

Curso 2008/2009 Estructura de Datos y de la Información

I. Informática, I. T. Informática de Gestión y de Sistemas

PRÁCTICA 1

1 El problema

Se trata de automatizar la gestión de un servicio de envío de mensajes para teléfonos móviles.

Para poder acceder al servicio los abonados han de estar bajo la cobertura de una antena, lo cual depende de varios factores, como la localización y el estado del teléfono. Así, un abonado puede estar bajo la cobertura de una antena, pero en otro momento podría estar sin cobertura o bajo la cobertura de otra antena diferente.

El servicio de envío de mensajes necesita conocer esta información, y para ello ha de mantener una lista actualizada de los abonados que están bajo la cobertura de cada antena, almacenando su número de teléfono y el operador de telefonía que tienen contratado. El objetivo de esta práctica es el diseño y la implementación de una nueva estructura de datos que nos permita almacenar y manejar esta información adecuadamente.

Para simplificar el problema, en esta práctica trabajaremos con UNA SOLA ANTENA.

2 Descripción de la tarea

El alumno diseñará un único programa principal (`principal.pas`) que lea, a través de un fichero, las distintas operaciones del servicio de gestión de mensajes—alta/baja de un número de teléfono en una antena, o modificación del operador asignado a un número de teléfono—las procesará y visualizará por pantalla los resultados. El objetivo de este trabajo es practicar la independencia de la implementación en los tipos abstractos de datos (TADs).

El programa funcionará en modo batch (sin interactividad con el usuario) y de manera transparente con cada una de las dos implementaciones del TAD Lista Ordenada (`unit ListaEstatica` y `unit ListaDinamica`), ejecutando las siguientes tareas:

1. Lectura de un fichero de texto `operaciones.txt` que contiene las operaciones a procesar. En la operación consta el código —A (alta), B (baja), M (modificación)— el número de teléfono y el nombre del operador de telefonía. El fichero se leerá línea a línea y se procesará cada operación tal y como sigue:

- a. Se imprimirá un mensaje del tipo:

```
*****
Nueva operación T: Número XXXXXXXXX - Operador YYYYY
*****
donde T es el tipo de operación —alta, baja, modificación—, XXXXXXXXX es el
nº de teléfono e YYYYY es el nombre del operador de telefonía.
```

- b. Si la operación es Alta, se incorporará el nuevo abonado a la lista de teléfonos a los que da servicio la antena. La incorporación del abonado debe realizarse *de manera ordenada* con respecto al número de teléfono. Además se imprimirá el mensaje:

```
Alta de Abonado: Número XXXXXXXXX Operador YYYYY
```

En caso de que ya exista un abonado con ese número, se imprimirá el mensaje:

```
*****  
Error en Alta: El número XXXXXXXXX ya dispone de servicio  
*****
```

- c. Si la operación es Baja, se permite dar de baja a un único abonado o bien a todos los abonados de un operador. En el primer caso se deberá buscar el abonado en la lista de teléfonos asociados a la antena, borrarlo e imprimir el mensaje:

Baja de Abonado: Número XXXXXXXXX Operador YYYYY

Si no existe un abonado con ese número, se imprimirá un mensaje del tipo:

```
*****  
Error en Baja: No existe el número XXXXXXXXX  
*****
```

Para dar de baja todos los teléfonos de una operadora, se deberá buscar en la lista todos los abonados asociados a la operadora y borrarlos de la lista de teléfonos de la antena, imprimiendo los teléfonos eliminados según el mensaje siguiente:

Bajas de Abonados del Operador YYYYY
Abonado: Número XXXXXXXXX
Abonado: Número ZZZZZZZZ
...

En caso de que no existan teléfonos de la operadora, se imprimirá el mensaje:

```
*****  
Error en Baja: No hay teléfonos del Operador YYY  
*****
```

- d. Si la operación es Modificación se deberá buscar el abonado en la lista de teléfonos asociados a la antena, actualizar el operador e imprimir un mensaje del tipo:

Modificación de Abonado: Número XXXXXXXXX. Nuevo Operador YYYYY

En caso de que no exista un abonado con ese número se imprimirá el mensaje:

```
*****  
Error en Modificación: No existe el número XXXXXXXXX  
*****
```

- 2. Visualizar la información de los abonados a la antena de telefonía con el formato:

Abonado Número XXXXXXXXX Operador YYYYY
...
Abonado Número XXXXXXXXX Operador YYYYY

3 TAD Lista Ordenada

Para mantener la lista de los teléfonos abonados al servicio de la antena se utilizará un TAD Lista Ordenada en el que se registran los abonados de manera ordenada con respecto a su número de teléfono, indicando el operador que facilita el servicio de telefonía. Se realizarán dos implementaciones del TAD Lista: una estática con arrays (unit ListaEstatica.pas) y otra dinámica, simplemente enlazada, con punteros (unit ListaDinamica.pas).

3.1 Tipos de datos incluidos en el TAD Lista

- **tLista** Representa una lista ordenada por número de teléfono.
- **tNumeroTelefono** Número de teléfono (string).
- **tOperadorTelefono** Operador de telefonía—movistar, vodafone, orange o yoigo (string).
- **tInfo** Dato de un elemento de la lista, compuesto por los campos *NumeroTelefono* y *OperadorTelefono* que contienen el número de teléfono y el operador de telefonía, respectivamente.
- **tPos** Posición de un elemento de la lista.
- **NULO** Constante utilizada para representar posiciones nulas.

3.2 Operaciones incluidas en el TAD LISTA

Una precondition común para todas las operaciones (salvo ListaVacía) es que la lista debe estar previamente inicializada.

- `ListaVacía (tLista) → tLista`
Crea una lista vacía.
- `esListaVacía (tLista) → Boolean`
Determina si la lista está vacía
- `Primero (tLista) → tPos`
Devuelve la posición del primer elemento de la lista (o NULO si la lista está vacía).
- `Ultimo (tLista) → tPos`
Devuelve la posición del último elemento de la lista (o NULO si la lista está vacía).
- `Siguiente (tLista, tPos) → tPos`
Devuelve la posición en la lista del siguiente elemento a la posición indicada (o NULO si la posición no tiene siguiente).
PreCD: La posición tiene que ser válida.
- `Anterior (tLista, tPos) → tPos`
Devuelve la posición en la lista del anterior elemento a la posición indicada (o NULO si la posición no tiene anterior).
PreCD: La posición tiene que ser válida.
- `InsertarOrdenado(tLista, tNumeroTelefono, tOperadorTelefono) → tLista, Boolean`
Inserta en la lista un nuevo teléfono con número `tNumeroTelefono` y operador `tOperadorTelefono`. La lista quedará ordenada con respecto al número. Devuelve un valor falso si no hay memoria suficiente para realizar la operación.
PreCD: No existe en la lista un teléfono con ese número.
- `Borrar (tLista, tPos) → tLista`
Borra de la lista el elemento que está en la posición indicada.
PreCD: La posición tiene que ser válida.

- `ObtenerDato (tLista, tPos) → tNumeroTelefono, tOperadorTelefono`
Devuelve la información que se encuentra en la posición indicada de la lista.
PreCD: La posición tiene que ser válida.
- `ActualizaDato(tLista, tPos, tOperadorTelefono) → tLista`
Actualiza con el nuevo operador `tOperadorTelefono` el número de teléfono de la posición `tPos` de la lista.
PreCD: La posición tiene que ser válida.
- `Buscar (tLista, tNumeroTelefono) → tPos`
Devuelve la posición del elemento con nombre `tNumeroTelefono` (o NULO si el elemento no existe).

4 Lectura de los ficheros

Para facilitar el desarrollo de la práctica se proporciona un fichero “lectura.pas”. Este fichero contiene un ejemplo de lectura y manejo del fichero de datos mencionado. También se proporciona un ejemplo de prueba del fichero `operaciones.txt`.

5 Normas de realización

- Estructura que deberá de tener cada programa/unit desarrollado
 - Encabezamiento del programa/unit. Constará la siguiente información entre comentarios


```
TÍTULO: Prácticas de EDI
SUBTÍTULO: Practica 1
AUTOR 1: _____ LOGIN 1: _____
AUTOR 2: _____ LOGIN 2: _____
GRUPO: E.I / E.T.I.X. / E.T.I.S
FECHA: __/__/____
```
 - Cuerpo del programa/unit
 - El código irá comentado. Los comentarios han de ser concisos pero explicativos.
 - Después de la cabecera de cada procedimiento o función se incluirá lo siguiente: objetivo, entradas, salidas, precondiciones (condiciones que han de cumplir las entradas para el correcto funcionamiento de la subrutina) y poscondiciones (otras consecuencias de la ejecución de la subrutina que no quedan reflejadas en la descripción del objetivo o de las salidas).
- Criterios de valoración de la práctica:
 - *Eficacia*: que se cumplan las especificaciones.
 - *Control de Errores*: Control intensivo de todos los errores de ejecución que sea posible detectar.
 - *Claridad*: que el programa se pueda entender con facilidad, que contenga comentarios oportunos, indentación adecuada, nombres de variables significativos, etc.
 - *Modular*: que esté construido con módulos intercambiables y reutilizables.

6 Normas de entrega

- Las prácticas son OBLIGATORIAS y serán realizadas en grupos de DOS PERSONAS.
- La entrega de TODAS las prácticas en las FECHAS indicadas es requisito IMPRESCINDIBLE para aprobar la asignatura en la convocatoria ordinaria de JUNIO. Para las convocatorias de SEPTIEMBRE y DICIEMBRE las prácticas serán las mismas pero se fijarán otras fechas de entrega.
- Las prácticas entregadas tendrán que ser compilables en FreePascal en los ordenadores del laboratorio de docencia asignado.
- Forma de entrega: Las prácticas quedarán depositadas en la red según el procedimiento del CECAFI. No se admitirán discos ni papel. El proceso para depositarlas será el siguiente:
 1. Conectarse a las máquinas **xurxo** o **limia**.
 2. Situarse en el directorio /PRACTICAS/XXXX/EDI/P1 donde XXXX es EI, ETIX o ETIS
 3. Situarse en el directorio que coincida con el login del usuario y copiar allí la práctica (sólo los ficheros fuentes .pas). **IMPORTANTE:** El nombre del fichero fuente que contenga el programa principal deberá ser `principal.pas`.
 4. Pasada la fecha de entrega no se permitirá el acceso a estos directorios.

7 Evaluación

- **Fecha límite de entrega: Viernes 8 de Mayo de 2009.**
- **Fecha de los Controles de Evaluación** (50% de la valoración de la práctica, véase guía docente de la asignatura en <http://campusvirtual.udc.es/moodle>):
 - Semana del **20 de Abril:** Implementación y prueba de unit ListaEstatica.
 - Semana del **27 de Abril:** Implementación y prueba de unit ListaDinámica.
- Las evaluaciones se realizarán en los laboratorios de prácticas en los horarios correspondientes a cada grupo.