

ENTREGA 2 - ED

Manuel Fernández La-Chica

Ejercicio1 - T.D.A Servidor de Red

Especificación

Representamos un servidor a partir de una IP, la IP está compuesta de 4 enteros positivos con valores entre 0 y 255.

Operaciones

- Constructor por defecto y de copia.
- Destructor
- Método get de la IP
- Método set de la IP
- Lectura y Escritura
- Operadores ==, !=

Tipo rep

```
class Servidor{
    private:
        struct IP{
            int valor1, valor2, valor3, valor4;
        };
        IP identificadorServidor;
```

Función abstracción

```
fA: {identificadorServidor} ->
"identificadorServidor.valor1"."identificadorServidor.valor2"."identificadorServidor.valor3"."identifi
cadorServidor.valor4"
```

Invariante de representación

$0 \leq (\text{valor1}, \text{valor2}, \text{valor3}, \text{valor4}) \leq 255$

Ejercicio2 - T.D.A SubRed

Especificación

Colección de servidores con posible conexión entre ellos.

Operaciones

- Constructor por defecto y de copia
- Destructor
- Consultar identificador IP
- Modificar identificador IP(añadir)
- Lectura y escritura
- Operadores ==, !=

Tipo rep

```
class Subred{
    private:
        map<Servidor, pair<bool,Servidor>> subredServidores;
```

Función abstracción

```
fA: Subred Sub
sub.size() >= 1
```

Invariante de representación

Las IP de los servidores van desde 0 a 255 cada uno de los 4 valores.
subredServidores[numeroServidor].second.first == false -> no existe enlace directo.
Numero de servidores >=1

— — —

```
class Subred {
    private:

        struct conexionDirecta {
            Servidor servidor;

            Bool conexDirect;

            Servidor conexionDirecta;

        };

        list<conexionDirecta> subredServidores;
```

Ejercicio3 - T.D.A Punto Geográfico

Especificación

Representación de puntos geográficos a partir de su latitud (con valores de entre -90 y 90) y longitud (con valores entre -180 y 180)

Operaciones

- Constructor por defecto y de copia
- Destructor
- Consulta de un punto geográfico, latitud y longitud
- Modificación de un punto geográfico, latitud y longitud.
- Lectura y escritura

Tipo rep

```
class PuntoGeografico{
    private:
        int latitud, longitud;
}
```

Función abstracción

fA: {latitud,longitud}-> (latitud,longitud)

Invariante de representación

-90 <= latitud <= 90
-180 <= longitud <= 180

```
class PuntoGeografico{
    private:
        pait<int,int> puntoGeografico;
}
```

Ejercicio4 - T.D.A Ruta

Especificación

Secuencia de puntos geográficos

Operaciones

- Constructor por defecto y de copia
- Destructor
- Consulta de los puntos geográficos
- Modificar o añadir un punto geográfico
- Lectura y escritura
- Operadores <,==,! =

Tipo Rep

```
class Ruta{
    private:
        list<PuntoGeografico> ruta;
}
```

Función abstracción

fA: Ruta r
r -> r.size() = n

Invariante de representación

Tenemos al menos un elemento
 $R[i] < R[i+1]$

```
class Ruta{
    private:
        vector<PuntoGeografico> ruta[TAM]
        static const int TAM = 1000;
}
```

Ejercicio5 - T.D.A Polinomio

Especificación

Sucesión de números reales que representan a polinomios con coeficientes reales del tipo $a_n x(n) + a_{n-1} x(n-1) + \dots + a_1 x + a_0$

Operaciones

- Constructor por defecto y de copia
- Destructor
- Consultar coeficiente y grado del polinomio
- Modificar coeficiente y grado del polinomio
- Lectura y Escritura

Tipo Rep

```
class Polinomio{
    private:
        float *coeficiente;
        int grado;
        int maximoGrado;
}
```

Función abstracción

fA: $p \rightarrow p.coef[0] + p.coef[1]x^1 + p.coef[2]x^2 + \dots + p.coef[n]x^n$

Invariante de representación

$p.coeficiente[p.grado] \neq 0$
 $p.coeficiente[i] = 0$ Para todo i perteneciente a $[p.grado+1, p.maximoGrado]$

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

Especificación

Almacenamos la derivada del polinomio p_origen en $p_derivada$.

La representación del polinomio viene dado por un puntero que va almacenando los coeficientes, empezando desde el primero con mayor grado hasta llegar al término independiente:

1	0	3	0	4
---	---	---	---	---

Para este ejemplo tendríamos:

$$4x^4 + 3x^2 + 1$$