

1. Defina el tipo *MENSAJES*.

Solución $MENSAJES ::= ok \mid numeroClienteEnUso$

2. Escriba en C un programa que implemente la operación *NuevoCliente*.

Solución

```
MENSAJES nuevocliente(DNI d) {
    if (!member(d,ca)) {
        add(ca,d,0);
        return OK;
    } else {
        return NUMEROCLIENTEENUSO;
    }
}
```

3. Respecto del ejercicio 2, ¿que diferencias tiene con el modelo? ¿Como implementó el tipo *DNI*? ¿Su implementación es isomorfa al tipo *DNI*? ¿Puede garantizar que su implementación verifica la especificación? ¿Como lo haría?

Solución COMPLETAR.

4. Suponga que el banco exige que para abrir una caja de ahorro el cliente debe depositar en el acto una suma de dinero no inferior a los \$200. Modele este requerimiento.

Solución

$$\frac{\text{montoMinimo} : \mathbb{N}}{\text{montoMinimo} = 200}$$

$NuevoCliente == NuevoClienteOk \vee ClienteExiste \vee MontoInsuficiente$

<i>NuevoClienteOk</i>	
$\Delta Banco$	
$d? : DNI$	
$m? : DINERO$	
$rep! : MENSAJES$	
$d? \notin dom(a)$	
$m? \geq montoMinimo$	
$ca' = ca \cup \{d? \mapsto m?\}$	
$rep! = ok$	
<i>MontoInsuficiente</i>	
$m? : DINERO$	
$rep! : MENSAJES$	
$m? < montoMinimo$	
$rep! = montoInsuficiente$	

5. Respecto del ejercicio 4, determine los cambios que se deberían dar en el entorno para que su estado coincida con el del sistema.

Solución COMPLETAR.

6. Complete la especificación de *ExtraerCA* con todos los esquemas de error necesarios. *Ayuda:* no olvide los esquemas ya definidos para *Extraer*.

Solución $ExtraerCA == ExtraerCAOk \vee ClienteInexistente \vee MontoIncorrecto \vee MontoInadmisibile$

<i>MontoInadmisibile</i>	
$\Xi Banco$	
$d? : DNI$	
$m? : DINERO$	
$rep! : MENSAJES$	
$m? > \min \{ca(d?), limiteExtraerCA\}$	
$rep! = montoInadmisibile$	

7. Indique los cambios que deberían hacerse si el BCRA decide eliminar el límite a las extracciones.

Solución Debe cambiarse $\text{mín}\{ca(d?), \text{limiteExtrCA}\}$ por $\{ca(d?)\}$.

8. Suponga, ahora, que el BCRA reglamenta que ningún banco puede permitir extracciones de caja de ahorro de más del 50 % del saldo. Modele este requerimiento.

Solución

ExtraerCAOk ΔBanco $d? : \text{DNI}$ $m? : \text{DINERO}$ $\text{rep!} : \text{MENSAJES}$	
$d? \in \text{dom}(a)$ $0 < m?$ $m? \leq \text{mín}\{ca(d?) * 50/100, \text{limiteExtrCA}\}$ $ca' = ca \cup \{d? \mapsto m?\}$ $\text{rep!} = \text{ok}$	

9. Modele una operación que muestre un listado con los saldos de un grupo de cajas de ahorro.

Solución

ListarSaldosOk ΞBanco $d? : \mathbb{P}\text{DNI}$ $\text{rep!} : \text{DNI} \rightarrow \text{DINERO}$	
$d? \subseteq \text{dom}(ca)$ $\text{rep!} = d? \triangleleft ca$	

10. Modele una operación que da de baja un cierto conjunto de clientes.

Solución

$CerrarCajasOk$	_____
$\Delta Banco$	
$d? : \mathbb{P}DNI$	
$d? \subseteq dom(ca)$	
$ran(d? \triangleleft ca) = \{0\}$	
$ca' = d? \triangleleft ca$	

11. Enumere algunos escenarios en los cuales haber elegido tipos básicos para los nombres y los números de cuenta sea un error. Para cada escenario, defina los tipos que enmendarían el error.

Solución COMPLETAR.

12. Definir el estado inicial para *CajaAhorros*.

Solución

$CajaAhorrosInit$	_____
$CajaAhorros$	
$saldo = 0$	
$his = \langle \rangle$	

13. Definir las operaciones de extracción y consulta de saldo. Seguir el patrón para los nombres de los esquemas.

Solución

$CAPedirSaldo$	
$\Xi CajaAhorros$	
$rep! : DINERO$	
$rep! = saldo$	
$CAExtraerOk$	
$\Delta CajaAhorros$	
$m? : DINERO$	
$rep! : MENSAJES$	
$0 < m? \leq saldo$	
$his' = his \cap \langle -m? \rangle$	
$saldo' = saldo - m?$	
$rep! = ok$	

14. ¿Por qué no incluimos $m?$ como variable de entrada y $rep!$ como variable de salida? Justifique.

Solución Ambas estarán definidas al haber incluido el subesquema $CADepositarOK$.

15. Modele la operación total $Depositar$; es decir, describa los casos erróneos.

Solución $Depositar == DepositarOk \vee DepositarE \vee CuentaInvalida$

$DepositarE$	
$\Xi Banco2$	
$CAMontoIncorrecto$	
$n? : NUMCTA$	
$n? \in dom(ca)$	
$ca(n?) = \Theta CajaAhorros$	

16. Defina el estado inicial del banco usando el esquema de estados *Banco3*.

Solución

<i>Banco3Init</i>	_____
<i>Banco3</i>	_____
<i>clis</i> = \emptyset	
<i>ca</i> = \emptyset	
<i>tits</i> = \emptyset	

17. Modele la operación de deposito pero usando el esquema de estados *Banco3*. ¿Necesita usar promoción de operaciones?

Solución

<i>DepositarOk</i>	_____
Δ <i>Banco3</i>	_____
<i>d?</i> : <i>DNI</i>	
<i>n?</i> : <i>NUMCTA</i>	
<i>m?</i> : <i>DINERO</i>	
<i>rep!</i> : <i>MENSAJES</i>	
<i>d?</i> \mapsto <i>n?</i> \in <i>tits</i>	
<i>n?</i> \in <i>dom</i> (<i>ca</i>)	
<i>m?</i> > 0	
$ca' = ca \oplus \left\{ n? \mapsto \left(first(ca(n?)) + m?, second(ca(n?))^{\wedge} \langle m? \rangle \right) \right\}$	
<i>clis'</i> = <i>clis</i>	
<i>tits'</i> = <i>tits</i>	
<i>rep!</i> = <i>ok</i>	

18. Defina el estado inicial del banco usando el esquema de estados *Banco4*.

Solución

<i>Banco4Init</i>	_____
<i>Banco4</i>	_____
<i>ca</i> = \emptyset	

19. Modele la operación de deposito pero usando el esquema de estados *Banco4*. ¿Necesita usar promoción de operaciones?

Solución COMPLETAR.

20. Complete la operación *CAExtraer*.

Solución $CAExtraer == CAExtraerOk \vee CAMontoIncorrecto \vee CASaldoInsuficiente$

21. ¿Por qué decimos que esta forma de definir *PedirSaldo* es artificiosa? ¿Qué es lo que la diferencia de las otras dos?

Solución COMPLETAR.

22. Modelar la operación *CAPedirSaldo*.

Solución ¿Ejercicio 13?

23. Modelar el alta y la baja de una caja de ahorros. ¿Puede utilizar el esquema marco de promoción? ¿Por que?

Solución

<i>NuevaCuentaOk</i>	
$\Delta Banco2$	
$d? : DNI$	
$n? : NUMCTA$	
$rep! : MENSAJES$	
$d? \in dom(clis)$	
$d? \mapsto n? \notin tits$	
$n? \notin dom(ca)$	
$ca' = ca \cup \{n? \mapsto CajaAhorrosInit\}$	
$clis' = clis$	
$tits' = tits$	
$rep! = ok$	

24. Modele las operaciones de extracción, alta y baja de un cliente y pedido de saldo utilizando el esquema de estados *Banco3*.

Solución

$PedirSaldoOk$ $\Xi Banco3$ $n? : NUMCTA$ $d? : DNI$ $rep! : DINERO$	
$(d? \mapsto n?) \in tits$ $n? \in dom(ca)$ $rep! = first(ca(n?))$	
$ExtraerOk$ $\Delta Banco3$ $n? : NUMCTA$ $d? : DNI$ $m? : DINERO$	
$(d? \mapsto n?) \in tits$ $n? \in dom(ca)$ $0 < m? \leq first(ca(n?))$ $ca' = ca \oplus \left\{ n? \mapsto \left(first(ca(n?)) - m?, second(ca(n?)) \wedge \langle -m? \rangle \right) \right\}$ $clis' = clis$ $tits' = tits$	
$NuevoClienteOk$ $\Delta Banco3$ $n? : NOMBRE$ $c? : DOMICILIO$ $d? : DNI$	
$d? \notin dom(clis)$ $clis' = clis \cup \{d? \mapsto (n?, c?)\}$ $ca' = ca$ $tits' = tits$	

<i>BorrarClienteOk</i>	
$\Delta Banco3$	
$d? : DNI$	
$d? \in dom(clis)$	
$d? \triangleleft tits = \emptyset$	
$clis' = d? \triangleleft clis$	
$ca' = ca$	
$tits' = tits$	

25. Modele las operaciones de extracción, alta y baja de un cliente y pedido de saldo utilizando el esquema de estados *Banco4*.

Solución

<i>PedirSaldoOk</i>	
$\Xi Banco4$	
$n? : NUMCTA$	
$d? : DNI$	
$rep! : DINERO$	
$n? \in dom(ca)$	
$ca(n?) = \Theta CajaAhorros2$	
$d? = \Theta Cliente2.dni$	
$\Theta Cliente2 \in tits$	
$rep! = saldo$	

COMPLETAR.

26. En el esquema marco de promoción *CajaAhorrosABanco* hay una precondición para la cual no se ha escrito ningún esquema de error para ninguna de las operaciones. ¿Por qué?

Solución COMPLETAR.

27. Falta modelar un esquema de error para la operación *Extraer*. Hágalo.

Solución COMPLETAR.

28. Muestre detalladamente la implementación de lo antedicho en lenguaje C++.

Solución COMPLETAR.

29. Muestre con algún detalle la forma de la implementación en caso de que exista más de un esquema marco de promoción.

Solución COMPLETAR.

30. ¿Por qué $CAExtraerOk[no?/n?]$ es igual a $CAExtraerOk$?

Solución COMPLETAR.

31. ¿Qué ocurre si $no?$ y $nd?$ son iguales? ¿Es necesario que sean diferentes? Si lo es, modifique el modelo para que lo sean.

Solución COMPLETAR.

32. Definir la operación de transferencia usando los esquemas de estados $Banco3$ y $Banco4$.

Solución COMPLETAR.

33. Determine, justificando formalmente su respuesta, si las dos expresiones que siguen son o no equivalentes.

$ExtraerOk[no?/n?]; DepositarOk[nd?/n?, saldo_1/saldo, saldo_2/saldo, his_1/his, his_2/his'']$
 $DepositarOk[no?/n?]; ExtraerOk[nd?/n?, saldo_1/saldo, saldo_2/saldo, his_1/his, his_2/his'']$

Solución COMPLETAR.

34. Modele, para los tres esquemas de estado que venimos considerando, operaciones para:
- a) Obtener todas las cuentas en las que un cliente dado es titular.
 - b) Todos los titulares de una cuenta dada.
 - c) Última extracción por cierre; es decir, se extrae el saldo completo de una cuenta y luego se la cierra.
 - d) Obligatoriedad de un depósito inicial; es decir, para que una persona pueda abrir una caja de ahorros debe depositar una suma de dinero en el momento de la apertura.
 - e) Eliminar un titular de una caja de ahorros, si es que hay más de uno; solo el mismo titular puede hacer esto.
 - f) Agregar un titular a una caja de ahorros; solo uno de los titulares puede hacerlo.

Solución COMPLETAR.

35. Complete en C++ la implementación de todas las operaciones definidas sobre *Banco2*.

Solución COMPLETAR.

36. Especifique el operador *sumSeq*, el cual retorna la suma de una secuencia de números enteros. ¿Qué parágrafo Z utilizará? ¿Un esquema, una definición axiomática, una definición genérica?

Solución COMPLETAR.

37. Defina el mismo invariante para los esquemas de estado *Banco3* y *Banco4*.

Solución COMPLETAR.

38. Intente formalizar el invariante: «en cualquier momento el saldo de una caja de ahorro es igual a la suma de todos los depósitos menos la suma de todas las extracciones».

Solución COMPLETAR.

39. Escriba las designaciones de los requerimientos enunciados en la sección 3.

Solución COMPLETAR.

40. Escriba las designaciones de los requerimientos enunciados en la sección 4.

Solución COMPLETAR.

41. ¿Necesita escribir nuevas designaciones o modificar las que escribió en el problema 40 para el modelo basando en *Banco3*?

Solución COMPLETAR.