(SIMULACIÓN TALLER) PROGRAMACIÓN ORIENTADA A OBJETOS BÁSICA

Angel Maroto Chivite
22/01/2023

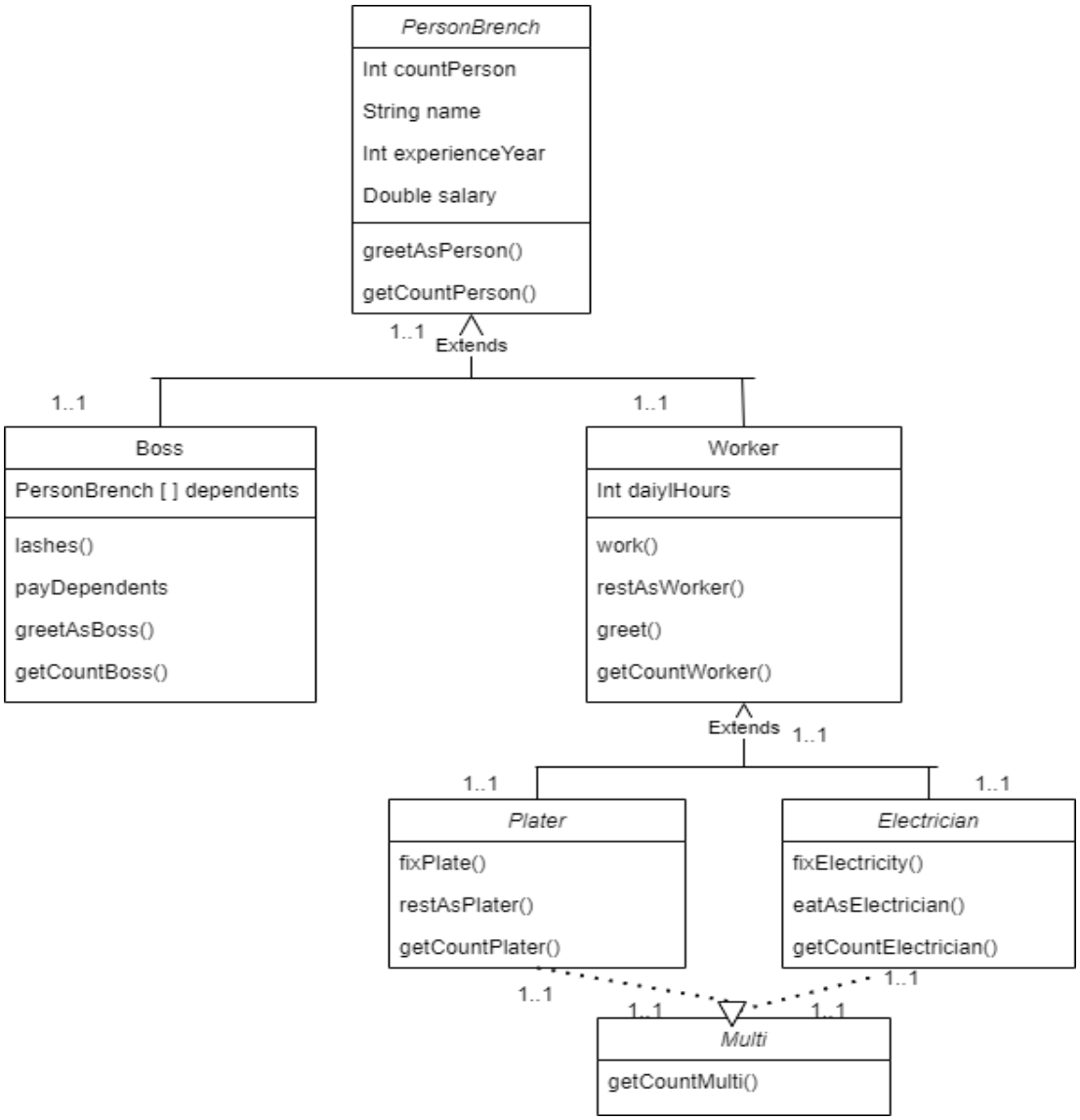
Índice

Contenido

1. Java & Kotlin

Primero realicé la práctica en Java, para practicar la herencia y el polimorfismo siguiendo el árbol de herencias presentado, e implementando las interfaces.

Para la versión de Kotlin me he apoyado del conversor del IntelIJ, a la par que debía ajustar diversos errores que iban surgiendo.



2. Ejecución de la simulación

2.1. Instanciar objetos

Instanciamos el Taller gracias a mi WorkShopFactory, está definida de tal manera que se puedan asignar el tamaño de las personas que trabajarán en el Taller, y los porcentajes en los que aparecerán, jefes, electricistas y chapistas, el resto del % que falte por rellenar serán trabajadores normales y una pequeña posibilidad de que haya Multi-Navajas, capaces de ser Electricistas y Chapistas.

Posteriormente se creará cada miembro con su correspondiente Factory, y se ejecutará un menú:

2.2 Menú

2.2.1 Listar

Se nos presentará desde la opción 1 hasta la 8 (incluidas), distintas formas de listar a todo el Taller generado.

2.2.2 Simulación

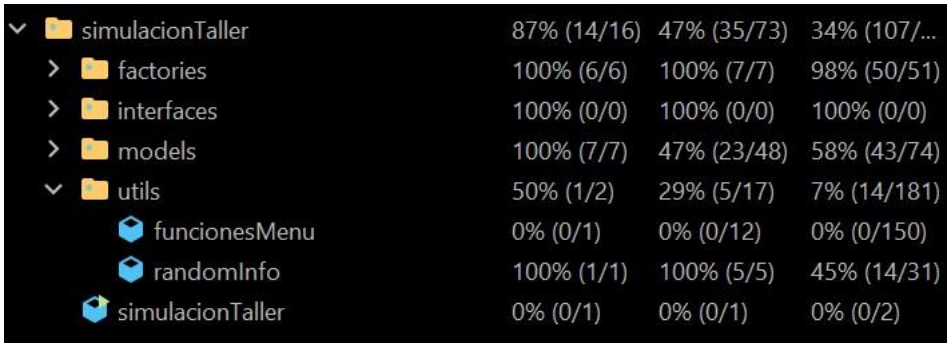
En la opción 9, dispongo de una pequeña simulación para hacer el ejercicio más amigable donde:

- En primer lugar, los Jefes aleatoriamente se decidirá, si se levantan de buen humor o no, si es la primera, ocurrán saludos entre el Jefe y sus Trabajadores así dando la paga que les corresponde, en el segundo lugar, el Jefe no pagará a nadie y elegirá aleatoriamente a un trabajador de los que él tiene al cargo y le latigueará.

- Después cada trabajador se irá presentando y realizando un breve acción de su día laboral, hasta acabar con la simulación.

3. Pruebas unitarias

Las Interface, Record y Data Class, y el código de impresión en pantalla no se encuentran probados, el resto del código está probado:



Para la versión en Java he utilizado el gestor de dependencias Maven mediante la dependencia de Test Junit5.

Para la versión de Kotlin he utilizado el gestor de dependencias Gradle mediante la dependencia de Test Junit5.

4. Diseño implementado

4.1 Factories Class

Creación de objetos aleatorios mediante factories, que se agregan a un Taller final llamado WorkShopFactory.

4.2 Models Class

Objetos que instanciaré a lo largo de la ejecución del programa.

Se puede detallar en la Clase PersonBrench, donde es **abstracta**, ya que nos sirve de molde inicial, siendo el padre principal de todas las clases.

4.3 Interfaces

La Clase Multi, al ser capaz de realizar las acciones de las Clases Electricista y Chapista, maquetamos gracias a la Interface para implementarla a la Clase Multi, así obligamos la incorporación de los métodos y comportamientos.