

<b>Prefazione</b>	<b>xiii</b>
<b>Introduzione</b>	<b>xv</b>
<b>1 Macchine astratte</b>	<b>1</b>
1.1 La nozione di macchina astratta e l'interprete . . . . .	1
1.1.1 Interprete . . . . .	3
1.1.2 Un esempio di macchina astratta: la macchina hardware . . . . .	6
1.2 Implementazione di un linguaggio . . . . .	9
1.2.1 Realizzazione di una macchina astratta . . . . .	10
1.2.2 Implementazione: il caso ideale . . . . .	13
1.2.3 Implementazione: il caso reale e la macchina intermedia . . . . .	18
1.3 Gerarchie di macchine astratte . . . . .	21
1.4 Sommario del capitolo . . . . .	24
1.5 Nota bibliografica . . . . .	25
1.6 Esercizi . . . . .	25
<b>2 Descrivere un linguaggio di programmazione</b>	<b>27</b>
2.1 Livelli di descrizione . . . . .	27
2.2 Grammatica e sintassi . . . . .	28
2.2.1 Grammatiche libere . . . . .	30
2.3 Vincoli sintattici contestuali . . . . .	40
2.4 Compilatori . . . . .	42
2.5 Semantica . . . . .	47
2.6 Pragmatica . . . . .	54
2.7 Implementazione . . . . .	55
2.8 Sommario del capitolo . . . . .	55
2.9 Nota bibliografica . . . . .	55
2.10 Esercizi . . . . .	56
<b>3 Fondamenti</b>	<b>57</b>
3.1 Il problema della fermata . . . . .	57
3.2 Espressività dei linguaggi di programmazione . . . . .	59
3.2.1 Formalismi per la calcolabilità . . . . .	60
3.3 Esistono più funzioni che algoritmi . . . . .	62
3.4 Sommario del capitolo . . . . .	64

---

3.5	Nota bibliografica . . . . .	64
3.6	Esercizi . . . . .	64
<b>4</b>	<b>I nomi e l'ambiente</b>	<b>67</b>
4.1	Nomi e oggetti denotabili . . . . .	67
4.1.1	Oggetti denotabili . . . . .	69
4.2	Ambiente e blocchi . . . . .	70
4.2.1	I blocchi . . . . .	72
4.2.2	Tipi di ambiente . . . . .	72
4.2.3	Operazioni sull'ambiente . . . . .	75
4.3	Regole di scope . . . . .	78
4.3.1	Scope statico . . . . .	79
4.3.2	Scope dinamico . . . . .	81
4.3.3	Alcuni problemi di scope . . . . .	84
4.4	Sommario del capitolo . . . . .	87
4.5	Nota bibliografica . . . . .	88
4.6	Esercizi . . . . .	88
<b>5</b>	<b>La gestione della memoria</b>	<b>93</b>
5.1	Tecniche di gestione della memoria . . . . .	93
5.2	Gestione statica della memoria . . . . .	95
5.3	Gestione dinamica mediante pila . . . . .	96
5.3.1	Record di attivazione per i blocchi in-line . . . . .	98
5.3.2	Record di attivazione per le procedure . . . . .	100
5.3.3	Gestione della pila . . . . .	102
5.4	Gestione dinamica mediante heap . . . . .	104
5.4.1	Blocchi di dimensione fissa . . . . .	105
5.4.2	Blocchi di dimensione variabile . . . . .	106
5.5	Implementazione delle regole di scope . . . . .	109
5.5.1	Scope statico: la catena statica . . . . .	109
5.5.2	Scope statico: il display . . . . .	114
5.5.3	Scope dinamico: lista di associazioni e CRT . . . . .	116
5.6	Sommario del capitolo . . . . .	121
5.7	Nota bibliografica . . . . .	121
5.8	Esercizi . . . . .	122
<b>6</b>	<b>Strutturare il controllo</b>	<b>125</b>
6.1	Le espressioni . . . . .	125
6.1.1	Sintassi delle espressioni . . . . .	126
6.1.2	Semantica delle espressioni . . . . .	129
6.1.3	Valutazione delle espressioni . . . . .	131
6.2	La nozione di comando . . . . .	136
6.2.1	La variabile . . . . .	137
6.2.2	L'assegnamento . . . . .	138
6.3	Comandi per il controllo di sequenza . . . . .	142
6.3.1	Comandi per il controllo di sequenza esplicito . . . . .	143

---

6.3.2	Comandi condizionali . . . . .	146
6.3.3	Comandi iterativi . . . . .	150
6.4	Programmazione strutturata . . . . .	157
6.5	La ricorsione . . . . .	159
6.5.1	La ricorsione in coda . . . . .	162
6.5.2	Ricorsione o iterazione? . . . . .	167
6.6	Sommario del capitolo . . . . .	168
6.7	Nota bibliografica . . . . .	168
6.8	Esercizi . . . . .	169
<b>7</b>	<b>Astrarre sul controllo</b> . . . . .	<b>171</b>
7.1	Sottoprogrammi . . . . .	172
7.1.1	Astrazione funzionale . . . . .	174
7.1.2	Passaggio dei parametri . . . . .	175
7.2	Funzioni di ordine superiore . . . . .	186
7.2.1	Funzioni come parametro . . . . .	187
7.2.2	Funzioni come risultato . . . . .	193
7.3	Eccezioni . . . . .	195
7.3.1	Implementare le eccezioni . . . . .	200
7.4	Sommario del capitolo . . . . .	202
7.5	Nota bibliografica . . . . .	203
7.6	Esercizi . . . . .	204
<b>8</b>	<b>Strutturare i dati</b> . . . . .	<b>207</b>
8.1	Tipi di dato . . . . .	207
8.1.1	Tipi come supporto all'organizzazione concettuale . . . . .	208
8.1.2	Tipi per la correttezza . . . . .	209
8.1.3	Tipi e implementazione . . . . .	210
8.2	Sistemi di tipi . . . . .	211
8.2.1	Controlli statici e dinamici . . . . .	212
8.3	Tipi scalari . . . . .	214
8.3.1	Booleani . . . . .	215
8.3.2	Caratteri . . . . .	215
8.3.3	Interi . . . . .	215
8.3.4	Reali . . . . .	216
8.3.5	Virgola fissa . . . . .	216
8.3.6	Complessi . . . . .	216
8.3.7	Void . . . . .	217
8.3.8	Enumerazioni . . . . .	217
8.3.9	Intervalli . . . . .	218
8.3.10	Tipi ordinali . . . . .	219
8.4	Tipi composti . . . . .	219
8.4.1	Record . . . . .	219
8.4.2	Record varianti e unioni . . . . .	221
8.4.3	Array . . . . .	226
8.4.4	Insiemi . . . . .	232

---

8.4.5	Puntatori . . . . .	232
8.4.6	Tipi ricorsivi . . . . .	238
8.4.7	Funzioni . . . . .	240
8.5	Equivalenza . . . . .	241
8.5.1	Equivalenza per nome . . . . .	242
8.5.2	Equivalenza strutturale . . . . .	242
8.6	Compatibilità e conversione . . . . .	245
8.7	Polimorfismo . . . . .	248
8.7.1	Overloading . . . . .	249
8.7.2	Polimorfismo universale parametrico . . . . .	250
8.7.3	Polimorfismo universale di sottotipo . . . . .	252
8.7.4	Cenni sull'implementazione . . . . .	254
8.8	Controllo e inferenza di tipo . . . . .	255
8.9	Sicurezza: un bilancio . . . . .	257
8.10	Evitare i dangling reference . . . . .	258
8.10.1	Tombstone . . . . .	258
8.10.2	Lucchetti e chiavi . . . . .	260
8.11	Garbage collection . . . . .	260
8.11.1	Contatori dei riferimenti . . . . .	263
8.11.2	Mark and sweep . . . . .	264
8.11.3	Intermezzo: rovesciare i puntatori . . . . .	265
8.11.4	Mark and compact . . . . .	267
8.11.5	Copia . . . . .	268
8.12	Sommario del capitolo . . . . .	270
8.13	Nota bibliografica . . . . .	271
8.14	Esercizi . . . . .	271
<b>9</b>	<b>Astrarre sui dati</b> . . . . .	<b>275</b>
9.1	Tipi di dato astratti . . . . .	276
9.2	Nascondere l'informazione . . . . .	279
9.2.1	Indipendenza dalla rappresentazione . . . . .	282
9.3	Moduli . . . . .	282
9.4	Sommario del capitolo . . . . .	284
9.5	Nota bibliografica . . . . .	286
9.6	Esercizi . . . . .	286
<b>10</b>	<b>Il paradigma orientato agli oggetti</b> . . . . .	<b>287</b>
10.1	Limiti dei tipi di dato astratti . . . . .	287
10.1.1	Un primo bilancio . . . . .	291
10.2	Concetti fondamentali . . . . .	291
10.2.1	Oggetti . . . . .	292
10.2.2	Classi . . . . .	294
10.2.3	Incapsulamento . . . . .	296
10.2.4	Sottotipi . . . . .	298
10.2.5	Ereditarietà . . . . .	303
10.2.6	Selezione dinamica dei metodi . . . . .	308

---

10.3	Aspetti implementativi . . . . .	312
10.3.1	Ereditarietà singola . . . . .	314
10.3.2	Il problema della classe base fragile . . . . .	316
10.3.3	Selezione dinamica dei metodi nella JVM . . . . .	317
10.3.4	Ereditarietà multipla . . . . .	320
10.4	Polimorfismo e generici . . . . .	326
10.4.1	Polimorfismo di sottotipo . . . . .	326
10.4.2	Generici in Java . . . . .	329
10.4.3	Implementazione dei generici in Java . . . . .	333
10.4.4	Generici, array e gerarchia di sottotipo . . . . .	334
10.4.5	Overriding covarianti e controvarianti . . . . .	336
10.5	Sommario del capitolo . . . . .	339
10.6	Nota bibliografica . . . . .	340
10.7	Esercizi . . . . .	340
<b>11</b>	<b>Il paradigma funzionale</b> . . . . .	<b>343</b>
11.1	Computazioni senza stato . . . . .	343
11.1.1	Espressioni e funzioni . . . . .	345
11.1.2	Computazione come riduzione . . . . .	347
11.1.3	Gli ingredienti fondamentali . . . . .	347
11.2	Valutazione . . . . .	349
11.2.1	Valori . . . . .	350
11.2.2	Sostituzione senza cattura . . . . .	350
11.2.3	Strategie di valutazione . . . . .	351
11.2.4	Confronto tra strategie . . . . .	353
11.3	Programmare in un linguaggio funzionale . . . . .	355
11.3.1	Ambiente locale . . . . .	355
11.3.2	Interattività . . . . .	355
11.3.3	Tipi . . . . .	356
11.3.4	Pattern matching . . . . .	357
11.3.5	Oggetti infiniti . . . . .	359
11.3.6	Aspetti imperativi . . . . .	360
11.4	Implementazione: la macchina SECD . . . . .	362
11.5	Il paradigma funzionale a confronto . . . . .	364
11.6	Fondamenti: $\lambda$ -calcolo . . . . .	367
11.7	Sommario del capitolo . . . . .	374
11.8	Nota bibliografica . . . . .	374
11.9	Esercizi . . . . .	375
<b>12</b>	<b>Il paradigma logico</b> . . . . .	<b>377</b>
12.1	Deduzione come computazione . . . . .	377
12.1.1	Un esempio . . . . .	378
12.2	Sintassi . . . . .	382
12.2.1	Il linguaggio della logica del prim'ordine . . . . .	382
12.2.2	I programmi logici . . . . .	384
12.3	Teoria dell'unificazione . . . . .	385

12.3.1	La variabile logica . . . . .	386
12.3.2	Sostituzione . . . . .	387
12.3.3	L'unificatore più generale . . . . .	390
12.3.4	Un algoritmo di unificazione . . . . .	392
12.4	Il modello computazionale . . . . .	395
12.4.1	L'universo di Herbrand . . . . .	395
12.4.2	Interpretazione dichiarativa e procedurale . . . . .	396
12.4.3	La chiamata di procedura . . . . .	397
12.4.4	Controllo: il non determinismo . . . . .	401
12.4.5	Alcuni esempi . . . . .	404
12.5	Estensioni . . . . .	407
12.5.1	PROLOG . . . . .	407
12.5.2	Programmazione logica e basi di dati . . . . .	411
12.5.3	Programmazione logica con vincoli . . . . .	413
12.6	Pregi e difetti del paradigma logico . . . . .	415
12.7	Sommario del capitolo . . . . .	417
12.8	Nota bibliografica . . . . .	418
12.9	Esercizi . . . . .	418
<b>13</b>	<b>Una breve panoramica storica</b>	<b>421</b>
13.1	Gli inizi . . . . .	421
13.2	Fattori evolutivi dei linguaggi . . . . .	424
13.3	Anni '50 e '60 . . . . .	425
13.4	Anni '70 . . . . .	429
13.5	Anni '80 . . . . .	434
13.6	Anni '90 . . . . .	437
13.7	Sommario del capitolo . . . . .	440
13.8	Nota bibliografica . . . . .	440
	<b>Bibliografia</b>	<b>443</b>
	<b>Indice analitico</b>	<b>449</b>