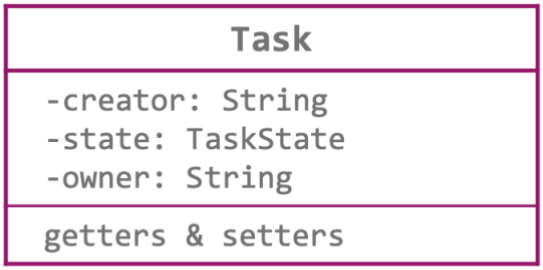
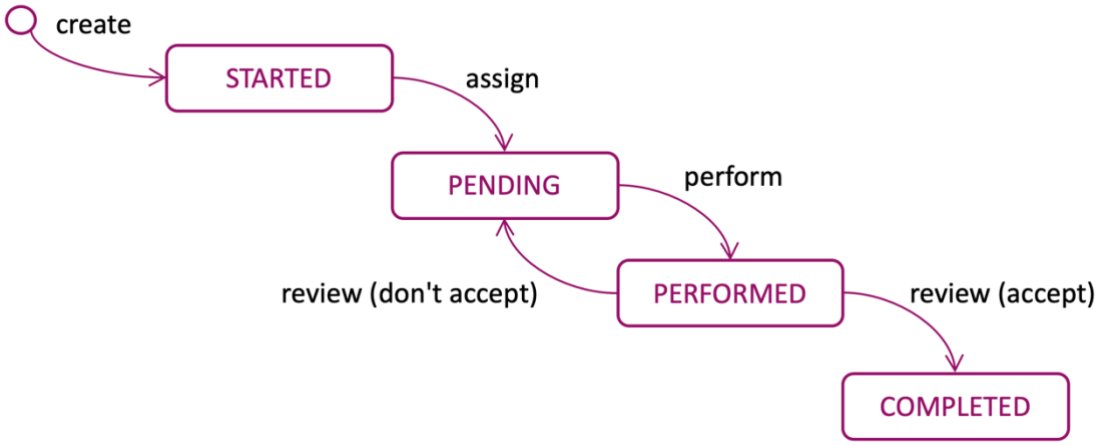
Se pide construir, en el paquete com.bpe.enums, un enum llamado TaskState, con los siguientes atributos:

- STARTED
- PENDING
- PERFORMED
- COMPLETED

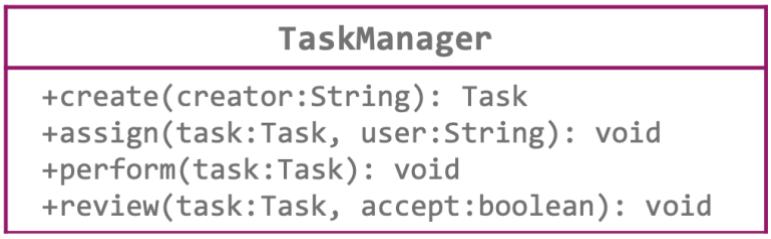
En el mismo paquete com.bpe.enums, construir una clase Task, que representa a una tarea, basada en el siguiente diagrama:



La idea es construir una clase que maneja la lógica de un flujo de trabajo, donde en cada transición se ejecuta una tarea. El flujo tiene el siguiente diagrama de estados, que corresponden a los valores del enum:



Para controlar la lógica de ejecución del flujo anterior, se pide construir una clase TaskManager, en el mismo paquete com.bpe.enums, basada en el siguiente diagrama:



El método assign(task, user) asigna la tarea sólo si está en estado STARTED, cambiando a PENDING, y asignando el parámetro user como el owner. Si no, lanza una unchecked exception con un mensaje. Una implementación del método es:

```
public void assign(Task task, String user) {
    if (task.getState() != TaskState.STARTED) {
        throw new RuntimeException("Task is not in STARTED state");
    }
    task.setState(TaskState.PENDING);
    task.setOwner(user);
}
```

Descripción de los métodos, que se pide implementar en TaskManager, basado en el ejemplo anterior:

- create: crea una nueva tarea, en estado STARTED, asignando el creator obtenido como parámetro.
- perform: sólo si está PENDING, la deja en PERFORMED. Si no, excepción.
- review: se ejecuta sólo si está PERFORMED, lanzando excepción si no. Si accept es true, queda COMPLETED, y en caso contrario vuelve a estado PENDING.

Para probar la lógica de ejecución del flujo anterior, se pide construir una clase TaskManagerTest, en el mismo paquete com.bpe.enums, que ejecute la siguiente secuencia de acciones en un main:

- Crear una instancia de TaskManager
- Crear una instancia de Task, utilizando TaskManager. Verificar que está STARTED.
- Asignarla (assign) a un usuario. Verificar que está PENDING.
- Ejecutarla (perform). Verificar que está PERFORMED.
- Revisarla (review), no aceptando el resultado. Verificar que vuelve a estar PENDING.
- Volver a ejecutarla. Verificar que está PERFORMED.
- Volver a revisarla, aceptando esta vez el resultado. Verificar que queda COMPLETED.