

Indice generale

Parte I – Le basi del linguaggio 1

Capitolo 1 – Introduzione a Java 3

1.1 Prerequisiti: concetti informatici di base	4
1.1.1 Hardware	5
1.1.1.1 Processore	5
1.1.1.2 Memoria	6
1.1.2 Software	6
1.1.2.1 Sistemi operativi	6
1.1.2.2 Tipologie di software	6
1.1.3 Linguaggi di programmazione	7
1.1.3.1 Linguaggio macchina	8
1.1.3.2 Linguaggi compilati	8
1.1.3.3 Linguaggi interpretati	10
1.2 Introduzione a Java	11
1.2.1 Cos'è Java	11
1.2.2 Caratteristiche del linguaggio Java	12
1.2.3 False convinzioni su Java	15
1.3 Il nostro primo programma	17
1.3.1 Ambienti di sviluppo	18
1.3.1.1 Soluzione classica	18
1.3.1.2 Soluzione per esperti	18
1.3.1.3 Soluzione per chi inizia	19
1.3.2 Programmiamo	20
1.3.2.1 Scrivere il codice sorgente	21
1.3.2.2 Salvare il file sorgente	26
1.3.2.3 Compilare il file sorgente	26
1.3.2.4 Eseguire il programma	27

1.4 Il nostro secondo programma	28
1.4.1 Un programma interattivo	28
1.4.2 Analisi del programma riga per riga	29
1.4.3 Testare il programma	33
1.5 Il Java Development Kit	34
1.5.1 Struttura del JDK	35
1.5.2 Guida per sviluppare passo dopo passo	36
1.5.3 Terzo programma con JDK e Blocco Note	37
1.5.4 La nostra interazione con il computer	38
Riepilogo	39

Capitolo 2 – Componenti fondamentali di un programma Java 43

2.1 Cosa significa creare un programma Java	44
2.1.1 Programmazione e paradigmi	45
2.1.2 Introduzione alla programmazione ad oggetti (OOP)	46
2.1.2.1 Astrazione	47
2.1.3 Concetti alla base della sintassi Java	48
2.1.4 Introduzione alle variabili in Java	52
2.1.4.1 Dichiarazione di una variabile	53
2.1.4.2 Assegnazione di un valore ad una variabile	54
2.1.5 Introduzione ai metodi in Java	56
2.1.5.1 Dichiarazione di un metodo	56
2.1.5.2 Invocazione di un metodo (statico)	58
2.2 Le basi della programmazione object oriented: classi ed oggetti	59
2.2.1 Classe ed oggetto: definizioni formali	60
2.2.1.1 La nostra prima vera classe	60
2.2.1.2 I nostri primi veri oggetti	62
2.2.1.3 Osservazione importante sulla classe Punto	65
2.2.1.4 Osservazione importante sulla classe TestOggettiPunto	66
2.2.1.5 La classe Auto	67
2.2.2 Ancora sui metodi in Java	69
2.2.2.1 Astrazione di un metodo	69
2.2.2.2 Invocazione di un metodo su un oggetto	72
2.2.2.3 Altre tipologie di metodi	73
2.2.3 Ancora sulle variabili in Java	77
2.2.3.1 Variabili d'istanza	77
2.2.3.2 Variabili locali	78
2.2.3.3 Parametri di un metodo	79
2.2.4 I metodi costruttori	80
2.2.4.1 Caratteristiche di un costruttore	80
2.2.4.2 Costruttore di default	83

2.3 Introduzione ai package	85
2.3.1 Le parole chiave <code>package</code> e <code>import</code>	85
2.3.2 Gestione manuale dei package	86
2.3.3 Tipico schema di un file sorgente	88
Riepilogo	88
Capitolo 3 – Stile di codifica, tipi di dato ed array	91
<hr/>	
3.1 Stile di codifica	92
3.1.1 Schema Libero e standard di formattazione	93
3.1.2 Commenti	94
3.1.2.1 Tipologie di commenti	95
3.1.2.2 Quando utilizzare i commenti	96
3.1.3 Identificatori	98
3.1.3.1 Regole per gli identificatori	98
3.1.3.2 Standard e convenzioni	101
3.2 Gestione della memoria	103
3.2.1 Byte	103
3.2.2 Memorizzazione dei numeri interi	105
3.2.2.1 Sistemi numerici	105
3.2.2.2 Metodo del complemento a due	106
3.2.3 Caratteri e simboli: standard Unicode	107
3.2.4 Numeri decimali e standard IEEE-754	108
3.3 Tipi di dati primitivi	108
3.3.1 Tipi di dati interi	109
3.3.1.1 Promozione automatica nelle espressioni	111
3.3.1.2 Cast	112
3.3.1.3 Operazioni con interi	112
3.3.2 Tipi di dati a virgola mobile, casting e promotion	114
3.3.2.1 Precisione dei numeri floating point (IEEE-754)	116
3.3.3 Underscore in tipi di dati numerici	119
3.3.4 Tipo di dato logico - booleano	119
3.3.5 Tipo di dato primitivo letterale	120
3.3.5.1 Caratteri sulla tastiera stampabili	121
3.3.5.2 Formato Unicode (notazione esadecimale)	122
3.3.5.3 Caratteri di escape speciali	122
3.4 Tipi di dati complessi: reference	124
3.4.1 Passaggio di parametri per valore	127
3.4.2 Inizializzazione delle variabili d'istanza	129
3.5 Introduzione alla libreria standard	130
3.5.1 Il comando <code>import</code>	131

3.5.2 La classe String	133
3.5.2.1 Istanziare una stringa	133
3.5.2.2 Operazioni con le stringhe	134
3.5.2.3 Immutabilità con le stringhe	135
3.5.2.4 Literal delle stringhe	136
3.5.3 La documentazione della libreria standard di Java	137
3.5.4 Lo strumento javadoc	138
3.6 Gli array in Java	139
3.6.1 Dichiarazione	139
3.6.2 Creazione	140
3.6.3 Inizializzazione	140
3.6.4 Sintassi alternative	141
3.6.5 Caratteristiche di un array	141
3.6.6 Array Multidimensionali	142
3.6.7 String args []	142
3.7 Deduzione del tipo per le variabili locali: var	144
3.7.1 Verbosità e leggibilità	144
3.7.2 Applicabilità	145
Riepilogo	147
Capitolo 4 – Operatori e gestione del flusso di esecuzione	151
<hr/>	
4.1 Operatori Java	152
4.1.1 Operatore d’assegnazione	152
4.1.2 Operatori aritmetici	153
4.1.2.1 Operatori di assegnazione composti	154
4.1.2.2 Operatori (unari) di pre e post-incremento (e decremento)	155
4.1.3 Operatori bitwise	157
4.1.4 Operatori relazionali o di confronto	157
4.1.5 Operatori logico - booleani	159
4.1.6 Concatenazione di stringhe con +	161
4.1.7 Priorità degli operatori	162
4.2 Costrutti di programmazione semplici	163
4.2.1 Il costrutto if	165
4.2.2 Il costrutto while	169
4.3 Costrutti di programmazione avanzati	170
4.3.1 Il costrutto for	170
4.3.1.1 for vs while ?	172
4.3.2 Il costrutto do	174
4.3.3 Ciclo for migliorato	175
4.3.4 L’operatore ternario	177

4.3.5 Supporto ai costrutti di programmazione break e continue	177
4.3.5.1 La parola chiave break	178
4.3.5.2 La parola chiave continue	178
4.3.5.3 Etichette (label)	178
4.4 Il nuovo switch	180
4.4.1 Perché switch è stato aggiornato	180
4.4.2 Il costrutto switch tradizionale (statement)	181
4.4.2.1 Sintassi	182
4.4.2.2 Fall through	184
4.4.2.3 Notazione freccia per switch	187
4.4.2.4 Come usare il costrutto switch	188
4.4.3 Espressione switch	189
4.4.3.1 Freccia vs due punti	191
4.4.3.2 Poly expression	192
4.4.3.3 Exhaustiveness (esaustività)	194
4.5 Programmare meglio	196
4.5.1 Approccio alla programmazione	196
4.5.2 Algoritmo	197
4.5.3 Introduzione ad UML	199
Riepilogo	202

Parte II – Programmazione Object Oriented 205

Capitolo 5 – Incapsulamento e visibilità 207

5.1 Programmazione orientata agli oggetti (OOP)	208
5.1.1 Paradigmi fondamentali	208
5.1.2 Astrazione	209
5.1.3 Riutilizzo	211
5.1.4 UML: diagramma delle classi	212
5.2 Incapsulamento	214
5.2.1 Separare l'interfaccia dall'implementazione degli oggetti	214
5.2.1.1 Case Study	215
5.2.1.2 Soluzione (implementazione dell'incapsulamento)	215
5.2.1.3 Accesso ai dati controllato	218
5.2.1.4 Evoluzione del software	219
5.2.1.5 Metodi accessor e mutator	219
5.2.2 Il reference this	220
5.2.2.1 Uso di this con variabili	220
5.2.2.2 Uso di this con metodi	222
5.2.2.3 Uso di this con i costruttori	223

5.2.3 Quando utilizzare l'incapsulamento	224
5.2.4 Come usare l'incapsulamento	225
5.2.4.1 IDE	225
5.2.4.2 Variazioni di incapsulamento	225
5.2.4.3 Record	226
5.3 Modificatori d'accesso	227
5.3.1 Modificatore public	227
5.3.2 Modificatore protected	228
5.3.3 Nessun modificatore d'accesso	230
5.3.4 Modificatore private	231
5.4 Il modificatore static	232
5.4.1 Metodi statici	233
5.4.2 Variabili statiche (di classe)	234
5.4.3 Inizializzatori statici ed inizializzatori d'istanza	237
5.4.3.1 Inizializzatore d'istanza	238
5.4.4 Static import	239
5.4.5 Quando usare static	240
5.4.6 Design pattern Singleton	240
Riepilogo	242
Capitolo 6 – Ereditarietà ed interfacce	245
<hr/>	
6.1 Ereditarietà	246
6.1.1 La parola chiave extends	246
6.1.2 Quando utilizzare l'ereditarietà: la relazione "is a"	248
6.1.3 Come usare l'ereditarietà: generalizzazione e specializzazione	249
6.1.4 La classe Object	250
6.2 Supporto del linguaggio all'ereditarietà	251
6.2.1 Rapporto ereditarietà-incapsulamento	251
6.2.2 Ereditarietà e costruttori	252
6.2.3 La parola chiave super	254
6.2.3.1 super e costruttori	254
6.2.3.2 super e metodi	257
6.2.4 Ereditarietà e modificatori	258
6.2.4.1 Modificatori d'accesso	259
6.2.4.2 Il modificatore static	261
6.2.4.3 Il modificatore final	262
6.2.4.4 Il modificatore sealed	263
6.2.5 Ereditarietà e record	264
6.3 Il modificatore abstract	265
6.3.1 Metodi astratti	266
6.3.2 Classi astratte	266

6.4 Interfacce	269
6.4.1 Definizione classica (pre-Java 8)	269
6.4.1.1 La parola chiave implements	270
6.4.2 Metodi statici	272
6.4.3 Metodi di default	273
6.4.4 Interfacce sealed	274
6.4.5 Interfacce funzionali	275
6.4.6 Ereditarietà multipla	276
6.4.6.1 Diamond Problem	277
6.4.6.2 Altri scenari	280
6.5 Differenze tra interfacce e classi astratte	282
6.5.1 Differenze tecniche	282
6.5.2 Differenza concettuale	283
Riepilogo	285
Capitolo 7 – Polimorfismo	289
7.1 Polimorfismo	290
7.1.1 Convenzione per i reference	290
7.2 Polimorfismo per metodi	292
7.2.1 Overload	292
7.2.1.1 Overload di metodi	292
7.2.1.2 Overload di costruttori	294
7.2.1.3 Ambiguità con overload	295
7.2.2 Varargs	297
7.2.2.1 Varargs vs Arrays	298
7.2.3 Override	300
7.2.3.1 Regole dell'override	302
7.2.3.2 Annotazione Override	303
7.2.3.3 Override e static	305
7.3 Polimorfismo per dati	306
7.3.1 Parametri polimorfi	308
7.3.2 Invocazione virtuale dei metodi	308
7.3.2.1 Il metodo toString	308
7.3.2.2 Come implementare il polimorfismo	311
7.3.2.3 Polimorfismo e interfacce	313
7.3.3 Collezioni eterogenee	315
7.3.3.1 L'operatore instanceof	316
7.3.3.2 Cast di oggetti	317
7.3.3.3 Pattern matching per instanceof	319
Riepilogo	320

Capitolo 8 – Eccezioni ed asserzioni	323
8.1 Eccezioni, errori ed asserzioni	323
8.2 Gerarchia di Throwable	324
8.2.1 Eccezioni Checked e Unchecked	325
8.2.2 Eccezioni più comuni	326
8.2.3 Eccezioni personalizzate	327
8.3 Meccanismo per la gestione delle eccezioni	328
8.3.1 Blocchi try - catch	328
8.3.1.1 Eccezioni come contenitori di informazioni	329
8.3.1.2 Helpfull NullPointerExceptions	330
8.3.2 Catturare le eccezioni	332
8.3.2.1 Gestire più eccezioni	332
8.3.2.2 Multi- catch	333
8.3.3 La clausola finally	335
8.3.3.1 Oggetti chiudibili	336
8.3.4 Try-with-resources	338
8.3.4.1 Interfaccia AutoCloseable	338
8.3.4.2 Variabili effettivamente non modificate	339
8.3.4.3 Eccezioni sopresse	339
8.3.5 Quando usare la gestione delle eccezioni	342
8.3.5.1 Lanciare eccezioni: la parola chiave throw	343
8.3.5.2 Propagazione delle eccezioni	345
8.3.5.3 La parola chiave throws	346
8.3.5.4 Altra regola sull'override	348
8.3.5.5 Warnings	350
8.4 Introduzione alle asserzioni	351
Riepilogo	353

Parte III – Caratteristiche avanzate 357

Capitolo 9 – Tipi Enumerazioni e Tipi Record 359

9.1 Tipi Enumerazioni	360
9.1.1 Sintassi	360
9.1.2 Enumerazioni vs classi	362
9.1.3 Ereditarietà ed enumerazioni	363
9.1.4 Polimorfismo ed enumerazioni	364
9.1.5 Metodi, variabili, costruttori in una enumerazione	365
9.1.6 Switch e enum	367

9.1.7 Quando utilizzare una enumerazione	369
9.1.8 Enumerazioni ed <code>import</code> statici	371
9.2 Tipi Record	372
9.2.1 Sintassi	373
9.2.2 Record vs classi	373
9.2.3 Ereditarietà e record	376
9.2.3.1 Il metodo <code>equals</code>	377
9.2.3.2 Il metodo <code>hashCode</code>	380
9.2.4 Polimorfismo e record	382
9.2.4.1 Override	382
9.2.4.2 Implementazione di interfacce	384
9.2.5 Metodi, variabili, costruttori in un record	388
9.2.5.1 Costruttore canonico compatto	388
9.2.5.2 Costruttore non canonico	389
9.2.6 Quando utilizzare un record	391
Riepilogo	392
Capitolo 10 – Tipi Generici e Collezioni	395
10.1 Generics e tipi parametro	396
10.2 Generics e framework Collections	400
10.2.1 La classe generica <code>ArrayList</code>	401
10.2.1.1 Metodi delle liste	403
10.2.1.2 Tipi primitivi e autoboxing-autounboxing	406
10.2.1.3 Tipi Raw	407
10.2.2 Interfaccia <code>Iterator</code>	408
10.2.3 Interfaccia <code>Map</code>	409
10.2.4 La classe <code>HashSet</code>	410
10.2.5 Ordinamento	412
10.2.5.1 L'interfaccia <code>Comparable</code>	412
10.2.5.2 L'interfaccia <code>Comparator</code>	414
10.3 Ereditarietà e tipi generici	416
10.3.1 Ereditarietà e tipo parametro	416
10.3.2 Type erasure	417
10.3.3 Wildcard	418
10.3.4 Metodi generici	420
10.3.5 Wildcard capture	420
10.3.5.1 Metodi helper	421
Riepilogo	422

Capitolo 11 – Reflection e Tipi Annotazione	425
11.1 La classe <code>Class</code> e Reflection	426
11.1.1 Istanziare un oggetto <code>Class</code>	427
11.1.2 <code>Class</code> è un tipo generico	428
11.1.3 Un esempio di reflection	429
11.1.4 Nomi dei parametri dei metodi	431
11.1.5 Quando usare la reflection	433
11.2 Definizione di annotazione (metadato)	434
11.2.1 Case study	435
11.2.2 Definizione di un tipo annotazione	436
11.2.3 Utilizzare annotazioni	438
11.2.4 Processare un'annotazione	439
11.2.5 Annotare annotazioni (meta-annotazioni)	440
11.2.5.1 La meta-annotazione <code>Retention</code>	440
11.2.5.2 La meta-annotazione <code>Target</code>	441
11.2.6 Tipologie di annotazioni e sintassi	441
11.2.6.1 Annotazione ordinaria (o completa)	442
11.2.6.2 Annotazione a valore singolo	445
11.2.6.3 Annotazione marcatrice	446
11.3 Annotazioni standard	446
11.3.1 <code>Override</code>	447
11.3.2 <code>FunctionalInterface</code>	447
11.3.3 <code>Deprecated</code>	448
11.3.4 <code>SuppressWarnings</code>	451
Riepilogo	454
Capitolo 12 – Gestione dei thread	457
12.1 Introduzione ai thread	458
12.1.1 Definizione provvisoria di thread	458
12.1.2 Cosa significa multithreading	459
12.2 La classe <code>Thread</code> e la dimensione temporale	460
12.2.1 Analisi di <code>ThreadExists</code>	461
12.2.2 L'interfaccia <code>Runnable</code> e la creazione dei thread	463
12.2.3 Analisi di <code>ThreadCreation</code>	464
12.2.4 La classe <code>Thread</code> e la creazione dei thread	469
12.3 Priorità, scheduler e sistemi operativi	470
12.3.1 Analisi di <code>ThreadRace</code>	472
12.3.2 Comportamento Windows (e la maggior parte dei sistemi Linux)	473

12.3.3 Comportamento Unix	473
12.3.4 Il modificatore <code>volatile</code>	477
12.4 Thread e sincronizzazione	478
12.4.1 Analisi di <code>Synch</code>	479
12.4.2 Monitor e lock	485
12.5 La comunicazione fra thread	486
12.5.1 Analisi di <code>IdealEconomy</code>	488
12.5.2 I metodi <code>wait</code> e <code>notify</code>	489
12.6 Concorrenza	492
12.6.1 Oggetti Immutabili	493
12.6.2 Tecnologie Java e classi standard ausiliarie	495
12.6.3 Introduzione alla libreria Java sulla concorrenza	496
12.6.3.1 Il package <code>java.util.concurrent.locks</code>	497
12.6.3.2 Il package <code>java.util.concurrent.atomic</code>	498
12.6.3.3 Interfacce <code>Executors</code>	499
12.6.3.4 La classe <code>Semaphore</code>	502
12.6.3.5 La classe <code>CyclicBarrier</code>	504
Riepilogo	506
Capitolo 13 – Espressioni lambda	509

13.1 Classi anonime	511
13.1.1 Tipi innestati: definizione	512
13.1.2 Classi anonime: definizione	513
13.1.3 Sintassi	513
13.1.4 Classi anonime vs classi interne	514
13.1.5 Classi anonime ed interfacce	515
13.1.6 Classi anonime locali e <code>var</code>	516
13.1.7 Quando usare le classi anonime	517
13.2 Espressioni lambda	522
13.2.1 Sintassi	523
13.2.2 Comprendere le espressioni lambda	524
13.2.3 Espressione lambda vs classe anonima	525
13.2.3.1 Sinteticità	525
13.2.3.2 Regole e visibilità	527
13.2.3.3 Dinamicità	529
13.2.4 Gestione delle eccezioni	531
13.2.5 Deduzione dei tipi per le espressioni lambda locali	533
13.2.6 Quando usare le espressioni lambda	533
13.3 Reference a metodi	534
13.3.1 Quando usare un reference a un metodo	535

13.3.2 Sintassi	535
13.3.2.1 Reference a un metodo statico	536
13.3.2.2 Reference a un metodo d'istanza	536
13.3.2.3 Reference a un metodo d'istanza di un certo tipo	537
13.3.2.4 Reference a un costruttore	539
13.3.3 Deduzione dei tipi per i reference a metodi	540
13.4 Le interfacce funzionali del package <code>java.util.function</code>	541
13.4.1 Predicate	542
13.4.2 Consumer	543
13.4.3 Supplier	544
13.4.4 Function	545
13.4.5 UnaryOperator e composizione di espressioni lambda	546
13.4.6 Interfacce funzionali e tipi primitivi	549
Riepilogo	549
Capitolo 14 – Stream API	553
<hr/>	
14.1 Introduzione alla libreria Stream API	554
14.2 Stream e pipeline	556
14.3 Classi Optional	558
14.3.1 Altri metodi or	560
14.3.2 Metodi get , filter , isPresent , map e flatMap	561
14.4 Stream Sources	564
14.4.1 Stream da valori (metodi of)	564
14.4.2 Stream da funzioni (metodi iterate e generate)	564
14.4.2.1 Metodi iterate , limit e skip	565
14.4.2.2 Metodo generate	566
14.4.3 Stream da array e collection	566
14.5 Aggregate Operations	567
14.5.1 Filtrare stream	567
14.5.1.1 Metodo filter	567
14.5.1.2 Metodi distinct , dropWhile e takeWhile	568
14.5.2 Fare debugging con gli stream	569
14.5.2.1 Metodo peek	570
14.5.3 Trasformare stream	570
14.5.3.1 Metodi sorted	571
14.5.3.2 Metodi map	571
14.5.3.3 Metodi flatMap	572
14.6 Terminal Operations	573

14.6.1 Operazioni di ricerca	574
14.6.1.1 Metodi <code>match</code>	574
14.6.1.2 Metodi <code>find</code>	575
14.6.2 Operazioni di riduzione	576
14.6.2.1 Metodi di riduzione semplici	576
14.6.2.2 Metodo <code>reduce</code>	576
14.6.2.3 Metodo <code>collect</code> e la classe <code>Collectors</code>	577
14.7 Stream paralleli e prestazioni	579
Riepilogo	582

Parte IV – Architettura e integrazione 585

Capitolo 15 – Input-Output 587

15.1 Introduzione all'input-output	587
15.2 Pattern Decorator	588
15.2.1 Descrizione del pattern	588
15.2.2 Esempio	589
15.3 Caratteristiche del package <code>java.io</code>	592
15.3.1 I Character Stream	593
15.3.2 I Byte Stream	595
15.3.3 Le interfacce alla base della gerarchia	597
15.3.4 Checked exception e chiusura degli stream	598
15.4 Input e output classici	600
15.4.1 Lettura di input da tastiera	601
15.4.1.1 <code>BufferedReader</code> e <code>InputStreamReader</code>	601
15.4.1.2 La classe <code>Scanner</code>	602
15.4.1.3 L'oggetto <code>Console</code>	603
15.4.2 Gestione dei file	604
15.4.3 Serializzazione di oggetti e modificatore <code>transient</code>	609
15.4.3.1 L'interfaccia <code>Serializable</code>	609
15.4.3.2 Il modificatore <code>Transient</code>	609
15.4.3.3 Esempio di serializzazione	610
15.4.3.4 Esempio di deserializzazione	610
15.4.3.5 Il campo <code>SerialVersionUID</code>	611
15.4.3.6 Personalizzare la serializzazione	613
15.4.3.7 L'interfaccia <code>Externalizable</code>	614
15.4.3.8 Codifica in Base64	617
15.5 New Input-Output	618
15.5.1 Package <code>java.nio</code>	618

15.5.2 NIO 2	620
15.5.2.1 L'interfaccia Path	620
15.5.2.2 La classe Files	623
15.6 Introduzione al Networking	628
15.6.1 Nozioni di base: Client e Server	629
15.6.2 Protocolli di rete	629
15.6.3 Conoscere l'indirizzo IP	630
15.6.4 Esempio (Socket API)	631
15.6.4.1 Server	631
15.6.4.2 Client	633
15.7 Introduzione a Java Native Interface (JNI)	634
15.7.1 Esempio	635
15.7.1.1 Creazione di file header	635
15.7.1.2 Compilazione e creazione di file header	636
15.7.1.3 Codice nativo come libreria	637
15.7.1.4 Esecuzione	637
Riepilogo	638
Capitolo 16 – Moduli	643
<hr/>	
16.1 Perché un sistema modulare?	644
16.1.1 Pro	645
16.1.2 Contro	646
16.2 Definizione di modulo	646
16.2.1 HelloModularWorld	648
16.2.2 Nozioni architetturali e di progettazione	651
16.2.2.1 Partizionamento verticale	651
16.2.2.2 Dipendenza, coesione ed accoppiamento	651
16.2.2.3 Legge di Demetra	652
16.2.2.4 Partizionamento orizzontale e framework	653
16.2.2.5 JAR e componenti software	654
16.2.3 Direttive	654
16.2.3.1 requires	655
16.2.3.2 exports	656
16.2.3.3 opens e open (forte incapsulamento)	657
16.2.3.4 uses e provides to	660
16.3 Servizi con ServiceLoader	660
16.3.1 Pattern Factory	661
16.3.2 Factory modulare	661
16.3.3 Evoluzione con ServiceLoader	665
16.3.4 Compilazione ed esecuzione	668
16.3.5 Metodo provider	669

16.4 Jar modulare	670
16.4.1 Strumento jar	671
16.4.2 Formato jmod	672
16.5 Migrazione di programmi pre-Java 9	672
16.5.1 Lo strumento jdeps	673
16.5.2 Modulo anonimo	674
16.5.3 Modulo automatico	675
16.5.4 Testare moduli esterni velocemente	675
16.6 Jlink	676
Riepilogo	677
Capitolo 17 – Java Database Connectivity	681
<hr/>	
17.1 Introduzione a JDBC	682
17.2 Le basi di JDBC	683
17.2.1 Implementazione del driver JDBC del fornitore	684
17.2.2 Implementazione dello sviluppatore (Applicazione JDBC)	686
17.2.3 Analisi dell'esempio JDBCApp	687
17.2.4 Database schema	690
17.3 Indipendenza dal database	690
17.4 Supporto a SQL	692
17.4.1 Operazioni DML	692
17.4.2 Operazioni DDL	694
17.4.3 Statement parametrizzati	695
17.4.4 Stored procedure	697
17.4.5 Mappatura dei tipi Java - SQL	697
17.5 Transazioni con JDBC	697
17.5.1 Preservare i dati con i livelli di isolamento	701
17.5.2 Savepoints	704
17.6 Evoluzione di JDBC	706
17.6.1 JDBC 2.0	708
17.6.2 JDBC 3.0	709
17.6.2.1 ResultSet aggiornabile	709
17.6.2.2 La Insert Row	710
17.6.3 JDBC 4.0	711
17.6.4 JDBC 4.1	712
17.6.5 JDBC 4.2	714
17.6.6 JDBC 4.3	714
17.7 Strati logici del software: Integrazione	714
17.7.1 Pattern Data Access Object (DAO)	715
Riepilogo	719

Capitolo 18 – Java e il mondo XML 723

18.1 JAXP: Modulo <code>java.xml</code>	724
18.1.1 Document Object Model (DOM)	726
18.1.1.1 Creare un documento DOM a partire da un file XML	726
18.1.1.2 Recuperare la lista dei nodi da un documento DOM	727
18.1.1.3 Recuperare particolari nodi	728
18.1.1.4 Modifica di un documento XML	730
18.1.2 Simple API for XML (SAX)	731
18.1.3 Streaming API for XML (StAX)	734
18.1.4 Trasformazioni con XSL	739
18.1.4.1 XPath	740
18.1.4.2 Scrivere un file con XSLT	742
18.1.4.3 Trasformare un file tramite XSLT	744
18.1.4.4 Leggere una stringa XML	747
18.2 Text block e formattazione	748
18.2.1 Introduzione	748
18.2.2 Sintassi	749
18.2.3 Normalizzazione dei terminatori di linea	750
18.2.4 Rimozione dei whitespace superflui	751
18.2.5 Interpretazione dei caratteri di escape	753
18.2.5.1 Nuovi caratteri di escape	754
18.2.6 Concatenazione di text block	755
Riepilogo	757

Parte V – Interfacce grafiche 759**Capitolo 19 – Introduzione alle Interfacce grafiche (GUI)** 761

19.1 Introduzione alla Graphical User Interface (GUI)	762
19.2 Introduzione ad Abstract Window Toolkit (AWT)	765
19.2.1 Esempi con AWT	766
19.2.1.1 Componenti grafici	767
19.2.1.2 Tipi di AWT utili anche senza AWT	768
19.2.1.3 Disegnare con Canvas	769
19.2.1.4 Composite pattern	769
19.3 Creazione di interfacce complesse con i layout manager	771
19.3.1 Il FlowLayout	772
19.3.2 Il BorderLayout	774
19.3.3 Il GridLayout	775

19.3.4 Creazione di interfacce grafiche complesse	776
19.3.5 Il <code>GridBagLayout</code>	777
19.3.6 Il <code>CardLayout</code>	778
19.4 Gestione degli eventi	780
19.4.1 Observer e Listener	780
19.4.2 Classi innestate e classi anonime	784
19.4.3 Espressioni lambda	786
19.4.4 Altri tipi di eventi	787
19.4.5 Classi Adapter	789
19.4.6 Errori classici	791
19.5 Introduzione a Swing	791
19.5.1 Swing vs AWT	792
19.5.2 Top-Level Containers	795
19.5.2.1 <code>SwingExample</code>	797
19.5.3 Concorrenza dei componenti Swing	799
19.5.3.1 Pluggable Look and Feel: <code>LookAndFeelExample</code>	800
19.5.4 Componenti grafici principali di Swing: <code>SwingMixExample</code>	803
19.5.4.1 <code>LabelsPanel</code> : classi <code>JLabel</code> e <code>ImageIcon</code>	804
19.5.4.2 <code>ButtonsPanel</code> : classe <code>JButton</code> e shortcuts	806
19.5.4.3 <code>PaintDialog</code> : classi <code>JDialog</code> , <code>JList</code> e <code>Color</code>	808
19.5.4.4 <code>TextPanel</code> : <code>JTextField</code> , <code>JTextArea</code> e <code>Border</code>	810
19.5.4.5 <code>BoxesPanel</code> : classi <code>JComboBox</code> , <code>JRadioButton</code> e <code>JCheckBox</code>	811
19.5.4.6 <code>TreePanel</code> : classe <code>JTree</code>	813
19.5.4.7 <code>TablePanel</code> : classe <code>JTable</code>	814
19.6 Altre caratteristiche delle GUI	816
19.6.1 Splash screen	816
19.6.2 Applicazioni in background: <code>SystemTrayExample</code>	816
19.6.3 Eseguire applicazioni con interfaccia grafica	817
19.6.3.1 Script di avvio	817
19.6.3.2 File JAR eseguibile	818
Riepilogo	819
Capitolo 20 – Introduzione a JavaFX	821
20.1 Storia delle GUI in Java	822
20.2 Caratteristiche di JavaFX	824
20.3 Primo programma con Java FX	825
20.3.1 Analisi di <code>FirstJFXExample</code>	826
20.3.2 Ciclo di vita di un'applicazione JavaFX	827
20.3.2.1 Il metodo <code>main</code> e argomenti da riga di comando	829

20.3.3 Esecuzione di un'applicazione JavaFX	830
20.3.3.1 Esecuzione da riga di comando non modulare	831
20.3.3.2 Creare un JAR eseguibile non modulare	832
20.3.3.3 Creare un modulo eseguibile	833
20.3.3.4 Creare un JAR modulare eseguibile	834
20.3.3.5 Creare un'immagine di runtime personalizzata	835
20.4 Interfacce complesse con i Layout	836
20.4.1 I Layout pane di JavaFX	837
20.4.1.1 Le classi VBox e HBox	837
20.4.1.2 La classe BorderPane	839
20.4.1.3 La classe GridPane	841
20.4.1.4 Altri Layout Pane	842
20.4.2 FXML	842
20.5 CSS	846
20.5.1 CSS e file FXML	846
20.5.2 CSS e file JavaFX	847
20.6 Gestione degli eventi	849
20.7 Proprietà JavaFX e Binding	849
20.7.1 Proprietà JavaFX	850
20.7.2 Binding	851
20.8 Effetti speciali	852
20.8.1 Blur	852
20.8.2 Fading	853
20.8.3 Traslazione	853
20.9 Componenti grafici avanzati	853
20.9.1 Grafico a torta	854
20.9.2 Media Player	855
20.9.3 Browser	855
Riepilogo	857
Indice degli argomenti per versione	859
<hr/>	
Indice analitico	861
<hr/>	