

1

Introducción

En este capítulo se realiza una introducción del libro y se presenta el contenido de cada uno de los capítulos. Además, también se explican las distintas formas de utilizar este libro.

1.1. Introducción

El problema fundamental en el diseño e instrumentación de grandes proyectos o aplicaciones informáticas es reducir su complejidad. Los atributos que permiten obtener esta característica deseable son los siguientes: legibilidad, corrección, eficiencia, facilidad de mantenimiento y reutilización.

A través de la abstracción de datos se pueden obtener estos objetivos. Además, la abstracción de datos permitirá el cambio de representación de los datos sin afectar por ello a los programas que los utilicen. Es además particularmente importante porque simplifica la estructura de los programas que la usan debido a que presenta un interfaz de alto nivel.

El primer mecanismo de abstracción en programación fue el concepto de procedimiento. Un procedimiento hace cierta tarea o función, que será realizada en el programa principal a través de una llamada a dicho procedimiento. Para utilizarlo, el programador sólo se preocupa de qué hace y no cómo está instrumentado. Con el paso de los años el énfasis en el diseño de programas fue cambiando del diseño basado en procedimientos a la organización de los datos. Esto permitió la aparición durante la década de los ochenta de la programación orientada a objetos. En esta forma de programar, son los datos los que constituyen la jerarquía básica. Un dato puede estar ligado a otro a través de una relación de parentesco o de cualquier otro tipo, lo que da lugar a la aparición de una red semejante a la que suele formarse en las aplicaciones procedimentales con los programas.

Los tipos de datos se definen mediante entidades abstractas (modelo matemático) y operaciones que se encapsulan de forma tal que la única vía para acceder o para modificar las entidades es a través de las operaciones abstractas. A este encapsulamiento se le llama Tipo Abstracto de Dato (TAD), haciendo referencia con el calificativo 'abstracto', a la idea de que, cuando se especifica, lo único fundamental son las cualidades, es decir, el qué se se puede hacer, y no cómo se hace. Informalmente, podríamos definir un tipo abstracto de datos como un ente que tiene la propiedad de instanciabilidad, más un conjunto de operaciones, las únicas aceptadas para acceder a la representación (privacidad). La especificación sirve para expresar las cualidades de un tipo abstracto de datos, es decir, qué debe producir, y no cómo lo hace. Es una descripción breve y precisa de

los datos y funciones de un módulo. La especificación nos va a permitir, por tanto separar la realización del uso. Es el enlace entre el usuario y el implementador.

La especificación formal de los Tipos Abstractos de Datos la vamos a plantear desde el punto de vista algebraico. En esta aproximación, la especificación de un TAD consta de dos partes: *sintaxis*, en la que se determinan los conjuntos de objetos que se van a utilizar y el perfil de las operaciones, y la *semántica*, es decir, un conjunto suficientemente completo de axiomas que especifique cómo interactúan las funciones. Además, es necesario un vocabulario que nos servirá para declarar los objetos que aparecen en la definición de la sintaxis y la semántica. Se denomina algebraica a esta aproximación por fundamentarse en la teoría de álgebras heterogéneas. Una vez obtenida la especificación del TAD, y la herramienta para poder instrumentarla, el programador decide qué estructura de datos utilizará para representar el tipo, y qué algoritmos utilizará para realizar las operaciones determinadas por la especificación.

1.2. Programación y Estructuras de Datos

Desde el curso 2011-2012, se imparte la asignatura “Programación y Estructuras de Datos” en la Universidad de Alicante. Esta asignatura obligatoria pertenece al plan de estudios de la titulación de Grado en Ingeniería Informática de la Universidad de Alicante y es impartida por el Departamento de Lenguajes y Sistemas Informáticos, adscrito a la Escuela Politécnica Superior.

La asignatura “Programación y Estructuras de Datos” equivale a la asignatura “Programación y Estructuras de Datos” del antiguo plan de estudios de 2001 de las titulaciones de Informática de la Universidad de Alicante.

Este libro pretende ser un material de apoyo para los alumnos de “Programación y Estructuras de Datos” para realizar preguntas tipo test y ejercicios del temario de la asignatura que servirán de preparación a las pruebas de evaluación de conocimientos a las que se tendrán que enfrentar al final de la asignatura.

1.3. Descripción de la asignatura

La asignatura “Programación y Estructuras de Datos” es obligatoria y de duración semestral, con una carga docente de 6 créditos ECTS (*European Credit Transfer System*), repartidos entre 3 créditos de teoría y 3 de prácticas. Se imparte en el segundo curso del Grado en Ingeniería Informática.

La asignatura “Programación y Estructuras de Datos” tiene una dependencia con la asignatura “Programación 2” y se debe coordinar con las asignaturas “Programación 3” y “Análisis y Diseño de Algoritmos”.

Los objetivos generales de la asignatura “Programación y Estructuras de Datos” son los siguientes:

- Que el alumno conozca:
 - Los mecanismos de abstracción y su importancia para la resolución de problemas.
 - Los tipos de datos más usuales en programación, sus representaciones más comunes y su utilidad.
- Que el alumno comprenda:
 - La necesidad de separación entre los niveles de especificación, implementación y uso.
 - La necesidad de adaptar la representación interna del tipo abstracto de datos (TAD) a los requerimientos de la aplicación a resolver.
- Que el alumno sea capaz de:
 - Distinguir entre las representaciones alternativas de una abstracción de datos y razonar sobre la solución escogida en cuanto a coste computacional se refiere.
 - Aplicar los tipos abstractos de datos básicos aprendidos a problemas prácticos reales.
 - Organizar un determinado volumen de datos de la forma más racional posible en función de los requerimientos del problema a resolver.
 - Crear nuevos tipos abstractos de datos, o que pueda elegir otra representación de los mismos para adaptarlos a una aplicación determinada.

- Evaluar cada representación de un tipo abstracto de datos en función de su consumo de recursos (eficiencia espacial y eficiencia temporal).
- Desarrollar el hábito de trabajar en equipos de programación.

1.4. Temario de la asignatura

La asignatura está dividida en cinco módulos principales, que a su vez se dividen en 17 unidades que tratan aspectos generales de la programación y las estructuras de datos:

Módulo I: Introducción a los tipos abstractos de datos. Los tipos lineales

Unidad 0: Presentación y objetivos de la asignatura

Unidad 1: Introducción a los tipos abstractos de datos

- 1.1 Introducción.
- 1.2 Especificación algebraica de los tipos abstractos de datos.
- 1.3 Implementación de tipos abstractos de datos.

Unidad 2: Vectores

- 2.1 Definición y especificación.
- 2.2 Representación secuencial.
- 2.3 Aplicaciones de vectores: matrices y polinomios.

Unidad 3: Pilas

- 3.1 Definición y especificación.
- 3.2 Representación secuencial y enlazada. Análisis de la eficiencia de las representaciones.
- 3.3 Aplicaciones de pilas: evaluación de expresiones y gestión de la llamada a procedimientos recursivos.

Unidad 4: Colas

- 4.1 Definición y especificación.
- 4.2 Representación secuencial y enlazada. Análisis de la eficiencia de las representaciones.

- 4.3 Aplicaciones de colas: simulación y gestión de colas de espera.

Unidad 5: Listas

- 5.1 Listas de acceso por posición: definición y especificación.
- 5.2 Representación secuencial.
- 5.3 Representación enlazada: listas enlazadas circulares, listas doblemente enlazadas y listas multienlazadas.
- 5.4 Análisis de la eficiencia de las representaciones.
- 5.5 Aplicaciones de listas: matrices dispersas y representación de polinomios.

Módulo II: La eficiencia de los algoritmos

Unidad 6: Definiciones generales de nociones de complejidad algorítmica

- 6.1 Noción de complejidad.
- 6.2 Cotas de complejidad.
- 6.3 Notación asintótica.
- 6.4 Obtención de cotas de complejidad. Ejemplos (algoritmos de búsqueda/ordenación en un vector, etc.).

Módulo III: Tipo árbol

Unidad 7: Definiciones generales de árboles

Unidad 8: Árboles binarios

- 8.1 Definición y especificación.
- 8.2 Representación secuencial y enlazada. Análisis de la eficiencia de las representaciones.
- 8.3 Recorridos de un árbol binario: preorden, inorden, postorden y niveles.
- 8.4 Operaciones complementarias: profundidad, número de elementos y altura de un nodo.
- 8.5 Aplicaciones de árboles binarios: árboles para la evaluación de expresiones aritméticas y códigos de Huffman.

Unidad 9: Árboles n-arios y árboles generales

- 9.1** Definición y especificación.
- 9.2** Representación secuencial y enlazada. Análisis de la eficiencia de las representaciones.
- 9.3** Recorridos.
- 9.4** Aplicaciones: árboles genealógicos.

Unidad 10: Árboles binarios de búsqueda

- 10.1** Definición y especificación.
- 10.2** Operaciones básicas: inserción, búsqueda y eliminación. Análisis de la eficiencia de las operaciones.
- 10.3** Árbol binario de búsqueda equilibrado en altura (AVL). Algoritmos de inserción y eliminación. Análisis de su eficiencia.
- 10.4** Aplicaciones.

Unidad 11: Árboles n-arios de búsqueda

- 11.1** Definición y especificación.
- 11.2** Árboles n-arios de búsqueda: árboles 2-3 y árboles 2-3-4.
- 11.3** Aplicación: utilización en la organización interna de bases de datos.

Módulo IV: Tipo conjunto*Unidad 12: El conjunto*

- 12.1** Definición y especificación.
- 12.2** Representación: secuencial y enlazada basada en listas. Análisis de la eficiencia de las representaciones.
- 12.3** Clasificación de los conjuntos según las operaciones.

Unidad 13: El diccionario

- 13.1** Definición y especificación.

- 13.2** Representación mediante tablas de dispersión (hashing). Dispersión cerrada y abierta; análisis comparativo entre ambas. Operaciones del diccionario con métodos de dispersión.
- 13.3** Otras representaciones: árbol 2-3 y árbol 2-3-4.

Unidad 14: Colas de prioridad

- 14.1** Definición y especificación.
- 14.2** Definición de árbol parcialmente ordenado. Representaciones: listas ordenadas y árboles parcialmente ordenados.
- 14.3** El montículo (heap) para representar árboles parcialmente ordenados. Tipos de montículos.
- 14.4** Algoritmos de inserción y borrado del elemento de prioridad mínima (máxima). Análisis de su eficiencia.

Unidad 15: Otros tipos de conjuntos

- 15.1** El trie. Definición e implementación. Algoritmos de búsqueda, inserción y borrado. Análisis de su eficiencia.

Módulo V: Tipo grafo

Unidad 16: Grafos

- 16.1** Definición de grafo y terminología. Especificación.
- 16.2** Algunas representaciones: matriz de adyacencia y lista de adyacencia. Análisis de la eficiencia de las representaciones.
- 16.3** Grafos dirigidos. Grafos acíclicos dirigidos. Grafos no dirigidos.
- 16.4** Recorridos en un grafo: profundidad y anchura. Comparación de los métodos.

Unidad 17: Aplicaciones con grafos

- 17.1** Clasificación de arcos. Bosque extendido en profundidad y en anchura de un grafo.
- 17.2** Coloreado de grafos y ordenación topológica.

1.5. Estructura del libro

Los capítulos principales del libro (del 2 al 7) se corresponden con los módulos del temario de la asignatura que se ha presentado en la sección anterior. Además, hay un capítulo adicional dedicado al lenguaje de programación C++, que se emplea en la realización de las prácticas de la asignatura. En estos seis capítulos, las preguntas están organizadas según el tema que tratan.

En el Capítulo 2 (**Introducción a los tipos abstractos de datos**) se incluyen preguntas y ejercicios sobre los tipos abstractos de datos y la complejidad.

En el Capítulo 3 (**El lenguaje C++**) se incluyen preguntas y ejercicios sobre el lenguaje de programación C++.

En el Capítulo 4 (**Tipos lineales**) se incluyen preguntas y ejercicios sobre los tipos lineales más importantes: la pila, la cola y la lista.

En el Capítulo 5 (**Tipo árbol**) se incluyen preguntas y ejercicios sobre diferentes tipos de árboles: árbol binario, árbol binario de búsqueda, árbol AVL, árbol 2-3, etc.

En el Capítulo 6 (**Tipo conjunto**) se incluyen preguntas y ejercicios sobre la representación de los conjuntos, el tipo unión-búsqueda y la cola de prioridad.

Por último, en el Capítulo 7 (**Tipo grafo**) se incluyen preguntas y ejercicios sobre los grafos dirigidos y no dirigidos.

Por otro lado, en el Capítulo 8 (**Exámenes**), se proponen cuatro exámenes sin solución similares a los empleados en la asignatura “Programación y Estructuras de Datos”. Para cada pregunta, ya sea de tipo test o de tipo ejercicio, se incluye una referencia (número de pregunta y página) a la explicación de la solución de la pregunta que aparece en los capítulos anteriores.

En el Capítulo 9 (**Solución de las preguntas de tipo test de los exámenes**), se incluye la solución (verdadero o falso) a las preguntas de tipo test de los exámenes propuestos en el Capítulo 8. Además, también se incluye una referencia a la explicación de la solución de la pregunta.

El libro se completa con un apéndice con la bibliografía recomendada y un índice alfabético que facilita la búsqueda de información.

Las preguntas planteadas para cada tema son:

- Introducción a los tipos abstractos de datos: 14 preguntas, 10 ejercicios.

- El lenguaje C++: 9 preguntas, 2 ejercicios.
- Tipos lineales: 7 preguntas, 8 ejercicios.
- Tipo árbol: 58 preguntas, 12 ejercicios.
- Tipo conjunto: 39 preguntas, 7 ejercicios.
- Tipo grafo: 11 preguntas, 5 ejercicios.

En total, este libro contiene 138 preguntas y 44 ejercicios.

1.6. Cómo utilizar este libro

Como ya se ha comentado, este libro pretende ser un material complementario que ayude a los alumnos a la preparación de los distintos exámenes de la asignatura. Teniendo en mente este objetivo, el libro se puede emplear de diversas formas:

- Para hacer ejercicios de forma individual. Utilizando el índice del libro (que se corresponde con el temario de la asignatura) el alumno puede seleccionar un tipo abstracto de datos concreto (por ejemplo, el árbol 2-3-4) y realizar tantos las preguntas como los ejercicios exclusivos de este tipo de datos.
- Para hacer exámenes tipo. Esta opción del libro permite al alumno realizar un examen global de todos los contenidos de la asignatura (tanto de test como de ejercicios). Sólo se presenta el enunciado del examen con los correspondientes enlaces a las soluciones; esto permite al alumno realizar un examen “en situación real” sin tener las soluciones delante.
- Cada una de las preguntas y ejercicios incluye su correspondiente solución, lo que permite al alumno realizar una autoevaluación de sus conocimientos adquiridos.

1.7. Agradecimientos

Queremos dar las gracias a todos los profesores que han impartido docencia en esta asignatura, cuya aportación a los contenidos de la misma ha sido fundamental. Asimismo, queremos agradecer a Francisco José Sánchez Alcaraz por su colaboración en los gráficos de este libro.