

## Práctica 6: Conjuntos en Turbo Pascal

### Estructura de Datos y de la Información

Crearemos la unidad Conjuntos que defina el TAD Conjunto dada su conocida especificación algebraica. Realizaremos tres implementaciones: la primera, una implementación estática con un vector de booleanos, la segunda una implementación estática que almacene los elementos en el vector y la tercera será la versión dinámica utilizando el TAD Listas. Utilizaremos una unidad ElemTAD que defina el tipo Entero creada en prácticas anteriores para parametrizar la estructura con el TipoElemento. A continuación se resolverá con ayuda de este TAD y a través del método de la Criba de Eratóstenes el problema de encontrar números primos menores que un entero dado N (ver ejemplo al final del enunciado).

#### ESPECIFICACIÓN Conjunto

##### PARÁMETROS GENÉRICOS

TIPOS TipoElemento

FIN PARÁMETROS

TIPOS TipoConjunto

##### OPERACIONES

###### (\* constructoras generadoras \*)

CrearConjuntoVacio:  $\rightarrow$  TipoConjunto

Poner: TipoConjunto x TipoElemento  $\rightarrow$  TipoConjunto

###### (\* observadoras selectoras \*)

PARCIAL Elegir: TipoConjunto  $\rightarrow$  TipoElemento

###### (\* observadoras no selectoras \*)

EsConjuntoVacio: TipoConjunto  $\rightarrow$  Booleano

Pertenece: TipoConjunto x TipoElemento  $\rightarrow$  Booleano

EsSubconjunto: TipoConjunto x TipoConjunto  $\rightarrow$  Booleano

Cardinal: TipoConjunto  $\rightarrow$  Natural

###### (\* constructoras no generadoras \*)

Quitar: TipoConjunto x TipoElemento  $\rightarrow$  TipoConjunto

Union: TipoConjunto x TipoConjunto  $\rightarrow$  TipoConjunto

Interseccion: TipoConjunto x TipoConjunto  $\rightarrow$  TipoConjunto

Diferencia: TipoConjunto x TipoConjunto  $\rightarrow$  TipoConjunto

##### VARIABLES

conjunto, conjunto2: TipoConjunto;

i, j, e1, e2: TipoElemento;

##### ECUACIONES DE DEFINITUD

DEF(Elegir(Poner(conjunto, i)))

##### ECUACIONES ENTRE GENERADORAS

###### (\* la operación 'Poner' es conmutativa \*)

Poner(Poner(conjunto, e1), e2) = Poner(Poner(conjunto, e2), e1)

###### (\* la operación 'Poner' es idempotente \*)

Poner(Poner(conjunto, e1), e1) = Poner(conjunto, e1)

##### ECUACIONES

###### (\* observadoras selectoras \*)

Elegir(Poner(conjunto, i)) =

<< seleccion aleatoria de un elemento j tal que: Pertenece(conjunto, j) >>

###### (\* observadoras no selectoras \*)

EsConjuntoVacio(CrearConjuntoVacio) = CIERTO

EsConjuntovacio(Poner(conjunto, i)) = FALSO

Pertenece(CrearConjuntoVacio, i) = FALSO  
 Pertenece(Poner(conjunto, i), j) = (i = j) **O** Pertenece(conjunto, j)  
 EsSubconjunto(CrearConjuntoVacio, conjunto2) = CIERTO  
 EsSubconjunto(Poner(conjunto, e1), conjunto2) =  
     Pertenece(conjunto2, e1) **Y** EsSubconjunto(conjunto, conjunto2)

Cardinal(CrearConjuntoVacio) = 0  
 Cardinal(Poner(conjunto, e1)) =  
     **SI** Pertenece(conjunto, e1)  $\rightarrow$  Cardinal(conjunto)  
     | 1 + Cardinal(conjunto)

(\* constructoras no generadoras \*)

Quitar(CrearConjuntoVacio, j) = CrearConjuntoVacio  
 Quitar(Poner(conjunto, i), j) =  
     **SI** i = j  $\rightarrow$  Quitar(conjunto, j)  
     | Poner(Quitar(conjunto, j), i)

Union(CrearConjuntoVacio, conjunto2) = conjunto2  
 Union(Poner(conjunto, i), conjunto2) = Poner(Union(conjunto, conjunto2), i)

Interseccion(CrearConjuntoVacio, conjunto2) = CrearConjuntoVacio  
 Interseccion(Poner(conjunto, i), conjunto2) =  
     **SI** Pertenece(i, conjunto2)  $\rightarrow$  Poner(Interseccion(conjunto, conjunto2), i)  
     | Interseccion(conjunto, conjunto2)

Diferencia(conjunto, CrearConjuntoVacio) = conjunto  
 Diferencia(conjunto, Poner(i, conjunto2)) = Diferencia(Quitar(conjunto, i), conjunto2)

## FIN ESPECIFICACIÓN

### EJEMPLO Criba de Eratóstenes:

Encontrar los números primos menores que N = 16

1. Se generan los enteros de 2 a N: 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
2. Se toma el menor y se eliminan sus múltiplos de la secuencia:  
     2  $\rightarrow$  eliminamos: 4, 6, 8, 10, 12, 14, 16  
     quedando la secuencia: 2, 3, 5, 7, 9, 11, 13, 15
3. Se toma el siguiente menor y se eliminan sus múltiplos de la secuencia:  
     3  $\rightarrow$  eliminamos: 9, 15  
     quedando la secuencia: 2, 3, 5, 7, 11, 13
4. Continuamos con el siguiente menor:  
     5  $\rightarrow$  No podemos eliminar ninguno
5. Continuamos con el siguiente menor: 7 y así sucesivamente
6. Los números primos menores que N = 16 son los que sobreviven a la criba:  
     2, 3, 5, 7, 11, 13

Nota: Haremos uso de las normas de estilo dictadas en clase (cabecera del fichero, interfaz de la unidad con precondiciones, postcondiciones, excepciones, implementaciones con el análisis de complejidad de cada operación, nombres coherentes de variables y operaciones,...)

Plantilla de cabecera del fichero:

```

{ *****
*
*      Módulo:
*      Tipo:  Programa()      Interfaz-Implementación TAD ()      Otros()
*      Autor/es:
*      Fecha de actualización:
*      Descripción:
*
*****
}
```