Programación Orientada a Objetos 2 - Práctica 1

Ejercicio 1

Para cada una de las siguientes expresiones, indique el orden de evaluación en el que Smalltalk ejecuta, indicando en cada paso el objeto receptor, y el mensaje (tipo de mensaje y colaboradores externos si tiene)

1. `(3 raisedTo: 2) even`
2. `3 + 2 raisedTo: 2 factorial`
3. `3 factorial + 1 factorial between: 4 * 2 + 3 and: (5 raisedTo: 2)`
4. `3 even and: [ 3 between: 2 factorial and: 4 * (4 - 2) ]`
5. `4 even not or: [ 3 even or: [ 'hello' size < 6 ] ]`

Ejercicio 2

Implementarse a ustedes mismos como objetos:

1. Implementar el mensaje `#saludar` que devuelve el string `'hola!'`
2. Implementar el mensaje `#nombre` que devuelve su nombre, e implementar el mensaje `#presentarse` para que diga: `'hola! me llamo Pepito'`
3. Implementar el mensaje `#edad` que devuelve un número con su edad, y cambiar el mensaje `#presentarse` para que diga: `'hola! me llamo Pepito y tengo 38 años'`.
4. Ahora también nos interesa decir la cantidad de materias aprobadas hasta el momento. Implementar el mensaje `#cantidadDeMateriasAprobadas` que inicialmente es `0` pero puede ir incrementando a través del mensaje `#aprobarMateria`.
5. Implementar el mensaje `#esNuevoEnLaCarrera`, que devuelve `true` si la cantidad de materias es `0`, y `false` en caso contrario.
6. Implementar otra persona con diferente edad, y que ambas personas puedan responder el mensaje `#esMasViejoQue:`. También que podamos comparar los nombres respecto a su longitud `#tieneNombreMasLargoQue:`.
7. Se puede, o no, tener un apodo. Cuando tienen apodo, entonces en la presentación aclarar: `'me llamo Pepito (pero me dicen Pepote)'`. Cuando no hay apodo, dejar la presentación como está.

Ejercicio 3

Implementar un pluralizador de palabras. Un pluralizador recibe una palabra en singular y un número y devuelve la expresión ya pluralizada. Ejemplo: `'3 veces'`, `'2 árboles'`, `'1 teléfono'`.

1. Hacer la primera versión que soporta únicamente poner una `'s'` al final de las palabras.
2. Hacer una versión en la que podamos dar como entrada la palabra en singular y en plural, y así soportar palabras con plurales "irregulares"
3. Implementar las siguientes validaciones:
    a. La cantidad debe ser un número mayor o igual a `0`

b. Las palabras sólo pueden contener letras. Los strings entienden el mensaje `#isWord`.

Ejercicio 4

Implementar una tarjeta similar a la SUBE. La misma inicia con un saldo de $0, pero puede recibir cargas de dinero. Debe responder al mensaje #realizar: que recibe un objeto que representa un viaje, y descuenta el monto del mismo (probar con varios viajes de distintos montos). También podemos preguntar si puede realizar un viaje o no. Podemos viajar siempre y cuando el saldo no quede menor a -$30.

Ejercicio 5

Utilizar objetos replicantes para eliminar las repeticiones de código generadas en los ejercicios 2 y 4.