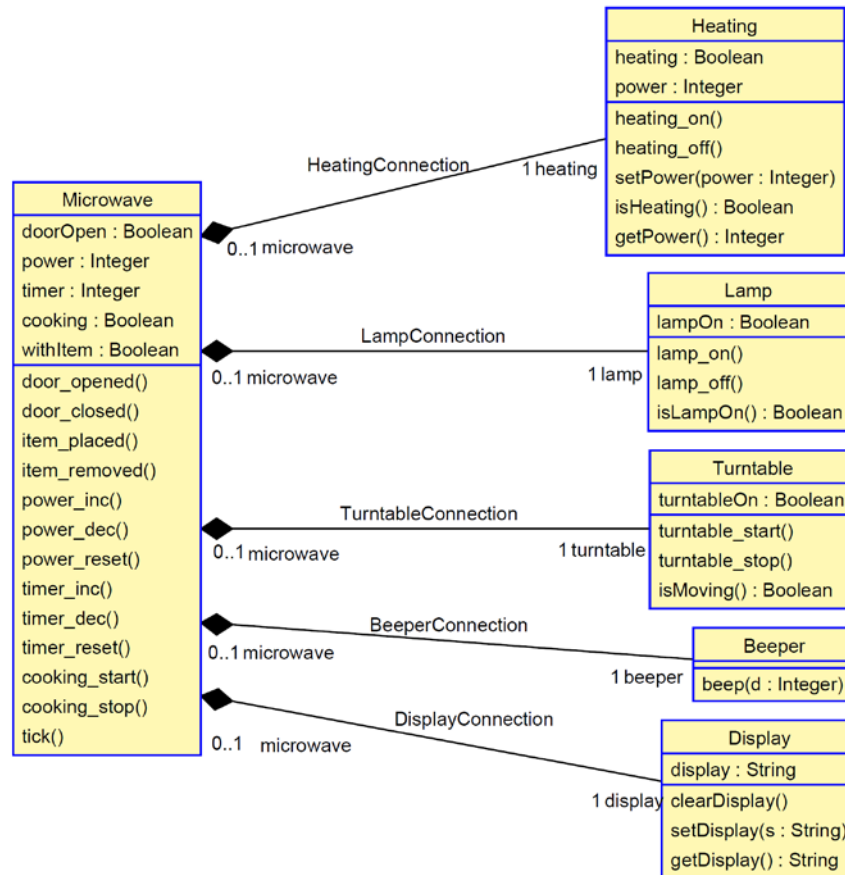


PROYECTO 1: EL HORNO MICROONDAS

Suponemos un horno de microondas que contiene diferentes componentes que interactúan a las órdenes de un componente principal, Microwave, mediante una comunicación basada en operaciones. Los componentes que forman parte del horno, y a los que el propio microondas invoca sus funciones, son los que se muestran en el modelo UML y se describen a continuación:



- Plato giratorio (Turntable). Se activa mediante la operación `turntable_start()` cuando el microondas está en funcionamiento y se para (mediante la operación `turntable_stop()`) cuando se abre la puerta o se acaba tiempo de cocinado. Implementa una operación de consulta `isMoving()` que permite saber en todo momento si el plato está girando o no.
- Campana (Beeper). Avisa cuando el temporizador haya llegado a cero. Su operación `beep(d:Integer)` hace que la campana suene tantas veces como indica el parámetro `d`.
- Lámpara (Lamp) que se enciende (`lamp_on()`) o apaga (`lamp_off()`) en función de distintos eventos, como pueden ser que la puerta esté abierta o que el microondas esté funcionando. Una operación de consulta, `isLampOn()` permite conocer al microondas si la luz está dada o no.
- Unidad de calor (Heating) que es el dispositivo de magnetrón que emite las microondas, encargado de calentar la comida a una determinada potencia (`power`). El microondas lo enciende y apaga usando las operaciones `heating_on()` y `heating_off()`, y también puede conocer si está encendido o no con la operación de consulta `isHeating()`. El componente microondas también puede establecer la potencia y conocerla con las operaciones `setPower()` y `getPower()`.
- Pantalla (Display) que le permite al microondas mostrar distintos mensajes (por ejemplo, "La comida está lista"). En este caso su función principal es mostrar el tiempo restante del temporizador. La operación `clearDisplay()` borra el contenido de la pantalla y la apaga. La pantalla se vuelve a encender cuando se invoca la operación `setDisplay()`.

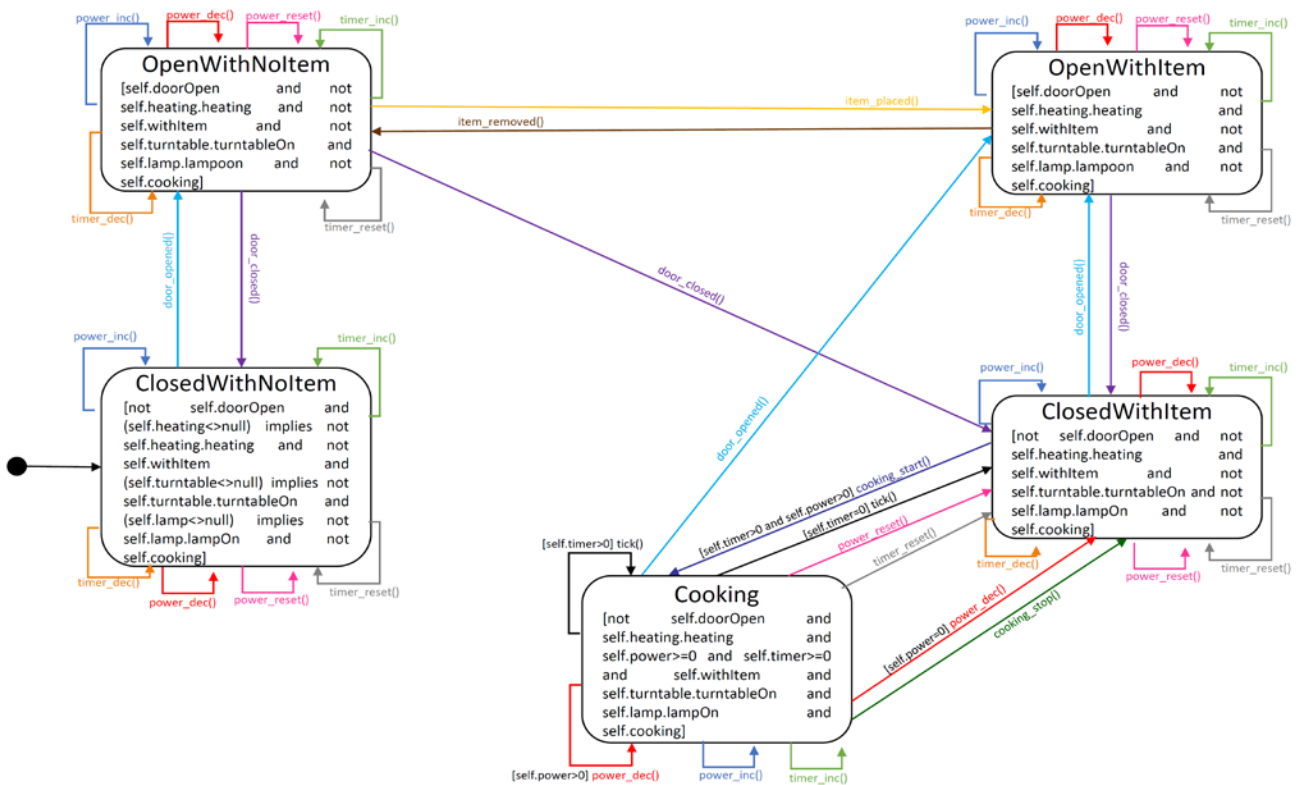
Los atributos que se muestran en los componentes son para su uso interno, a su valor solo puede conocerse y establecerse a través de las correspondientes operaciones.

Por su parte, el propio microondas admite una serie de operaciones que el usuario puede invocar a través de una Interfaz de Usuario (UI), y que sirven para controlar tanto su estado como su comportamiento. Dicha interfaz no se muestra en el diagrama de clases, sino que inicialmente supondremos que es el propio usuario

quien invoca las operaciones del microondas, bien mediante acciones físicas (abrir y cerrar la puerta, meter un alimento o retirarlo) o a través de un panel de control que le permite incrementar o disminuir el tiempo de cocinado y la potencia de calentamiento, así como iniciar o parar el cocinado. El estado interno del microondas viene determinado por sus atributos, que solo pueden ser modificados a través de dichas operaciones. Finalmente, un reloj externo es el encargado de invocar la operación `tick()` del microondas cada segundo, permitiéndole conocer así el paso del tiempo.

El modo de funcionamiento normal del microondas es que el usuario abre la puerta, coloca un artículo dentro, cierra la puerta, establece la potencia y el tiempo de cocinado, y presiona el botón de `start_cooking()`. A partir de ese momento el microondas pone a girar el plato, apaga la luz, muestra en el display el tiempo que queda por cocinar, y da la orden a la unidad de calor para que empiece a calentar. Cada vez que se recibe un `tick()` del reloj, el horno decrementa el tiempo restante, y actualiza el valor mostrado en el display. Cuando el timer llega a 0, el horno da la orden a la unidad de calor para que se detenga, la campana suena tres veces, el plato giratorio se para, se enciende la luz, y en el display se muestra un mensaje que indica que la comida está lista. Si el horno está funcionando y se abre la puerta, el resultado es el mismo, salvo que el temporizador deja de avanzar, manteniendo el valor actual. En cualquier momento, independientemente de si la puerta está abierta o cerrada o el horno está cocinando o no, es posible modificar los valores del temporizador o la potencia de cocinado.

Por supuesto, las operaciones del microondas no pueden invocarse en cualquier orden. Por ejemplo, si la puerta está cerrada no pueden invocarse las operaciones que permiten meter o sacar la comida (`item_removed()` e `item_placed()`). Por razones de seguridad, cuando el usuario pulsa el botón de cocinado (operación `start_cooking()`) y la puerta está abierta, dicha operación no tiene efecto alguno. Igualmente, si no hay ningún artículo en el horno, dicha acción tampoco tiene efecto, aunque la puerta esté cerrada. El siguiente diagrama de estados especifica los comportamientos válidos en cada estado, así como los invariantes que se cumplen en cada uno de los estados.



Se pide: (a) Implementar el sistema en Java, usando el patrón de diseño Estado; (b) definir pruebas unitarias con Junit para cada uno de los componentes que conforman el sistema; (c) definir un conjunto de escenarios de prueba para el sistema completo con Gherkin, e implementarlas en Cucumber; (d) [opcional] definir e implementar tres interfaces de usuario que, a través de botones, permitan interactuar con el microondas de forma concurrente: uno con el panel de control, otro que simule la puerta y el hecho de meter y sacar un alimento del microondas, y un tercero que permita simular el tick de reloj.