

Conteúdo

Prefácio	xi
1 Introdução à Programação	1
1.1 Computadores	1
1.2 Programação	2
1.3 Algoritmos: resolvendo problemas	3
1.3.1 Regras do jogo	4
1.3.2 Desenvolvimento e demonstração de correcção	10
1.4 Programas	16
1.5 Resumo: resolução de problemas	17
2 Conceitos básicos de programação	19
2.1 Introdução	19
2.1.1 Consola e canais	22
2.1.2 Definição de variáveis	25
2.1.3 Controlo de fluxo	26
2.2 Variáveis	27
2.2.1 Memória e inicialização	27
2.2.2 Nomes de variáveis	29
2.2.3 Inicialização de variáveis	29
2.3 Tipos básicos	30
2.3.1 Tipos aritméticos	31
2.3.2 Booleanos ou lógicos	37
2.3.3 Caracteres	37
2.4 Valores literais	39

2.5	Constantes	40
2.6	Expressões e operadores	43
2.6.1	Operadores aritméticos	44
2.6.2	Operadores relacionais e de igualdade	46
2.6.3	Operadores lógicos	47
2.6.4	Operadores <i>bit-a-bit</i>	48
2.6.5	Operadores de atribuição	49
2.6.6	Operadores de incrementação e decrementação	51
2.6.7	Precedência e associatividade	51
2.6.8	Efeitos laterais e mau comportamento	53
2.6.9	Ordem de cálculo	54
3	Modularização: funções e procedimentos	57
3.1	Introdução à modularização	57
3.2	Funções e procedimentos	60
3.2.1	Abordagens descendente e ascendente	61
3.2.2	Definição de rotinas	68
3.2.3	Sintaxe das definições de funções	71
3.2.4	Contrato e documentação de uma rotina	72
3.2.5	Integração da função no programa	74
3.2.6	Sintaxe e semântica da invocação ou chamada	75
3.2.7	Parâmetros	76
3.2.8	Argumentos	77
3.2.9	Retorno e devolução	77
3.2.10	Significado de <code>void</code>	77
3.2.11	Passagem de argumentos por valor e por referência	79
3.2.12	Variáveis locais e globais	87
3.2.13	Blocos de instruções ou instruções compostas	87
3.2.14	Âmbito ou visibilidade de variáveis	88
3.2.15	Duração ou permanência de variáveis	91
3.2.16	Nomes de rotinas	92
3.2.17	Declaração vs. definição	93
3.2.18	Parâmetros constantes	97

3.2.19	Instruções de asserção	100
3.2.20	Melhorando módulos já produzidos	110
3.3	Rotinas recursivas	113
3.3.1	Mecanismo de invocação de rotinas	115
3.4	Sobrecarga de nomes	122
4	Controlo do fluxo dos programas	125
4.1	Instruções de selecção	125
4.1.1	As instruções <code>if</code> e <code>if else</code>	126
4.1.2	Instruções de selecção encadeadas	129
4.1.3	Problemas comuns	132
4.2	Asserções	133
4.2.1	Dedução de asserções	133
4.2.2	Predicados mais fortes e mais fracos	135
4.2.3	Dedução da pré-condição mais fraca de uma atribuição	135
4.2.4	Asserções em instruções de selecção	137
4.3	Desenvolvimento de instruções de selecção	142
4.3.1	Escolha das instruções alternativas	143
4.3.2	Determinação das pré-condições mais fracas	143
4.3.3	Determinação das guardas	144
4.3.4	Verificação das guardas	145
4.3.5	Escolha das condições	145
4.3.6	Alterando a solução	146
4.3.7	Metodologia	147
4.3.8	Discussão	148
4.3.9	Outro exemplo de desenvolvimento	151
4.4	Variantes das instruções de selecção	154
4.4.1	O operador <code>? :</code>	154
4.4.2	A instrução <code>switch</code>	155
4.5	Instruções de iteração	159
4.5.1	A instrução de iteração <code>while</code>	159
4.5.2	Variantes do ciclo <code>while</code> : <code>for</code> e <code>do while</code>	162
4.5.3	Exemplo simples	168

4.5.4	return, break, continue, e goto em ciclos	170
4.5.5	Problemas comuns	175
4.6	Asserções com quantificadores	176
4.6.1	Somas	177
4.6.2	Produtos	178
4.6.3	Conjunções e o quantificador universal	179
4.6.4	Disjunções e o quantificador existencial	180
4.6.5	Contagens	181
4.6.6	O resto da divisão	182
4.7	Desenvolvimento de ciclos	182
4.7.1	Noção de invariante	186
4.7.2	Correcção de ciclos	191
4.7.3	Melhorando a função potência()	194
4.7.4	Metodologia de Dijkstra	196
4.7.5	Um exemplo	209
4.7.6	Outro exemplo	221
5	Matrizes, vectores e outros agregados	227
5.1	Matrizes clássicas do C++	228
5.1.1	Definição de matrizes	229
5.1.2	Indexação de matrizes	230
5.1.3	Inicialização de matrizes	232
5.1.4	Matrizes multidimensionais	233
5.1.5	Matrizes constantes	234
5.1.6	Matrizes como parâmetros de rotinas	234
5.1.7	Restrições na utilização de matrizes	239
5.2	Vectores	240
5.2.1	Definição de vectores	241
5.2.2	Indexação de vectores	241
5.2.3	Inicialização de vectores	242
5.2.4	Métodos	243
5.2.5	Acesso aos itens de vectores	243
5.2.6	Alteração da dimensão de um vector	244

5.2.7	Inserção e remoção de itens	245
5.2.8	Vectores multidimensionais?	246
5.2.9	Vectores constantes	247
5.2.10	Vectores como parâmetros de rotinas	247
5.2.11	Passagem de argumentos por referência constante	249
5.2.12	Outras operações com vectores	253
5.3	Algoritmos com matrizes e vectores	254
5.3.1	Soma dos elementos de uma matriz	254
5.3.2	Soma dos itens de um vector	258
5.3.3	Índice do maior elemento de uma matriz	258
5.3.4	Índice do maior item de um vector	264
5.3.5	Elementos de uma matriz num intervalo	264
5.3.6	Itens de um vector num intervalo	269
5.3.7	Segundo elemento de uma matriz com um dado valor	270
5.3.8	Segundo item de um vector com um dado valor	278
5.4	Cadeias de caracteres	280
5.4.1	Cadeias de caracteres clássicas	280
5.4.2	A classe <code>string</code>	283
6	Classes e enumerações	289
6.1	Enumerações ou tipos enumerados	290
6.1.1	Sobrecarga de operadores	292
6.2	Agregados heterogéneos	293
6.2.1	Definição de classes	294
6.2.2	Seleccção de membros	294
6.2.3	Alguma nomenclatura	295
6.2.4	Operações suportadas	296
6.2.5	Membros públicos e privados	297
6.3	Classes	297
6.3.1	Representação de racionais por fracções	297
6.3.2	Implementação do conceito de racional	301
6.3.3	Corrigindo a classe <code>Racional</code>	308
6.3.4	Usando sobrecarga de operadores	311

6.3.5	Construtores: conversões implícitas e valores literais	312
6.3.6	Exercícios	313
6.4	Desenho de classes	314
6.4.1	Conceito de pilha	315
6.4.2	Concretização em C++	316
6.4.3	O programa completo	319
6.4.4	Condição invariante de instância	319
6.5	Mais sobre classes em C++	322
6.5.1	Funções e procedimentos inline	322
6.5.2	Funções e procedimentos membro constantes	324
6.5.3	Referências	326
6.5.4	Destrutores	329
6.5.5	Membros de classe	329
6.5.6	Funções amigas	330
6.5.7	Construtores por omissão e matrizes de classes	332
6.5.8	Conversões para outros tipos	334
6.5.9	Novamente os racionais	335
7	Programação baseada em objectos	339
8	Modularização de alto nível	341
8.1	Modularização física	343
8.1.1	Constituição de um módulo	344
8.2	Fases da construção do ficheiro executável	344
8.2.1	Pré-processamento	345
8.2.2	Compilação	352
8.2.3	Fusão	354
8.2.4	Arquivos	357
8.3	Ligação dos nomes	360
8.3.1	Vantagens de restringir a ligação dos nomes	361
8.3.2	Espaços nominativos sem nome	364
8.3.3	Ligação de rotinas, variáveis e constantes	366
8.3.4	Ligação de classes e tipos enumerados	373

8.4	Conteúdo dos ficheiros de interface e implementação	377
8.4.1	Relação entre interface e implementação	378
8.4.2	Ferramentas de utilidade interna ao módulo	378
8.4.3	Rotinas não-membro	379
8.4.4	Variáveis globais	379
8.4.5	Constantes globais	379
8.4.6	Tipos enumerados não-membro	380
8.4.7	Classes não-membro	381
8.4.8	Métodos (rotinas membro)	381
8.4.9	Variáveis e constantes membro de instância	382
8.4.10	Variáveis membro de classe	382
8.4.11	Constantes membro de classe	383
8.4.12	Classes membro (embutidas)	383
8.4.13	Enumerados membro	384
8.4.14	Evitando erros devido a inclusões múltiplas	385
8.4.15	Ficheiro auxiliar de implementação	387
8.5	Construção automática do ficheiro executável	387
8.6	Modularização em pacotes	394
8.6.1	Colisão de nomes	394
8.6.2	Espaços nominativos	396
8.6.3	Directivas de utilização	397
8.6.4	Declarações de utilização	399
8.6.5	Espaços nominativos e modularização física	400
8.6.6	Ficheiros de interface	401
8.6.7	Ficheiros de implementação e ficheiros auxiliares de implementação . . .	402
8.6.8	Pacotes e espaços nominativos	402
8.7	Exemplo final	403
9	Listas e iteradores	409
9.1	Listas	409
9.1.1	Operações com listas	410
9.2	Iteradores	413
9.2.1	Operações com iteradores	414

9.2.2	Operações se podem realizar com iteradores e listas	415
9.2.3	Itens fictícios	415
9.2.4	Operações que invalidam os iteradores	419
9.2.5	Conclusão	419
9.3	Interface	420
9.3.1	Interface de <code>ListaInt</code>	420
9.3.2	Interface de <code>ListaInt::Iterador</code>	425
9.3.3	Usando a interface das novas classes	428
9.3.4	Teste dos módulos	430
9.4	Implementação simplista	431
9.4.1	Implementação de <code>ListaInt</code>	432
9.4.2	Implementação de <code>ListaInt::Iterador</code>	434
9.4.3	Implementação dos métodos públicos de <code>ListaInt</code>	437
9.4.4	Implementação dos métodos públicos de <code>ListaInt::Iterador</code>	441
9.5	Uma implementação mais eficiente	442
9.5.1	Cadeias ligadas	443
10	Ponteiros e variáveis dinâmicas	447
11	Herança e polimorfismo	449
12	Programação genérica	451
13	Excepções e tratamento de erros	453
A	Notação e símbolos	471
A.1	Notação e símbolos	471
A.2	Abreviaturas e acrónimos	474
B	Um pouco de lógica	475
C	Curiosidades e fenómenos estranhos	477
C.1	Inicialização	477
C.1.1	Inicialização de membros	478
C.2	Rotinas locais	478
C.3	Membros acessíveis "só para leitura"	479
C.4	Variáveis virtuais	481

<i>CONTEÚDO</i>	ix
D Nomes e seu formato: recomendações	489
E Palavras-chave do C++	491
F Precedência e associatividade no C++	495
G Tabelas de codificação ISO-8859-1 (Latin-1) e ISO-8859-15 (Latin-9)	499
H Listas e iteradores: listagens	505
H.1 Versão simplista	505
H.1.1 Ficheiro de interface: <code>lista_int.H</code>	505
H.1.2 Ficheiro de implementação auxiliar: <code>lista_int_impl.H</code>	510
H.1.3 Ficheiro de implementação: <code>lista_int.C</code>	515
I Listas e iteradores seguros	523

