

Indice

Prefazione	IX
Introduzione	XV
1 Definizione e implementazione dei linguaggi di programmazione	1
1.1 Analisi sintattica	1
1.1.1 Stringhe e linguaggi	1
1.1.2 Grammatiche libere dal contesto	4
1.1.3 Alberi di derivazione	7
1.1.4 Ambiguità	9
1.1.5 Algebre sintattiche e sintassi astratta	13
1.1.6 Grammatiche come definizioni induttive	16
1.1.7 Grammatiche ed espressioni regolari	16
1.1.8 Analisi lessicale e scanner per FL	22
1.1.9 Rappresentazione interna	27
1.1.10 Schema di parsing e parser per FL	32
1.2 Type checking	39
1.2.1 Vincoli contestuali	39
1.2.2 Semantica statica di FL	44
1.2.3 Type checker per FL	48
1.3 Esecuzione	53
1.3.1 Semantica dinamica di FL	53
1.3.2 Interprete per FL	57
1.4 Esercizi riassuntivi	62
1.5 Note bibliografiche	64
2 Il paradigma object-oriented	65
2.1 Introduzione	65
2.1.1 Definizione e motivazioni	65

2.1.2	Note storiche	68
2.2	Oggetti e classi	69
2.2.1	Classi	69
2.2.2	Riferimento all'oggetto ricevente	73
2.2.3	Confronto tra il paradigma object-oriented e il paradigma ADT	75
2.2.4	Uguaglianza tra oggetti	77
2.2.5	Copia di oggetti	81
2.2.6	Clientship, ricorsione tra classi e confronto tra oggetti e puntatori	82
2.2.7	Accessibilità nelle classi clienti	88
2.2.8	Inizializzazione e deallocazione di oggetti	90
2.2.9	Variabili e metodi di classe	94
2.2.10	Un esempio di algoritmo "a oggetti"	98
2.3	Ereditarietà e sottotipo	100
2.3.1	Classi eredi	100
2.3.2	Ridefinizione e binding dinamico	102
2.3.3	Sottotipo	105
2.3.4	Tipi statici e dinamici	106
2.3.5	Interfacce e classi astratte	109
2.3.6	Ereditarietà e programmazione "per casi"	112
2.3.7	Esempi di classi che codificano tipi primitivi	123
2.3.8	Ereditarietà e accessibilità	128
2.3.9	Uso di <code>super</code>	128
2.3.10	Costruttori ed ereditarietà	131
2.3.11	Varianti della nozione di ereditarietà	133
2.4	Il linguaggio Java	135
2.4.1	Generalità	135
2.4.2	Array	141
2.4.3	Hiding	142
2.4.4	Regole sulla ridefinizione	144
2.4.5	Risoluzione dell'overloading	145
2.4.6	Casting	151
2.4.7	Eccezioni	155
2.4.8	Package	166
2.4.9	Accessibilità e modificatori	168
2.4.10	Inizializzatori	173
2.4.11	Classi nested	174
2.4.12	Tipi enumerazione	179
2.4.13	Cenni ai tipi generici	181

2.4.14	Cenni alle classi predefinite	189
2.4.15	Riassunto, caratteristiche non trattate, confronto con C, C++ e C#	197
2.5	Esercizi riassuntivi	200
2.6	Note bibliografiche	213
3	Paradigmi dichiarativi	215
3.1	Introduzione	215
3.1.1	Definizione e motivazioni	215
3.1.2	Note storiche	217
3.2	Il paradigma funzionale	219
3.2.1	Concetti di base	219
3.2.2	Simulazione di algoritmi imperativi	230
3.2.3	Liste	232
3.2.4	Parametri di accumulazione	235
3.2.5	Funzioni di ordine superiore su liste	237
3.2.6	Tipi user-defined	241
3.2.7	Tipi Caml e inferenza di tipo	244
3.2.8	Caratteristiche non trattate	248
3.3	Il paradigma logico	248
3.3.1	Sintassi e concetti base	248
3.3.2	Semantica dichiarativa	253
3.3.3	Semantica operativa e risoluzione SLD	261
3.3.4	Predicati su liste	275
3.3.5	Cenni ai predicati di sistema	277
3.3.6	Un esempio di applicazione	279
3.3.7	Relazioni con la logica del primo ordine	281
3.3.8	Uso dell'interprete Prolog	282
3.4	Esercizi riassuntivi	284
3.5	Note bibliografiche	291
A	Appendice tecnica	293
A.1	Nozioni base	293
A.2	Definizioni induttive	297
A.3	Algebre eterogenee	301
B	Soluzione di alcuni esercizi	305
B.1	Capitolo 1	305
B.2	Capitolo 2	313
B.3	Capitolo 3	338

Bibliografia	347
Indice Analitico	350