

# TADs ESTRUCTURAS DE DATOS Y RESOLUCION DE PROBLEMAS CON C++

por NYHOFF

Isbn 8420546399

## Indice del Contenido

### PREFACIO

#### 1. DESARROLLO DE SOFTWARE

1.1. Análisis y especificación de problemas

1.2. Diseño

Diseño descendente

Diseño orientado a objetos

Diseño a pequeña escala

1.3. Codificación

1.4. Pruebas, ejecución y depuración

Especificación de la búsqueda binaria

1.5. Mantenimiento

Repaso rápido 1.5

Ejercicios 1.5

Resumen

Conceptos

Pautas de programación

Consejos sobre TADs

Problemas de programación

#### 2. INTRODUCCIÓN A LOS TIPOS ABSTRACTOS DE DATOS

2.1. Una primera mirada a los TADs y sus implementaciones

2.2. Tipos de datos simples en C++

Datos enteros

Números reales

Caracteres

Valores booleanos

Repaso rápido 2.2

Ejercicios 2.2

2.3. Tipos de datos definidos por el programador

Typedef

Tipos enumerados

Repaso rápido 2.3

Ejercicios 2.3

2.4. Punteros

Declaración de punteros

Operaciones básicas sobre punteros

Reserva de memoria dinámica: la operación new

Repaso rápido 2.4

Ejercicios 2.4

Resumen

Conceptos

Pautas de programación

Consejos sobre TADs

Problemas de programación

### 3. ESTRUCTURAS DE DATOS Y TIPOS ABSTRACTOS DE DATOS

#### 3.1. Estructuras de datos, tipos abstractos de datos e implementaciones

#### 3.2. Vectores estáticos

Vectores estáticos unidimensionales

La operación de acceso

Vectores como parámetros

Errores fuera de rango

Problemas con vectores

Repaso rápido 3.2

Ejercicios 3.2

#### 3.3. Vectores multidimensionales (opcional)

Vectores bidimensionales

Vectores con más dimensiones

Declaraciones de vectores de vectores

Vectores multidimensionales como parámetros

Repaso rápido 3.3

Ejercicios 3.3

#### 3.4. Vectores dinámicos

El operador new: vectores dinámicos

Otros usos de punteros

Repaso rápido 3.4

Ejercicios 3.4

#### 3.5. Registros al estilo de C (opcional)

Punteros a registros

Repaso rápido 3.5

#### 3.6. Programación con procedimientos

Ejemplo de programación procedimental

Resumen

Conceptos

Pautas de programación

Consejos sobre TADs

Problemas de programación

### 4. MÁS SOBRE POO Y TADS: CLASES

#### 4.1. Programación con procedimientos

La programación orientada a objetos

Repaso rápido 4.1

#### 4.2. Clases

Diferencias entre registros "tradicionales" (C) y clases y registros POO (C++)

Declaración de clases

Repaso rápido 4.2

#### 4.3. Una primera visión de una clase Hora

¿Por qué no hacer públicos todos los miembros de una clase?

Implementación de una clase

Algunas observaciones

#### 4.4. Constructores de clases

#### 4.5. Otras operaciones de clases

Operaciones de copia: inicialización y asignación

Observadores y modificadores

Sobrecarga de operadores

Sobrecarga de operadores de entrada/salida

Otras operaciones: adelantar y los operadores relacionales

Resumen y algunos otros datos

Punteros a objetos

El puntero this

Repaso rápido 4.5

Ejercicios 4.5  
Resumen  
Conceptos  
Pautas de programación  
Consejos sobre TADs  
Problemas de programación

## 5. LAS CLASES ESTÁNDAR DE C++ PARA ENTRADA/SALIDA Y CADENAS DE CARACTERES

### 5.1. Las clases estándar de C++ para entrada/salida

La clase istream  
La clase ostream  
E/S a través de ficheros: clases ifstream y ofstream  
La jerarquía de clases de E/S  
Repaso rápido 5.1

### 5.2. Los tipos de cadenas de caracteres en C++

Cadenas al estilo de C

La clase string de C++

Canales de cadenas

Repaso rápido 5.2

Ejercicios 5.2

### 5.3. Caso de estudio: edición de textos

### 5.4. Introducción al encaje de patrones (opcional)

### 5.5. Introducción a la encriptación de datos (opcional)

Estándar de encriptación de datos

Encriptación de clave pública

Resumen

Conceptos

Pautas de programación

Consejos sobre TADs

Problemas de programación

## 6. LISTAS

### 6.1. Las listas como TAD

Diseño y construcción de una clase Lista

### 6.2. Una implementación de listas basada en vectores estáticos

Implementación de las operaciones

Una clase Lista con almacenamiento en un vector estático

### 6.3. Una implementación de listas basada en vectores dinámicos

Memoria dinámica en las clases: destructores, constructores de copia y asignación

Nota final

Repaso rápido 6.3

Ejercicios 6.3

### 6.4. Introducción a las listas enlazadas

¿Qué son?

Implementación de las operaciones básicas de las listas

Resumen

Repaso rápido 6.4

Ejercicios 6.4

### 6.5. Una implementación de listas enlazadas basada en punteros en C++

Estructura de los nodos

Atributos para la implementación de listas enlazadas

Métodos para la implementación de listas enlazadas

Ejercicios 6.5

### 6.6. Una implementación de listas enlazadas basada en vectores (opcional)

Estructura de los nodos

Organización de repositorio

Ejercicios 6.6  
Resumen  
Conceptos  
Pautas de programación  
Consejos sobre TADs  
Problemas de programación

## 7. PILAS

7.1. Introducción a las pilas  
Problema 1  
Problema 2  
Problema 3  
Problema 4  
7.2. Diseño y construcción de una clase  
Pila basada en vector  
Seleccionando estructuras de almacenamiento  
Implementando las operaciones  
La clase Pila completa  
Utilizando un vector dinámico para almacenar los elementos de la pila  
Una mirada hacia delante  
Repaso rápido 7.2  
Ejercicios 7.2  
7.3. Pilas enlazadas  
Seleccionando una estructura de almacenamiento  
Implementando las operaciones  
La clase Pila completa: versión con lista enlazada  
Ejercicios 7.3  
7.4. Uso de pilas en llamadas de función  
Repaso rápido 7.4  
Ejercicios 7.4  
7.5. Caso de estudio: notación postfija (RPN)  
Evaluando expresiones postfijas  
Transformando expresiones infijas en postfijas  
Repaso rápido 7.5  
Ejercicios 7.5  
Resumen  
Conceptos  
Pautas de programación  
Consejos sobre TADs  
Problemas de programación

## 8. COLAS

8.1. Introducción a las colas  
Problemas de entrenamiento  
8.2. Diseño y construcción de una clase Cola basada en vectores  
Utilización de un vector estático para almacenar los elementos de la cola  
Utilización de un vector dinámico para almacenar los elementos de la cola  
Repaso rápido 8.2  
Ejercicios 8.2  
8.3. Colas enlazadas  
Una primera implementación enlazada  
Utilización de una lista enlazada circular  
Repaso rápido 8.3  
Ejercicios 8.3  
8.4. Aplicaciones de las colas: buffers y planificación  
Repaso rápido 8.4

Ejercicios 8.4

8.5. Caso de estudio: Simulación de un centro de información

Análisis y especificación del problema

Construcción de una clase

Simulación

Las clases Reloj y Llamada

Resumen

Conceptos

Pautas de programación

Consejos sobre TADs

Problemas de programación

## 9. IMPLEMENTACIÓN DE TADS: PLANTILLAS Y CONTENEDORES ESTÁNDAR

9.1. Introducción: la evolución de la reusabilidad y la genericidad

De los algoritmos a los algoritmos

De los datos a los contenedores

9.2. Genericidad de funciones: sobrecarga y plantillas

Sobrecarga

Plantillas de funciones

Mostrando un vector

9.3. Genericidad de clases: plantillas

¿Qué tiene de malo typedef?

Plantillas de clases

Una versión alternativa de la plantilla de clases Pila

Una mirada a las plantillas de clases contenedoras estándar de C++

Repaso rápido 9.3

Ejercicios 9.3

9.4. El contenedor vector

Declarando objetos vector

Algunas operaciones sobre vector

Un primer vistazo a las entrañas: aumentando la capacidad

Un primer vistazo a los iteradores

Algunos métodos de vector que utilizan iteradores

Concluyendo: vectores frente a vectores (al estilo de C)

Repaso rápido 9.4

Ejercicios 9.4

9.5. Caso de estudio: contando las conexiones a un sistema informático

9.6. Vectores multidimensional (opcional)

Objetos vector bidimensionales

Operaciones de los vectores bidimensionales

Ejercicios 9.6

9.7. Otros contenedores estándar: deque, stack y queue

La plantilla de clases deque de la STI

Una nueva (pero innecesaria) versión de la plantilla de clases

El adaptador stack de la STI

El adaptador queue de la STI

Repaso rápido 9.7

9.8. bit: sets y valarrays (opcional)

Bbitsets

Valarrays

Slices, masks y vectores indirectos

Resumen

Conceptos

Pautas de programación

Consejos sobre TADs

## Problemas de programación

### 10. IMPLEMENTACIÓN DE TADS: RECURSIÓN, ANÁLISIS DE ALGORITMOS Y ALGORITMOS ESTÁNDAR

#### 10.1. Recursión

Ejemplos de funciones recursivas

Codificación de funciones recursivas

Ejemplo de recursión inapropiada: números de Fibonacci

Búsqueda binaria

Comprobación de palíndromo

Repaso rápido 10.1

Ejercicios 10.1

#### 10.2. Ejemplos de recursión: Torres de Hanoi y análisis sintáctico

Torres de Hanoi

Análisis sintáctico

Ejercicios 10.2

#### 10.3. Implementación de la recursión

#### 10.4. Eficiencia de algoritmos

Repaso rápido 10.4

Ejercicios 10.4

#### 10.5. Algoritmos estándar en C++

Algoritmo sort de la STI

Algunos algoritmos de la STI

Algoritmos de la librería

Puntuación en el patinaje artístico

Repaso rápido 10.5

#### 10.6. Demostración de la corrección de algoritmos (opcional)

Cálculo de la media

Función recursiva para calcular potencias

Resumen

Ejercicios 10.6

Resumen

Conceptos

Pautas de programación

Consejos sobre TADs

Problemas de programación

### 11. MÁS ENLACES EN LAS LISTAS ENLAZADAS

#### 11.1. Algunas variantes de las listas enlazadas simples

Listas enlazadas con nodos de cabeza

Listas enlazadas circulares

Repaso rápido 11.1

Ejercicios 11.1

#### 11.2. Implementación con listas enlazadas de polinomios dispersos

Ejercicios 11.2

#### 11.3. Listas doblemente enlazadas y el contenedor estándar list de C++

Listas doblemente enlazadas

La plantilla de clases estándar List

Direcciones de Internet

Un vistazo a las entrañas del contenedor list de C++

Ejercicios 11.3

#### 11.4. Caso de estudio: aritmética de grandes números

Problema

Diseño

Implementación

Ejercicios 11.4

## 11.5. Otras listas múltiplemente enlazadas

Listas múltiplemente enlazadas

Matrices dispersas

Listas generalizadas

Ejercicios 11.5

Resumen

Conceptos

Pautas de programación

Consejos sobre TADs

Problemas de programación

## 12. BÚSQUEDA: ÁRBOLES BINARIOS Y TABLAS DISPERSAS

### 12.1. Repaso de las búsquedas lineal y binaria

Búsqueda lineal

Búsqueda binaria

Ejercicios 12.1

### 12.2. Introducción a los árboles binarios

Terminología de árboles

Algunos ejemplos de árboles binarios

Representación de árboles binarios en un vector

Representación enlazada de árboles binarios

Repaso rápido 12.2

Ejercicios 12.2

### 12.3. Los árboles binarios como estructura de datos recursiva

Recorridos

Repaso rápido 12.3

Ejercicios 12.3

### 12.4. Árboles binarios de búsqueda

Implementación de los ABBs

Recorridos de un ABB

Búsquedas en un ABB

Inserción en un ABB

Eliminación de un nodo de un ABB

El problema del desequilibrio

Repaso rápido 12.4

Ejercicios 12.4

### 12.5. Caso de estudio: validación de accesos a una computadora

Problema

Diseño

Codificación

### 12.6. Árboles binarios de búsqueda hilvanados (opcional)

Repaso rápido 12.6

Ejercicios 12.6

### 12.7. Tablas dispersas

Funciones de dispersión

Estrategias para resolver colisiones

Mejoras

Repaso rápido 12.7

Ejercicios 12.7

Resumen

Conceptos

Pautas de programación

Consejos sobre TADs

Problemas de programación

## 13. ORDENACIÓN

- 13.1. Algunos esquemas de ordenación en  $O(n^2)$ 
  - Ordenación por selección
  - Ordenación por intercambio
  - Ordenación por inserción
  - Evaluación de estos esquemas de ordenación
  - Ordenación indirecta
  - Repaso rápido 13.1
  - Ejercicios 13.1
- 13.2. Montículos, ordenación con montículo y colas con prioridad
  - Montículos
  - Operaciones básicas de los montículos
  - Ordenación con montículo
  - Algoritmos sobre montículos en la STI
  - Montículos y colas con prioridad
  - Repaso rápido 13.2
  - Ejercicios 13.2
- 13.3. Ordenación rápida
  - La operación de repartir
  - Ordenación rápida
  - Mejoras
  - Repaso rápido 13.3
  - Ejercicios 13.3
- 13.4. Ordenación por mezclas
  - Mezclando listas
  - Ordenación por mezclas binaria
  - Ordenación por mezclas natural
  - Repaso rápido 13.4
  - Ejercicios 13.4
- 13.5. Ordenación por radicales
  - Ejercicios 13.5
- Resumen
- Conceptos
- Pautas de programación
- Consejos sobre TADs
- Problemas de programación

## 14. POO Y TADS

- 14.1. Una breve visión general e histórica de la POO y los TAD
  - Encapsulación
  - Herencia
  - Polimorfismo y vinculación dinámica
  - Repaso rápido 14.1
- 14.2. Herencia y diseño orientado a objetos
  - Carnets y licencias
  - Secciones públicas, privadas y protegidas
  - La forma de las clases derivadas
  - Las relaciones es-una, tiene-una y usa-una entre clases
  - Repaso rápido 14.2
  - Ejercicios 14.2
- 14.3. Construyendo clases derivadas
  - Constructores de la clase derivada
  - Accediendo a los atributos heredados
  - Reutilizando operaciones
  - Pilas y pilas acotadas
- 14.4. Caso de estudio: paga de nóminas
  - Problema



Diseño

14.5. Polimorfismo, funciones virtuales y TADs

Por qué se necesita el polimorfismo: un problema de vinculación

Funciones virtuales y vinculación dinámica

Utilizando asideros

Pilas y pilas acotadas

Funciones virtuales puras y clases abstractas

Repaso rápido 14.5

Ejercicios 14.5

14.6. Caso de estudio: una estructura de datos heterogénea

La necesidad de funciones virtuales

Resumen

Conceptos

Pautas de programación

Consejos sobre TADs

Problemas de programación

## 15. ÁRBOLES

15.1. Caso de estudio: códigos de Huffman

Códigos de longitud variable

Descodificabilidad inmediata

Códigos de Huffman

Ejercicios 15.1

15.2. Árboles equilibrados: árboles AVL

Un ABB de abreviaturas de ciudades

Las rotaciones básicas

Repaso rápido 15.2

Ejercicios 15.2

15.3. Árboles 2-3-4, árboles rojinegros, B-árboles y otros árboles

Árboles 2-3-4

Árboles rojinegros

B-árboles

Representación de árboles y bosques como árboles binarios

Repaso rápido 15.3

Ejercicios 15.3

15.4. Contenedores asociativos en la STL: maps (opcional)

Repaso rápido 15.4

Resumen

Conceptos

Pautas de programación

Consejos sobre TADs

Problemas de programación

## 16. GRAFOS Y DIGRAFOS

16.1. Grafos dirigidos

Representación con una matriz de adyacencia

Representación con listas de adyacentes

Repaso rápido 16.1

Ejercicios 16.1

16.2. Búsqueda y recorrido de digrafos

Búsqueda en profundidad

Búsqueda en anchura

Recorridos y problemas de camino mínimo

Problemas NP-completos

Repaso rápido 16.2

Ejercicios 16.2

16.3. Grafos  
Representación por medio de listas de aristas  
Conectividad  
Repaso rápido 16.3  
Ejercicios 16.3  
Resumen  
Conceptos  
Pautas de programación  
Consejos sobre TADs  
Problemas de programación

## APÉNDICES

- A. Conjunto de caracteres ASCII
- B. Sistemas de numeración
- C. C++ básico
  - C.1. Estructura de un programa en C++
  - C.2. Directivas del compilador
  - C.3. Librerías estándar
  - C.4. Comentarios
  - C.5. Identificadores y palabras clave
  - C.6. Tipos de datos fundamentales
  - C.7. Constantes
  - C.8. Declaraciones
  - C.9. Operadores y expresiones
  - C.10. Estructuras de control
  - C.11. Funciones
  - C.12. Tiempos de vida, ámbitos y espacios de nombres
  - C.13. Ficheros
- D. Otras características de C++
  - D.1. Operaciones de canales
  - D.2. Operaciones de cadenas
  - D.3. Excepciones
  - D.4. Más sobre las plantillas de funciones
  - D.5. Otras aplicaciones de los punteros
- E. De Java a C++
  - E.1. Estructura de un programa
  - E.2. Directivas using y del compilador C++
  - E.3. Entrada y salida (secciones C.9, C.13 y D.1)
  - E.4. Tipos de datos
  - E.5. Variables y constantes
  - E.6. Funciones
  - E.7. Sobrecarga de operadores
  - E.8. Aspectos similares en C++ y Java
  - E.9. Más diferencias entre C++ y Java
- F. Respuestas a los repastos

## ÍNDICE