

### Ejercicio 1 – Patrones

Sea un lavarropas semiautomático que posee tres programas de lavado: lavado diario, ropa delicada y ropa muy sucia. Cada programa consiste de lo siguiente:

- Lavado diario: llena de agua hasta el 50%, agrega jabón en polvo, 20 minutos de lavado, cambia el agua, agrega enjuague, 7 minutos de enjuague, 3 minutos de centrifugado.
- Ropa delicada: llena de agua hasta el 100%, agrega jabón en polvo, 40 minutos de lavado, cambia el agua, agrega enjuague, 15 minutos de enjuague, no hay centrifugado.
- Ropa muy sucia: llena de agua caliente hasta el 100%, agrega jabón en polvo, 100 minutos de lavado, cambia el agua, agrega enjuague, 15 minutos de enjuague, 10 minutos de centrifugado.

Considere que el lavarropas posee los siguientes mensajes, los cuales puede utilizar sin implementar.

```
Lavarropas>agregarAgua: porcentaje  
Lavarropas>agregarJabonEnPolvo  
Lavarropas>lavar: minutos  
Lavarropas>vaciarAgua  
Lavarropas>agregarEnjuague  
Lavarropas>enjuagar: minutos  
Lavarropas>centrifugar: minutos
```

Usted debe implementar los siguientes mensajes y todos los que necesite:

```
Lavarropas>elegirPrograma: unPrograma  
Lavarropas>iniciarLavado
```

#### Tareas:

- a.- Diseñe un Lavarrropas programable. Indique que patrones utiliza mediante estereotipos y roles.
- b.- Implemente en smalltalk.
- c.- Escriba las instrucciones para lanzar un lavado diario en un workspace.

### Ejercicio 2 – Refactoring

Horacio desea solicitar licencia por vacaciones a su empleador, por lo cual, su empleador le provee ciertos rangos de fechas en los que puede solicitar la licencia. Por su parte, Horacio quiere tomarse el rango más grande posible, sin embargo, tiene que verificar que en ese rango no cae ninguna fecha de cumpleaños de sus seres queridos.

Dadas las siguientes clases:

```
RangoDeFechas con variables de instancia fechaInicial y fechaFinal  
AgendaPersonal con variable de instancia cumpleDeSeresQueridos
```

Sea el siguiente código:

```
AgendaPersonal>rangoMasGrandeDe: rangos  
| temp encuentre maximo |  
"elimina los que incluyen alguna fecha"  
temp := OrderedCollection new.  
rangos do: [:rango | encuentre := false. cumpleDeSeresQueridos do: [:fecha | fecha between:  
rango fechaInicial and: rango fechaFinal ifTrue: [encuentre := true] ].  
encuentre ifFalse:[temp add: rango] ]  
"Obtiene el maximo"  
maximo := temp first  
temp do: [:rango | rango fechaFinal - rango fechaInicial > (maximo fechaFinal - maximo  
fechaInicial) ifTrue: [maximo := rango] ]  
^maximo
```

#### Tareas:

- a.- identifique los defectos (y la/ las líneas que presentan el defecto)
- b.- indique como corregirlos
- c.- Realice un nuevo diseño e implementación.

### Ejercicio 3 – Frameworks

- 1.- Justifique de forma concisa porque SUnit es un Framework
- 2.- Justifique de forma concisa porque el patrón template method tiene relación con la inversión de control y la extensión de un framework de caja blanca.
- 3.- Indique que parte de SUnit considera que es un frozen spot (puede describir coloquialmente que parte de la funcionalidad o mencionar metodos específicos).