

Oigo y olvido; veo y recuerdo; hago y comprendo - Confucio

Práctica nº 8

Ejercicio 1

La fabricación de una pizza implica los siguientes pasos, que se ejecutan en orden: amasar, poner la base, poner el queso, hacer el toque especial, cocinarla, y hacer el paso final en frío.

La cantidad de tiempo de amasado y cocción depende de cada tipo de pizza.

Contamos con las siguientes clases y métodos

<code>Pizzero</code> <code>amasar: pizza durante: minutos</code> <code>echarSalsaDeTomatesA: pizza</code> <code>echarSalsaGolf: pizza</code> <code>echarAjoEn: pizza</code> <code>echarAceiteEn: pizza</code> <code>hacerTajitosEn: pizza</code> <code>Pizza (abstracta)</code> <code>pizzero</code> <code>horno</code> <code>agregar: cantidad de: producto</code>	<code>Horno</code> <code>cocinar: producto durante: minutos</code> <code>Productos</code> <code>jamón (mc)</code> <code>muzza (mc)</code> <code>tomates (mc)</code> <code>morron (mc)</code> <code>cebolla (mc)</code> <code>palmitos (mc)</code> <code>aceitunas (mc)</code>
---	--

Implementar la fabricación de las pizzas, y hacerla andar para las siguientes variedades:

–Napo: se amasa 6 minutos, poner la base consiste en echar salsa de tomates y hacerle tajitos, poner el queso consiste en agregar 250 gramos de muzza, el toque especial consiste en echar ajo, se cocina 10 minutos, el paso final en frío es agregar 400 gramos de tomates.

–Tropical: se amasa 5 minutos, poner la base consiste en echar salsa de tomates, poner el queso consiste en agregar 300 gramos de muzza, el toque especial consiste en agregar 100 gramos de jamón y 100 de morrones, se cocina 6 minutos, el paso final en frío es agregar 150 gramos de palmitos y echar salsa golf.

–Musa: se amasa 4 minutos, poner la base consiste en echar salsa de tomates, poner el queso consiste en agregar 300 gramos de muzza, no tiene toque especial, se cocina 12 minutos, el paso final en frío consiste en agregar 50 gramos de aceitunas.

–Fugazza: se amasa 9 minutos, poner la base consiste en hacer tajitos y agregar 250 gramos de cebolla, no lleva queso, el toque especial es echar aceite, se cocina 14 minutos, no tiene paso final en frío.

Ejercicio 2

Tenemos la clase `Caja` con estos métodos

```
pesoContenidos
  ^self contenidos inject: 0 into: [:total :cont | total + cont peso]
estaSobrecargada
  ^self pesoContenidos > self pesoMaximo
necesitaSerAsegurada
  ^(self contenidos size > 6) |
```

```
((self contenidos select: [:cont | cont costo > 1000]) size > 3)
```

Tenemos la clase `Bulto` cuyas instancias entienden los mensajes `precio` y `pesoTotal`.

Lograr que en una caja puedan entrar objetos que representen bultos, aunque no van a ser precisamente instancias de la clase `Bulto`. No se puede tocar la clase `Bulto`. Lo mismo con pizzas, lograr que puedan entrar objetos que representan pizzas sin tocar la clase `Pizza`. El costo de una pizza es el peso de sus ingredientes * 3.

Ejercicio 3

Analizar los usos del patrón *template method* en la clase `Magnitude`.

Ejercicio 4

Nos piden modelar las preguntas de un examen, vamos a tener distintos tipos de pregunta. Cada pregunta lleva un puntaje.

Queremos saber cuánto suma una pregunta dada una respuesta. Si la respuesta es correcta, entonces suma el puntaje. Si la respuesta es incorrecta, entonces suma un número (que puede ser negativo) que depende del tipo de pregunta.

Queremos considerar estos tipos de pregunta:

- multiple choice, la respuesta es correcta si está entre las opciones verdaderas (puede ser más de una, p.ej. una pregunta con a,b,c,d,e donde son verdaderas a y d). Si es incorrecta resta la mitad del puntaje.
- exacta, la respuesta es correcta si coincide con la respuesta adecuada. Si es incorrecta resta un punto.
- en rango, la respuesta es correcta si está entre los valores mínimo y máximo aceptables. Si es incorrecta no suma ni resta.

Modelar las preguntas tal que se pueda preguntar a una pregunta cuánto suma para una determinada respuesta.

Ejercicio 5

En un sistema que se usa para armar la programación de radios tenemos la siguiente clase

`Radio`

```
>>agregarPrograma: unPrograma
    (self sePuedeIncorporar: unPrograma)
        ifTrue: [programas add: unPrograma]
        ifFalse: [Error raiseSignal]
>>sePuedeIncorporar: unPrograma
    ^ (self estaLibreEntre: unPrograma horaInicio y: unPrograma horaFin)
    & (self tiposDeProgramaAceptables includes: unPrograma tipoDePrograma)
>>presupuesto
    ^self programas inject: 0 into: [:prog | prog presupuesto]
```

y la clase `Programa` cuyas instancias entienden los mensajes: `horaInicio`, `horaFin`, `tipoDePrograma`, `presupuesto`.

Nos llega una clase `BloqueDeNoticias`, cuyas instancias entienden los siguientes mensajes: `duracion`, `horaInicio`, `costo`.

Lograr que se puedan agregar bloques de noticias como programas, aunque no van a ser representados exactamente por instancias de BloqueDeNoticias. El presupuesto de un bloque de noticias es su costo + 100 pesos, el tipo de programa es 'Noticias'. No se puede tocar la clase `BloqueDeNoticias`.

Ejercicio 6

En una empresa se manejan distintos productos, de los cuales queremos manejar costo y precio de venta. El precio de venta de un producto es: $\text{costo} + \text{impuestos} + (1 * \text{porcentaje de ganancia})$; el porcentaje se establece para cada familia de productos, de cada producto se sabe a qué familia pertenece.

Hay tres tipos de producto:

- 1.producto fabricado. Su costo se calcula como: costo de materias primas más costo de fabricación. El costo de fabricación es: $\text{costo diario de un trabajador (que se establece para cada familia de productos), cantidad de trabajadores que hacen falta para producirlo (que se establece para cada producto), cantidad de unidades del producto que se producen en un día (que se establece para cada producto)}$. No pagan impuestos.
- 2.producto comprado: Su costo es el precio de compra, que se establece para cada producto. Los impuestos son del 20% del precio de compra.
- 3.producto conservado: es como uno comprado, pero se le suma el costo de almacenamiento, que es $\text{costo diario del almacenamiento (que se establece para cada familia de productos)} * \text{cantidad de día que hay que conservarlo después de comprado (que se establece para cada producto conservado)}$. Los impuestos son del 5% del costo de almacenamiento.

Modelar los productos y familias de forma tal de poder conocer para cada producto su costo y su precio de venta.