

No hay cosa tan difícil que, a fuerza de estudiarla, no parezca fácil – Terencio

Práctica nº 4

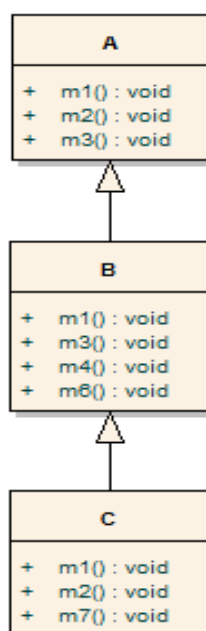
Ejercicio 1

Implemente en Smalltalk cuentas bancarias como las vistas en la teoría. Se necesita tener cajas de ahorro y cuentas corrientes. De ambas puede extraerse y depositarse dinero y pedir el saldo. La única diferencia entre ellas es que las cajas de ahorro no permiten extraer más de 500 pesos de una vez ni en descubierto (es decir, si el saldo queda bajo cero como resultado de la extracción), mientras que las cuentas corrientes no tienen un límite predefinido de monto para cada extracción y tienen un límite de descubierto permitido. Puede que distintas cuentas corrientes tengan distintos límites de descubierto, y ese dato se indica al momento de crear la cuenta.

- Implementar los mensajes para crear e inicializar estos objetos.
- En un workspace cree una caja de ahorro, inspeccione y compruebe que se ha inicializado correctamente. Idem para una cuenta corriente.
- Implemente los métodos de extracción, depósito y saldo y verifique su funcionamiento en el workspace.
- Nos agregan el requerimiento de saber, para cada cuenta corriente, cuántas veces hizo extracciones en descubierto. Agregar el código correspondiente. Reflexionar acerca de cómo la herencia ayuda a que los cambios se localicen.

Ejercicio 2

- Suponga la siguiente jerarquía:



Los métodos se han definido de la siguiente manera:

A	B	C
m1 ^3	m1 ^6	m1 ^self m4
m2 ^5	m4 ^self m2 + super m3	m2 ^9
m3 ^self m6 + self m2	m3 ^4	m7 ^super m6
	m6 ^self m2	

¿qué retornan las siguientes expresiones?

| unObjeto |

unObjeto := C new.

a) unObjeto m7

b) unObjeto m1

Ejercicio 3

Modele la clase Dice (dado) para que sea posible crear dados y tirarlos. Analice las clases Random y FastRandom y vea si les son útiles. Extienda luego la clase Dice para que sea posible definir dados de número arbitrario de caras y con cualquier objeto en sus caras. Ej: dadoMultiple := Dice withFacesIn: #('simple' 'x2' 'pierde turno' 'x3' 'x4')

Ejercicio 4

Modele e implemente la clase NumeroComplejo (que tiene un componente real y otro imaginario) para que sus instancias extiendan los mensajes:

+ unNumeroComplejo

- unNumeroComplejo

= unNumeroComplejo

modulo

parteReal

parteImaginaria

- ¿De quién es subclase NumeroComplejo? ¿Por qué?
- Modifique la implementación de la clase NumeroComplejo para que internamente se modele como módulo y argumento. Para quienes hacen uso de la clase, ¿cuál fue el impacto?

Ejercicio 5

Se desea modelar e implementar el examen de ingreso a una facultad.

Para ello, se sabe que existen 4 tipos de ejercicios posibles que pueden ocurrir en el examen:

- Ejercicios de multiple-choice con 3 opciones (sólo una correcta).

- Ejercicios de preguntas teórico-prácticas que el aspirante deberá desarrollar.
- También habrá preguntas que se contestan con un valor numérico resultante de un ejercicio de ingenio.
- Afirmaciones que se contestan Verdadero o Falso.

Se debe modelar e implementar cada uno de los tipos de ejercicios, sabiendo que de cada uno se conoce el puntaje total que suma al examen y el área de conocimiento del tema al que aplican: Lengua, Matemática, o Lógica. También queremos saber para cada ejercicio cuánto tiempo insume su corrección; para cada pregunta teórico-práctica se indica, para el resto es 0.

Piense con detalle cómo deben crearse o instanciarse las preguntas teórico-prácticas, los ejercicios multiple-choice, y los otros dos tipos.

Ejercicio 6:

a) Se desea modelar una aplicación para guardar información sobre discos de música (CD) y películas (DVD). De los CDs nos interesa conocer el título del cd, el intérprete, la cantidad de pistas, el tiempo total que dura, y un comentario. De los DVDs nos interesa el título de la película, el rubro, el tiempo total que dura, el director y un comentario.

Modelar también los locales de venta, para cada local se indica qué ítems (CDs y/o DVDs) se venden. Para un local se tiene que poder saber: el conjunto de títulos de todos sus ítems, los ítems donde aparece una determinada persona, y la cantidad de ítems largos. Decimos que en un CD aparece el intérprete, y en un DVD aparece el director. Un CD se considera largo si tiene más de 15 canciones, en cambio un DVD se considera largo si dura más de 140 minutos.

b) Agregar a cada DVD los autores de la película, teniendo en cuenta que los actores deben incluirse entre las personas que aparecen en el DVD.

Ejercicio 7:

Una fábrica de relojes fabrica un sólo modelo de relojes y nos piden modelar su reloj. Dicho reloj debe poder responder a los siguientes mensajes:

`mostrarFecha`, devuelve el día, mes y año

`mostrarHorario`, devuelve la hora, y minutos del día

`mostrarHora`, devuelve la hora del día

`mostrarMinutos`, devuelve los minutos que pasaron de la hora actual

Debido a una nueva campaña de marketing para deportistas la fábrica decide sacar un nuevo modelo. A este nuevo modelo se lo denomina "reloj cronómetro". Cómo modelaría este nuevo reloj si debería responder a los mismos mensajes del reloj original y además, debería agregar estos 3 mensajes:

`activarCronometro`, activa el cronómetro y empieza a controlar el tiempo transcurrido.

`detenerCronometro`, desactiva el cronómetro, y ya no controla el tiempo transcurrido.

`mostrarCronometro`, devuelve la cantidad de horas, minutos y segundos que pasaron desde que se activó el cronómetro.

Ayuda: para modelar la parte del cronómetro quizás alcanza con guardar el instante de tiempo en que se activa y después realizar el calculo cada vez que se pide mostrar el cronometro.

Ejercicio 8:

Modele el ejercicio de los “micros empresarios” de la práctica anterior usando los conceptos de jerarquía de clase, herencia, métodos y clases abstractas, etc.

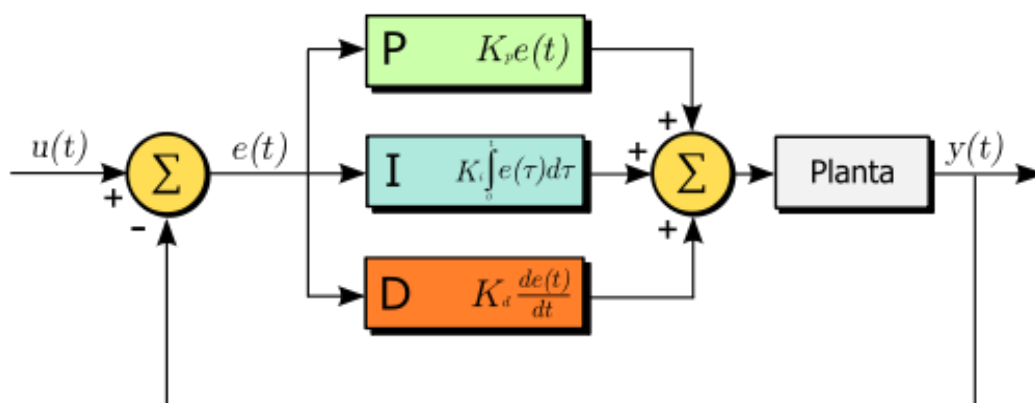
Para eso considerar

- que queremos agregar el nombre, apellido y fecha de nacimiento de cada persona, pudiendo preguntar para cada persona la edad y si es o no mayor (los mayores son los que tienen al menos 21 años).
- agregar a los “muy moderados”, que son como los moderados, pero además para subirse a un micro tiene que haber en el momento en que se suben más personas sentadas que paradas.
- Agregar un factor de selectividad, que será de 0 para los apurados, 3 para los claustrofóbicos, 8 para los fiacas, para los moderados el número de asientos libres que exigen, y para los muy moderados, el mismo criterio que para los moderados sumando dos. La selectividad de un obsecuente es la misma que la de su jefe.
- Agregar el índice de alegría de una persona, que es $(200 - \text{edad}) / 4$. Para los moderados restar su factor de selectividad.

Ejercicios de Automatización y control

IACI 1 - Un controlador PID tiene el mismo comportamiento que un controlador PI, pero agrega la acción Derivativa. Lo mismo, un PI tiene el mismo comportamiento que un controlador Proporcional con acción Integral.

En la siguiente figura se ilustra la acción de control de estos controladores:



Modele los controladores de tipo P, PI y PID aproximando la integral como la suma de los valores recibidos y la derivada como la resta del valor actual menos el valor anterior (es decir, que el tiempo de muestreo es igual a 1).

IACI 2- Un PWM (Modulación por Ancho de Pulso) es una señal binaria, función de un período y un ciclo de trabajo, siendo:

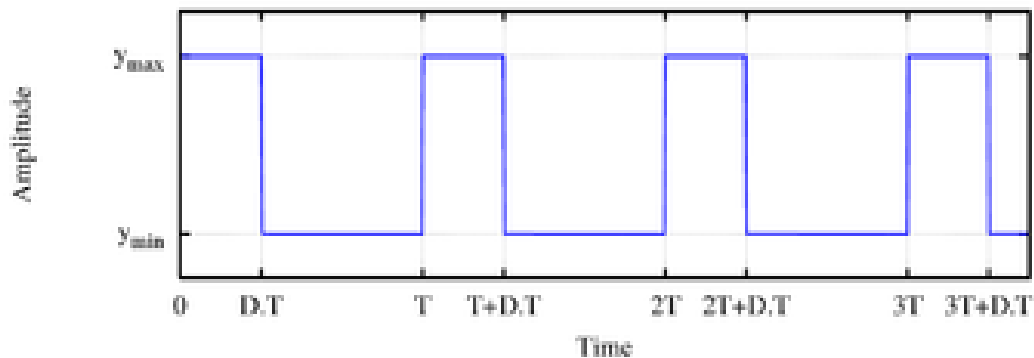
$$D = \frac{\tau}{T}$$

D es el ciclo de trabajo

τ es el tiempo en que la función es positiva (ancho del pulso)

T es el período de la función

Un generador de señales PWM genera salidas como esta:



Modelar el generador de señales PWM considerando que el período y el ancho de pulso son valores enteros y que cada vez que se consulta su valor de salida avanzamos 1 unidad de tiempo.

Considere que no tiene sentido hablar de un PWM sin período y ancho de pulso.

IACI 3 - Un PLC es un dispositivo electrónico que posee: entradas y salidas analógicas y digitales y registros de almacenamientos (analógicos también).

A su vez, un PLC tiene funciones internas capaces de, por ejemplo, generar una señal PWM, un control PID, etc. El resultado de estas funciones puede conectarse a una salida física.

Modele la clase PLC.

PLC
<ul style="list-style-type: none">- entradaAnalogica- entradaDigital- salidaAnalogica- salidaDigital- registroAlmacenamiento
<ul style="list-style-type: none">+ generarPWM: unPeriodo anchoDePulso: unAnchoDePulso+ senialPWM+ controlarProp: unKp int: unKi deriv: unKd+ senialControl: <u>unError</u>