

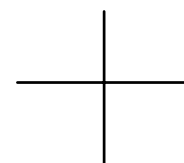
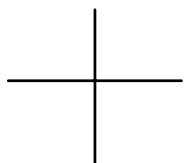
PRÓLOGO

Uno de los ambientes de computadora más usados en nuestros días es Microsoft Windows. Hasta hace poco, los programadores que desarrollaban software para Windows tenían que aprender "Microsoft Windows Software Development Kit (SDK)" y cientos de funciones y mensajes. Debido a esto, escribir software para Windows en C estándar es posible, pero el costo es demasiado alto.

En este libro se trata de aprovechar las características de la programación orientada a objetos para facilitar la programación en ambiente Windows. En particular se aprovecha una de las características más importantes de la orientación a objetos, como lo es la reutilización de código. Esto a través de la biblioteca de clases Object Windows Library (OWL) y de la jerarquía de clases parametrizadas BIDS de Borland C++.

Tanto la programación orientada a objetos como la programación Windows son temas suficientemente amplios como para escribir un libro de cada uno. El objetivo central de este libro es presentar los aspectos de la programación orientada a objetos que son particularmente útiles para el desarrollo de aplicaciones de alta calidad en Windows y con un tiempo de desarrollo relativamente bajo.

Por esta razón se dejan de lado temas de la orientación a objetos que son interesantes, como son la sobrecarga de operadores, los "streams", entre otros. Estos temas se pueden consultar en [Stroustrup-2].



C es un lenguaje de programación considerado de un nivel intermedio, es decir, no es un lenguaje de bajo nivel como ensamblador ni de alto nivel como Pascal. C es similar a un lenguaje de alto nivel en sus estructuras y procedimientos, pero también es de bajo nivel pues tiene operadores que permiten manipular bits en forma individual.

En el primer capítulo de este libro se presentan las características más importantes del lenguaje C que serán útiles en la programación orientada a objetos en C++. Este primer capítulo va dirigido especialmente a personas que ya conocen algún lenguaje de programación, o bien a personas que ya conocen C y desean repasar los aspectos más importantes de este lenguaje que serán útiles en la programación orientada a objetos. Para una presentación completa de C consulte el libro *Programación en C* [Gottfried].

En general el objetivo del libro no es enseñar la programación estructurada, por lo que se recomienda que el lector conozca previamente algún lenguaje de alto nivel, como C o Pascal.

Siendo C uno de los lenguajes más utilizados en el desarrollo de aplicaciones de muy diversos propósitos, es natural pensar que este lenguaje debería ser extendido a un nuevo lenguaje con las características de la orientación a objetos. Es así como Bjarne Stroustrup en la década de 1980 a 1990 diseñó un lenguaje llamado *C con clases*, que posteriormente evolucionó a C++.


C++ fue diseñado de modo que conserva las características más importantes del lenguaje C, como son: la flexibilidad, es decir, C++ es un lenguaje que se puede aplicar a gran cantidad de problemas. La eficiencia, la característica de bajo nivel de C permite que la mayoría de los programas C++ se ejecuten en tiempos bastante cortos. La disponibilidad y la portabilidad, ya que compiladores de C y C++ están disponibles en casi todos los sistemas operativos conocidos.

En el capítulo II se presentan con detalle las características de la orientación a objetos presentes en C++. Este capítulo es prácticamente independiente del compilador de C++ que se utilice.

El capítulo III está dedicado a la programación en ambiente Windows mediante la reutilización de código. Se presentan en este capítulo los aspectos más importantes de la jerarquía de objetos Object-Windows 2.0 de Borland C++, entre otros, cómo crear menús, cajas de diálogo, gráficos.

Con Borland C++ 4.0 o 4.5 se pueden crear programas para Windows95 o para Windows 3.1. Esto dependerá de la forma en que el programa sea compilado. Si se compila a 32 bits el programa se ejecutará adecuadamente en Windows95 o en WindowsNT, mientras que si se compila a 16 bits el programa se ejecutará correctamente en Windows 3.1. En el capítulo 1 se explican las diferentes opciones de compilación que tiene Borland C++.

Los ejemplos presentados a lo largo de todo el texto son programas completos, lo cual permite realmente tener la idea global sobre cómo debe estructurarse un programa. El libro se basa en un ejemplo central el cual aparece por primera vez, en forma muy reducida, en el capítulo 1 y se va ampliando a lo largo de los siguientes capítulos, hasta tener un programa completo. Esto tiene como objetivo introducir la técnica de programación conocida como *programación incremental*, la cual es sumamente útil cuando se pretenden construir programas complejos.

¿Cómo se pueden utilizar estos ejemplos? Todos los ejemplos precedidos del símbolo  están en el disquete asociado al libro. Este disquete puede ser copiado en forma gratuita a través de INTERNET desde el sitio FTP anónimo de la Universidad Nacional localizado en la dirección <ftp://ftp.una.ac.cr> en el archivo cuya ruta es `/pub/Textos/ejemplos.zip` (note que la letra T de la palabra Textos es en mayúscula).

Una vez copiado el disquete, simplemente copie todos los archivos del disquete en un directorio de su disco duro; luego descomprima el

archivo ejemplos.zip usando el comando PKUNZIP ejemplos.zip. Para ejecutar los programas cargue los proyectos extensión *.IDE tal como se explica en el capítulo 1. Si compila los ejemplos con Borland C++ 4.5 cambie, en el proyecto, el directorio de las librerías C:\bc4\lib por C:\bc45\lib y el directorio "include" C:\bc4\include por C:\bc45\include.

Cualquier duda, sugerencia o recomendación sobre este libro puede escribirle al autor por medio de INTERNET al e-mail orodrigu@irazu.una.ac.cr o bien al e-mail orodrigu@cariari.ucr.ac.cr.