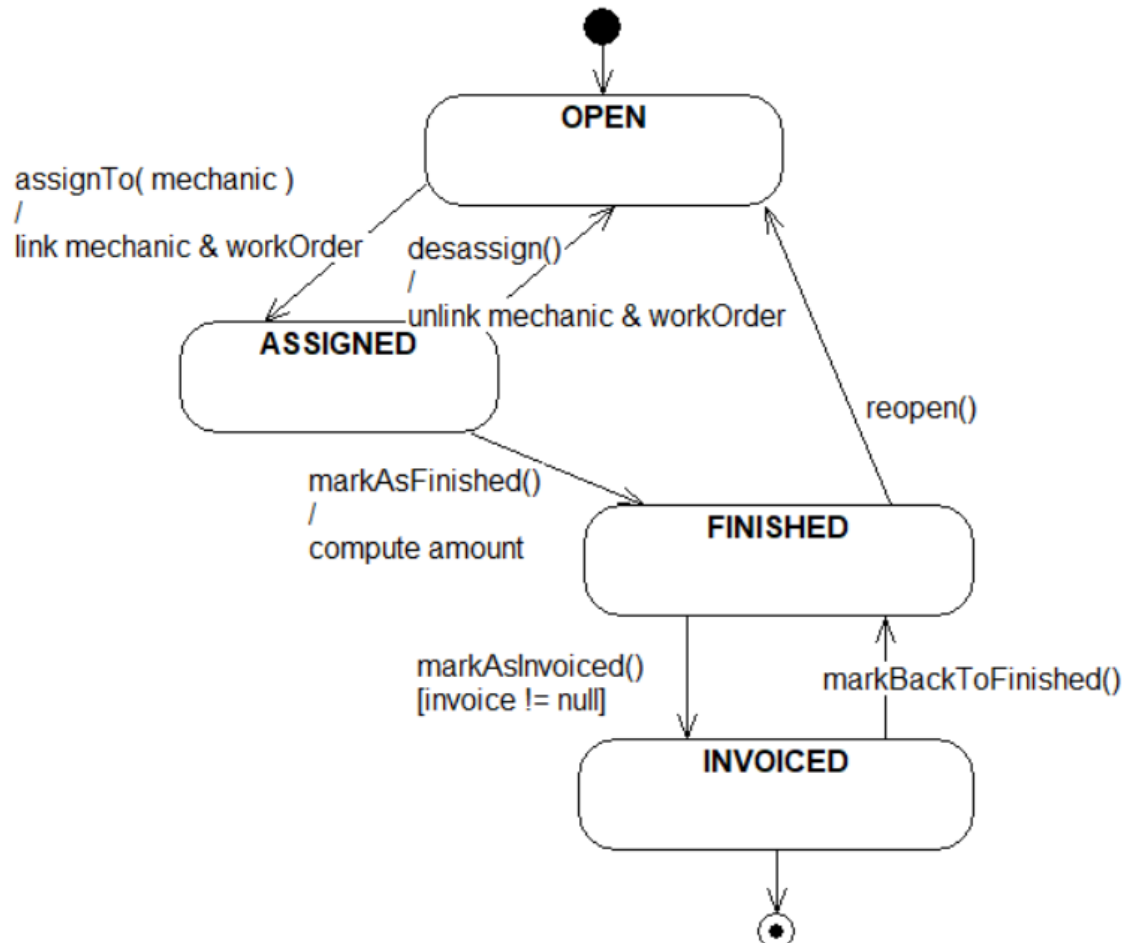
- Calcular el importe de:
  - \* Una sustitución
  - \* Los repuestos de una intervención
  - \* Una intervención (mano de obra y repuestos)
  - \* Una avería
  - \* Una factura
- Crear una factura para varias averías
- Permitir las transiciones entre estados de una orden de trabajo y una factura. Usa para ello los diagramas UML de estados mostrados debajo. Déjate guiar por los tests unitarios.

Parte del esqueleto de código que se entrega con la práctica. Añade tus clases en el paquete `uo.ri.cws.domain`.

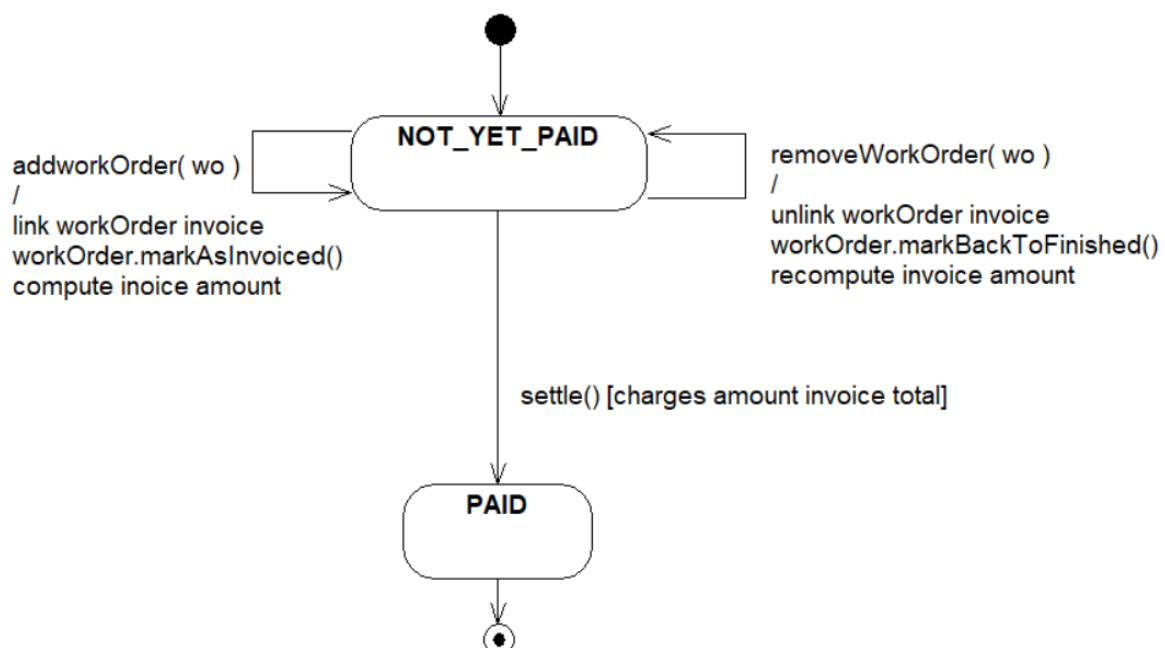
La práctica estará terminada cuando se ejecuten todos los test de la carpeta /test.



## Diagrama de estados de una orden de trabajo



## Diagrama de estados de una factura





## Modelo del dominio

