

Contenido



UNIVERSIDAD NACIONAL DE ENTRE RÍOS
FACULTAD DE INGENIERÍA
CENTRO DE ESTUDIOS CIENTÍFICOS Y TECNOLÓGICOS

9932

Prefacio, xix

Introducción, xxxvii

Prólogo a la edición española, xxxix

Capítulo 1

La computación y la metodología de diseño orientado a objetos, 1

Introducción, 1

Conceptos fundamentales, 1

- 1.1. Terminología básica en computación, 2
 - 1.1.1. Unidades de medida en computación, 2
 - 1.1.2. Organización de una computadora, 3
 - 1.1.3. Programación, 14
- 1.2. Software, 20
 - 1.2.1. Software de sistema, 20
 - 1.2.2. Software de aplicación, 25
- 1.3. Ingeniería del software, 28
 - 1.3.1. Principios de la ingeniería del software, 31
- 1.4. Diseño orientado a objetos, 35
 - 1.4.1. Programación orientada a objetos, 38
- 1.5. Puntos a recordar, 45
- 1.6. Para ampliar conocimientos, 48
- 1.7. Ejercicios, 48

Capítulo 2

C++: los fundamentos, 55

Introducción, 55

Conceptos fundamentales, 55

- 2.1. Organización de un programa, 56
- 2.2. Un primer programa, 56
- 2.3. Un segundo programa, 59
- 2.4. Comentarios, 60

- 2.5. Asignando un valor, 63
- 2.6. Objetos fundamentales de C++, 64
 - 2.6.1. Tipos de objetos enteros, 65
 - 2.6.2. Tipos de objetos carácter, 66
 - 2.6.3. Tipos de objetos en coma flotante, 67
- 2.7. Constantes, 68
 - 2.7.1. Constantes de tipo carácter y de cadenas, 69
 - 2.7.2. Constantes enteras, 72
 - 2.7.3. Constantes en coma flotante, 75
- 2.8. Nombres, 78
 - 2.8.1. Palabras reservadas, 80
 - 2.8.2. Identificadores, 81
- 2.9. Definiciones, 82
- 2.10. Expresiones, 86
 - 2.10.1. Expresiones simples, 87
 - 2.10.2. Operaciones aritméticas binarias, 88
 - 2.10.3. Operaciones aritméticas unarias, 93
 - 2.10.4. Área de un círculo, 93
 - 2.10.5. Expresiones mixtas, 94
 - 2.10.6. Precedencia, 95
 - 2.10.7. Asociatividad, 97
- 2.11. Sentencias de salida, 98
- 2.12. Cálculo de la velocidad media, 102
- 2.13. Puntos a recordar, 105
- 2.14. Ejercicios, 109

Capítulo 3

Modificación de objetos, 115

Introducción, 115

Conceptos fundamentales, 115

- 3.1. Asignación, 116
 - 3.1.1. Conversiones en la asignación, 118
 - 3.1.2. Precedencia y asociatividad de la asignación, 119
- 3.2. Definiciones de constantes, 120
- 3.3. Sentencias de entrada, 121
- 3.4. Cálculo del número de moléculas de un hidrocarburo, 124
- 3.5. Asignación compuesta, 128
- 3.6. Incremento y decremento, 129
- 3.7. Estimación del ahorro anual en calderilla, 131
- 3.8. La clase string, 134
- 3.9. EzWindows, 141
 - 3.9.1. La clase VentanaSencilla, 142
 - 3.9.2. La clase Rectángulo, 144
- 3.10. La siega del jardín, 147

- 3.11. Puntos a recordar, 153
- 3.12. Ejercicios, 158

Capítulo 4

Construcciones de control, 165

- Introducción, 165
- Conceptos fundamentales, 165
 - 4.1. Álgebra booleana, 166
 - 4.1.1. Tablas de verdad, 166
 - 4.1.2. Expresiones lógicas, 168
 - 4.2. Un tipo booleano, 169
 - 4.2.1. Operadores booleanos, 169
 - 4.2.2. Operadores relacionales, 171
 - 4.2.3. Revisión de la precedencia de los operadores, 173
 - 4.2.4. Evaluación con cortocircuito, 174
 - 4.3. Ejecución condicional usando la sentencia if, 176
 - 4.3.1. La sentencia if-else, 179
 - 4.3.2. Ordenamiento de tres números, 184
 - 4.4. Ejecución condicional utilizando la sentencia switch, 186
 - 4.5. Cálculo de una expresión solicitada, 189
 - 4.6. La validación de una fecha, 192
 - 4.7. La iteración usando la sentencia while, 199
 - 4.8. Procesamiento sencillo de caracteres y cadenas, 206
 - 4.9. La iteración usando la construcción for, 215
 - 4.10. Visualización de datos simple, 223
 - 4.11. La solución del acertijo del vagabundo perezoso, 225
 - 4.12. La iteración usando la construcción do, 227
 - 4.13. Puntos a recordar, 231
 - 4.14. Ejercicios, 233

Capítulo 5

Fundamentos del uso de funciones y las bibliotecas, 245

- Introducción, 245
- Conceptos fundamentales, 245
 - 5.1. Fundamentos de las funciones, 246
 - 5.1.1. Especificación de la interfaz, 248
 - 5.1.2. Prototipado de funciones, 251
 - 5.1.3. Invocación y flujo de control, 252
 - 5.2. El preprocesador, 257
 - 5.2.1. Definiciones de macro, 257
 - 5.2.2. La compilación condicional, 258
 - 5.2.3. Directivas de inclusión de archivos, 259
 - 5.3. Utilización de bibliotecas de software, 261
 - 5.4. La biblioteca iostream, 263

- 5.4.1. Los flujos de datos estándar, 264
- 5.4.2. Objetos de flujo de datos de error estándar, 265
- 5.4.3. Los manipuladores de iostream, 266
- 5.5. La biblioteca iomanip, 268
- 5.6. La biblioteca fstream, 277
- 5.7. La biblioteca math, 283
 - 5.7.1. Visualización de una tabla de valores de una función, 285
- 5.8. La biblioteca ctype, 288
- 5.9. Las macros assert, 288
- 5.10. Puntos a recordar, 291
- 5.11. Para profundizar, 295
- 5.12. Ejercicios, 295

Capítulo 6 **Funciones definidas por el programador, 301**

Introducción, 301

Conceptos fundamentales, 301

6.1. Fundamentos, 302

6.1.1. Sintaxis de la definición de función, 302

6.1.2. Invocación y flujo de control, 304

6.2. Un problema delicioso, 306

6.3. Algunas funciones útiles, 312

6.4. Integración de un polinomio de segundo grado, 315

6.5. El ámbito local, 318

6.5.1. Las reglas del ámbito local, 318

6.5.2. Reutilización de nombres con objetos, 319

6.6. El ámbito global, 322

6.6.1. Reglas de ámbito y reutilización de nombres con objetos globales, 322

6.6.2. Iniciación de los objetos globales, 324

6.7. Visualización de un diagrama de precios/intervalos, 324

6.8. Funciones recursivas, 335

6.9. Puntos a recordar, 341

6.10. Ejercicios, 343

Capítulo 7 **Aspectos avanzados del paso de parámetros, 353**

Introducción, 353

Conceptos fundamentales, 353

7.1. Parámetros por referencia, 354

7.2. Pasando objetos por referencia, 364

7.3. Validando el código de acceso telefónico, 366

7.4. Parámetros constantes, 369

7.5. Parámetros por defecto, 372

7.6. Conversión de los parámetros de una función, 374

- 7.7. Sobrecarga de funciones, 376
- 7.8. Números aleatorios, 378
- 7.9. Un entrenador para la automatización de una fábrica, 385
- 7.10. Para ampliar conocimientos, 395
- 7.11. Puntos a recordar, 396
- 7.12. Ejercicios, 398

Capítulo 8

La construcción clase y el diseño orientado a objetos, 411

Introducción, 411

Conceptos fundamentales, 411

- 8.1. Introducción a los tipos de datos definidos por el programador, 412
- 8.2. La clase Rectángulo, 414
 - 8.2.1. Acceso privado y público, 421
- 8.3. La utilización de la clase Rectángulo, 423
- 8.4. Constructores, 427
- 8.5. Construcción de un caleidoscopio, 428
- 8.6. Análisis y diseño orientado a objetos, 433
- 8.7. Puntos a recordar, 445
- 8.8. Para profundizar, 446
- 8.9. Ejercicios, 449

Capítulo 9

Implementación de tipos abstractos de datos, 455

Introducción, 455

Conceptos fundamentales, 455

- 9.1. Introducción a los tipos de datos abstractos, 456
- 9.2. ADT para números racionales, 457
 - 9.2.1. Un programa cliente que utiliza la biblioteca racional, 460
- 9.3. Descripción de la interfaz de la clase racional, 465
 - 9.3.1. Restricciones en el acceso, 466
 - 9.3.2. Funciones miembro constructoras, 467
 - 9.3.3. Funciones miembro de utilidad, 467
 - 9.3.4. Funciones miembro inspectoras, 470
 - 9.3.5. Funciones miembro modificadoras, 470
 - 9.3.6. Miembros de datos, 471
 - 9.3.7. Operadores sobrecargados, 472
- 9.4. Implementación de la clase racional, 474
 - 9.4.1. Definición de los constructores, 475
 - 9.4.2. Definición de las funciones inspectoras, 476
 - 9.4.3. Definición de las funciones modificadoras, 477
 - 9.4.4. Definición de las funciones de utilidad aritméticas, 478
 - 9.4.5. Definición de las funciones de utilidad de inserción y extracción, 479

- 9.4.6. Definición de los operadores aritméticos auxiliares, 481
- 9.4.7. Definición de los operadores auxiliares de flujos, 482
- 9.5. Construcción de copia, asignación de miembros y destrucción, 483
- 9.6. Una ADT para enteros pseudoaleatorios, 488
 - 9.6.1. Implementación de EnteroAleatorio, 493
- 9.7. Juego del rojo-amarillo-verde, 497
 - 9.7.1. Abstracción e interfaz, 498
 - 9.7.2. Implementación de la clase Elemento, 505
 - 9.7.3. Implementación de la clase Conjetura, 506
 - 9.7.4. Implementación de la clase RespuestaRav, 509
 - 9.7.5. Implementación de la clase RAV, 511
- 9.8. Puntos a recordar, 519
- 9.9. Ejercicios, 521

Capítulo 10

✓ Listas, 529

- ✓ Introducción, 529
- ✓ Conceptos fundamentales, 529
- ✓ 10.1. Colecciones con nombre, 530
- ✓ 10.2. Vectores unidimensionales, 531
 - ✓ 10.2.1. Iniciación de vectores, 534
 - ✓ 10.2.2. Vectores constantes, 535
 - ✓ 10.2.3. Procesamiento de vectores simple, 535
 - ✓ 10.2.4. Cadenas de caracteres, 541
- 10.3. Vectores como parámetros, 544
- 10.4. Ordenación, 548
 - 10.4.1. El método InserciónEnOrden(), 549
 - 10.4.2. Calidad de la función InserciónEnOrden(), 553
- 10.5. Clases contenedoras, 554
- 10.6. Clase vector, 557
 - 10.6.1. Acceso aleatorio a los elementos de vector, 560
 - 10.6.2. Métodos de acceso secuencial, 564
 - 10.6.3. Paso de objetos de tipo vector a funciones, 569
- 10.7. QuickSort(), 574
- 10.8. Búsqueda binaria, 580
- 10.9. Clase string revisada, 583
- ✓ 10.10. Búsqueda de una palabra. Exploración de una lista bidimensional, 585
- 10.11. El juego del laberinto, 591
 - 10.11.1. Especificación de la clase Laberinto, 594
 - 10.11.2. Especificación de la clase Caminante, 597
 - 10.11.3. Implementación de la clase Laberinto, 601
 - 10.11.4. Implementación de la clase Caminante, 604
 - 10.11.5. Representación del camino, 606
 - 10.11.6. Representación gráfica del laberinto, 609

- 10.11.7. Programa de control, 612
- 10.11.8. Implementación de la estrategia de la mano derecha, 616
- ✓ 10.12. Vectores multidimensionales, 619
 - ✓ 10.12.1. Matrices, 622
- 10.13. Puntos a recordar, 623
- 10.14. Ejercicios, 629

Capítulo 11

La API de EzWindows: un examen detallado, 639

Introducción, 639

Conceptos fundamentales, 639

- 11.1. Interfaces de programación de aplicaciones, 640
- 11.2. Una clase de ventanas sencillas, 641
 - 11.2.1. Programación basada en eventos, 642
 - 11.2.2. Sistema de coordenadas de VentanaSencilla, 643
 - 11.2.3. Hola EzWindows, 646
 - 11.2.4. Mecanismos de la API de EzWindows, 649
- 11.3. La clase MapaDeBits, 653
- 11.4. Eventos de ratón, 655
- 11.5. Mapas de Bits y eventos del ratón, 660
- 11.6. Eventos de temporizador, 663
- 11.7. Mensajes de alerta, 667
- 11.8. Simón dice, 668
- 11.9. Puntos a recordar, 687
- 11.10. Ejercicios, 687

Capítulo 12

✓ Punteros y memoria dinámica, 695

Introducción, 695

Conceptos fundamentales, 695

- 12.1. Valores-l y valores-r, 696
- 12.2. Conceptos básicos de punteros, 697
 - 12.2.1. Direccionamiento e indirección, 699
 - 12.2.2. Punteros a punteros, 703
 - 12.2.3. Punteros como parámetros, 704
- 12.3. Constantes punteros y punteros a constantes, 706
- ✓ 12.4. Vectores y punteros, 708
- ✓ 12.5. Procesamiento de tiras de caracteres, 712
- 12.6. Parámetros de mandatos de línea, 715
- 12.7. Punteros a funciones, 717
- 12.8. Objetos dinámicos, 721
- 12.9. Un ADT sencillo para representar listas de valores enteros, 727
 - 12.9.1. Especificación de ListaEnteros, 729
 - 12.9.2. Implementación del constructor de ListaEnteros, 733

- 12.9.3. Implementación del destructor de ListaEnteros, 735
- 12.9.4. Implementación del operador de indexación, 735
- 12.9.5. Asignación de miembros en ListaEnteros y el puntero this, 736
- 12.9.6. Sobrecarga del operador de inserción para objetos ListaEnteros, 738
- 12.10. Puntos a recordar, 738
- 12.11. Ejercicios, 742

Capítulo 13 Herencia, 751

Introducción, 751

Conceptos fundamentales, 751

- 13.1. Diseño orientado a objetos usando herencia, 751
- 13.2. Reutilización vía herencia, 753
- 13.3. Una jerarquía de formas, 756
 - 13.3.1. Declarando una clase derivada, 761
 - 13.3.2. Implementando una clase derivada, 763
- 13.4. Miembros protegidos y herencia, 771
- 13.5. Controlando la herencia, 774
 - 13.5.1. Herencia pública, 774
 - 13.5.2. Herencia privada, 775
 - 13.5.3. Herencia protegida, 776
- 13.6. Herencia múltiple, 776
- 13.7. Un caleidoscopio más bonito, 784
- 13.8. Puntos a recordar, 801
- 13.9. Ejercicios, 803

Capítulo 14 Plantillas y polimorfismo, 809

Introducción, 809

Conceptos fundamentales, 809

- 14.1. Acciones genéricas y tipos, 810
- 14.2. Plantillas para funciones, 810
- 14.3. Plantillas de clases, 814
- 14.4. Una clase de listas sencillas usando una plantilla de clase, 815
 - 14.4.1. Especificación, 817
 - 14.4.2. Implementación, 820
- 14.5. Listas secuenciales, 825
 - 14.5.1. Plantilla de la clase ListaSecuencial, 825
 - 14.5.2. Fundamentos de los miembros de función de ListaSecuencial, 829
 - 14.5.3. Función miembro display(), 833
 - 14.5.4. Añadiendo un elemento, 834
 - 14.5.5. Borrando un elemento, 838

- 14.5.6. Borrando todos los elementos, 840
- 14.5.7. Un pequeño programa que usa la plantilla de clase ListaSecuencial, 841
- 14.5.8. Una clase de iteradores ListaSecuencial, 842
- 14.5.9. Implementación de la clase IteradorSecuencial, 846
- 14.6. Polimorfismo, 850
- 14.7. Matices de las funciones virtuales, 853
- 14.8. Clases base abstractas, 857
- 14.9. Herencia virtual múltiple, 860
- 14.10. Puntos a recordar, 862
- 14.11. Ejercicios, 865

Capítulo 15 Proyecto software: la Caza de la pulga, 873

Introducción, 873

Conceptos fundamentales, 873

- 15.1. Caza de la pulga, 873
- 15.2. Clase base pulga, 875
 - 15.2.1. Clase derivada PulgaLenta, 888
 - 15.2.2. Clase derivada PulgaRápida, 890
- 15.3. Clase del controlador del juego, 895
- 15.4. Caza de la pulga, 899
- 15.5. Puntos a recordar, 900
- 15.6. Ejercicios, 902

Apéndice A Tablas, 905

- A.1. Juego de caracteres ASCII, 906
- A.2. Precedencia de operadores 907

Apéndice B Bibliotecas estándar, 909

- B.1. Nombres de bibliotecas y accesos, 909
- B.2. Biblioteca iostream, 910
- B.3. Biblioteca stdlib, 912
- B.4. Biblioteca time, 913
- B.5. Biblioteca cstring, 914
- B.6. Biblioteca algorithm, 915

Apéndice C Las clases estándar, 921

- C.1. Clases contenedoras, 921
 - C.1.1. Contenedor vector, 923
 - C.1.2. Contenedor list, 926
 - C.1.3. Contenedor dequeue, 926

- C.1.4. Adaptador contenedor `priority_queue`, 927
- C.1.5. Adaptador contenedor `queue`, 927
- C.1.6. Adaptador contenedor `stack`, 927
- C.1.7. Contenedor asociativo `map`, 927
- C.1.8. Contenedor asociativo `set`, 928
- C.2. La clase `string`, 928
 - C.2.1. Funciones miembro de `string`, 928
 - C.2.2. Funciones de `string` auxiliares, 930

Apéndice D

Temas avanzados, 933

- D.1. Espacios de nombres, 933
 - D.1.1. Definiciones, 934
 - D.1.2. Uso de los espacios de nombres, 937
- D.2. Manejo de excepciones, 940
 - D.2.1. Manejo de excepciones básico, 940
 - D.2.2. Manejo de excepciones genérico, 944
 - D.2.3. Especificación de posibles excepciones, 945
 - D.2.4. Captura de excepciones de asignación de memoria dinámica, 946
 - D.2.5. Código heredado, 947
- D.3. `Friends`, 948
 - D.3.1. Una implementación alternativa de `Rational`, 948

Apéndice E

Manual de referencia de la API de `EzWindows`, 951

- E.1. Tipos enumerados, 951
- E.2. Sistema de coordenadas, 952
- E.3. Clase `VentanaSencilla`, 953
- E.4. Clase `ObjetoVentana`, 957
- E.5. Clase `SegmentoDirigido`, 958
- E.6. Clase `Forma`, 960
- E.7. Clase `Elipse`, 961
- E.8. Clase `Círculo`, 962
- E.9. Clase `Rectángulo`, 962
- E.10. Clase `Triángulo`, 964
- E.11. Clase `Cuadrado`, 964
- E.12. Clase `Etiqueta`, 965
- E.13. Clase `MapaDeBits`, 967
- E.14. Clase `EnteroAleatorio`, 969
- E.15. Funciones misceláneas, 969

Apéndice F

Proyectos y `makefiles`, 971

- F.1. Aspectos básicos sobre proyectos y `makefiles`, 971
- F.2. IDE de C++ de Borland, 973

- F.2.1. Creación de un archivo de proyecto, 974
- F.2.2. Inclusión de los archivos fuente en el proyecto, 976
- F.2.3. Configuración de las opciones del proyecto, 977
- F.2.4. Edición, compilación y enlace, 979
- F.2.5. Salvar un archivo de proyecto, 980
- F.2.6. Ejecución de una aplicación de EzWindows, 980
- F.3. IDE de Visual C++ de Microsoft, 981
 - F.3.1. Creación de un espacio de trabajo, 982
 - F.3.2. Inclusión de archivos fuente en el proyecto, 983
 - F.3.3. Configuración de las opciones del proyecto, 985
 - F.3.4. Edición, compilación y enlace, 987
 - F.3.5. Salvar un archivo de proyecto, 987
 - F.3.6. Ejecución de una aplicación de EzWindows, 987
- F.4. Makefiles de UNIX, 988
 - F.4.1. Ejecución de una aplicación de EzWindows, 993

Índice, 995