

Orientación Objetos 2 - Parcial 2da fecha

Refactoring

1. Refactorice para **eliminar el condicional** del siguiente código:

```
ClasificadorDeEmpleados>>clasificarEstadoCivil: unEmpleado
unEmpleado situacion == 'Casado'
ifTrue: [ self casados add: unEmpleado]
ifFalse: [ unEmpleado situacion == 'Soltero'
ifTrue: [self solteros add: unEmpleado ]
ifFalse: [unEmpleado situacion == 'divorciado'
ifTrue: [self divorciados add: unEmpleado ]
ifFalse: [ self error: 'No clasificable']]]
```

2. Aplique el refactoring **extraer método** al siguiente código

```
Cuenta>>imprimirEnTranscript
'imprime en el Transcript: titular, numero de cuenta, saldo inicial
y saldo actual'
self imprimirDatosTitular.
Transcript show: "Numero de Cuenta: ", self numeroCuenta, cr.
Transcript show: "Saldo inicial: ", self saldoInicial, " --- ",
"Saldo actual: ", self saldoActual;cr.
```

3. Sea la jerarquía de empleados, en donde de la clase abstracta Empleado heredan las subclases EmpleadoDePlanta y EmpleadoTemporario. **Refactorice los métodos #sueldo y #salarioFamiliar** de la clase Empleado a partir del código que se provee.

```
Clase Empleado
variables de instancia 'tieneConyuge cantidadDeHijos horasPorSemana'
#sueldo
| sueldoBase descuentos |
"calculo del sueldo base"
self esEmpleadoDePlanta ifTrue: [ sueldoBase:= self horasPorSemana * 20]
ifFalse: [ sueldoBase:= self horasPorSemana * 15]
"calculo de los descuentos"
descuentos:= self descuento * self sueldoBase * 100.
"calculo final del sueldo"
^sueldoBase + self salarioFamiliar - descuentos
#salarioFamiliar
self esEmpleadoDePlanta ifTrue:[
self tieneConyuge ifTrue:[
^self cantidadDeHijos * 30 + 50]
ifFalse: [
^self cantidadDeHijos * 30]
ifFalse:[ ^0]
]
```

```
-----
Clase EmpleadoTemporario subclase de Empleado
variables de instancia ''
#esEmpleadoDePlanta
^false
#descuento
^0.10
-----
```

```
Clase EmpleadoDePlanta subclase de Empleado
variables de instancia ''
#esEmpleadoDePlanta
```

```
^true
#descuento
^0.30
```

Frameworks

1. Indique si el framework SUnit es de caja blanca o caja negra y justifique explicando por qué.

2. De un ejemplo de inversión de control en el framework de GUIs de VisualWorks, indicando claramente cual es el método del framework y cuál es el método de aplicación que es llamado desde el método del framework.

3. Indique el orden adecuado para realizar los siguientes pasos en el diseño de una aplicación interactiva con el frameworks de GUIs de VisualWorks:

1. Definir una clase para la aplicación (subclase de ApplicationModel)
2. Dibujar la ventana de la aplicación con el GUI Painter e instalarla en la aplicación
3. Definir los métodos en la aplicación que devuelvan los value models de los widgets
4. Vincular la aplicación con el modelo de dominio
5. Definir métodos en la aplicación para procesar las acciones del usuario
6. Programar y probar el modelo de dominio