

-Ejercicios propuestos- Spring Core y AOP

INDICE

INDICE	2
1.- SPRING CORE	3
EJERCICIO 1	3
EJERCICIO 2	3
EJERCICIO 3	3
EJERCICIO 4	4
EJERCICIO 5	4
EJERCICIO 6	4
EJERCICIO 7	4
2.- SPRING AOP	5
EJERCICIO 1	5
EJERCICIO 2	5
EJERCICIO 3	6
EJERCICIO 4	6
INDICE DE GRÁFICOS	¡Error! Marcador no definido.

1.- SPRING CORE

EJERCICIO 1

Declarar un bean de la clase Saludo con una propiedad de tipo String que contenga un mensaje de bienvenida.

Recuperar el bean creado desde la clase Main y mostrar el mensaje de bienvenida.

EJERCICIO 2

Crear la clase Cliente con las propiedades: nombre, nif y dirección todas ellas de tipo String.

Crear la clase Factura con las propiedades: cliente de tipo Cliente, importe de tipo double y fecha de tipo Date.

Declarar varios beans de tipo Factura inyectando las propiedades de la siguiente forma:

- Haciendo referencia a sus propiedades
- A través del constructor:
 - Por tipo de datos
 - Por posición
 - Por nombre

La propiedad fecha se debe inyectar con un bean interno.

EJERCICIO 3

Rehacer el ejercicio anterior utilizando un método en la clase Cliente de tipo factoría, esto es que me devuelva instancias de Cliente.

EJERCICIO 4

Crear la clase Colegio con una propiedad de tipo colección para sus alumnos. Rellenar esta propiedad con el nombre de los alumnos utilizando un List, Set, Map y Properties.

Recuperar los beans declarados y mostrar los alumnos por pantalla.

EJERCICIO 5

Modificar el ejercicio 1 para declarar dos beans, uno será de tipo Singleton y otro Prototype.

Recuperar los beans varias veces desde la clase principal y comprobar si son el mismo o diferentes utilizando el método equals y el operador ==.

EJERCICIO 6

Agregar métodos de ciclo de vida al ejercicio 2 y forzar la eliminación de estos en el método main.

EJERCICIO 7

Rehacer el ejercicio 2 generando un solo bean de Cliente y Factura e inyectando las propiedades con el espacio de nombres p.

2.- SPRING AOP

EJERCICIO 1

Vamos a crear la clase Taller con un método arreglarCoche().

En otra clase Operaciones crearemos los siguientes métodos para recoger las operaciones transversales:

- comprobarCliente(); para saber si es antiguo cliente o hay que darlo de alta.
- emitirFactura(); con el total de horas que dura la reparación emitiremos la factura correspondiente.
- asignarCita(); se anotará en la agenda la cita para la reparación
- guardarHistorial(); si el coche se repara en el taller se guarda constancia de la reparación efectuada.
- devolucionPieza(); cuando la pieza es defectuosa y no se puede reparar el coche procedemos a su devolución.
- encuestaCliente(); Todo cliente debe rellenar una encuesta al recoger el coche en el taller.
- recogerCoche(); Los coches son retirados tras su reparación.

Utilizar AOP para ejecutar los métodos anteriores.

EJERCICIO 2

Modificamos el ejercicio anterior incorporando los siguientes parámetros al método arreglarCoche:

- String matrícula
- String nombreCliente

Estos datos deben utilizarse en los métodos de la clase Operaciones.

EJERCICIO 3

Rehacer el ejercicio 1 utilizando anotaciones.

EJERCICIO 4

Rehacer el ejercicio 2 utilizando anotaciones.