

Contenido

1	Encapsulamiento a través de clases y objetos	1
1.1	Introducción	2
1.2	Clases y objetos	2
1.2.1	Las clases	2
1.2.2	Miembros de la clase	3
1.2.3	Interfaz y encapsulamiento	5
1.2.4	Estructura de una clase	6
1.2.5	El constructor y el destructor	6
1.2.6	Los métodos	7
1.2.7	Los objetos	8
1.2.8	Instanciar objetos	8
1.2.9	Operadores new, delete y punteros a objetos	9
1.2.10	Sobrecarga de métodos	9
1.3	Encapsulamiento de estructuras lineales	10
1.3.1	Análisis de la clase Pila	10
1.3.2	Templates y generalizaciones	12
1.4	El lenguaje de programación Java	15
1.4.1	El programa principal en Java	17
1.4.2	Templates en C++, generics en Java	18
1.4.3	Los wrappers (envoltorios) de los tipos de datos primitivos	20
1.4.4	Autoboxing	20
1.5	Resumen	21
2	Introducción al lenguaje de programación Java	23
2.1	Introducción	24
2.2	Comencemos a programar	24
2.2.1	El Entorno Integrado de Desarrollo (IDE)	25
2.2.2	Entrada y salida estándar	25
2.2.3	Comentarios en el código fuente	27
2.3	Tipos de datos, operadores y estructuras de control	27
2.3.1	El bit de signo para los tipos de datos enteros	27
2.3.2	El compilador y la máquina virtual (JVM o JRE)	27
2.3.3	Estructuras de decisión	28
2.3.4	Estructuras iterativas	31
2.3.5	El tipo de datos boolean y las expresiones lógicas	32
2.3.6	Las constantes	33
2.3.7	Arrays	34
2.3.8	Matrices	36
2.3.9	Literales de cadenas de caracteres	38
2.3.10	Caracteres especiales	40
2.3.11	Argumentos en línea de comandos	41
2.4	Tratamiento de cadenas de caracteres	42
2.4.1	Acceso a los caracteres de un string	42
2.4.2	Mayúsculas y minúsculas	43
2.4.3	Ocurrencias de caracteres	44

2.4.4	Subcadenas	44
2.4.5	Prefijos y sufijos	45
2.4.6	Posición de un substring dentro de la cadena	45
2.4.7	Conversión entre números y cadenas	46
2.4.8	Representación en diferentes bases numéricas	46
2.4.9	La clase StringTokenizer	47
2.4.10	Comparación de cadenas	48
2.5	Resumen	50
3	Programación orientada a objetos	51
3.1	Introducción	52
3.2	Clases y objetos	52
3.2.1	Los métodos	53
3.2.2	Herencia y sobrescritura de métodos	56
3.2.3	El método toString	56
3.2.4	El método equals	57
3.2.5	Declarar y "crear" objetos	58
3.2.6	El constructor	59
3.2.7	Repaso de lo visto hasta aquí	60
3.2.8	Convenciones de nomenclatura	62
3.2.9	Sobrecarga de métodos	63
3.2.10	Encapsulamiento	66
3.2.11	Visibilidad de los métodos y los atributos	68
3.2.12	Packages (paquetes)	69
3.2.13	Estructura de paquetes y la variable CLASSPATH	70
3.2.14	Las APIs (Application Programming Interface)	71
3.2.15	Representación gráfica UML	71
3.2.16	Importar clases de otros paquetes	72
3.3	Herencia y polimorfismo	73
3.3.1	Polimorfismo	76
3.3.2	Constructores de subclases	78
3.3.3	La referencia super	79
3.3.4	La referencia this	82
3.3.5	Clases abstractas	83
3.3.6	Constructores de clases abstractas	86
3.3.7	Instancias	90
3.3.8	Variables de instancia	91
3.3.9	Variables de la clase	93
3.3.10	El garbage collector (recolector de residuos)	93
3.3.11	El método finalize	94
3.3.12	Constantes	95
3.3.13	Métodos de la clase	95
3.3.14	Clases utilitarias	97
3.3.15	Referencias estáticas	98
3.3.16	Colecciones (primera parte)	99
3.3.17	Clases genéricas	104
3.4	Interfaces	107
3.4.1	Desacoplamiento de clases	109

3.4.2	El patrón de diseño de la factoría de objetos	111
3.4.3	Abstracción a través de interfaces.....	111
3.4.4	La interface Comparable.....	112
3.4.5	Desacoplar aún más	116
3.4.6	La interface Comparator	119
3.5	Colecciones de objetos	119
3.5.1	Cambio de implementación	122
3.5.2	El método Collections.sort	122
3.6	Excepciones	126
3.6.1	Errores lógicos vs. errores físicos.....	129
3.6.2	Excepciones declarativas y no declarativas.....	129
3.6.3	El bloque try-catch-finally.....	131
3.6.4	El método printStackTrace.....	134
3.7	Resumen	134
4	Estructuras de datos lineales	135
4.1	Introducción	136
4.2	Estructuras estáticas	136
4.3	Estructuras dinámicas	137
4.3.1	Operaciones asociadas a las estructuras	138
4.4	Estructuras dinámicas en Java	138
4.4.1	Memoria dinámica y punteros.....	138
4.4.2	Estructura Nodo	139
4.4.3	Lista enlazada.....	139
4.4.4	Pila	144
4.4.5	Cola	145
4.4.5.1	Cola (implementación mejorada).....	146
4.4.6	Clases LinkedList, Stack y Queue.....	147
4.4.7	Tablas de dispersión (Hashtable).....	148
4.4.8	Estructuras de datos combinadas	150
4.4.9	Árboles.....	152
4.4.10	Árbol Binario de Búsqueda (ABB)	153
4.4.11	La clase TreeSet	154
4.5	Resumen	154
5	Compresión de archivos mediante el algoritmo de Huffman	157
5.1	Introducción	158
5.2	El algoritmo de Huffman	159
5.2.1	Paso 1 - Contar la cantidad de ocurrencias de cada carácter.....	160
5.2.2	Paso 2 - Crear una lista enlazada	160
5.2.3	Paso 3 - Convertir la lista enlazada en el árbol Huffman	161
5.2.4	Paso 4 - Asignación de códigos Huffman	163
5.2.5	Paso 5 - Codificación del contenido.....	164
5.2.6	Proceso de decodificación y descompresión	164
5.3	Aplicación práctica.....	164
5.3.1	Compresor de archivos (szip.java).....	164
5.3.2	Descompresor de archivos (szunzip.java).....	165

5.3.3	Estructura del archivo .szcod (árbol Huffman)	165
5.3.4	Estructura del archivo .szdat (archivo comprimido)	166
5.3.5	El setup del ejercicio	166
5.4	Análisis de clases y objetos	167
5.4.1	szip.java	169
5.4.2	szunzip.java	170
5.5	Interfaces e implementaciones	171
5.5.1	ICode.java - Interface de los códigos Huffman	171
5.5.2	ITree.java - Interface del árbol binario o árbol Huffman	171
5.5.3	IList.java - Interface de la lista enlazada	172
5.5.4	ITable.java - Interface de la tabla de ocurrencias	172
5.5.5	IFileInpu.java - Interface del archivo que vamos a comprimir o a restaurar	173
5.5.6	IFileCode.java - Interface del archivo de códigos Huffman	173
5.5.7	IFileCompressed.java - Interface del archivo comprimido	174
5.6	Manejo de archivos en Java	174
5.6.1	Leer un archivo (clase FileInputStream)	175
5.6.2	Bytes sin bit de signo	176
5.6.3	Escribir un archivo (clase FileOutputStream)	176
5.6.4	Buffers de entrada y salida	177
5.7	Clases utilitarias	179
5.7.1	La clase UTree - Recorrer el árbol binario	179
5.7.2	La clase UFile - Leer y escribir bits en un archivo	181
5.7.3	La clase UFactory - Factoría de objetos	183
5.8	Resumen	184
6	Recursividad	185
6.1	Introducción	186
6.2	Conceptos iniciales	186
6.2.1	Funciones recursivas	186
6.2.2	Finalización de la recursión	187
6.2.3	Invocación a funciones recursivas	187
6.2.4	Funcionamiento de la pila de llamadas (stack)	188
6.2.5	Funciones recursivas vs. funciones iterativas	191
6.3	Otros ejemplos de recursividad	192
6.4	Permutar los caracteres de una cadena	193
6.5	Búsqueda binaria	197
6.6	Ordenamiento por selección	199
6.7	La función de Fibonacci	201
6.7.1	Optimización del algoritmo recursivo de Fibonacci	207
6.8	Resumen	208
7	Árboles	209
7.1	Introducción	210
7.1.1	Tipos de árbol	210
7.1.2	Implementación de la estructura de datos	210
7.2	Árbol binario	211

7.2.1	Niveles de un árbol binario.....	212
7.2.2	Recorrer los nodos de un árbol binario.....	212
7.2.3	Recorrido en amplitud o “por niveles”.....	212
7.2.4	Recorridos en profundidad (preorden, postorden e inorden)	212
7.2.5	Implementación iterativa del recorrido en preorden.....	214
7.2.6	Implementación iterativa del recorrido en postorden.....	215
7.4	Árbol Binario de Búsqueda	217
7.4.1	Crear un Árbol Binario de Búsqueda (ABB).....	218
7.4.2	Encapsulamiento de la lógica y la estructura de datos (clase Abb).....	220
7.4.3	Agregar un elemento al ABB (método agregar).....	221
7.4.4	Ordenar valores mediante un ABB (recorrido inOrden).....	223
7.4.5	Búsqueda de un elemento sobre un ABB (método buscar).....	224
7.4.6	Eliminar un elemento del ABB (método eliminar).....	225
7.5	Árbol n-ario	227
7.5.1	Nodo del árbol n-ario	227
7.5.2	Recorridos sobre un árbol n-ario	228
7.5.3	Permutar los caracteres de una cadena.....	229
7.5.4	Implementación de un “AutoSuggest”	229
7.6	Resumen	232
8	Complejidad algorítmica	233
8.1	Introducción	234
8.2	Conceptos iniciales	234
8.2.1	Análisis del algoritmo de la búsqueda secuencial.....	234
8.3	Notación O grande (cota superior asintótica)	235
8.3.1	Análisis del algoritmo de la búsqueda binaria.....	236
8.3.2	Análisis del algoritmo de ordenamiento por burbujeo	238
8.4	Cota inferior (Ω) y cota ajustada asintótica (Θ)	239
8.5	Resumen	239
9	Algoritmos de ordenamiento	241
9.1	Introducción	242
9.2	Bubble sort (ordenamiento por burbujeo)	243
9.2.1	Bubble sort optimizado	246
9.3	Selection sort (ordenamiento por selección)	246
9.4	Insertion sort (ordenamiento por inserción)	247
9.5	Quicksort (ordenamiento rápido)	248
9.5.1	Implementación utilizando arrays auxiliares	248
9.5.2	Implementación sin arrays auxiliares.....	249
9.6	Heapsort (ordenamiento por montículos)	250
9.6.1	Árbol binario semicompleto.....	250
9.6.2	Representar un árbol binario semicompleto en un array	251
9.6.3	Montículo (heap).....	252
9.6.4	Transformar un árbol binario semicompleto en un montículo.....	252
9.6.5	El algoritmo de ordenamiento por montículos	253
9.7	Shellsort (ordenamiento Shell)	257

9.8	Binsort (ordenamiento por cajas).....	258
9.9	Radix sort (ordenamiento de raíz).....	259
9.9.1	Ordenar cadenas de caracteres con radix sort	260
9.10	Resumen	261
10	Estrategia algorítmica	263
10.1	Introducción.....	264
10.2	Divide y conquista.....	264
10.3	Greedy, algoritmos voraces.....	264
10.3.1	Seleccionar billetes de diferentes denominaciones	265
10.3.2	Problema de la mochila.....	267
10.4	Programación dinámica.....	269
10.4.1	Problema de los billetes según la programación dinámica.....	270
10.5	Resumen	277
11	Algoritmos sobre grafos	279
11.1	Introducción.....	280
11.2	Definición de grafo	281
11.2.1	Grafos conexos y no conexos	281
11.2.2	Grafos dirigidos y no dirigidos.....	282
11.2.3	El camino entre dos vértices	282
11.2.4	Ciclos.....	282
11.2.5	Árboles.....	283
11.2.6	Grafos ponderados	283
11.2.7	Vértices adyacentes y matriz de adyacencias	284
11.2.8	La clase Grafo	286
11.2.10	Determinar la existencia de un ciclo.....	289
11.2.11	Los nodos "vecinos".....	293
11.3	El problema de los caminos mínimos.....	296
11.3.1	Algoritmo de Dijkstra	296
11.3.2	Dijkstra, enfoque greedy.....	297
11.3.3	Dijkstra, implementación por programación dinámica.....	301
11.4	Árbol de cubrimiento mínimo (MST)	305
11.4.1	El algoritmo de Prim	306
11.4.2	Algoritmo de Kruskal	310
11.5	Resumen	314
	Bibliografía.....	315