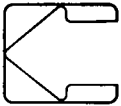


INDICE DE MATERIAS



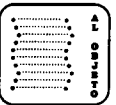
PROLOGO	13
----------------------	----



Capítulo 1. INTRODUCCION	15
1.1. A quién va dirigido este libro	15
1.2. Objetivos de este libro	16
1.3. Cómo leer este libro según sus conocimientos	16



Capítulo 2. UNA VISION GENERAL DE LA OOP	19
2.1. La evolución de los lenguajes de programación	20
2.1.1. Evolución en cuanto a tecnología	20
2.1.2. Evolución en cuanto a conceptualización	23
2.1.3. Evolución en cuanto a enfoque	25
2.2. Qué es la OOP	27
2.3. Breve historia de la OOP	28

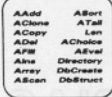


Capítulo 3. PROGRAMACION ORIENTADA AL OBJETO	31
3.1. Conceptos básicos	31
3.1.1. Definición de Clase	31
3.1.2. Definición de Objeto	33
3.1.3. Herencia	38
3.1.4. Datos, mensajes y métodos: Encapsulación	39
3.1.5. Polimorfismo	43



INDICE DE MATERIAS

3.2.	Consideraciones previas a la implementación de las clases	44
3.3.	De la teoría a la realidad	45
3.4.	Limitaciones e inconvenientes de la OOP	51



Capítulo 4. ALGUNAS HERRAMIENTAS NECESARIAS 55

4.1.	Sugerencias preliminares	55
4.1.1.	Nomenclatura sintáctica	55
4.1.2.	Notación Húngara	57
4.1.3.	Tipos de variables	60
4.1.4.	Directivas del compilador	63
4.1.5.	Estilo de programación	65
4.2.	El preprocesador	67
4.2.1.	Directiva #include	68
4.2.2.	Directivas #define y #undef	69
4.2.3.	Directivas #command y #translate	71
4.2.4.	Directivas #ifdef y #ifndef	76
4.2.5.	Observación final	76
4.3.	Arrays multidimensionales	77
4.3.1.	Definición	77
4.3.2.	Creación de un array	78
4.3.3.	Arrays multidimensionales	80
4.3.4.	Funciones asociadas al manejo de arrays	82
4.4.	Bloques de código	90
4.4.1.	Macros: antepasados de los bloques de código	90
4.4.2.	Definición y uso de los bloques de código	93
4.4.3.	Uso y abuso	94
4.4.4.	Anidación de bloques de código	95
4.4.5.	Funciones relacionadas con los bloques de código	96
4.4.6.	Precauciones en el uso de bloques de código	101



Capítulo 5. EL PROGRAMA GENERADOR DE CLASES:		
OBJECTS		105
5.1.	Introducción	105
5.2.	Utilización de Objects	107
5.2.1.	Cómo incluir objects en sus programas	107
5.2.2.	Cómo utilizar las clases creadas por usted	108
5.2.3.	Cómo utilizar el Debugger con Objects	110
5.3.	Sintaxis de Objects	112



Capítulo 6. IMPLEMENTACION DE UNA JERARQUIA DE CLASES	117
6.1. Planteamiento de las clases	117
6.2. Definición de la clase padre	118
6.3. Definición de las clases herederas	121
6.4. Incorporación y uso en un programa	124
6.5. Sugerencias para ampliar y mejorar la clase ventana	128



Capítulo 7. PSEUDO-OBJETOS	131
7.1. Definición	131
7.2. Utilización	135
7.3. El pseudo-objeto MENU	136



Capítulo 8. LAS CLASES PREDEFINIDAS EN CLIPPER 5	143
8.1. Limitaciones	143
8.2. Clase Error	144
8.2.1. Introducción	144
8.2.2. Datos y Métodos	144
8.2.3. Ejemplos	146
8.3. Clase Get	148
8.3.1. Introducción	148
8.3.2. Datos y Métodos	150
8.3.3. Ejemplos	155
8.4. Clase TBColumn	155
8.4.1. Introducción	155
8.4.2. Datos y Métodos	156
8.4.3. Ejemplos	158
8.5. Clase TBrowse	158
8.5.1. Introducción	158
8.5.2. Datos y Métodos	160
8.5.3. Ejemplos	166



Capítulo 9. TODOS A UNA	171
9.1. Introducción	171
9.2. Normativa C.U.A.	172
9.2.1. Ventanas	173
9.2.2. Menús desplegables	174



9.2.3. Cajas de diálogo	175
9.2.4. Sistemas de ayuda	177
9.3. Implementación de cajas de diálogo	178



Apéndice 1. EJEMPLO DE IMPLEMENTACION OOP EN C++	195
1.1. Ejemplo básico	195
1.2. Ejemplo para programadores de C	200
1.3. Código fuente de la librería Objects	237
1.4. CuaWise: Una aproximación a la normativa CUA	247

Apéndice 2. GLOSARIO	271
-----------------------------------	------------

Apéndice 3. FICHEROS DEL DISCO ANEXO	281
---	------------



Bibliografía	285
---------------------------	------------



Relación de marcas nombradas	287
---	------------



Indice alfabético	289
--------------------------------	------------