Tipos abstractos de datos básicos

Algoritmos y Estructuras de Datos II, DC, UBA.

Índice

1.	TAD NOMBRECLIENTE	2
2.	TAD NombreTítulo	2
3.	TAD PRECIO	2
4.	TAD TÍTULO	2
5 .	TAD PROMESA	3
6.	TAD WOLFIE	5

1. TAD Nombrecliente

TAD NOMBRECLIENTE

TAD NombreCliente es String

Fin TAD

2. TAD Nombre Título

TAD NombreTítulo

TAD NombreTítulo es String

Fin TAD

3. TAD Precio

TAD PRECIO

TAD Precio es Nat

Fin TAD

4. TAD TÍTULO

TAD TÍTULO

géneros título

usa Nat, NombreTítulo, Precio

exporta nombre, maxCant, cotizaciónInicial

igualdad observacional

$$(\forall t_1, t_2 : \text{título}) \ \left(t_1 =_{\text{obs}} t_2 \Longleftrightarrow \begin{pmatrix} (nombre(t_1) = nombre(t_2)) \land \\ (maxCant(t_1) = maxCant(t_2)) \land \\ (cotizaci\'onInicial(t_1) = cotizaci\'onInicial(t_2)) \end{pmatrix} \right)$$

generadores

crear Título : Nombre Título \times Nat \times Precio \longrightarrow Título

observadores básicos

 ${\tt nombre} \hspace{1.5cm} : \hspace{.1cm} \texttt{T\'{i}tulo} \hspace{1.5cm} \longrightarrow \hspace{.1cm} \texttt{NombreT\'{i}tulo}$

axiomas

 $(\forall \ nT \colon \mathbf{NombreTitulo}, \forall \ m \colon \mathbf{Nat}, \ \forall \ p \colon \mathbf{Precio})$

nombre(crearTitulo(nT,m,p)) $\equiv nT$

 $\max Cant(crearTitulo(nT,m,p)) \equiv m$

 $cotizaciónInicial(crearTítulo(nT,m,p)) \equiv p$

Fin TAD

5. TAD PROMESA

TAD PROMESA

géneros promesa

usa Nombre Título, Nombre Cliente, Nat, Bool

exporta nombreDueño, nombreEmpresa, cota, cantidad, esDeCompra?, esDeVenta?

igualdad observacional

 $(\forall p_1, p_2 : \text{promesa}) \quad \left(p_1 =_{\text{obs}} p_2 \iff \begin{pmatrix} (nombreDue\~no(p_1) = nombreDue\~no(p_2)) \land \\ (nombreEmpresa(p_1) = nombreEmpresa(p_2)) \land \\ (cota(p_1) = cota(p_2)) \land \\ (cantidad(p_1) = cantidad(p_2)) \land \\ (esDeCompra?(p_1)) = esDeCompra?(p_2) \end{pmatrix} \right)$

generadores

promesa De
C : Nombre Cliente × Nombre Título × Nat × Nat \longrightarrow Promesa

promesaDeV : Nombre $Cliente \times NombreT$ ítulo $\times Nat \times Nat$ $\longrightarrow Promesa$

observadores básicos

nombre Dueño : Promesa \longrightarrow Nombre Cliente

nombre Empresa : Promesa \longrightarrow Nombre Título

cota : Promesa \longrightarrow Nat

cantidad : Promesa \longrightarrow Nat

es DeCompra? : Promesa \longrightarrow Bool

otras operaciones

esDeVenta? : Promesa \longrightarrow Bool

axiomas

 $(\forall nC: nombreCliente, \forall nT: nombreTitulo, \forall cot: Nat, \forall cant: Nat)$ $nombreDue\~no(promesaDeC(nC,nT,cot,cant))$ \equiv nC $nombreDue\~no(promesaDeV(nC,nT,cot,cant))$ \equiv nC nombreEmpresa(promesaDeC(nC,nT,cot,cant)) \equiv nT nombre Empresa(promesa DeV(nC,nT,cot,cant)) \equiv nT $\cot a(\operatorname{promesaDeC}(nC,nT,\cot,\operatorname{cant}))$ $\equiv \cot$ $\cot a(\operatorname{promesaDeV}(\operatorname{nC},\operatorname{nT},\cot,\operatorname{cant}))$ $\equiv \cot$ cantidad(promesaDeC(nC,nT,cot,cant)) \equiv cant cantidad(promesaDeV(nC,nT,cot,cant)) \equiv cant esDeCompra?(promesaDeC(nC,nT,cot,cant)) ≡ true esDeCompra?(promesaDeV(nC,nT,cot,cant)) \equiv false esDeVenta?(p) $\equiv \neg esDeCompra?(p)$

Fin TAD

6. TAD WOLFIE

TAD WOLFIE

géneros Wolfie

usa $Conj(\alpha)$, Nat, Título, NombreCliente, NombreTítulo, Promesa, Precio, Bool

exporta nombresTítulos, nombresClientes, dameTítulo, promesasSinCumplir, cotización, accionesPorCliente, enAlza

igualdad observacional

$$\left(\forall w_1, w_2 : \text{wolfie} \right) \left(\begin{array}{c} (nombresTitulos(w_1) =_{\text{obs}} nombresTitulos(w_2)) \land \\ (nombresClientes(w_1) =_{\text{obs}} nombresClientes(w_2)) \land_{\text{L}} \\ (\forall nT : nombreTitulo \in nombresTitulos(w_1)) \\ promesasSinCumplir(nT, w_1) \\ promesasSinCumplir(nT, w_2) \\ dameTitulo(nT, w_1) =_{\text{obs}} dameTitulo(nT, w_2) \land \\ cotización(nT, w_1) = cotización(nT, w_2) \land \\ enAlza(nT, w1) = enAlza(nT, w2) \land_{\text{L}} \\ (\forall nC : nombreCliente \in nombresClientes(w_1)) \\ accionesPorCliente(nC, nT, w_1) \\ accionesPorCliente(nC, nT, w_2) \\ \end{array} \right)$$

generadores

observadores básicos

nombresTítulos : Wolfie \longrightarrow Conjunto(NombreTítulo)

```
nombresClientes: Wolfie \to Conjunto(NombreCliente)
       dameTítulo \;:\; NombreTítulo \;nT \;\times\; Wolfie \;w \;\;\longrightarrow\; Título
                                                                                                       \{nT \in nombresT \hat{t}ulos(w)\}
       cotización : NombreTítulo nT × Wolfie w --> Precio
                                                                                                       \{nT \in nombresTítulos(w)\}
       en
Alza : Nombre
Título n<br/>T\timesWolfie w \,\longrightarrow\,Bool
                                                                                                       \{nT \in nombresTitulos(w)\}
       promesas
Sin<br/>Cumplir : Nombre
Título n<br/>T\timesWolfie w \longrightarrow Conjunto<br/>(Promesa)
                                                                                                       \{nT \in nombresTitulos(w)\}
       acciones
Por<br/>Cliente : Nombre
Cliente n<br/>C\timesNombre
Título n<br/>T\timesWolfie w~\longrightarrow~ Nat
                                                                      \{nT \in nombresTitulos(w) \land nC \in nombresClientes(w)\}
     otras operaciones
       disponibles : Nombre
Título n<br/>T\timesWolfie w \;\longrightarrow\; Nat
                                                                                                       \{nT \in nombresTitulos(w)\}
       acciones
Vendidas : Nombre<br/>Título nT × Conjunto(NombreCliente) c × Wolfie w \longrightarrow Nat
                                                                        \{nT \in nombresTitulos(w) \land c \subseteq nombresClientes(w)\}
       sePuedeEjecutar : Precio \times Nat \times Promesa \longrightarrow Bool
       revisarPendientes : Conjunto(Promesa) × Precio × Nat × Conjunto(Promesa) → Conjunto(Promesa)
       sumarCumplidas : NombreCliente \times Conjunto(Promesa) \times Conjunto(Promesa) \longrightarrow Nat
       ya
Existe : Promesa \times Conjunto
(Promesa) \longrightarrow Bool
     axiomas
     (\forall d: \text{nat}, \forall m: \text{precio}, \forall nC, cl: \text{nombreCliente}, \forall nT, nT': \text{nombreTítulo}, \forall t: Título}, \forall p: \text{promesa}, \forall cCl:
conjunto(nombreCliente), \forall sinCumplir, cP, cViejo, cNuevo: conjunto(promesa), \forall w: wolfie)
       nombresTítulos(crearW)
       nombresTítulos(agregarTítulo(t,w))
                                                                \equiv Ag(nombre(t), nombresTítulos(w))
       nombresTítulos(agregarCliente(cl,w))
                                                                \equiv nombresTítulos(w)
       nombresTítulos(agregarPromesa(p,w))
                                                                \equiv nombresTítulos(w)
```

```
nombresTítulos(recotizarTítulo(nT, p,w))
                                                  \equiv \text{nombresTítulos(w)}
nombresClientes(crearW)
                                                  \equiv \emptyset
nombresClientes(agregarTitulo(t,w))\\
                                                  \equiv nombresClientes(w)
nombresClientes(agregarCliente(cl, w))
                                                  \equiv Ag(cl,nombresClientes(w))
nombresClientes(agregarPromesa(p ,w))
                                                  \equiv nombresClientes(w)
nombresClientes(recotizarTítulo(nT, p ,w))
                                                  \equiv nombresClientes(w)
dameTítulo(nT, agregarTítulo(t,w))
                                                   \equiv if nT == nombre(t) then
                                                      else
                                                         dameTítulo(nT, w)
                                                      fi
dameTítulo(nT, agregarCliente(cl, w))
                                                  \equiv dameTítulo(nT, w)
dameTítulo(nT, agregarPromesa(p,w))
                                                  \equiv dameTitulo(nT, w)
dameTítulo(nT, recotizarTítulo(nT', p,w))
                                                  \equiv dameTitulo(nT, w)
cotización(nT, agregarTítulo(t,w))
                                                   \equiv if nT = = nombre(t) then
                                                         cotizaciónInicial(t)
                                                         cotización (nT,w)
                                                      fi
cotización(nT, agregarCliente(cl,w))
                                                  \equiv \cot i zación(nT, w)
cotización(nT, agregarPromesa(p,w))
                                                  \equiv \cot i zación(nT, w)
cotización(nT, recotizarTítulo(nT', m ,w))
                                                  ≡ if nT==nT' then m else cotización(nT,w) fi
promesasSinCumplir(nT,crearW)
                                                  \equiv \emptyset
promesasSinCumplir(nT,agregarTítulo(t,w))
                                                  \equiv promesasSinCumplir(nT,w)
promesasSinCumplir(nT,agregarCliente(cl, w))
                                                  \equiv promesasSinCumplir(nT,w)
promesasSinCumplir(nT,agregarPromesa(p,w)) \equiv
```

```
if nT== nombreEmpresa(p) then
         if sePuedeEjecutar (cotización(nombreEmpresa(p),w), disponibles(nombreEmpresa(p),w),p) then
            if esDeCompra(p) then
               promesasSinCumplir(nT,w)
            else
               revisarPendientes (Ø, cotización (nT,w), (disponibles (nT,w) + cantidad (p)), promesasSinCum-
      plir (nT,w))
         else
            Ag(p, promesasSinCumplir(nT,w))
         fi
      else
         promesasSinCumplir(nT,w)
      fi
promesasSinCumplir(nT,recotizarTítulo(nT',m,w)) =
      if nT == nT' then
         revisarPendientes(\emptyset, m, disponibles(nT,w), promesasSinCumplir(nT,w))
         promesasSinCumplir(nT,w)
      fi
enAlza(nT, agregarTítulo(t,w))
                                                 \equiv if nT = = nombre(t) then true else enAlza(nT, w) fi
enAlza(nT, agregarCliente(cl, w))
                                                 \equiv \text{enAlza}(nT, w)
enAlza(nT, agregarPromesa(p ,w))
                                                 \equiv \text{enAlza(nT,w)}
enAlza(nT, recotizarTítulo(nT', m ,w))
                                                 \equiv if nT = = nT' then
                                                       (m>cotización(n,w))
                                                    else
                                                       enAlza(nT,w)
                                                    fi
accionesPorCliente(nC,nT,agregarTítulo(t,w))
                                                 \equiv if nT = = nombre(t) then
                                                       0
                                                    else
                                                       accionesPorCliente(nC, nT, w)
                                                    fi
accionesPorCliente(nC,nT,agregarCliente(c,w))
                                                \equiv if nC = = c then
                                                       0
                                                    else
                                                       accionesPorCliente(nC,nT,w)
                                                    fi
accionesPorCliente(nC,nT,agregarPromesa(p,w)) \equiv
```

```
if nT== nombreEmpresa(p) ∧ nC== nombreDueño(p) ∧ sePuedeEjecutar(cotización(nT,w), disponi-
      bles(nT, w),p) then
         if esDeVenta(p) then
            sumarCumplidas(nC, promesasSinCumplir(nT,w), promesasSinCumplir(nT, agregarProme-
      sa(p,w)) - cantidad(p) + accionesPorCliente(nC, nT, w)
         else
            cantidad(p)+ accionesPorCliente(nC, nT, w)
         fi
      else
            sumarCumplidas (nC, promesasSinCumplir(nT,w), promesasSinCumplir (nT, agregarPromesa (p,
      w))) + accionesPorCliente(nC, nT, w)
accionesPorCliente(nC,nT,recotizarTítulo(nT',c,w)) =
      if nT == nT' then
         sumarCumplidas (nC, promesasSinCumplir (nT, w), promesasSinCumplir (nT, RecotizarTítulo(nT',
         + accionesPorCliente (nC,nT,w)
      else
         accionesPorCliente(nC, nT, w)
      fi
disponibles(nT, w)
                                                \equiv \max Cant(dameTitulo(nT,w)) - accionesVendidas(nT,
                                                   nombresClientes(w), w)
acciones Vendidas (nT, cCl, w)
                                                \equiv if \emptyset?(cCl) then
                                                      0
                                                   else
                                                       accionesPorCliente(dameUno(cCl), nT, w) + acciones-
                                                       Vendidas(nT, sinUno(cCl), w)
                                                   fi
sePuedeEjecutar(m, d, p)
                                                \equiv if esDeCompra(p) then
                                                       (\cot a(p) \leq m \, \wedge \, \mathrm{cantidad}(p) \leq d)
                                                   else
                                                       \cot a(p) > m
                                                   fi
revisarPendientes(sinCumplir, m, d, cP)
                                                =
     if\neg(\emptyset?(cP)) then
         if sePuedeEjecutar(m, d, dameUno(cP)) then
            if esDeVenta(dameUno(cP)) then
               revisarPendientes(\emptyset, m, d + cantidad(dameUno (cP)), sinUno(cP) \cup sinCumplir)
            else
               revisarPendientes(sinCumplir, m, d - cantindad(dameUno(cP)), sinUno(cP))
         else
            revisarPendientes(Ag (dameUno(cP),sincumplir), m, d, sinUno(cP))
         fi
      else
```

```
\sin Cumplir
      fi
sumarCumplidas(nC, cViejo, cNuevo)
      if \emptyset?(cViejo) then
      else
         if nC==nombreDue\~no(dameUno(cViejo)) \land dameUno(cViejo) \notin cNuevo then
            if esDeVenta(dameUno(cViejo)) then
               sumarCumplidas(nC, sinUno(cViejo), cNuevo) - cantidad(dameUno(cViejo))
            else
               cantidad(dameUno(cViejo)) + sumarCumplidas(nC, sinUno(cViejo), cNuevo)
            fi
         {f else}
            sumarCumplidas(nC, sinUno(cViejo), cNuevo)
         fi
      fi
yaExiste(p, cP)
                                                \equiv
      if Ø?(cP) then
         false
      else
         if nombreDue\~no(p) == nombreDue\~no(dameUno(cP)) AND (esDeVenta?(p) == esDeVenta?(p)
      ta?(dameUno(c)))then
            true
         \mathbf{else}
            yaExiste(p, sinUno(cP))
         fi
      fi
```

Fin TAD