

# LA PROGRAMACIÓN BASADA EN OBERON

D. Enrique Hernández Hernández, D. Jose Luis Vicén Cruz, D. Eduardo Falces Larraga  
Grupo de Programación de la Escuela Universitaria Politécnica de La Almunia (Zaragoza)  
C\ Mayor s/n E-Mail: eduardo@eupla.unizar.es

*RESUMEN Los lenguajes que se usan tradicionalmente para un primer curso de programación se quedan cortos en el enfoque de una programación moderna. No nos parece conveniente utilizar lenguajes como C y Ada. Oberon/F es una alternativa seria que se ha realizado en la Escuela Universitaria Politécnica de La Almunia de Doña Godina (Zaragoza).*

## 1. INTRODUCCIÓN.

El objetivo fundamental de la asignatura de Programación es enseñar a programar. Lo ideal es que se utilice un lenguaje que debe reunir, a ser posible, las siguientes características:

- Fuertemente tipado.
- Legible, claro y sencillo de mantener.
- Estructurado.
- Modular.
- Extensible.
- Orientado a objetos.
- Fuertemente reutilizable.
- Sobre S.O. Windows.

Precisamente estas son algunas cualidades de Oberón, razón por la cual se tomó como lenguaje de partida para la enseñanza de la programación en Primer Curso. En numerosas Universidades de todo el mundo se utiliza este lenguaje en varias asignaturas de los currícula de Informática.

## 2. ORGANIZACIÓN DE LA ASIGNATURA.

La asignatura se imparte a lo largo de cinco horas semanales, en las que tres son de algorítmica y dos de prácticas con codificación en Oberón.

En las prácticas de codificación se procura hacer especial hincapié en aquellos puntos fundamentales que se desarrollan en las clases de teoría algorítmica. Se ha diseñado una algorítmica basada en Oberón.

Respecto a las prácticas, se dividen en tres bloques, que debe realizar el alumno y examinarse oralmente. Es preceptivo su aprobado para poder presentarse al examen final de la asignatura.

- El primer bloque trata con tipos no estructurados y hace especial hincapié en el diseño descendente y en las abstracciones desarrollando procedimientos y funciones.
- El segundo bloque trata ya con tipos estructurados (incluyendo tipos procedimiento). Sigue haciendo hincapié en abstracciones mediante procedimientos y funciones y en el análisis descendente. Asimismo debe usar descomposición modular. Las prácticas de este bloque utilizan formas y programación visual.
- En el tercer bloque los ejercicios ya presentan una mayor envergadura. Se debe usar variables dinámicas, extensión de tipos y métodos. Se realiza con programación visual y menús.

### 3. PROGRAMA DE LA ASIGNATURA.

La primera parte del programa trata acerca de la especificación de algoritmos, tipos de datos no estructurados, procedimientos y funciones (incluyendo recursividad) y descomposición modular. Dos módulos llamados *entradas* y *salidas* nos permiten tratar las secuencias de datos. Un tercer módulo denominado *planoXY* nos permite realizar gráficos.

En la segunda parte del programa se estudian los tipos estructurados, los tipos procedimiento sin parámetros y la creación de formas bajo Windows. Se hace especial hincapié en las técnicas de programación descendente y en la descomposición modular (creación de módulos de servicio con parámetros de tipo vector abierto).

La tercera parte trata acerca de variables dinámicas (listas, colas, etc.), la extensión de tipos, los módulos genéricos, las estructuras abstractas de datos y los tipos abstractos de datos. Finalmente se hace una introducción a los objetos que en Oberón se basan en la extensión de tipos (herencia) y en la asociación dinámica de procedimientos.

### 4. RESULTADOS OBTENIDOS.

Los resultados son muy alentadores y nos invitan a seguir en esta línea de trabajo. Los alumnos llegan a dominar la programación visual bajo Windows tras un Primer Curso de Programación. Además se introducen en ella con un lenguaje muy potente en el que quedan claramente de manifiesto los tipos abstractos de datos, la programación orientada al objeto y la reusabilidad del software.

### 5. CARACTERÍSTICAS DEL LENGUAJE OBERON.

El lenguaje Oberón sigue la línea de su antecesor el Módula-2 y la filosofía de trabajo de Smalltalk.

Cabe citar, entre otras características:

- Gran facilidad para generar formas (programación visual) que pueden contener objetos OLE.
- Sencilla creación de menús.
- Puede generar hipertexto.
- Implementa el concepto de secuencias de datos a través de los canales.
- El tipo conjunto es predefinido, limitándose a conjuntos de enteros.
- No hay subrangos.
- Los índices de los vectores tienen que ser enteros.
- Permite el paso como parámetros de vectores sin dimensión.
- Permite crear vectores cuyo tamaño se fija en tiempo de ejecución.
- En el ámbito modular, los distintos objetos pueden ser exportables de lectura/escritura y de sólo lectura. Los objetos que no sean exportables permanecen ocultos a los módulos clientes.
- La liberación de memoria reservada dinámicamente es automática y la realiza el Garbage Collector.
- Cuando se producen errores en tiempo de ejecución se generan traps indicando el estado de todo el sistema.
- Permite código en ensamblador.
- Las operaciones matemáticas están implementadas directamente con instrucciones del coprocesador.
- Soporta objetos OLE.
- La programación orientada a objetos se basa en los siguientes conceptos:
  - Extensión de tipos basada en registros.
  - Hay compatibilidad entre el puntero a un registro base y el puntero a uno extendido.
  - Identificación de tipos en tiempo de ejecución.
  - Tipos procedimiento.
- Hay librerías de componentes software para manejo de formas, ventanas, ficheros.

## 6. REFERENCIAS.

- [1] "The Oberon System, User Guide and Programmer's Manual"  
Martin Reiser  
Addison-Wesley, Wokingham, Inglaterra, 1991, ISBN 0-201-54422-9
- [2] "Programming in Oberon, Steps beyond Pascal and Modula"  
Martin Reiser, Niklaus Wirth  
Addison-Wesley, Wokingham, Inglaterra, 1992, ISBN 0-201-56543-9
- [3] "Insight ETHOS: On Object-Orientation in Operating Systems"  
Clemens Szyperski  
vdf, Zürich, 1992, ISBN 3 7281 1948 2

- [4] "**Object-Oriented Programming in Oberon-2**"  
Hanspeter Mössenböck  
Springer Verlag, Nueva York, 1993, ISBN 0-387-56411-X
- [5] "The Oberon/F System"  
Dick Pountain  
Byte Magazine, Vol. 20, No. 1, Enero 1995, pp. 227-228
- [6] "Effective Technology Transfer"  
Prof. Niklaus Wirth  
Institut für Computersysteme, ETH Zürich, Switzerland  
The Oberon Tribune (see below), No 1, Vol 1, 1995
- [7] "Component Software and Oberon: a Perspective on Oberon/F"  
Bert Fitié  
Former account manager at IBM, The Netherlands  
The Oberon Tribune (see below), No 1, Vol 1, 1995
- [8] "Component-Oriented Programming: a Refinement on Object-Oriented Programming"  
Prof. Clemens Szyperski  
School of Computing Science, Queensland University of Technology, Australia  
The Oberon Tribune (see below), No 2, Vol 1, 1995
- [9] "The Oberon/F White Paper"  
Oberon microsystems, Inc.  
The Oberon Tribune (see below), No 2, Vol 1, 1995