UT 1.2 – Ejercicios

Ejercicio 05 – Inyección de dependencias en Spring

Copia el proyecto del ejercicio 04, y modifica los nombres de artefactos, paquetes y clases que consideres adecuado, para que se refleje que este es un ejercicio distinto.

Crea:

- MessageService. Intefaz con un único método abstracto “void showMessage(String message)”.
- SystemOutMessageService. Clase que implemente la interfaz, y que:
    - En el método showMessage mostrará el mensaje usando System.out
    - Debe anotarse para que sea un bean.

Modifica las clases ProcesoA, ProcesoB, ProcesoC, para que reciban un objeto MessageService (un objeto de la interfaz, no de la clase que la implementa) como dependencia, usando inyección por constructor. Estas clases usarán el objeto recibido para mostrar el mensaje en el que figura el nombre completo de la clase.

Ejercicio 06 – Inyección de dependencias en Spring

Copia el proyecto del ejercicio 05, y modifica los nombres de artefactos, paquetes y clases que consideres adecuado, para que se refleje que este es un ejercicio distinto.

Crea una clase “SystemErrMessageService”, que será igual que la clase “SystemOutMessageService”, pero que usará System.err en lugar de System.out para mostrar los mensajes. Esta clase tiene que ser también un bean, no olvides anotarla.

¿Hay algún problema por tener dos clases que implementan la misma interfaz?

Soluciónalo usando una anotación en alguna de las clases que implementan la interfaz (SystemOutMessageService o SystemErrMessageService) para hacer que se use esa clase y no la otra.

Una vez resuelto, haz que uno de los procesos (de los CommandLineRunner) use la clase que no se ha establecido por defecto, con anotaciones en el constructor del proceso.