

Sumário

Prefácio	17
Sobre o livro	18
Como fazer download dos arquivos-fonte	19
Como ler este livro	19
Experiência do autor	21
Parte I – Filosofia e Modelagem Orientada a Objetos	23
Capítulo 1 – Introdução à Programação Orientada a Objetos	25
1.1 Passado/Presente/Futuro da programação	25
1.2 Exemplo de objeto	27
1.3 Visão desorganizada versus orientada a objeto	28
Capítulo 2 – Conceitos Básicos de POO	30
2.1 Abstração	30
2.2 Objeto (ou Instância)	30
2.3 Classes	31
2.4 Encapsulamento	32
2.5 Atributos (Propriedades/Variáveis)	33
2.6 Métodos (Funções)	33
2.7 Herança (Hereditariedade)	34
2.8 Polimorfismo	35
Capítulo 3 – Modelagem TMO/UML	36
3.1 Introdução à modelagem orientada a objeto	36
3.2 Programas para desenho dos diagramas UML	37
3.3 Especificação e cenários	40
3.4 Modelo de objetos	40
3.5 Modelo dinâmico	46
3.6 Modelo funcional	50
3.7 Verificando os componentes e a execução	51
Capítulo 4 – Etapas para o Desenvolvimento de um Programa	53
4.1 Concepção – Especificação do programa	54
4.2 Elaboração	55
4.3 Análise orientada a objeto (AOO)	55
4.4 AOO – Modelagem de objetos	56
4.5 AOO – Modelagem dinâmica	61
4.6 AOO – Modelagem funcional	62
4.7 Projeto do sistema	63
4.8 Projeto orientado a objeto	64
4.9 Implementação do programa	65

4.10 Testes	65
4.11 Manutenção do programa	66
4.12 Documentação de um programa	68
Parte II – POO Utilizando C++	71
Capítulo 5 – Introdução ao C++	73
5.1 Um pouco de história	73
5.2 O que é o ANSI C++?	74
5.3 Quais as novidades e vantagens de C++?	74
5.4 Tipos de programação em C++	74
5.5 Seleção da plataforma de programação	75
5.6 Ambientes de desenvolvimento	76
5.7 Compilar, linkar, debugar e profiler	79
5.8 Diferenças de nomenclatura (POO e C++)	80
5.9 Layout de um programa	81
5.10 Exemplo de um programa orientado a objeto em C++	82
Capítulo 6 – Conceitos Básicos de C++	84
6.1 Sobre a sintaxe de C++	84
6.2 Palavras-chave do C++	87
6.3 Nome dos objetos (identificadores)	88
6.4 Declarações	88
6.5 Definições	90
Capítulo 7 – Tipos	93
7.1 Introdução ao conceito de tipos	93
7.2 Uso de tipos predefinidos de C++	93
7.3 Uso de tipos do usuário	97
7.4 Uso de tipos definidos em bibliotecas externas (STL)	98
7.5 Vantagem da tipificação forte do C++	100
7.6 Sentenças para tipos	100
Capítulo 8 – Namespace	101
8.1 O que é um namespace?	101
8.2 Utilizando o namespace da biblioteca-padrão de C++ (std)	101
8.3 Definindo um namespace	102
8.4 Compondo namespace	103
8.5 Sentenças para namespace	103
Capítulo 9 – Classes	105
9.1 Protótipo para declarar e definir classes	105
9.2 Encapsulamento em C++ utilizando o especificador de acesso	106
9.3 Classes aninhadas	106
9.4 Sentenças para classes	107
Capítulo 10 – Atributos	108
10.1 Protótipo para declarar e definir atributos	108

10.2 Atributos de objeto	108
10.3 Atributos de classe (estáticos)	109
10.4 Atributos const	111
10.5 Atributos com mutable	111
10.6 Atributos com volatile	112
10.7 Inicialização dos atributos da classe nos construtores	112
10.8 Sentenças para atributos	112
Capítulo 11 – Métodos	113
11.1 Protótipo para declarar e definir métodos	113
11.2 Declaração, definição e retorno de um método	114
11.3 Métodos normais	118
11.4 Métodos const	121
11.5 Métodos estáticos	122
11.6 Métodos inline	125
11.7 Sentenças para métodos	128
Capítulo 12 – Sobrecarga de Métodos	129
12.1 O que é a sobrecarga de métodos?	129
12.2 Exemplos de sobrecarga	129
12.3 Sentenças para métodos sobrecarregados	130
Capítulo 13 – Utilização de Ponteiros e Referências	131
13.1 Ponteiros	131
13.2 Criação e uso de objetos dinâmicos com ponteiros	132
13.3 Ponteiros const e ponteiros para const	134
13.4 Conversão de ponteiros	135
13.5 Ponteiro this	136
13.6 Utilizando auto_ptr	136
13.7 Ponteiros para métodos e atributos da classe	139
13.8 Referências (&)	140
Capítulo 14 – Métodos Construtores e Destrutores	142
14.1 Protótipo para construtores e destrutores	142
14.2 Métodos construtores	142
14.3 Construtor default	143
14.4 Construtor de cópia X (const X& obj)	144
14.5 Métodos destrutores	150
14.6 Sentenças para construtores e destrutores	152
Capítulo 15 – Herança	153
15.1 Protótipo para herança	153
15.2 Especificador de herança	155
15.3 Chamando construtores da classe-base explicitamente	157
15.4 Ambigüidade	157
15.5 Redecaração de método ou atributo na classe-derivada	158
15.6 Sentenças para herança	159

Sumário

Capítulo 16 – Herança Múltipla	160
16.1 Protótipo para herança múltipla	160
16.2 Herança múltipla	160
16.3 Ambigüidade em herança múltipla	161
16.4 Herança múltipla virtual	163
16.5 Sentenças para herança múltipla	165
16.6 Ordem de criação e destruição dos objetos em heranças	166
16.7 Exemplo de herança simples e herança múltipla	168
16.8 Análise dos erros emitidos pelo compilador	171
Capítulo 17 – Polimorfismo	173
17.1 Métodos não virtuais (normais, estáticos)	173
17.2 Métodos virtuais	174
17.3 Como implementar o polimorfismo	176
17.4 Métodos virtuais puros (classes abstratas)	179
17.5 Exemplo completo com polimorfismo	179
Capítulo 18 – Friend	190
18.1 Introdução ao conceito de friend	190
18.2 Classes friend	191
18.3 Métodos friend	191
18.4 Funções friend	192
18.5 Sentenças para friend	192
Capítulo 19 – Sobrecarga de Operador	194
19.1 Introdução à sobrecarga de operador	194
19.2 Protótipo para sobrecarga de operador	195
19.3 Operadores que podem ser sobrecarregados	195
19.4 Sobrecarga de operador como função friend	196
19.5 Sobrecarga de operador como método membro da classe	197
19.6 Alguns protótipos de sobrecarga	197
19.7 Sentenças para sobrecarga de operador	198
Capítulo 20 – Implementando Associações em C++	204
20.1 Introdução às associações em C++	204
20.2 Associação sem atributo	204
20.3 Associação com atributo	205
Capítulo 21 – Conversões	207
21.1 Protótipos para conversões	207
21.2 Necessidade de conversão	208
21.3 Construtor de conversão	209
21.4 Métodos de conversão (cast)	209
21.5 Conversão explícita nos construtores com explicit	210
21.6 Sentenças para métodos de conversão	211
21.7 Conversão dinâmica com dynamic_cast	212
21.8 Conversão estática com static_cast	213

21.9 Conversão com reinterpret_cast	214
21.10 Usando typeid	214
21.11 Verificação do tamanho de um objeto com sizeof	215
21.12 Referências e dynamic_cast	215
Capítulo 22 – Exceções	216
22.1 Introdução às exceções	216
22.2 Conceitos básicos de exceções	217
22.3 Seqüência de controle	220
22.4 Como fica a pilha (heap)	221
22.5 Exceções não tratadas	222
22.6 Exceção para new	222
22.7 Exceções-padrão	224
22.8 Sentenças para exceções	224
Capítulo 23 – Templates (ou gabaritos)	226
23.1 Introdução aos templates (gabaritos)	226
23.2 Protótipo para funções templates	226
23.3 Protótipo para classes templates (ou tipos paramétricos)	228
23.4 Especialização	229
23.5 Usando templates e especializações para obter mais desempenho3	230
23.6 Usando typename com templates	233
23.7 Sentenças para templates	233
Capítulo 24 – Entrada e Saída com C++	235
24.1 Introdução à entrada e saída de dados no C++	235
24.2 Biblioteca de entrada e saída	236
24.3 A classe locale	237
24.4 A classe <ios_base>	240
24.5 A classe <ios>	242
24.6 A classe <iomanip>	244
24.7 A classe <istream>	246
24.8 A classe <ostream>	249
24.9 A classe <sstream>	252
24.10 Sentenças para stream	253
Capítulo 25 – Entrada e Saída com Arquivos de Disco	254
25.1 Introdução ao acesso a disco	254
25.2 A classe <fstream>	254
25.3 Armazenando e lendo objetos	258
25.4 Posicionando ponteiros de arquivos com seekg(), seekp(), tellg(), tellp() 2	261
25.5 Acessando a impressora e a saída auxiliar	263
25.6 Arquivos de disco binários	264
25.7 Executando e enviando comandos para outro programa	264
25.8 Redirecionamento de entrada e saída	265

Capítulo 26 – A classe <string>	268
26.1 Introdução a strings	268
26.2 Métodos de string	268
26.3 Sentenças para strings	273
Capítulo 27 – Classes <complex>, <bitset> e as Funções Matemáticas de <cmath>	274
27.1 A classe <complex>	274
27.2 A classe <bitset>	276
27.3 Usando a biblioteca <cmath> de C	280
Parte III – Introdução à STL	283
Capítulo 28 – Introdução à Biblioteca-Padrão de Gabaritos de C++ (STL)	285
28.1 O que é a STL?	285
28.2 Introdução aos containers	286
28.3 Introdução aos iteradores (iterators)	291
Capítulo 29 – Containers Sequenciais <vector>, <list> e <deque>	294
29.1 A classe container <vector>	294
29.2 A classe container <list>	301
29.3 A classe container <deque>	305
Capítulo 30 – Containers Adaptativos <stack>, <queue> e <priority_queue>	308
30.1 A classe container <stack>	308
30.2 A classe container <queue>	310
30.3 A classe container <priority_queue>	312
Capítulo 31 – Containers Associativos <set>, <multiset>, <map> e <multimap>	313
31.1 A classe <pair>	313
31.2 A classe container <set>	313
31.3 A classe container <multiset>	316
31.4 A classe container <map>	317
31.5 A classe container <multimap>	322
Capítulo 32 – Programação Genérica	323
32.1 Introdução à programação genérica	323
32.2 Classificação das funções genéricas	323
32.3 Funções genéricas	325
Capítulo 33 – Objetos Funções da STL	336
33.1 Introdução aos objetos funções da STL	336
33.2 Objetos funções fornecidos pela STL	337
Parte IV – Programação Multiplataforma com Software Livre	347
Capítulo 34 – Introdução à Programação Multiplataforma com Software Livre	349
34.1 Termos e conceitos utilizados	350
34.2 O básico do GNU/Linux, Unix	351
34.3 Diretórios úteis para programadores	362
34.4 Programas úteis para programadores	362

Capítulo 35 – Edição de texto com emacs	364
35.1 Introdução ao editor emacs	364
35.2 Comandos do editor emacs	365
Capítulo 36 – Os programas diff, patch, indent	369
36.1 O programa diff	369
36.2 O programa patch	372
36.3 O programa indent	372
Capítulo 37 – Compilando com gcc e g++	375
37.1 Protótipo e parâmetros do gcc/g++	375
37.2 Arquivos gerados pelo gcc/g++	376
37.3 Exemplo de uso do gcc/g++	376
37.4 Código obsoleto	377
37.5 Sentenças para o compilador da GNU	377
Capítulo 38 – Make	378
38.1 Protótipo e parâmetros do make	378
38.2 Um arquivo de projeto	379
38.3 Formato de um arquivo Makefile	379
38.4 Exemplo de um arquivo Makefile	380
38.5 Sentenças para o make	381
Capítulo 39 – Bibliotecas	382
39.1 Introdução à montagem de bibliotecas	382
39.2 Convenção para nomes de bibliotecas	385
39.3 Bibliotecas usuais	385
39.4 Montando uma biblioteca estática (libNome.a)	386
39.5 Montando uma biblioteca dinâmica (libNome.so)	387
39.6 Sentenças para bibliotecas	389
Capítulo 40 – Libtool	390
40.1 Introdução ao libtool	390
40.2 Forma de uso do libtool	390
40.3 Compilando com o libtool	391
40.4 Criando uma biblioteca estática com o libtool	392
40.5 Criando uma biblioteca dinâmica com o libtool	392
40.6 Linkando programas executáveis	393
40.7 Comparando o tamanho dos executáveis e os tempos de processamento	395
40.8 Sentenças para o libtool	395
Capítulo 41 – Como Evitar Bugs e Utilizar o Debugger da GNU (gdb)	396
41.1 O que é um bug?	396
41.3 Utilização de assert	397
41.4 Sentenças para evitar bugs	397
41.5 O debugger da GNU – gdb	399

Capítulo 42 – Otimização de Programas e o Profiler da GNU	404
42.1 Sentenças para obter um programa otimizado	404
42.2 Profiler (gprof)	406
Capítulo 43 – Versão de Depuração e Final e Formas de Distribuição do Programa	409
43.1 Versões debug e release	409
43.2 Distribuição dos programas e bibliotecas	409
Capítulo 44 – Documentação de Programas Utilizando JAVA_DOC e doxygen	413
44.1 Introdução à documentação de programas	413
44.2 Documentação embutida no código com JAVA_DOC	413
44.3 Tutorial de configuração e utilização do doxygen	418
44.4 Documentação profissional com sgml/xml	423
Capítulo 45 – Portabilidade	425
45.1 Introdução à portabilidade	425
45.2 Sentenças para desenvolver programas portáveis	425
Capítulo 46 – Montagem de um Programa Multiplataforma com Ferramentas da GNU	427
46.1 Introdução à programação multiplataforma com as ferramentas da GNU	427
46.2 autoscan	429
46.3 aclocal	430
46.4 ifnames	431
46.5 autoheader	431
46.6 automake	432
46.7 autoconf	436
46.8 autoupdate	445
46.9 autoreconf	446
46.10 configure	446
46.11 Como incluir instruções do programa libtool em seu pacote GNU	446
46.11 Exemplo completo	448
Capítulo 47 – Introdução ao Controle de Versões com o cvs	451
47.1 O que é o cvs?	451
47.2 Comandos do cvs	452
47.3 Sequência de trabalho	453
47.4 Versões, tags e releases	460
47.5 Para verificar diferenças entre arquivos	465
47.6 Verificando o estado do repositório	465
47.7 Ramos e misturas (branching and merging)	468
47.8 Configuração do cvs no sistema cliente-servidor	469
47.9 Como baixar programas de terceiros usando o cvs	470
47.10 Frontends (interfaces gráficas)	471
47.11 Sentenças para o cvs	472
47.12 Um diagrama com os comandos do cvs	473

Parte V – Cluster de Computadores e Processamento Paralelo	475
Capítulo 48 – Introdução ao Processamento Paralelo em um Cluster de Computadores	477
48.1 Introdução à utilização de clusters de computadores	477
48.2 Introdução ao Processamento Paralelo	482
48.3 Resumo deste capítulo	487
Capítulo 49 – Introdução ao Processamento Paralelo com Múltiplos Processos	488
49.1 Introdução aos processos	488
49.2 Estrutura e características de um processo	488
49.3 Estados de um processo	489
49.4 Processos sincronizados, recursos e dados	490
49.5 Funcionamento de um programa com múltiplos processos	490
49.6 Exemplo de código comum	492
49.7 Exemplo de código usando múltiplos processos	493
Capítulo 50 – Introdução ao Processamento Paralelo com Múltiplas Threads	497
50.1 Introdução às threads	497
50.2 Estrutura e características de uma thread	498
50.3 Estados de uma thread	499
50.4 Tipos de threads	499
50.5 Cooperação e sincronização	499
50.6 Funções básicas para criar threads	502
50.7 Roteiro básico para desenvolvimento de um programa com threads	504
50.8 Exemplo de código usando múltiplas threads	505
50.9 A biblioteca common C++	506
50.10 Sentenças sobre o uso de threads e processos	512
Capítulo 51 – Exemplo de cluster	513
51.1 Sobre o cluster	513
51.2 Comparação dos tempos de processamento	513
51.3 Como se aprofundar	514
51.4 Glossário para processamento paralelo	515
Parte VI – Apêndices	517
Apêndice A – Diretrizes de Pré-Processador	519
A.1 Introdução às diretrizes de pré-processador	519
A.2 Compilação condicional	519
Apêndice B – Conceitos úteis para Programação em C	522
B.1 Classes de armazenamento	522
B.2 Modificadores de acesso	523
B.3 Escopo das variáveis	524
B.4 Sentenças para classes de armazenamento, escopo e modificadores de acesso	526
Apêndice C – Operadores	527
C.1 Introdução aos operadores	527

Sumário

C.2 Operadores de uso geral	527
C.3 Operadores de uso específico	530
Apêndice D – Estruturas de Controle	534
D.1 if	534
D.2 if...else	534
D.3 if...else if...else if...else	535
D.4 switch...case	535
D.5 expressão ? ação_verdadeira : ação_falsa;	536
D.6 for (inicialização ; teste ; ação1) ação2;	536
D.7 while (teste) {ação;}	537
D.8 do {ação} while (teste);	537
D.9 break	537
D.10 continue	538
D.11 Sentenças para estruturas de controle	542
Apêndice E – Funções	543
E.1 A função main() e a entrada na linha de comando	543
E.2 Funções recursivas	544
E.3 Uso de elipse ... em funções	545
E.4 Sentenças para funções	546
E.5 Exemplos de funções	546
Apêndice F – Ponteiros – Parte II	550
F.1 Operações com ponteiros (+/-)	550
F.2 Ponteiro void	551
F.3 Ponteiro para ponteiro 3	551
Apêndice G – Estruturas, Uniões e Enumerações	555
G.1 Introdução às estruturas (struct)	555
G.2 Uniões (union)	559
G.3 Enumerações (enumerated)	560
Apêndice H – Glossário	562
Apêndice I – Links para Sites em C++	572
Apêndice J – Arquivos de cabeçalho	576
J.1 Arquivos de cabeçalho	576
Bibliografia	578