

RELACIÓN TEMA 2 - ABSTRACCIÓN

1. Definir el T.D.A Servidor de red. Un servidor es un punto de red que se encuentra identificado por una dirección *ip*. Una dirección *ip* viene definida por cuatro dígitos que pueden tener valores que van desde 0 a 255. Se pide:
 - Dar la especificación del tipo Servidor. Además establecer las operaciones que manejan al T.D.A
 - Definir al menos dos *tipo rep*
 - Escoger uno de los *tipo rep* y para este establecer la función de abstracción e invariante de representación

·Especificación: representa a un punto de red identificado por una dirección *ip*, la cual viene definida por 4 dígitos. Cada dígito puede tener valores entre 0-255.

·Operaciones:

- Constructores: por defecto, por parámetros, copia.
- Consulta: GetIp
- Operaciones ==, !=
- Operaciones de E/S

·Tipo Rep:

```
class Servidor{  
  
    int ip[4];
```

```
class Servidor{  
  
    VD<int> ip;
```

·Función Abstracción: *r* es un objeto de tipo Servidor

$$f_a(r) = r.ip[0], r.ip[1], r.ip[2], r.ip[3]$$

·Invariante de la representación:

$$r.ip[i] \in [0...255] \quad \forall i = 0..3$$

2. Definir el T.D.A Subred. Este T.D.A es una colección de servidores (s_1, s_2, \dots, s_n), según se ha definido el T.D.A Servidor en el ejercicio anterior. En el T.D.A. Subred también se almacena si dos servidores están conectados (existe un enlace directo) entre ellos. Se pide:

- Dar la especificación del tipo Subred. Además establecer las operaciones que manejan a T.D.A
- Definir al menos dos **tipo rep**
- Escoger uno de los **tipo rep** y para este establecer la función de abstracción e invariante de representación

·Especificación: representa una colección de objetos de tipo Servidor (s_1, s_2, \dots, s_n). Además almacena si 2 servidores están conectados, es decir, si entre ellos existe un enlace directo. No puede haber servidores repetidos.

·Operaciones:

- Constructores: por defecto, por parámetros, copia.
- Consulta: GetServidor
- Modificador: set
- Destructor
- Operaciones de E/S
- bool Conectados(Servidor s1, Servidor s2) const: devuelve true si 2 servidores están conectados directamente y false en caso contrario.

·Tipo Rep:

class Subred{	class Servidor{
VD <Servidor> subred;	Servidor *subred;
VD<VD<int>> conectados;	int n_elem;
	VD<VD<int>> conectado

·Función Abstracción: r es un objeto de tipo subred

$f_a(r) = \{r.subred[0], r.subred[1], \dots, r.subred[r.size()-1], r.conectados[0], r.conectados[1], \dots, r.conectados[r.size()-2]\}$

·Invariante de la representación:

$\forall i, j \ r.subred[i] \neq r.subred[j] \ i \neq j$
 $\forall i, j \ i < j \ r.conectados[i] < r.conectados[j]$

3 . Definir el T.D.A Punto Geográfico. Un punto Geografico se define por una latitud y longitud. La latitud es la distancia en grados desde la línea del Ecuador a los Polos. Su rango va desde -90° a 90°. La longitud es la distancia desde el meridiano 0 al punto donde estamos. El rango de valores que adopta va desde -180° a 180°. Se pide:

- Dar la especificación del tipo Punto Geografico. Ademas establecer las operaciones que manejan a T.D.A.
- Definir al menos dos **tipo rep**
- Escoger uno de los **tipo rep** y para este establecer la función de abstracción e invariante de representación

·Especificación: Un objeto de tipo PuntoGeografico esta formado por una latitud y una longitud. La latitud es la distancia en grados desde la línea del Ecuador a los Polos, comprendida entre [-90°,90°]. La longitud es la distancia desde el meridiano 0 al punto desde donde estamos. Su rango de valores esta entre [-180°, 180°].

·Operaciones:

- Constructores: por defecto, por parámetros, copia.
- Consulta: GetLatitud, GetLongitud
- Modificador: SetLatitud, SetLongitud
- Destructor
- Operadores ==, !=
- int Distancia(PuntoGeografico p1, PuntoGeografico p2);

·Tipo Rep.:

<pre>class PuntoGeografico{ int longitud; int latitud;</pre>	<pre>class PuntoGeografico{ int v[2];</pre>
--	---

·Función Abstracción: r es un objeto de tipo PuntoGeografico

$$f_a(r) = \{r.latitud, r.longitud\}$$

·Invariante de la representación:

$$\begin{aligned} -90 &\leq r.latitud \leq 90 \\ -180 &\leq r.longitud \leq 180 \end{aligned}$$

4 . Definir el T.D.A Ruta. Una ruta es una swecuencia de puntos geográficos (ver ejercicio anterior) Se pide:

- Dar la especificación del tipo Ruta. Ademas establecer las operaciones que manejan a T.D.A.
- Definir al menos dos **tipo rep**
- Escoger uno de los **tipo rep** y para este establecer la función de abstracción e invariante de representación

·Especificación: Un objeto de tipo Ruta es una secuencia de puntos geográficos (de tipo PuntoGeografico, ejercicio anterior). Será de la forma $\{p_1, p_2, \dots, p_n\}$ pudiendo hacer puntos geográficos repetidos.

·Operaciones:

- Constructores: por defecto, por parámetros, copia.
- Consulta: GetRuta()
- Modificador: SetRuta(PuntoGeografico &nuevo);
- Destructor
- Operadores ==, !=
- Ruta RutaEficiente(Ruta r1, Ruta r2);

·Tipo Rep.:

class Ruta{

VD<PuntoGeografico> ruta

class PuntoGeografico{

PuntoGeografico *ruta;
int nrutas;
int rutasreservadas;

·Función Abstracción: r es un objeto de tipo Ruta

$f_a(r) = \{r.ruta[0], r.ruta[1], \dots, r.ruta[r.size()-1]\}$

·Invariante de la representación:

$r.size() > 1$

5. Dar una especificación para la función que permite derivar un polinomio. Suponiendo que tenemos el TDA Polinomio, la cabecera de la función derivada sería así:

```
void Derivar(const Polinomio & p_origen, Polinomio & p_derivada);
```

Un polinomio viene definido por un grado y por los coeficientes, el grado es una variable entera que nos dice la potencia más alta de un coeficiente del polinomio, y los coeficientes se almacenan en un vector de float llamado coef

·Función Abstracción: r es un objeto de tipo Ruta

$f_a = \text{rep} \rightarrow p(x)$
 $(\text{coef}, \text{grado}) \rightarrow \text{coef}[0] + \text{coef}[1]*x + \text{coef}[2] *x^2 + \dots + \text{coef}[\text{grado}] * x^{\text{grado}}.$

·Invariante de la representación: Cualquier objeto del tipo rep: {coef, grado} debe verificar:

- grado ≥ 0 .
- coef apunta a un vector con grado+1 componentes.
- coef[grado] $\neq 0$ ó grado $= 0$.