



Προγραμματισμός δικτυακών εφαρμογών με Java

Elliote Rusty Harold

Επιστημονική επιμέλεια ελληνικής έκδοσης:
Στέργιος Παπαδημητρίου, ΤΕΙ ΑΜΘ

Περιεχόμενα

Πρόλογος.....	15
Τι νεότερο υπάρχει στην τέταρτη έκδοση	16
Οργάνωση του βιβλίου	17
Απαιτήσεις από τους αναγνώστες	18
Εκδόσεις της Java	19
Τα παραδείγματα	19
Συμβάσεις του βιβλίου	20
Αιτήσεις για Σχόλια	22
Χρήση παραδειγμάτων κώδικα	22
Επικοινωνία	23
Ευχαριστίες.....	23
Πρόλογος ελληνικής έκδοσης	24
1 Βασικές έννοιες δικτύων	27
Δίκτυα.....	28
Τα επίπεδα ενός δικτύου.....	30
Το επίπεδο υπολογιστή υπηρεσίας-δικτύου	33
Το επίπεδο διαδικτύου	34
Το επίπεδο μεταφοράς	35
Το επίπεδο εφαρμογών	36
IP, TCP και UDP	36
Διευθύνσεις IP και ονόματα περιοχών	38
Θύρες	40
Το Διαδίκτυο	40
Μπλοκ διευθύνσεων Διαδικτύου	42
Μετάφραση διευθύνσεων δικτύου.....	42
Τείχη προστασίας	43
Διακομιστές μεσολάβησης	43
Το μοντέλο πελάτη/διακομιστή.....	45
Πρότυπα του Διαδικτύου.....	47
Έγγραφα RFC της IETF	47
Συστάσεις της W3C	51

2	Ρεύματα.....	53
	Ρεύματα εξόδου	54
	Ρεύματα εισόδου	59
	Σήμανση και επαναφορά	63
	Ρεύματα φίλτρων	64
	Αλυσιδωτή σύνδεση φίλτρων.....	65
	Προσωρινά αποθηκευμένα ρεύματα	67
	Η κλάση PrintStream.....	68
	Ρεύματα δεδομένων.....	70
	Αναγνώστες και εγγραφείς.....	73
	Εγγραφείς	74
	Η κλάση OutputStreamWriter	76
	Αναγνώστες.....	76
	Αναγνώστες και εγγραφείς φίλτρων.....	78
	Η κλάση PrintWriter.....	80
3	Νήματα	83
	Εκτέλεση νημάτων	85
	Δημιουργία υποκλάσεων της κλάσης Thread.....	86
	Υλοποίηση της διασύνδεσης Runnable	88
	Επιστροφή πληροφοριών από ένα νήμα.....	90
	Συνθήκες ανταγωνισμού.....	92
	Περίοδευση.....	93
	Επανακλήσεις	94
	Οι διασυνδέσεις Future, Callable και Executor	98
	Συγχρονισμός	100
	Συγχρονισμένα μπλοκ	102
	Συγχρονισμένες μέθοδοι.....	105
	Εναλλακτικές λύσεις αντί του συγχρονισμού	105
	Αδιέξοδο.....	107
	Χρονοπρογραμματισμός νημάτων	108
	Προτεραιότητες	109
	Προεκτοπισμός.....	110
	Δεξαμενές νημάτων και η κλάση Executors	121
4	Διευθύνσεις Διαδικτύου	125
	Η κλάση InetAddress.....	127
	Δημιουργία νέων αντικειμένων της κλάσης InetAddress.....	127
	Μέθοδοι get	132
	Τύποι διευθύνσεων	135
	Έλεγχος προσπελασιμότητας	138
	Μέθοδοι της κλάσης Object	139

Οι κλάσεις Inet4Address και Inet6Address	140
Η κλάση NetworkInterface	141
«Εργοστασιακές» μέθοδοι	141
Μέθοδοι get	143
Μερικά χρήσιμα προγράμματα	143
SpamCheck	143
Επεξεργασία αρχείων καταγραφής των διακομιστών Ιστού	145
5 URL και URI.....	151
Τα URI.....	151
Τα URL.....	155
Σχετικά URL.....	156
Η κλάση URL	157
Δημιουργία νέων URL.....	157
Ανάκτηση δεδομένων από URL	163
Διαχωρισμός URL σε τμήματα.....	169
Ισότητα και σύγκριση	174
Μετατροπή.....	175
Η κλάση URI	176
Κατασκευή ενός URI.....	176
Τμήματα του URI	179
Ανάλυση σχετικών URI.....	182
Ισότητα και σύγκριση	183
Αναπαράσταση με συμβολοσειρές.....	183
Η μορφή x-www-form-urlencoded.....	184
Η κλάση URLEncoder.....	185
Η κλάση URLDecoder.....	188
Διακομιστές μεσολάβησης	189
Ιδιότητες συστήματος	189
Η κλάση Proxy	190
Η κλάση ProxySelector	191
Επικοινωνία με προγράμματα στην πλευρά του διακομιστή μέσω της μεθόδου GET	192
Προσπέλαση ιστότοπων που προστατεύονται με κωδικό πρόσβασης	196
Η κλάση Authenticator	196
Η κλάση PasswordAuthentication	198
Η κλάση JPasswordField	199
6 Το πρωτόκολλο HTTP	203
Το πρωτόκολλο.....	203
Keep-Alive.....	211
Μέθοδοι HTTP	212
Το σώμα της αίτησης.....	215

Cookies	216
Η υποκλάση CookieManager	220
Η κλάση CookieStore	221
7 Συνδέσεις με την κλάση URLConnection	223
Άνοιγμα συνδέσεων με την κλάση URLConnection	224
Ανάγνωση δεδομένων από διακομιστή	225
Ανάγνωση της κεφαλίδας	226
Ανάκτηση συγκεκριμένων πεδίων της κεφαλίδας	227
Ανάκτηση αυθαίρετων πεδίων της κεφαλίδας	233
Κρυφή μνήμη	236
Κρυφή μνήμη Ιστού για την Java	239
Διευθέτηση της σύνδεσης	244
protected URL url	245
protected boolean connected	246
protected boolean allowUserInteraction	246
protected boolean doInput	248
protected boolean doOutput	248
protected boolean ifModifiedSince	249
protected boolean useCaches	250
Χρόνος αναμονής	251
Διαμόρφωση κεφαλίδας HTTP της αίτησης πελάτη	252
Εγγραφή δεδομένων σε διακομιστή	254
Ζητήματα ασφαλείας για συνδέσεις URLConnection	260
Εικασίες για τους τύπους μέσων MIME	260
Η κλάση HttpURLConnection	261
Η μέθοδος αιτήσεων	262
Αποσύνδεση από τον διακομιστή	266
Χειρισμός απαντήσεων του διακομιστή	267
Διακομιστές μεσολάβησης	272
Κατάσταση συνεχούς ροής	272
8 Υποδοχές για πελάτες	275
Χρήση υποδοχών	275
Διερεύνηση πρωτοκόλλων με το Telnet	276
Ανάγνωση δεδομένων από διακομιστές με υποδοχές	278
Εγγραφή δεδομένων σε διακομιστές με υποδοχές	285
Κατασκευή και σύνδεση υποδοχών	289
Βασικές κατασκευάστριες μέθοδοι	289
Επιλογή τοπικής διασύνδεσης από την οποία θα πραγματοποιηθεί η σύνδεση	291
Κατασκευή χωρίς πραγματοποίηση σύνδεσης	293
Διευθύνσεις υποδοχών	294
Διακομιστές μεσολάβησης	295

Λήψη πληροφοριών για μια υποδοχή.....	296
Κλειστή ή συνδεδεμένη;.....	297
toString().....	298
Καθορισμός τιμών για επιλογές υποδοχών	298
TCP_NODELAY	299
SO_LINGER	300
SO_TIMEOUT	300
SO_RCVBUF και SO_SNDBUF	301
SO_KEEPALIVE	303
OOBINLINE	303
SO_REUSEADDR	304
Κατηγορία υπηρεσίας (IP_TOS)	305
Εξαιρέσεις υποδοχών.....	307
Υποδοχές σε εφαρμογές με GUI	308
Whois.....	308
Μια βιβλιοθήκη πελατών δικτύου	312
9 Υποδοχές για διακομιστές.....	321
Χρήση υποδοχών ServerSocket.....	321
Χειρισμός δυαδικών δεδομένων.....	326
Πολυνηματικοί διακομιστές	327
Εγγραφή σε διακομιστές με υποδοχές	331
Κλείσιμο υποδοχών διακομιστών	333
Καταγραφή	334
Πληροφορίες για καταγραφή.....	334
Τρόποι καταγραφής.....	335
Κατασκευή υποδοχών διακομιστών	340
Κατασκευή χωρίς δέσμευση.....	342
Εξαγωγή πληροφοριών για μια υποδοχή διακομιστή.....	343
Επιλογές υποδοχών.....	344
SO_TIMEOUT	345
SO_REUSEADDR	346
SO_RCVBUF	346
Κατηγορία υπηρεσίας.....	347
Διακομιστές HTTP	348
Ένας διακομιστής ενός αρχείου.....	349
Ένας ανακατευθυντήρας.....	352
Ένας πλήρως ανεπτυγμένος διακομιστής HTTP	357
10 Ασφαλείς υποδοχές.....	363
Ασφαλείς επικοινωνίες.....	364
Δημιουργία ασφαλών υποδοχών για πελάτες.....	366
Επιλογή της σουίτας κρυπτογραφημάτων	370

Χειριστές συμβάντων	374
Διαχείριση συνόδων	375
Κατάσταση πελάτη.....	376
Δημιουργία ασφαλών υποδοχών για διακομιστές.....	377
Διευθέτηση υποδοχών SSLServerSocket.....	381
Επιλογή της σουίτας κρυπτογραφημάτων	382
Διαχείριση συνόδων	383
Κατάσταση πελάτη.....	383
11 Είσοδος/έξοδος χωρίς μπλοκάρισμα	385
Ένα παράδειγμα πελάτη	387
Ένα παράδειγμα διακομιστή	391
Περιοχές προσωρινής αποθήκευσης	398
Δημιουργία περιοχών προσωρινής αποθήκευσης	401
Γέμισμα και άδειασμα	402
Μαζικές μέθοδοι.....	404
Μετατροπή δεδομένων	404
Περιοχές προσωρινής αποθήκευσης για προβολή	407
Σύμπτυξη περιοχών προσωρινής αποθήκευσης	409
Δημιουργία αντιγράφων για περιοχές προσωρινής αποθήκευσης	412
Τεμαχισμός περιοχών προσωρινής αποθήκευσης	416
Σήμανση και επαναφορά	416
Μέθοδοι αντικειμένων.....	417
Κανάλια	418
Η κλάση SocketChannel.....	418
Η κλάση ServerSocketChannel	421
Η κλάση Channels	423
Ασύγχρονα κανάλια (Java 7).....	424
Επιλογές υποδοχών (Java 7).....	426
Επιλογή ετοιμότητας	428
Η κλάση Selector.....	428
Η κλάση SelectionKey	430
12 Το πρωτόκολλο UDP.....	433
Το πρωτόκολλο UDP	433
Πελάτες UDP.....	435
Διακομιστές UDP	438
Η κλάση DatagramPacket	439
Οι κατασκευάστριες μέθοδοι.....	441
Οι μέθοδοι get.....	444
Οι μέθοδοι set.....	447
Η κλάση DatagramSocket	449
Οι κατασκευάστριες μέθοδοι.....	450
Αποστολή και λήψη αυτοδύναμων πακέτων.....	453
Διαχείριση συνδέσεων.....	457

Επιλογές υποδοχών.....	458
SO_TIMEOUT	459
SO_RCVBUF	460
SO_SNDBUF	461
SO_REUSEADDR	461
Μερικές χρήσιμες εφαρμογές.....	463
Απλοί πελάτες UDP.....	463
Η κλάση UDPServer.....	467
Ένας πελάτης UDP για το πρωτόκολλο echo.....	470
Η κλάση DatagramChannel.....	473
Χρήση της κλάσης DatagramChannel.....	474
13 Πολυεκπομπή IP	485
Πολυεκπομπή	486
Διευθύνσεις και ομάδες πολυεκπομπής.....	489
Πελάτες και διακομιστές	493
Δρομολογητές και δρομολόγηση.....	495
Εργασία με υποδοχές πολυεκπομπής	496
Οι κατασκευάστριες μέθοδοι.....	498
Επικοινωνία με μια ομάδα πολυεκπομπής	498
Δύο απλά παραδείγματα	503
Ευρετήριο	508

Πρόλογος

Η ανάπτυξη της Java τα 20 τελευταία χρόνια είναι πρωτοφανής. Με δεδομένη τη ραγδαία άνοδό της στην κορυφή και την ακόμα πιο εντυπωσιακή ανάπτυξη του Διαδικτύου, προκαλεί έκπληξη το γεγονός ότι ο δικτυακός προγραμματισμός σε Java εξακολουθεί να αποτελεί τόσο μεγάλο μυστήριο για πολλούς. Στην πραγματικότητα, η δημιουργία δικτυακών προγραμμάτων με την Java είναι αρκετά απλή, όπως θα διαπιστώσετε από το βιβλίο που κρατάτε στα χέρια σας. Οι αναγνώστες που έχουν πείρα στον δικτυακό προγραμματισμό σε περιβάλλοντα Unix, Windows ή Macintosh θα βρεθούν μπροστά σε μια ευχάριστη έκπληξη όταν συνειδητοποιήσουν πόσο πιο εύκολο είναι να γράψουν τα ίδια ή παραπλήσια προγράμματα χρησιμοποιώντας την Java. Η βασική διασύνδεση προγραμματισμού εφαρμογών (application programming interface, API) της Java περιλαμβάνει καλοσχεδιασμένες διασυνδέσεις για τις περισσότερες δικτυακές δυνατότητες. Μάλιστα, στη συντριπτική πλειοψηφία των περιπτώσεων, είναι πιο εύκολο να αναπτυχθεί στην Java οποιοδήποτε λογισμικό δικτύου επιπέδου εφαρμογών το οποίο μπορείτε να αναπτύξετε με τις γλώσσες προγραμματισμού C ή C++. Στην τέταρτη έκδοση του βιβλίου *Προγραμματισμός δικτυακών εφαρμογών με Java*, καταβάλλουμε κάθε δυνατή προσπάθεια να σας δείξουμε πώς να αξιοποιείτε τη βιβλιοθήκη δικτυακών κλάσεων της Java προκειμένου να γράφετε γρήγορα και εύκολα κώδικα για προγράμματα που εκτελούν επιτυχώς πολλές από τις συνήθεις δικτυακές εργασίες. Σε αυτές περιλαμβάνονται, μεταξύ άλλων, οι εξής:

- Περιήγηση στον Ιστό με το πρωτόκολλο HTTP (Hypertext Transfer Protocol, Πρωτόκολλο Μεταφοράς Υπερ-κειμένου)
- Συγγραφή κώδικα για πολυνηματικούς διακομιστές (multithreaded servers)
- Κρυπτογράφηση επικοινωνιών για σκοπούς εμπιστευτικότητας (confidentiality), πιστοποίησης ταυτότητας (authentication) και εγγυημένης ακεραιότητας (integrity) των μηνυμάτων
- Σχεδιασμός πελατών που διαθέτουν GUI (Graphical User Interface, διασύνδεση γραφικών με τον χρήστη) για δικτυακές υπηρεσίες
- Δημοσίευση δεδομένων σε προγράμματα που εκτελούνται στην πλευρά του διακομιστή
- Αναζήτηση υπολογιστών υπηρεσίας με χρήση του Συστήματος Ονομάτων Περιοχών (Domain Name System, DNS)
- Κατέβασμα αρχείων με ανώνυμο FTP (File Transfer Protocol, Πρωτόκολλο Μεταφοράς Αρχείων)
- Σύνδεση υποδοχών για δικτυακή επικοινωνία χαμηλού επιπέδου
- Μετάδοση δεδομένων σε όλους τους υπολογιστές υπηρεσίας του δικτύου με πολυεκπομπή

Η Java είναι η πρώτη (αν και όχι η μόνη) γλώσσα προγραμματισμού που παρέχει μια τέτοια πανίσχυρη δικτυακή βιβλιοθήκη για τον χειρισμό όλων αυτών των διαφορετικών εργασιών σε ετερογενή υπολογιστικά περιβάλλοντα. Στο βιβλίο *Προγραμματισμός δικτυακών εφαρμογών με Java* αποκαλύπτεται η ισχύς και η ποιότητα της συγκεκριμένης βιβλιοθήκης. Στόχος του βιβλίου είναι να σας δώσει τη δυνατότητα να χρησιμοποιήσετε την Java ως περιβάλλον για σοβαρό δικτυακό προγραμματισμό. Προς αυτή την κατεύθυνση, το βιβλίο παρέχει μια γενική εισαγωγή στις βασικές έννοιες των δικτύων, καθώς και λεπτομερείς περιγραφές και αναλύσεις των δυνατοτήτων και των ευκολιών που παρέχει η Java για την ανάπτυξη δικτυακών προγραμμάτων. Θα μάθετε πώς να δημιουργείτε προγράμματα σε Java τα οποία χρησιμοποιούν από κοινού δεδομένα στο Διαδίκτυο για παιχνίδια, συνεργατικά εγχειρήματα, ενημερώσεις λογισμικού, μεταφορά αρχείων και πολλά άλλα. Θα σας δοθεί επίσης η ευκαιρία να ρίξετε μια ματιά στα παρασκήνια των πρωτοκόλλων HTTP, SMTP (Simple Mail Transfer Protocol, Απλό Πρωτόκολλο Μεταφοράς Ταχυδρομείου) και TCP/IP (Transmission Control Protocol/Internet Protocol, Πρωτόκολλο Ελέγχου Μετάδοσης/Πρωτόκολλο Διαδικτύου), αλλά και των υπόλοιπων πρωτοκόλλων που υποστηρίζουν το Διαδίκτυο και τον Ιστό. Όταν ολοκληρώσετε τη μελέτη του βιβλίου, θα έχετε τις απαιτούμενες γνώσεις και τα εργαλεία για να δημιουργήσετε την επόμενη γενιά λογισμικού που θα εκμεταλλεύεται πλήρως τις δυνατότητες του Διαδικτύου.

Τι νεότερο υπάρχει στην τέταρτη έκδοση

Το 1996, στο εναρκτήριο κεφάλαιο της πρώτης έκδοσης αυτού του βιβλίου, αναφερθήκαμε εκτεταμένα στον τύπο των δυναμικών, κατανεμημένων (distributed) δικτυακών εφαρμογών που πιστεύαμε ότι θα μπορούσατε να αναπτύξετε με την Java. Μία από τις πιο συναρπαστικές πτυχές της συγγραφής των επόμενων εκδόσεων ήταν το να βλέπουμε με τα ίδια μας τα μάτια να υλοποιούνται σχεδόν όλες οι εφαρμογές που είχαμε προβλέψει. Οι προγραμματιστές χρησιμοποιούν την Java για να υποβάλλουν ερωτήματα σε διακομιστές βάσεων δεδομένων, να παρακολουθούν ιστοσελίδες, να ελέγχουν τηλεσκόπια, να διαχειρίζονται παιχνίδια για πολλούς παίκτες, και πολλά άλλα· και όλα αυτά τα επιτυγχάνουν χρησιμοποιώντας την εγγενή ικανότητα της Java να προσπελάζει το Διαδίκτυο. Η Java γενικότερα, και ο δικτυακός προγραμματισμός σε Java ειδικότερα, έχουν ξεφύγει οριστικά από το στάδιο του ενθουσιασμού και έχουν εισέλθει στον κόσμο των πραγματικών, λειτουργικών εφαρμογών.

Παρόμοια πορεία έχει ακολουθήσει και αυτό το βιβλίο. Η τέταρτη έκδοση εστιάζει ακόμα πιο πολύ στο πρωτόκολλο HTTP και την αρχιτεκτονική REST (REpresentational State Transfer, Μεταφορά Παραστατικής Κατάστασης). Το HTTP, ένα από τα πολλά διαθέσιμα δικτυακά πρωτόκολλα, έχει εξελιχθεί σταδιακά στο κύριο δικτυακό πρωτόκολλο. Όπως θα διαπιστώσετε, συχνά αποτελεί το πρωτόκολλο πάνω στο οποίο στηρίζεται η δημιουργία άλλων πρωτοκόλλων, σχηματίζοντας έτσι το δικό του επίπεδο στη στοίβα δικτύου (network stack).

Στο βιβλίο περιγράφονται επίσης οι πολλές άλλες μικροαλλαγές και ενημερώσεις που έχουν γίνει τόσο στο πακέτο `java.net`, όσο και στα υποστηρικτικά πακέτα των εκδόσεων 6, 7 και 8 της Java. Στις νέες κλάσεις που παρουσιάζονται σε αυτή την έκδοση περιλαμβάνονται οι κλάσεις `CookieManager`, `CookiePolicy`, `CookieStore`, `HttpCookie`, `SwingWorker`, `Executor`, `ExecutorService`, `AsynchronousSocketChannel`, `AsynchronousServerSocketChannel` κ.ά. Στις τρεις πιο πρόσφατες εκδόσεις της Java έχουν προστεθεί και πολλές νέες μέθοδοι σε υπάρχουσες κλάσεις, μέθοδοι που αναλύονται στα σχετικά κεφάλαια. Μεγάλα τμήματα του βιβλίου έχουν γραφεί εκ νέου ώστε να αντικατοπτρίζουν τις διαρκώς μεταβαλλόμενες τάσεις στον προγραμματισμό με Java γενικότερα και στον δικτυακό προγραμματισμό ειδικότερα. Ελπίζουμε να διαπιστώσετε και μόνοι

σας ότι, σε σχέση με την προηγούμενη έκδοση, η τέταρτη έκδοση αποτελεί ένα ακόμα πιο ισχυρό, πιο χρήσιμο σε βάθος χρόνου, πιο ακριβές, και πιο διασκεδαστικό διδακτικό βοήθημα και εγχειρίδιο αναφοράς για τον δικτυακό προγραμματισμό με Java.

Οργάνωση του βιβλίου

Στο Κεφάλαιο 1, *Βασικές έννοιες δικτύων*, θα εξηγήσουμε λεπτομερώς όλα όσα χρειάζεται να γνωρίζει ένας προγραμματιστής σχετικά με τον τρόπο λειτουργίας των δικτύων και του Διαδικτύου. Στο ίδιο κεφάλαιο θα περιγράψουμε τα υποκείμενα πρωτόκολλα του Διαδικτύου, όπως το TCP/IP και το UDP/IP.

Τα δύο επόμενα κεφάλαια ρίχνουν φως σε δύο πτυχές του προγραμματισμού με Java οι οποίες είναι κρίσιμες για όλα σχεδόν τα δικτυακά προγράμματα, αλλά συχνά γίνονται αντικείμενο παρανόησης και κακής χρήσης: την είσοδο/έξοδο (input/output, I/O) και τη νημάτωση (threading). Στο Κεφάλαιο 2, *Ρεύματα*, θα διερευνήσουμε την κλασική είσοδο/έξοδο της Java η οποία –παρά την έλευση των νέων διασυνδέσεων API εισόδου/εξόδου– δεν πρόκειται να εγκαταλειφθεί στο άμεσο μέλλον, και εξακολουθεί να αποτελεί τον προτιμώμενο τρόπο χειρισμού της εισόδου/εξόδου στην πλειονότητα των εφαρμογών-πελατών. Η κατανόηση του τρόπου με τον οποίο η Java χειρίζεται την είσοδο/έξοδο στη γενική περίπτωση αποτελεί προαπαιτούμενο για να κατανοήσει κανείς την ειδικότερη περίπτωση που αφορά τον αντίστοιχο τρόπο χειρισμού της δικτυακής εισόδου/εξόδου. Στο Κεφάλαιο 3, *Νήματα*, θα μελετήσουμε την πολυνημάτωση και τον συγχρονισμό (synchronization), δίνοντας ιδιαίτερη έμφαση στο πώς μπορούν να χρησιμοποιηθούν για ασύγχρονη είσοδο/έξοδο (asynchronous I/O) και διακομιστές δικτύου.

Οι έμπειροι προγραμματιστές της Java θα μπορούσαν να φυλλομετρήσουν ή να παραλείψουν αυτά τα δύο κεφάλαια. Ωστόσο, το Κεφάλαιο 4, *Διευθύνσεις Διαδικτύου*, συνιστά υποχρεωτική μελέτη για όλους τους αναγνώστες. Στο συγκεκριμένο κεφάλαιο, θα δείξουμε πώς τα προγράμματα της Java αλληλεπιδρούν με το Σύστημα Ονομάτων Περιοχών (DNS) μέσω της κλάσης `InetAddress`, η οποία αποτελεί και τη μόνη κλάση που απαιτείται από όλα ουσιαστικά τα δικτυακά προγράμματα. Αφού ολοκληρώσετε την ανάγνωση του κεφαλαίου, μπορείτε να διαβάσετε τα επόμενα κεφάλαια με όποια σειρά επιθυμείτε, ανάλογα με τα ενδιαφέροντα και τις ανάγκες σας.

Στο Κεφάλαιο 5, *URL και URI*, θα μελετήσουμε την κλάση `URL` της Java, μια πανίσχυρη αφαίρεση (abstraction) για το κατέβασμα πληροφοριών και αρχείων από δικτυακούς διακομιστές πολλών ειδών. Η κλάση `URL` σας δίνει τη δυνατότητα να συνδέεστε σε έναν δικτυακό διακομιστή και να κατεβάζετε αρχεία από αυτόν χωρίς να ανησυχείτε για τις λεπτομέρειες του πρωτοκόλλου που χρησιμοποιεί ο διακομιστής. Σας επιτρέπει να συνδέεστε σε έναν διακομιστή FTP χρησιμοποιώντας τον ίδιο κώδικα που χρησιμοποιείτε για να «συνομιλείτε» με διακομιστές HTTP ή να διαβάζετε αρχεία από τον τοπικό σκληρό δίσκο. Θα περιγράψουμε επίσης τη νεότερη κλάση `URI`, μια εναλλακτική επιλογή για τον προσδιορισμό –αλλά όχι και την ανάκτηση– πόρων η οποία συμμορφώνεται περισσότερο με τα διάφορα πρότυπα.

Στο Κεφάλαιο 6, *Το πρωτόκολλο HTTP*, θα εμβαθύνουμε αποκλειστικά στο HTTP. Στα σχετικά θέματα περιλαμβάνονται η αρχιτεκτονική REST, οι κεφαλίδες HTTP και τα cookies («μπισκότα»). Στο Κεφάλαιο 7, *Συνδέσεις με την κλάση `URLConnection`*, θα σας δείξουμε πώς να χρησιμοποιείτε τις κλάσεις `URLConnection` και `HttpURLConnection` όχι μόνο για να κατεβάζετε δεδομένα από διακομιστές Ιστού, αλλά και για να «ανεβάζετε» έγγραφα και να ρυθμίζετε συνδέσεις.

Στα Κεφάλαια 8 έως 10 θα περιγράψουμε τις κλάσεις υποδοχών χαμηλού επιπέδου της Java για δικτυακή πρόσβαση. Στο Κεφάλαιο 8, *Υποδοχές για πελάτες*, θα παρουσιάσουμε τη διασύνδεση API υποδοχών της Java γενικά και την κλάση `Socket` ειδικότερα. Θα σας δείξουμε πώς να

γράφετε κώδικα για πελάτες δικτύου που αλληλεπιδρούν με διακομιστές TCP όλων των ειδών, συμπεριλαμβανομένων διακομιστών για τα πρωτόκολλα whois, dict και HTTP. Στο Κεφάλαιο 9, *Υποδοχές για διακομιστές*, θα μάθετε πώς να χρησιμοποιείτε την κλάση `ServerSocket` για να αναπτύσσετε διακομιστές γι' αυτά και άλλα πρωτόκολλα. Τέλος, στο Κεφάλαιο 10, *Ασφαλείς Υποδοχές*, θα μελετήσουμε τρόπους με τους οποίους μπορείτε να προστατεύετε τις επικοινωνίες πελάτη-διακομιστή χρησιμοποιώντας το Ασφαλές Επίπεδο Υποδοχών (Secure Sockets Layer, SSL) και την Επέκταση Ασφαλών Υποδοχών της Java (Java Secure Sockets Extension, JSSE).

Το Κεφάλαιο 11, *Είσοδος/έξοδος χωρίς μπλοκάρισμα*, αποτελεί μια εισαγωγή στις νέες διασυνδέσεις API εισόδου/εξόδου που έχουν σχεδιαστεί αποκλειστικά για δικτυακούς διακομιστές. Οι συγκεκριμένες διασυνδέσεις επιτρέπουν σε ένα πρόγραμμα να εξακριβώσει αν μια σύνδεση είναι έτοιμη προτού επιχειρήσει να διαβάσει ή να εγγράψει δεδομένα από την/στην υποδοχή. Αυτό καθιστά δυνατή την ταυτόχρονη διαχείριση πολλών διαφορετικών συνδέσεων από ένα μόνο νήμα (thread), περιορίζοντας έτσι σημαντικά τον φόρτο της εικονικής μηχανής (virtual machine, VM). Οι νέες διασυνδέσεις API εισόδου/εξόδου δεν βοηθούν ιδιαίτερα στην περίπτωση μικρών διακομιστών ή πελατών που δεν ανοίγουν πολλές ταυτόχρονες συνδέσεις· ωστόσο, ενδέχεται να βελτιώσουν την απόδοση διακομιστών που διακινούν υψηλούς όγκους δεδομένων, οι οποίοι θέλουν να μεταδίδουν όσο το δυνατόν πιο γρήγορα τη μέγιστη ποσότητα δεδομένων που δύναται να χειριστεί το δίκτυο.

Στο Κεφάλαιο 12, *Το πρωτόκολλο UDP*, θα παρουσιάσουμε το Πρωτόκολλο Αυτοδύναμων Πακέτων Χρήστη (User Datagram Protocol, UDP) και τις σχετικές κλάσεις `DatagramPacket` και `DatagramSocket`, οι οποίες παρέχουν γρήγορη και αναξιόπιστη επικοινωνία. Τέλος, στο Κεφάλαιο 13, *Πολυεκπομπή IP*, θα σας δείξουμε πώς να χρησιμοποιείτε το UDP για να επικοινωνείτε με πολλούς υπολογιστές υπηρεσίας (hosts) ταυτόχρονα.

Απαιτήσεις από τους αναγνώστες

Το βιβλίο προϋποθέτει ότι είστε εξοικειωμένοι όχι μόνο με τον αντικειμενοστρεφή προγραμματισμό (object-oriented programming) γενικά, αλλά και με τη γλώσσα και το περιβάλλον προγραμματισμού της Java ειδικότερα. Το σύγγραμμα αυτό δεν προσπαθεί να παίξει τον ρόλο του βασικού εγχειριδίου για μια γλώσσα προγραμματισμού. Θα πρέπει να είστε καλοί γνώστες της σύνταξης που χρησιμοποιεί η Java. Θα πρέπει να έχετε ήδη γράψει κώδικα για απλές εφαρμογές. Δεν θα έβλαπτε επίσης καθόλου αν είχατε βασικές γνώσεις προγραμματισμού με τη βιβλιοθήκη Swing, αν και κάτι τέτοιο είναι ουσιαστικά απαραίτητο μόνο σε ελάχιστα παραδείγματα. Σε περίπτωση που χρειαστεί να εντρυφήσετε σε ένα θέμα το οποίο απαιτεί βαθύτερη κατανόηση του δικτυακού προγραμματισμού από ό,τι είθισται –για παράδειγμα, τα νήματα και τα ρεύματα (streams)– το συγκεκριμένο θέμα θα καλύπτεται επίσης, έστω και συνοπτικά.

Από την άλλη πλευρά, το βιβλίο που κρατάτε στα χέρια σας δεν απαιτεί να έχετε προηγούμενη εμπειρία με τον δικτυακό προγραμματισμό. Θα διαπιστώσετε ότι αποτελεί μιας πρώτης τάξης εισαγωγή στις έννοιες των δικτύων και την ανάπτυξη δικτυακών εφαρμογών. Επίσης, δεν θεωρεί δεδομένο ότι θα γνωρίζετε απέξω και ανακατωτά τα χιλιάδες αρκτικόλεξα της δικτύωσης (TCP, UDP, SMTP κλπ.). Στα αντίστοιχα κεφάλαια του βιβλίου, θα μάθετε όλα όσα χρειάζεται να ξέρετε για όλα αυτά.

Εκδόσεις της Java

Σε σχέση με άλλα τμήματα της βασικής διασύνδεσης API, οι δικτυακές κλάσεις της Java έχουν αλλάξει με πολύ πιο αργό ρυθμό από την έκδοση 1.0 και έπειτα. Συγκριτικά με τη βιβλιοθήκη AWT ή την εισόδο/έξοδο, έχουν γίνει ελάχιστες προσθήκες και σχεδόν καθόλου αλλαγές. Φυσικά, όλα τα δικτυακά προγράμματα κάνουν εκτεταμένη χρήση των κλάσεων εισόδου/εξόδου, ενώ κάποια από αυτά χρησιμοποιούν πολύ και διασυνδέσεις GUI. Η συγγραφή του βιβλίου έχει στηριχθεί στην υπόθεση ότι γράφετε κώδικα με την Java 5.0 ή μεταγενέστερες εκδόσεις. Γενικά, θα παρατηρήσετε ότι στο βιβλίο χρησιμοποιούνται αφειδώς –και χωρίς περαιτέρω εξηγήσεις– χαρακτηριστικά και δυνατότητες της Java 5, όπως οι γενικεύσεις (generics) και ο βελτιωμένος βρόχος for.

Για τους σκοπούς του δικτυακού προγραμματισμού, οι διαφορές μεταξύ της Java 5 και της Java 6 δεν είναι μεγάλες. Τα περισσότερα παραδείγματα είναι σχεδόν πανομοιότυπα στις δύο εκδόσεις. Όταν γίνεται χρήση οποιασδήποτε μεθόδου ή κλάσης η οποία εμφανίζεται για πρώτη φορά στην έκδοση 6, 7 ή 8 της Java, αυτό επισημαίνεται με ένα σχόλιο που ακολουθεί τη δήλωσή της, όπως φαίνεται παρακάτω:

```
public void setFixedLengthStreamingMode(long contentLength) // Java 7
```

Στην περίπτωση της Java 7, είμαστε λίγο πιο φειδωλοί. Μπορεί να μην αποφεύγουμε τη χρήση χαρακτηριστικών και δυνατοτήτων που παρουσιάστηκαν για πρώτη φορά στη συγκεκριμένη έκδοση, όπου αυτά κρίνονται ιδιαίτερα χρήσιμα ή βολικά, αλλά επισημαίνουμε εμφανώς τη χρήση τους. (Για παράδειγμα, η εντολή try-with-resources και η πολλαπλή «σύλληψη» εξαιρέσεων με εντολές catch αποδείχθηκαν και οι δύο πολύ χρήσιμες στην προσπάθειά μας να χωρέσουμε όλα τα παραδείγματα στον περιορισμένο διαθέσιμο χώρο της έντυπης έκδοσης.)

Στο σύνολό της, ωστόσο, η διασύνδεση API της Java για τη δικτύωση έχει παραμείνει σχετικά σταθερή από την έκδοση 1.0 και έπειτα. Στις μεταγενέστερες εκδόσεις, είναι ελάχιστο το ποσοστό της διασύνδεσης που έχει κριθεί απαρχαιωμένο και άρα πρέπει να αποφεύγεται, ενώ και οι προσθήκες δεν είναι ιδιαίτερα σημαντικές. Κανονικά, θα έπρεπε να είστε σε θέση να χρησιμοποιήσετε αυτό το βιβλίο και με την Java 8. Αντίθετα, η εμφάνιση νέων διασυνδέσεων API παρατηρείται με κάπως μεγαλύτερη συχνότητα στις υποστηρικτικές κλάσεις, ειδικά εκείνες που σχετίζονται με το κομμάτι της εισόδου/εξόδου, το οποίο έχει υποστεί τρεις σημαντικές αναθεωρήσεις από τότε που κυκλοφόρησε η Java 1.0.

Τα παραδείγματα

Οι περισσότερες μέθοδοι και κλάσεις που περιγράφονται στο βιβλίο συνοδεύονται από τουλάχιστον ένα πλήρες λειτουργικό πρόγραμμα, όσο απλό και αν είναι αυτό. Κατά την εμπειρία μας, ένα πλήρες λειτουργικό πρόγραμμα είναι απαραίτητο όταν θέλει κανείς να δείξει τη σωστή χρήση μιας μεθόδου. Διαφορετικά, είναι πολύ εύκολο για τον συγγραφέα να καταφύγει στην υπερβολική χρήση ορολογίας ή να παραλείψει επιμελώς λεπτομέρειες τις οποίες ο ίδιος ενδέχεται να μην έχει ξεκαθαρίσει στο μυαλό του. Η τεκμηρίωση των διασυνδέσεων API της Java δείχνει, επίσης, να πάσχει συχνά από το σύνδρομο της υπερβολικής λακωνικότητας στις περιγραφές των κλήσεων των μεθόδων. Στο βιβλίο που κρατάτε στα χέρια σας δείχνουμε μια σαφή προτίμηση στις λεπτομέρειες επεξηγήσεις, ακόμα και όταν αυτό φαντάζει υπερβολικό. Αν κάτι είναι προφανές για εσάς, δεν έχετε παρά να το αγνοήσετε. Δεν χρειάζεται να πληκτρολογήσετε και να εκτελέσετε τον κώδικα όλων των παραδειγμάτων του βιβλίου· αν, όμως, αντιμετωπίζετε δυσκολίες με μια συγκεκριμένη

κριμένη μέθοδο, τότε η άποψή μας είναι ότι πρέπει να έχετε στη διάθεσή σας τουλάχιστον ένα λειτουργικό παράδειγμα.

Κάθε κεφάλαιο περιλαμβάνει τουλάχιστον ένα (ή περισσότερα) πιο πολύπλοκα προγράμματα που δείχνουν τη χρήση των κλάσεων και των μεθόδων του κεφαλαίου σε ένα πιο ρεαλιστικό περιβάλλον. Σε αρκετές περιπτώσεις, τα συγκεκριμένα προγράμματα στηρίζονται σε χαρακτηριστικά και δυνατότητες της Java που δεν περιγράφονται στο βιβλίο. Μάλιστα, σε πολλά από αυτά οι υπομονάδες δικτύωσης καλύπτουν μόνο ένα μικρό ποσοστό του πηγαίου κώδικα, ενώ συχνά αποτελούν και τα λιγότερο δύσκολα τμήματα του κώδικα. Μολαταύτα, κανένα από τα εν λόγω προγράμματα δεν θα μπορούσε να αναπτυχθεί πιο εύκολα σε γλώσσες προγραμματισμού οι οποίες δεν εστιάζουν στη δικτύωση τόσο πολύ όσο η Java. Στην ουσία, η προφανής απλότητα των σχετικών τμημάτων του κώδικα δεν αντανakλά την ασημαντότητα του προγράμματος, αλλά τον βαθμό στον οποίο η δικτύωση αποτελεί πλέον θεμελιώδες χαρακτηριστικό της Java. Όλα τα προγράμματα των παραδειγμάτων που παρουσιάζονται στο βιβλίο είναι διαθέσιμα στο Διαδίκτυο, συχνά με διορθώσεις και προσθήκες. Μπορείτε να κατεβάσετε τον πηγαίο κώδικα από τη διεύθυνση <http://www.cafeaulait.org/books/jnp4/>.

Όλα τα παραδείγματα έχουν δοκιμαστεί σε υπολογιστές με λειτουργικό σύστημα Linux, ενώ πολλές αντίστοιχες δοκιμές έχουν γίνει και σε μηχανήματα με Windows και Mac OS X. Η πλειονότητα των παραδειγμάτων που δίνονται στο βιβλίο *θα έπρεπε* να λειτουργεί επίσης σε άλλα υπολογιστικά περιβάλλοντα, καθώς και με άλλους μεταγλωττιστές (compilers) και εικονικές μηχανές που υποστηρίζουν την Java 5 ή μεταγενέστερες εκδόσεις. Οι συνηθέστεροι λόγοι για τους οποίους ένα παράδειγμα ενδέχεται να μην μεταγλωττιστεί με την Java 5 ή την Java 6 είναι η χρήση της εντολής `try-with-resources` και η πολλαπλή «σύλληψη». Ο κώδικας των σχετικών παραδειγμάτων μπορεί να ξαναγραφεί εύκολα ώστε να υποστηρίζει προγενέστερες εκδόσεις της Java· το αντίτιμο είναι η αυξημένη (και μάλλον περιττή) έκταση του κώδικα.

Δεν μπορούμε να μην επισημάνουμε, επίσης, και κάποιους συμβιβασμούς που ο καθένας μας θα ήταν αναγκασμένος να κάνει λόγω του περιορισμένου χώρου μιας έντυπης έκδοσης. Πρώτον, σπανίως ελέγχονται οι συνθήκες εισόδου (preconditions). Οι περισσότερες μέθοδοι υποθέτουν ότι μεταβιβάζονται σε αυτές σωστά δεδομένα, οπότε παρακάμπτουν τους ελέγχους για τιμές `null` και άλλες παρόμοιες πρακτικές που σχετίζονται με την ορθή συγγραφή κώδικα. Επιπλέον, έχουμε ελαττώσει τις εσοχές (indentation) σε δύο χαρακτήρες για κάθε νέο μπλοκ κώδικα και τέσσερις χαρακτήρες για κάθε γραμμή που συνεχίζεται στην επόμενη, σε αντίθεση με το πρότυπο της Java (τέσσερις και οκτώ χαρακτήρες, αντίστοιχα). Ελπίζουμε ότι αυτές οι παραβλέψεις δεν θα σας αποσπάσουν από την προσεκτική μελέτη. Από την άλλη πλευρά, όμως, οι ίδιοι συμβιβασμοί μάς βοήθησαν να περιορίσουμε σημαντικά το μέγεθος της τέταρτης έκδοσης (κατά αρκετές εκατοντάδες σελίδες) σε σχέση με την προηγούμενη έκδοση.

Συμβάσεις του βιβλίου

Μια ειδική γραμματοσειρά με σταθερό πλάτος χρησιμοποιείται για τα εξής:

- Παραδείγματα και αποσπάσματα κώδικα
- Οτιδήποτε μπορεί να εμφανιστεί μέσα σε ένα πρόγραμμα της Java, μεταξύ άλλων δεσμευμένες λέξεις, τελεστές, τύποι δεδομένων, καθώς και ονόματα μεθόδων, μεταβλητών, κλάσεων και διασυνδέσεων
- Έξοδος προγραμμάτων
- Ετικέτες που ενδέχεται να εμφανίζονται σε ένα έγγραφο HTML

Μια έντονη γραμματοσειρά σταθερού μεγέθους χρησιμοποιείται για

- Γραμμές εντολών και επιλογές που πρέπει να πληκτρολογηθούν κατά λέξη στην οθόνη

Μια πλάγια γραμματοσειρά χρησιμοποιείται στις ακόλουθες περιπτώσεις:

- Νέοι όροι, στα σημεία όπου δίνονται οι ορισμοί
- Ονόματα διαδρομών, αρχείων και προγραμμάτων (όμως, αν το όνομα του προγράμματος είναι επίσης το όνομα μιας κλάσης της Java, τότε γράφεται με την ειδική γραμματοσειρά σταθερού μεγέθους όπως όλα τα υπόλοιπα ονόματα κλάσεων)
- Ονόματα υπολογιστών υπηρεσίας και περιοχών (*www.hpamor.com*)
- Διευθύνσεις URL (*http://www.cafeaulait.org/slides/*)
- Τίτλοι άλλων βιβλίων (*Java I/O*)



Υποδεικνύει μια συμβουλή, πρόταση, ή γενική σημείωση.



Υποδεικνύει μια προειδοποίηση ή σύσταση.

Γενικά, τα μεγάλα αποσπάσματα κώδικα και τα πλήρη προγράμματα τοποθετούνται σε ξεχωριστή παράγραφο, ως εξής:

```
Socket s = new Socket("java.oreilly.com", 80);
if (!s.getTcpNoDelay()) s.setTcpNoDelay(true);
```

Όταν ο κώδικας παρουσιάζεται τμηματικά και όχι με τη μορφή πλήρους προγράμματος, πρέπει να θεωρείται δεδομένη η ύπαρξη των κατάλληλων εντολών εισαγωγής (`import`). Για παράδειγμα, στο προηγούμενο απόσπασμα κώδικα, μπορείτε να υποθέσετε ότι έχει γίνει εισαγωγή της κλάσης `java.net.Socket`.

Σε κάποια παραδείγματα, συνδυάζονται δεδομένα που εισάγει ο χρήστης με δεδομένα εξόδου του προγράμματος. Σε τέτοιες περιπτώσεις, τα δεδομένα του χρήστη θα εμφανίζονται με έντονη γραφή, όπως στο παρακάτω παράδειγμα από το Κεφάλαιο 9:

```
% telnet rama.poly.edu 7
Trying 128.238.10.212...
Connected to rama.poly.edu.
Escape character is '^]'.
This is a test
This is a test
This is another test
This is another test
9876543210
9876543210
^]
telnet> close
Connection closed.
```

Τέλος, αν και πολλά από τα παραδείγματα που παρουσιάζονται στο βιβλίο είναι «τεχνητά» παραδείγματα τα οποία είναι απίθανο να χρησιμοποιηθούν στον πραγματικό κόσμο, μερικές από τις κλάσεις που θα αναπτύξουμε έχουν πραγματική αξία. Μπορείτε να τις ενσωματώσετε στον δι-

κό σας κώδικα, είτε ολόκληρες είτε τμήματα αυτών. Δεν απαιτείται ειδική άδεια. Η χρήση τους είναι ελεύθερη (αν και προφανώς δεν ισχύει το ίδιο για το επεξηγηματικό κείμενο!).

Αιτήσεις για Σχόλια

Πάντα χαιρόμαστε να λαμβάνουμε μηνύματα από τους αναγνώστες, ανεξάρτητα από το αν περιέχουν γενικά σχόλια για το βιβλίο, συγκεκριμένες διορθώσεις, άλλα θέματα που θα ήθελαν να συμπεριληφθούν στο βιβλίο, ή απλώς ιστορίες τρόμου για τις δικές τους επώδυνες εμπειρίες με τον δικτυακό προγραμματισμό. Μπορείτε να έρθετε σε επαφή με τον συγγραφέα στέλνοντας ένα μήνυμα ηλεκτρονικού ταχυδρομείου στη διεύθυνση elharo@ibiblio.org. Είναι αναγκαίο να τονιστεί, ωστόσο, ότι ο συγγραφέας λαμβάνει εκατοντάδες μηνύματα καθημερινά και είναι αδύνατο να απαντήσει προσωπικά σε καθένα από αυτά. Για να αυξήσετε τις πιθανότητές σας να λάβετε προσωπική απάντηση, σας παρακαλούμε να επισημάνετε ότι είστε αναγνώστες του βιβλίου. Αν έχετε κάποια ερώτηση σχετικά με ένα συγκεκριμένο πρόγραμμα το οποίο δεν λειτουργεί όπως ίσως θα περιμένατε, προσπαθήστε να το φέρετε στην πιο απλή μορφή που αναπαράγει το σφάλμα –κατά προτίμηση μία μόνο κλάση– και επικολλήστε τον κώδικα ολόκληρου του προγράμματος στο σώμα του μηνύματος. Τυχόν αυθαίρετα συνημμένα θα διαγράφονται χωρίς να ανοιχθούν. Και μια θερμή παράκληση: Σας παρακαλούμε να στέλνετε τα μηνύματά σας από τον λογαριασμό στον οποίο θέλετε να σας απαντήσει ο συγγραφέας, αφού πρώτα βεβαιωθείτε ότι η διεύθυνση Reply-to (Απάντηση) είναι ρυθμισμένη σωστά! Δεν υπάρχει τίποτα πιο εκνευριστικό από το να ξοδέψεις μία ώρα ή περισσότερο σε ενδελεχή έρευνα προκειμένου να απαντήσεις σε κάποια ενδιαφέρουσα ερώτηση και να συντάξεις μια λεπτομερή απάντηση, και αμέσως μετά να ανακαλύψεις με τρόπο ότι δεν μπορείς να επικοινωνήσεις με τον συνομιλητή σου. Γιατί; Επειδή επέλεξε να στείλει τις απόψεις του από κάποιο δημόσιο τερματικό και αμέλησε να ρυθμίσει τις σχετικές προτιμήσεις του φυλλομετρητή ώστε να φαίνεται η πραγματική διεύθυνση ηλεκτρονικού ταχυδρομείου του.

Επίσης, ο συγγραφέας είναι ένθερμος οπαδός της διαχρονικής ρήσης «Αν σας αρέσει το βιβλίο, πείτε το στους φίλους σας. Αν δεν σας αρέσει, πείτε το σε μένα». Ενδιαφέρεται ιδιαίτερα να μαθαίνει για λάθη. Αυτή είναι η τέταρτη έκδοση του βιβλίου. Υπάρχουν πάντα περιθώρια βελτίωσης, γι' αυτό και δεν σταματούμε ποτέ να προσπαθούμε. Ανεξάρτητα από το πόσο σκληρά εργαστήκαμε όλοι –και ο συγγραφέας και οι επιμελητές του εκδοτικού οίκου O'Reilly– είμαστε σίγουροι ότι θα υπάρχουν λάθη και τυπογραφικά σφάλματα που μας ξέφυγαν σε κάποια σημεία. Και δεν έχουμε καμία αμφιβολία ότι τουλάχιστον ένα από αυτά θα προκαλέσει κάποιο πραγματικά ντροπιαστικό και τεράστιο πρόβλημα. Αν εντοπίσετε λάθη ή τυπογραφικά σφάλματα, ενημερώστε τον συγγραφέα ώστε να τα διορθώσει. Η διόρθωση θα δημοσιευθεί στον ιστότοπο του εκδοτικού οίκου O'Reilly, στη διεύθυνση http://bit.ly/java_ntwk_program_errata. Προτού υποβάλλετε μια αναφορά, σας παρακαλούμε να ελέγξετε τις συγκεκριμένες ιστοσελίδες για να δείτε αν ο συγγραφέας είναι ήδη ενήμερος για το λάθος και έχει δημοσιεύσει τη σχετική διόρθωση. Όλα τα λάθη που επισημαίνονται με αυτόν τον τρόπο θα διορθωθούν σε μελλοντικές εκδόσεις.

Χρήση παραδειγμάτων κώδικα

Στόχος του βιβλίου είναι να σας βοηθήσει να κάνετε τη δουλειά σας. Γενικά, αν ο κώδικας των παραδειγμάτων που παρουσιάζονται στο βιβλίο σας φαίνεται χρήσιμος, μπορείτε να τον ενσωματώσετε στα προγράμματα και την τεκμηρίωσή σας. Δεν χρειάζεται να επικοινωνήσετε μαζί μας για να ζητήσετε άδεια, εκτός και αν έχετε επιλέξει να αναπαραγάγετε σημαντική ποσότητα κώδικα. Για παράδειγμα, η ανάπτυξη ενός προγράμματος που χρησιμοποιεί αρκετά αποσπάσματα κώ-

δικα από το βιβλίο δεν απαιτεί άδεια. Ωστόσο, δεν ισχύει το ίδιο για την πώληση ή διανομή ενός CD-ROM με παραδείγματα από βιβλία του εκδοτικού οίκου O'Reilly. Αν θέλετε να απαντήσετε σε μια ερώτηση μνημονεύοντας το βιβλίο και παραθέτοντας κώδικα από κάποιο παράδειγμα, δεν χρειάζεται να ζητήσετε άδεια. Αν, όμως, θέλετε να ενσωματώσετε μια σημαντική ποσότητα από τον κώδικα των παραδειγμάτων του βιβλίου στην τεκμηρίωση του προϊόντος σας, τότε πρέπει όντως να ζητήσετε άδεια.

Εκτιμούμε, αλλά δεν απαιτούμε, τις βιβλιογραφικές αναφορές. Παρά ταύτα, μια σωστή βιβλιογραφική αναφορά περιλαμβάνει συνήθως τον τίτλο, τον συγγραφέα, τον εκδότη και τον κωδικό ISBN. Για παράδειγμα: “*Java Network Programming*, Fourth Edition, Elliotte Rusty Harold (O'Reilly). Copyright 2014 Elliotte Rusty Harold, 978-1-449-35767-2.”

Αν αισθάνεστε ότι η χρήση που επιλέγετε να κάνετε όσον αφορά τον κώδικα των παραδειγμάτων του βιβλίου ξεφεύγει από τα όρια της θεμιτής χρήσης ή της άδειας που παρέχεται εδώ, μην διστάσετε να επικοινωνήσετε μαζί μας στη διεύθυνση permissions@oreilly.com.

Επικοινωνία

Μπορείτε να στείλετε σχόλια και ερωτήσεις για το βιβλίο είτε στις Εκδόσεις Κλειδάριθμος (info@klidarithmos.gr) είτε στις Εκδόσεις O'Reilly, στην παρακάτω διεύθυνση (στα αγγλικά) και τηλέφωνα:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

Υπάρχει μια συνοδευτική ιστοσελίδα (στα Αγγλικά) με παροράματα, παραδείγματα και άλλες πρόσθετες πληροφορίες. Μπορείτε να βρείτε τη σελίδα αυτή στη διεύθυνση <http://oreil.ly/java-network-prgammimg>.

Αν θέλετε να υποβάλετε σχόλια ή τεχνικές ερωτήσεις για το βιβλίο, σας παρακαλούμε να στείλετε ένα μήνυμα ηλεκτρονικού ταχυδρομείου (στα αγγλικά) στη διεύθυνση bookquestions@oreilly.com.

Ευχαριστίες

Στην παραγωγή αυτού του βιβλίου συμμετείχαν πολλά άτομα. Ο επιμελητής μου, Mike Loukides, ήταν ο ιθύνων νους και παρείχε στην πορεία πολλά χρήσιμα σχόλια που το βελτίωσαν αισθητά. Η συμβολή του Δρ. Peter “Peppar” Parnes στο Κεφάλαιο 13 ήταν παραπάνω από καθοριστική. Όλοι οι διορθωτές συνέβαλαν ανεκτίμητα στον εντοπισμό λαθών και παραλείψεων. Ο Simon St. Laurent παρείχε κρίσιμες συμβουλές όσον αφορά τα θέματα που χρειαζόνταν αναλυτικότερη κάλυψη. Ο Scott Oaks είχε την ευγενή καλοσύνη να μας «δανείσει» την πολύτιμη πείρα του στο Κεφάλαιο 3, αποδεικνύοντας για μία ακόμα φορά –με τα πολλά δυσδιάκριτα σφάλματα που εντόπισε– ότι η πολυνημάτωση εξακολουθεί να απαιτεί τη φροντίδα ενός ειδικού. Ο Ron Hitchens έριξε φως σε πολλές από τις πιο σκοτεινές περιοχές των νέων διασυνδέσεων API εισόδου/εξόδου. Ο Marc Loy και ο Jim Elliott ανέλαβαν να επιθεωρήσουν και να κρίνουν ένα κομμάτι της πρωτοποριακής ύλης που παρουσιάζεται στο βιβλίο. Ο Timothy F. Rohaly επέδειξε απaráμιλλη αφοσί-

ωση στο να κλείσει όλες τις υποδοχές μου και να «συλλάβει» όλες τις πιθανές εξαιρέσεις, και γενικά στο να με βοηθήσει να γράψω τον πιο καθαρό, ασφαλή και υποδειγματικό κώδικα που θα μπορούσε να γράψει κανείς. Ο John Zukowski εντόπισε αρκετές παραλείψεις, που πλέον δεν υφίστανται χάρη σε εκείνον. Και ο Avner Gelb, ένας πραγματικός «αετός», επέδειξε μια εκπληκτική ικανότητα στον εντοπισμό λαθών που είχαν καταφέρει να περάσουν απαρατήρητα από τον συγγραφέα, όλους τους άλλους επιμελητές/διορθωτές, και τους δεκάδες χιλιάδες αναγνώστες της πρώτης έκδοσης του βιβλίου. Ο Alex Stangl και ο Ryan Cuprak παρείχαν περαιτέρω βοήθεια στον εντοπισμό νέων λαθών σε αυτή την έκδοση, καθώς και παλαιότερων λαθών που δεν είχαν διορθωθεί.

Δεν είθισται να απευθύνονται ευχαριστίες στον εκδότη, αλλά ο εκδότης είναι εκείνος που ορίζει τα στάνταρ για όλη την υπόλοιπη εταιρεία, τους συγγραφείς, τους επιμελητές και το προσωπικό παραγωγής. Νομίζω ότι ο Tim O'Reilly αξίζει ιδιαίτερα εύσημα για το γεγονός ότι έχει καταφέρει να εξελίξει τις εκδόσεις O'Reilly Media σε έναν από τους καλύτερους εκδοτικούς οίκους για οποιονδήποτε συγγραφέα. Αν υπάρχει ένα άτομο χωρίς τη συμβολή του οποίου δεν θα ήταν εφικτή η συγγραφή και έκδοση του βιβλίου που κρατάτε στα χέρια σας, αυτός είναι ο Tim O'Reilly. Αν εσείς, οι αναγνώστες, κρίνετε επανειλημμένα ότι τα βιβλία του εκδοτικού οίκου O'Reilly είναι σταθερά και με διαφορά καλύτερα από πολλά άλλα βιβλία αμφιβόλου ποιότητας που κυκλοφορούν στην αγορά, τότε οφείλω να σας ενημερώσω ότι οι ρίζες της επιτυχίας εντοπίζονται σε ένα και μοναδικό άτομο, τον Tim.

Ο ατζέντης μου, ο David Rogelberg, με έπεισε ότι θα μπορούσα να βγάλω τα προς το ζην γράφοντας βιβλία, χωρίς να χρειάζεται να δουλεύω μέρα-νύχτα σε ένα γραφείο. Όλοι οι υπάλληλοι που έχουν δουλέψει κατά καιρούς στον ιστότοπο *ibiblio.org* με έχουν βοηθήσει πραγματικά να επικοινωνώ καλύτερα με τους αναγνώστες μου με ποικίλους τρόπους. Οι έπαινοι και οι αποδοκιμασίες που έστειλε ακόμα και ο τελευταίος αναγνώστης συνέβαλαν καθοριστικά στη συγγραφή αυτή της αισθητά βελτιωμένης έκδοσης. Όλοι αυτοί οι άνθρωποι αξίζουν πολλές ευχαριστίες και εύσημα. Τέλος, όπως πάντα, θα ήθελα να ευχαριστήσω μέσα από την καρδιά μου τη σύζυγό μου Beth· χωρίς την αγάπη και την υποστήριξή της, το συγκεκριμένο βιβλίο δεν θα έφτανε ποτέ στα χέρια σας.

—Elliote Rusty Harold
elharo@ibiblio.org
 5 Ιουλίου 2013

Πρόλογος ελληνικής έκδοσης

Κατά τη διάρκεια της τελευταίας δεκαετίας έχουμε γίνει μάρτυρες της εκρηκτικής εξέλιξης του Διαδικτύου, με ολοένα και περισσότερες εφαρμογές του να αποτελούν τμήμα της καθημερινότητάς μας. Αυτή η άνθηση αναμένεται να κορυφωθεί στο πλαίσιο των εφαρμογών του «*Διαδικτύου των Πραγμάτων*» (Internet of Things), οι οποίες θα αλλάξουν ριζικά τον τρόπο που χρησιμοποιούμε τους υπολογιστές, ενσωματώνοντάς τους σε όλες σχεδόν τις πτυχές των δραστηριοτήτων μας. Ο δικτυακός προγραμματισμός παρέχει τις βασικές γνώσεις για την υλοποίηση τόσο των σημερινών εφαρμογών του Διαδικτύου, όσο και των μελλοντικών εφαρμογών του «*Διαδικτύου των Πραγμάτων*».

Η Java επινοήθηκε αρχικά για τον προγραμματισμό συσκευών, και καθ' όλη την διάρκεια της ταχύτατης ανάπτυξής της παρέμεινε στενά συνδεδεμένη με το Διαδίκτυο, ίσως περισσότερο από οποιαδήποτε άλλη γλώσσα προγραμματισμού. Η ανάπτυξη προηγμένων τεχνικών JIT Compilation (Just In Time Compilation, Μεταγλώττιση κατά τον Χρόνο Εκτέλεσης) οδήγησε στο

εντυπωσιακό αποτέλεσμα να υπάρχουν προγράμματα γραμμένα σε Java τα οποία εκτελούνται σχεδόν με την ίδια, ή και μερικές φορές μεγαλύτερη, ταχύτητα από τη βελτιστοποιημένη C/C++. Η Java έχει πλέον ξεπεράσει τα προβλήματα των αρχικών εκδόσεων που σχετίζονταν με την ταχύτητα των εκτελέσιμων της, διατηρώντας ταυτόχρονα το πλήθος των σημαντικών πλεονεκτημάτων της: Είναι πλήρως αντικειμενοστραφής· από την έκδοση 8 και μετέπειτα, υποστηρίζει επιπλέον και τον συναρτησιακό προγραμματισμό (functional programming)· έχει σχεδιαστεί προσεκτικά με γνώμονα την ανάπτυξη αξιόπιστου και ασφαλούς κώδικα· είναι σχετικά απλή στην εκμάθησή της· είναι άμεσα φορητή (portable) σε όλα τα βασικά λειτουργικά συστήματα· υποστηρίζει εγγενώς την πολυνημάτωση (multithreading), τον δικτυακό προγραμματισμό (network programming) και τις συλλογές (collections)· και διαθέτει πλήθος εξαιρετικών βιβλιοθηκών που υποστηρίζουν όλους σχεδόν τους τύπους εφαρμογών. Η ευκολία και η παραγωγικότητα στην ανάπτυξη κώδικα με γλώσσες σεναρίων (scripting languages) αναμένεται να είναι βασικό χαρακτηριστικό της Java 9, η οποία θα διαθέτει ένα ενιαίο πλαίσιο ισχυρού περιβάλλοντος σεναρίων, μαζί με πολλές άλλες βελτιώσεις. Αν στα παραπάνω προσθέσουμε και το γεγονός ότι η Java αποτελεί τη δημοφιλέστερη γλώσσα τόσο στον ακαδημαϊκό όσο και στον βιομηχανικό χώρο, είναι απόλυτα δικαιολογημένη η επιλογή της για τη διδασκαλία του δικτυακού προγραμματισμού.

Η αλήθεια είναι ότι υπάρχουν λίγα καλά εκπαιδευτικά εγχειρίδια για τον δικτυακό προγραμματισμό. Ορισμένα από αυτά επικεντρώνονται στον δικτυακό προγραμματισμό με τη βιβλιοθήκη υποδοχών (socket library) του BSD Unix –κλασικό παράδειγμα αποτελεί το *Unix Network Programming* του W. Richard Stevens. Ωστόσο, μια τέτοια προγραμματιστική προσέγγιση «χαμηλού επιπέδου» καθιστά πολύπλοκη και χρονοβόρα την ανάπτυξη εφαρμογών. Για τον λόγο αυτόν προτιμήσαμε την Java, με την οποία μπορούμε εύκολα να υλοποιούμε φορητές διαδικτυακές εφαρμογές χωρίς να θυσιάζουμε την αποτελεσματικότητα. Επιλέξαμε για μετάφραση το εξαιρετικό βιβλίο *Java Network Programming* του Elliotte Rusty Harold, το οποίο εισάγει με προσιτό και κατανοητό τρόπο όλες τις βασικές πτυχές του δικτυακού προγραμματισμού. Ο συγγραφέας παρέχει αφενός μεν ακαδημαϊκές παρουσιάσεις και εξηγήσεις των βασικών εννοιών, και αφετέρου μια πληθώρα καλά σχολιασμένων παραδειγμάτων κώδικα και μικροεφαρμογών για την εμπέδωση των εννοιών.

Εντόπισα το βιβλίο *Java Network Programming* στα πλαίσια μιας πτυχιακής εργασίας στο τμήμα Μηχανικών Πληροφορικής του ΤΕΙ ΑΜΘ (πρώην Καβάλας). Συγκεκριμένα, σε συνεργασία με τον φοιτητή Γεωργιάδη Ιορδάνη Χρήστο του τμήματος Μηχανικών Πληροφορικής και στο πλαίσιο της υλοποίησης ενός συστήματος ανίχνευσης κίνησης (motion detection) με χρήση web camera, χρειάστηκε να μεταφέρουμε αναπαραστάσεις ψηφιακών εικόνων σε Java και σήματα συναγερμού (alarm) μέσω του Διαδικτύου. Η βοήθεια που μας παρείχε το βιβλίο στην υλοποίηση υποδοχών (sockets) σε Java ήταν καθοριστική για τη δημιουργία του συστήματος. Στη συνέχεια ήρθα σε επαφή με τις Εκδόσεις Κλειδάριθμος και τον κ. Γιάννη Αϊναλίδη, ο οποίος από την αρχή αντιμετώπισε θετικά την ιδέα για την έκδοση του βιβλίου στην ελληνική γλώσσα και στήριξε τη σχετική προσπάθεια.

Σε αυτό το σημείο θα ήθελα να επισημάνω τη συμβολή του Γιώργου Στάμου στη δημιουργία της ελληνικής έκδοσης, ο οποίος έκανε εξαιρετική δουλειά όσον αφορά την απόδοση του αγγλικού κειμένου στην ελληνική γλώσσα –ένα δύσκολο έργο που χρειάζεται επιστημονική κατάρτιση, εκφραστικές ικανότητες και πολύ υπομονή! Η πολύ καλή μετάφραση διευκόλυνε πολύ το δικό μου έργο, περιορίζοντας τις σχετικές διορθώσεις. Θα ήθελα να ευχαριστήσω τόσο τον κ. Αϊναλίδη όσο και τον κ. Στάμου για την άριστη συνεργασία που είχα μαζί τους καθ' όλη τη διάρκεια της υλοποίησης αυτού του εγχειρήματος.

Ελπίζουμε ότι το βιβλίο αυτό θα αποτελέσει πολύτιμο εγχειρίδιο για τη διδασκαλία του δικτυακού προγραμματισμού σε τμήματα των Ελληνικών ΑΕΙ και ΑΤΕΙ. Επιπλέον, ενδείκνυται για επαγγελματίες προγραμματιστές και ερευνητές που θέλουν να μάθουν ή να βελτιώσουν τις γνώσεις τους στον δικτυακό προγραμματισμό με αυτοδιδασκαλία.

Καλή ανάγνωση!

–Στέργιος Παπαδημητρίου, PhD
Καθηγητής
Τμήμα Μηχανικών Πληροφορικής
ΑΤΕΙ ΑΜΘ
sterg@teiemt.gr
Καβάλα, 2016

Βασικές έννοιες δικτύων

Πλέον, ο δικτυακός προγραμματισμός δεν ανήκει στη σφαίρα δραστηριότητας μόνο των ειδικών. Έχει εξελιχθεί σε θεμελιώδες στοιχείο της εργαλειοθήκης όλων των προγραμματιστών. Σήμερα, τα προγράμματα που έχουν δικτυακές δυνατότητες είναι περισσότερα από εκείνα που δεν έχουν. Πέρα από κλασικές εφαρμογές όπως το ηλεκτρονικό ταχυδρομείο, οι φυλλομετρητές Ιστού (web browsers) και η απομακρυσμένη σύνδεση (remote login), οι περισσότερες σημαντικές εφαρμογές διαθέτουν κάποιο επίπεδο ενσωματωμένης δικτύωσης. Για παράδειγμα:

- Διορθωτές κειμένου, όπως το πρόγραμμα BBEdit, ανοίγουν και αποθηκεύουν αρχεία απευθείας από/σε διακομιστές FTP.
- Ολοκληρωμένα περιβάλλοντα ανάπτυξης λογισμικού (Integrated Development Environment, IDE), π.χ. το Eclipse και το IntelliJ IDEA, επικοινωνούν με αποθετήρια πηγαίου κώδικα, όπως το GitHub και το Sourceforge.
- Επεξεργαστές κειμένου, όπως το Microsoft Word, ανοίγουν αρχεία από URL (Uniform Resource Locators, Ενιαίοι Εντοπιστές Πόρων).
- Προγράμματα προστασίας από ιούς, όπως το Norton AntiVirus, συνδέονται με τον ιστότοπο του κατασκευαστή λογισμικού κάθε φορά που εκκινείται ο υπολογιστής προκειμένου να ελέγξουν αν υπάρχουν ορισμοί νέων ιών.
- Προγράμματα αναπαραγωγής μουσικής, όπως το Winamp και το iTunes, «ανεβάζουν» στοιχεία για τη διάρκεια των μουσικών κομματιών ενός CD στη βάση δεδομένων CDDb και κατεβάζουν τους αντίστοιχους τίτλους.
- Όσοι παίζουν παιχνίδια βολών πρώτου προσώπου (first-person shooters, FPS) για πολλούς παίκτες, όπως το Halo, αλληλοσκοτώνονται με χαρά σε πραγματικό χρόνο.
- Στα σουπερμάρκετ, ταμειακές μηχανές με το λειτουργικό σύστημα SurePOS ACE της IBM επικοινωνούν με τον διακομιστή του καταστήματος σε πραγματικό χρόνο για κάθε συναλλαγή. Κάθε βράδυ, ο διακομιστής «ανεβάζει» τις αποδείξεις της ημέρας στους κεντρικούς υπολογιστές της αλυσίδας σουπερμάρκετ.
- Εφαρμογές χρονικού προγραμματισμού, όπως το Microsoft Outlook, συγχρονίζουν αυτόματα τα ημερολόγια των υπαλλήλων μιας εταιρείας.

Η Java ήταν η πρώτη γλώσσα προγραμματισμού που σχεδιάστηκε από το μηδέν για δικτυακές εφαρμογές. Παρότι προοριζόταν αρχικά για τα ιδιόκτητα δίκτυα της καλωδιακής τηλεόρασης και όχι για το Διαδίκτυο, είχε πάντα πρωτίστως κατά νου την έννοια του δικτύου. Μία από τις δύο πρώτες πραγματικές εφαρμογές Java ήταν ένας φυλλομετρητής Ιστού. Καθώς το Διαδίκτυο συνεχίζει να εξαπλώνεται, η Java έχει εξελιχθεί στην πλέον κατάλληλη γλώσσα προγραμματισμού για την ανάπτυξη της επόμενης γενιάς δικτυακών εφαρμογών.

Ένα από τα μεγαλύτερα μυστικά σχετικά με την Java είναι ότι διευκολύνει σε μεγάλο βαθμό τη συγγραφή κώδικα για δικτυακά προγράμματα. Στην πραγματικότητα, είναι πολύ πιο εύκολο να γράψει κάποιος δικτυακά προγράμματα σε Java από ό,τι σε οποιαδήποτε άλλη γλώσσα προγραμματισμού. Στο συγκεκριμένο βιβλίο θα βρείτε δεκάδες ολοκληρωμένα προγράμματα που αξιοποιούν το Διαδίκτυο. Κάποια από αυτά είναι απλά παραδείγματα παρόμοια με εκείνα που περιέχονται σε πολλά εγχειρίδια, ενώ άλλα αποτελούν πλήρως λειτουργικές εφαρμογές. Ένα στοιχείο που θα παρατηρήσετε στις πλήρως λειτουργικές εφαρμογές είναι η ελάχιστη ποσότητα του κώδικα που σχετίζεται με τη δικτύωση. Ακόμα και σε «βαριά» δικτυακά προγράμματα όπως οι διακομιστές και πελάτες Ιστού, σχεδόν όλος ο κώδικας αφορά τον χειρισμό δεδομένων ή τη διασύνδεση με τον χρήστη (user interface). Το τμήμα του προγράμματος που ασχολείται με το δίκτυο είναι συνήθως το μικρότερο και το πιο απλό. Με λίγα λόγια, είναι εύκολο για τις εφαρμογές Java να στέλνουν και να λαμβάνουν δεδομένα στο Διαδίκτυο.

Σε αυτό το κεφάλαιο θα αναφερθούμε σε βασικές έννοιες της δικτύωσης τις οποίες πρέπει να κατανοήσετε προτού ξεκινήσετε να αναπτύσσετε δικτυωμένα προγράμματα σε Java (ή σε οποιαδήποτε άλλη γλώσσα προγραμματισμού). Ακολουθώντας μια λογική διδασκαλίας που κινείται από το γενικότερο προς το ειδικότερο, θα εξηγήσουμε τι χρειάζεται να γνωρίζετε 1) για τα δίκτυα γενικά, 2) για τα δίκτυα που βασίζονται στα πρωτόκολλα IP και TCP/IP και 3) για το Διαδίκτυο. Δεν θα προσπαθήσουμε εδώ να σας διδάξουμε πώς να εγκαθιστάτε την καλωδίωση ενός δικτύου ή να διευθετείτε έναν δρομολογητή (router), αλλά θα σας δείξουμε όλα όσα πρέπει να ξέρετε προκειμένου να γράφετε κώδικα για εφαρμογές που επικοινωνούν μέσω του Διαδικτύου. Τα θέματα που καλύπτονται στο κεφάλαιο περιλαμβάνουν τη φύση των δικτύων· το μοντέλο επιπέδων (layer model) του TCP/IP· τα πρωτόκολλα IP, TCP και UDP· τα τείχη προστασίας (firewalls) και τους διακομιστές μεσολάβησης (proxy servers)· το Διαδίκτυο· και τη διαδικασία προτυποποίησης του Διαδικτύου. Οι έμπειροι «γκουρού» των δικτύων μπορούν κάλλιστα να παραλείψουν το συγκεκριμένο κεφάλαιο και να ξεκινήσουν από το επόμενο, στο οποίο θα αρχίσουμε να αναπτύσσουμε τα εργαλεία που θα χρειαστείτε για να δημιουργείτε τα δικά σας δικτυακά προγράμματα σε Java.

Δίκτυα

Ένα *δίκτυο* είναι μια συλλογή υπολογιστών και άλλων συσκευών που μπορούν να στέλνουν και να λαμβάνουν δεδομένα μεταξύ τους, περίπου σε πραγματικό χρόνο. Για τη σύνδεση του δικτύου συχνά χρησιμοποιούνται σύρματα (wires), και τα bit των δεδομένων μετατρέπονται σε ηλεκτρομαγνητικά κύματα τα οποία ταξιδεύουν μέσα από τα σύρματα. Από την άλλη πλευρά, τα ασύρματα δίκτυα μεταδίδουν δεδομένα χρησιμοποιώντας ραδιοκύματα, ενώ η πλειονότητα των μεταδόσεων σε μεγάλες αποστάσεις πραγματοποιείται με καλώδια οπτικών ινών τα οποία στέλνουν κύματα φωτός μέσα από υαλονήματα. Δεν υπάρχει κάποιος απαράβατος νόμος που να περιορίζει τη μετάδοση δεδομένων με χρήση συγκεκριμένων φυσικών μέσων. Θεωρητικά, τα δεδομένα θα μπορούσαν να μεταδίδονται από υπολογιστές που λειτουργούν με κάρβουνο, οι οποίοι στέλνουν σήματα καπνού ο ένας στον άλλο. Ωστόσο, ο χρόνος απόκρισης ενός τέτοιου δικτύου (και ο σχετικός αντίκτυπος στο περιβάλλον του πλανήτη) μάλλον δεν θα ήταν και τόσο εντυπωσιακός.

Κάθε μηχανήμα το οποίο ανήκει σε δίκτυο ονομάζεται *κόμβος* (node). Οι περισσότεροι κόμβοι είναι υπολογιστές· ωστόσο, στους κόμβους μπορεί να συγκαταλέγονται και εκτυπωτές, δρομολογητές, γέφυρες (bridges), πύλες δικτύου (gateways), μη νοήμονα τερματικά, ή ακόμα και μηχανήματα της Coca-ColaTM. Θα μπορούσατε να χρησιμοποιήσετε την Java για να δημιουργήσετε μια διασύνδεση με ένα μηχανήμα αναψυκτικών, αλλά κατά κανόνα θα την εκμεταλλεύεστε για να

«συνομιλείτε» με άλλους υπολογιστές. Οι κόμβοι που αποτελούν πλήρως λειτουργικούς υπολογιστές είναι επίσης γνωστοί ως *υπολογιστές υπηρεσίας* (hosts). Θα χρησιμοποιούμε τον όρο *κόμβος* για να αναφερόμαστε σε οποιαδήποτε συσκευή του δικτύου και τον όρο *υπολογιστές υπηρεσίας* για να αναφερόμαστε σε κόμβους που είναι υπολογιστές γενικής χρήσης.

Κάθε κόμβος του δικτύου διαθέτει *διεύθυνση*, η οποία είναι μια ακολουθία byte που τον ταυτοποιεί με μοναδικό τρόπο. Μπορείτε να θεωρήσετε αυτή την ομάδα των byte ως έναν αριθμό· γενικά, όμως, δεν υπάρχει κάποια εγγύηση ότι το πλήθος των byte σε μια διεύθυνση ή η διάταξή τους –μεγάλου άκρου (big endian) ή μικρού άκρου (little endian)– θα ταιριάζει με οποιονδήποτε στοιχειώδη αριθμητικό τύπο δεδομένων της Java. Όσο περισσότερα byte υπάρχουν σε κάθε διεύθυνση, τόσο περισσότερες θα είναι οι διαθέσιμες διευθύνσεις, και άρα πιο πολλές συσκευές θα μπορούν να συνδέονται στο δίκτυο ταυτόχρονα.

Οι διευθύνσεις χορηγούνται με διαφορετικό τρόπο ανάλογα με το είδος του δικτύου. Οι διευθύνσεις Ethernet προσαρτώνται στον εξοπλισμό Ethernet. Κάθε κατασκευαστής υλικού Ethernet χρησιμοποιεί ξεχωριστούς κωδικούς που έχουν χορηγηθεί εκ των προτέρων γι' αυτόν τον σκοπό προκειμένου να εξασφαλίζει ότι δεν θα υπάρχουν διενέξεις μεταξύ των διευθύνσεων που αντιστοιχούν στο δικό του υλικό και στο υλικό άλλων κατασκευαστών. Ο κατασκευαστής πρέπει επίσης να λαμβάνει τα κατάλληλα μέτρα ώστε να αποτραπεί το ενδεχόμενο κυκλοφορίας δύο διαφορετικών καρτών Ethernet με την ίδια διεύθυνση. Υπό φυσιολογικές συνθήκες, οι διευθύνσεις του Διαδικτύου χορηγούνται στους υπολογιστές από τον οργανισμό που είναι υπεύθυνος για τη λειτουργία τους. Ωστόσο, οι διευθύνσεις που επιτρέπεται να επιλέγει ένας οργανισμός για τους υπολογιστές του χορηγούνται από τον πάροχο υπηρεσιών Διαδικτύου (Internet Service Provider, ISP) με τον οποίο έχει συνάψει σύμβαση. Την ευθύνη για τη χορήγηση διευθύνσεων IP έχουν τέσσερις περιφερειακοί φορείς χορήγησης διευθύνσεων Διαδικτύου –π.χ. αρμόδιος φορέας για τη Βόρεια Αμερική είναι ο Αμερικανικός Φορέας για την Εκχώρηση Αριθμών του Διαδικτύου (American Registry for Internet Numbers), ή ARIN (<http://www.arin.net>)· οι σχετικές διευθύνσεις έχουν χορηγηθεί στους περιφερειακούς φορείς από τον Οργανισμό για την Εκχώρηση Ονομάτων και Αριθμών του Internet (Internet Corporation for Assigned Names and Numbers), ή ICANN (<http://www.icann.org>).

Σε κάποια είδη δικτύων, οι κόμβοι έχουν και ονόματα τα οποία βοηθούν τους ανθρώπους να τους αναγνωρίζουν, π.χ. “www.elharo.com” ή “George’s Computer”. Σε μια ορισμένη χρονική στιγμή, ένα συγκεκριμένο όνομα αντιστοιχεί σε ακριβώς μία διεύθυνση. Μολαταύτα, τα ονόματα δεν είναι οριστικά αντιστοιχισμένα σε διευθύνσεις. Τα ονόματα μπορούν να αλλάζουν ενώ οι διευθύνσεις παραμένουν ίδιες· ομοίως, οι διευθύνσεις μπορούν να αλλάζουν ενώ τα ονόματα παραμένουν ίδια. Μία διεύθυνση ενδέχεται να διαθέτει πολλά αντιστοιχισμένα ονόματα, και ένα όνομα ενδέχεται να αντιστοιχεί σε αρκετές διαφορετικές διευθύνσεις.

Όλα τα σύγχρονα δίκτυα υπολογιστών είναι *δίκτυα μεταγωγής πακέτων* (packet-switched networks): Τα δεδομένα που μετακινούνται στο δίκτυο έχουν διαιρεθεί σε τεμάχια που ονομάζονται *πακέτα* (packets), και ο χειρισμός κάθε πακέτου γίνεται ξεχωριστά. Κάθε πακέτο περιέχει πληροφορίες για την προέλευση και τον προορισμό του. Το πιο σημαντικό πλεονέκτημα της διαίρεσης των δεδομένων σε πακέτα τα οποία έχουν ξεχωριστές διευθύνσεις είναι το γεγονός ότι πακέτα από πολλές ανταλλαγές που βρίσκονται σε εξέλιξη μπορούν να «ταξιδεύουν» σε ένα σύρμα, κάτι που μειώνει αισθητά το κόστος κατασκευής ενός δικτύου: πολλοί υπολογιστές μπορούν να χρησιμοποιούν το ίδιο σύρμα χωρίς παρεμβολές. (Αντιθέτως, όταν πραγματοποιείτε μια αστική κλήση εντός του ίδιου τηλεφωνικού κέντρου χρησιμοποιώντας μια παραδοσιακή τηλεφωνική γραμμή, ουσιαστικά δεσμεύετε ένα σύρμα που ενώνει την τηλεφωνική συσκευή σας με τη συ-

σκευή του ατόμου το οποίο καλείτε. Όταν όλα τα σύρματα βρίσκονται σε χρήση, όπως συμβαίνει συνήθως κατά τη διάρκεια μιας μείζονος κατάστασης έκτακτης ανάγκης ή μιας αργίας, υπάρχει πιθανότητα να σηκώσετε το ακουστικό του τηλεφώνου σας και να μην ακούσετε τον χαρακτηριστικό τόνο επιλογής. Αν κρατήσετε το ακουστικό σηκωμένο, θα ακούσετε τελικά τον τόνο επιλογής όταν αποδεσμευθεί κάποια γραμμή. Σε κάποιες χώρες όπου οι τηλεφωνικές υπηρεσίες δεν έχουν την ίδια ποιότητα όπως στις Ηνωμένες Πολιτείες, μπορεί να χρειαστεί να περιμένετε μισή ώρα ή περισσότερο για να ακούσετε τον τόνο επιλογής.) Ένα άλλο πλεονέκτημα των πακέτων είναι ότι μπορούν να χρησιμοποιούνται αθροίσματα ελέγχου (checksums) ώστε να εξακριβώνεται αν ένα πακέτο αλλοιώθηκε κατά τη μεταφορά.

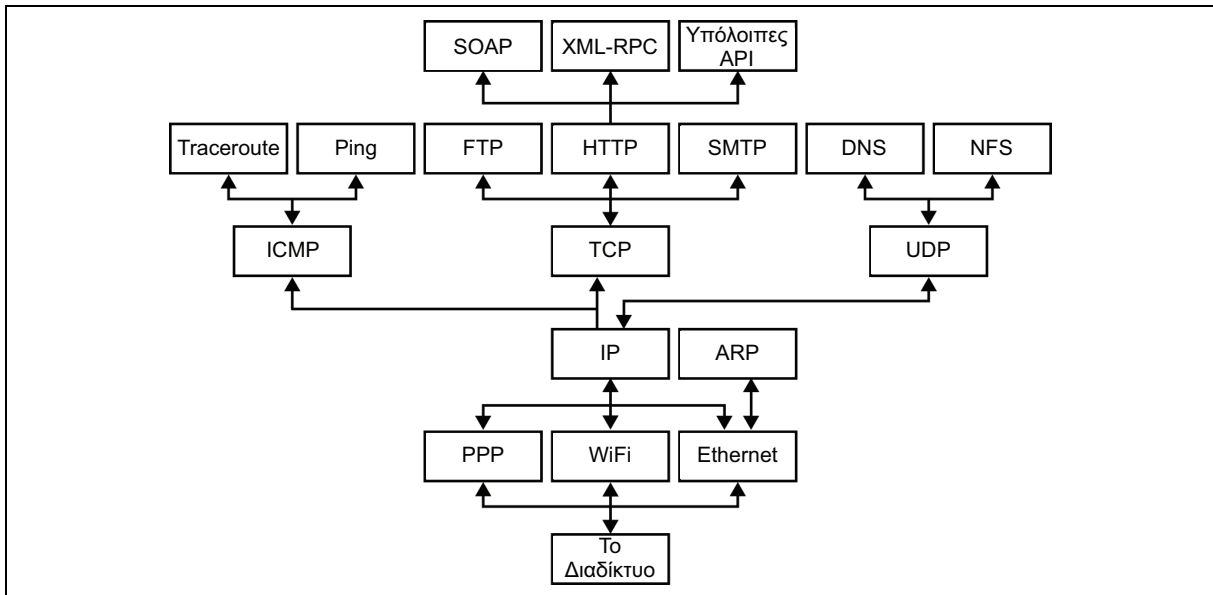
Ωστόσο, εξακολουθεί να λείπει ένα σημαντικό κομμάτι: κάποια έννοια που να περιγράφει τι πρέπει να «λένε» οι υπολογιστές για να μεταφέρουν τα δεδομένα από και προς τους εαυτούς τους. Το *πρωτόκολλο* (protocol) είναι ένα ακριβές σύνολο κανόνων που ορίζουν πώς επικοινωνούν οι υπολογιστές: τη μορφή των διευθύνσεων, τον τρόπο με τον οποίο τα δεδομένα διαιρούνται σε πακέτα κ.ο.κ. Υπάρχουν πολλά διαφορετικά πρωτόκολλα που αφορούν διαφορετικές πτυχές της δικτυακής επικοινωνίας. Για παράδειγμα, το Πρωτόκολλο Μεταφοράς Υπερ-κειμένου (Hypertext Transfer Protocol, HTTP) ορίζει τον τρόπο επικοινωνίας μεταξύ φυλλομετρητών και διακομιστών Ιστού. Στο άλλο άκρο του φάσματος, το πρότυπο IEEE 802.3 ορίζει ένα πρωτόκολλο για την κωδικοποίηση των bit ως ηλεκτρικών σημάτων σε ένα σύρμα συγκεκριμένου τύπου. Τα ανοικτά, δημοσιευμένα πρότυπα πρωτοκόλλων επιτρέπουν την επικοινωνία μεταξύ λογισμικού και εξοπλισμού που προέρχεται από διαφορετικούς κατασκευαστές. Για έναν διακομιστή Ιστού, δεν παίζει ρόλο αν ο πελάτης (client) είναι σταθμός εργασίας με λειτουργικό σύστημα Unix, κινητό τηλέφωνο με λειτουργικό σύστημα Android ή iPad, επειδή όλοι οι πελάτες «μιλούν» το ίδιο πρωτόκολλο HTTP ανεξάρτητα από το υπολογιστικό περιβάλλον.

Τα επίπεδα ενός δικτύου

Η αποστολή δεδομένων μέσω ενός δικτύου είναι μια πολύπλοκη λειτουργία που πρέπει να ρυθμιστεί προσεκτικά με βάση τόσο τα φυσικά χαρακτηριστικά του δικτύου, όσο και τον λογικό χαρακτήρα των αποστέλλομενων δεδομένων. Λογισμικό το οποίο στέλνει δεδομένα μέσω ενός δικτύου πρέπει να κατανοεί πώς μπορεί να αποφεύγει τις συγκρούσεις πακέτων, να μετατρέπει ψηφιακά δεδομένα σε αναλογικά σήματα, να ανιχνεύει και να διορθώνει σφάλματα, να δρομολογεί πακέτα από τον έναν υπολογιστή υπηρεσίας στον άλλο κ.λπ. Η διαδικασία περιπλέκεται περισσότερο όταν προστεθεί και η απαίτηση για την υποστήριξη πολλών λειτουργικών συστημάτων και την καλωδίωση ετερογενών δικτύων.

Για να αποκρυβεί το μεγαλύτερο μέρος αυτής της πολυπλοκότητας από τον προγραμματιστή και τον τελικό χρήστη των εφαρμογών, οι διαφορετικές πτυχές της δικτυακής επικοινωνίας χωρίζονται σε πολλά επίπεδα (layers). Κάθε τέτοιο επίπεδο αναπαριστά ένα διαφορετικό επίπεδο αφάιρησης (abstraction level) μεταξύ του φυσικού υλικού –δηλαδή των συρμάτων και του ηλεκτρικού ρεύματος– και των μεταδιδόμενων πληροφοριών. Θεωρητικά, κάθε επίπεδο «συνομιλεί» μόνο με τα επίπεδα που βρίσκονται ακριβώς από πάνω και από κάτω του. Ο διαχωρισμός του δικτύου σε επίπεδα μάς επιτρέπει να τροποποιήσουμε, ή ακόμα και να αντικαταστήσουμε, το λογισμικό σε ένα επίπεδο χωρίς να επηρεάσουμε τα υπόλοιπα, με την προϋπόθεση ότι οι διασυνδέσεις μεταξύ των επιπέδων παραμένουν ίδιες.

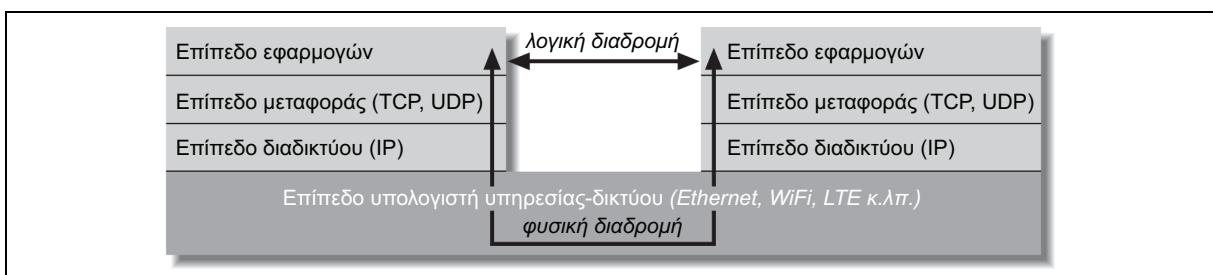
Στην Εικόνα 1-1 μπορείτε να δείτε μια στοίβα πιθανών πρωτοκόλλων τα οποία ενδέχεται να υπάρχουν στο δίκτυό σας. Ενώ τα πρωτόκολλα των μεσαίων επιπέδων εμφανίζουν μια σχετική



Εικόνα 1-1. Πρωτόκολλα σε διαφορετικά επίπεδα ενός δικτύου

συνέπεια στο μεγαλύτερο μέρος του σημερινού Διαδικτύου, δεν ισχύει το ίδιο για το ανώτερο και το κατώτερο επίπεδο. Ορισμένοι υπολογιστές υπηρεσίας χρησιμοποιούν το Ethernet, άλλοι το WiFi, κάποιιοι το πρωτόκολλο PPP (Point-to-Point Protocol, Πρωτόκολλο από Σημείο σε Σημείο), και άλλοι κάτι διαφορετικό. Ομοίως, το τι θα βρίσκεται στην κορυφή της στοίβας εξαρτάται απολύτως από τα προγράμματα που εκτελούνται σε έναν υπολογιστή υπηρεσίας. Η κρίσιμη λεπτομέρεια είναι το γεγονός ότι, όταν βρίσκεσαι στην κορυφή της στοίβας, δεν έχει ουσιαστικά σημασία τι υπάρχει στη βάση και αντίστροφα. Το μοντέλο των επιπέδων διαχωρίζει τα πρωτόκολλα εφαρμογών (που αποτελούν το κύριο θέμα αυτού του βιβλίου) από τα φυσικά χαρακτηριστικά του δικτυακού υλικού και την τοπολογία των συνδέσεων δικτύου.

Υπάρχουν πολλά διαφορετικά μοντέλα επιπέδων· κάθε μοντέλο είναι οργανωμένο ώστε να ικανοποιεί τις ανάγκες ενός συγκεκριμένου είδους δικτύου. Σε αυτό το βιβλίο χρησιμοποιούμε το –κατάλληλο για το Διαδίκτυο– καθιερωμένο μοντέλο τεσσάρων επιπέδων του TCP/IP, το οποίο φαίνεται στην Εικόνα 1-2. Στο συγκεκριμένο μοντέλο, εφαρμογές όπως ο φυλλομετρητής Ιστού Firefox και το παιχνίδι Warcraft εκτελούνται στο επίπεδο εφαρμογών (application layer) και «συνομιλούν» μόνο με το επίπεδο μεταφοράς (transport layer). Το επίπεδο μεταφοράς συνομιλεί μόνο με το επίπεδο εφαρμογών και το επίπεδο διαδικτύου (internet layer). Το επίπεδο διαδικτύου, με τη σειρά του, συνομιλεί μόνο με το επίπεδο υπολογιστή υπηρεσίας-δικτύου (host-to-network layer) και το επίπεδο μεταφοράς, και ποτέ απευθείας με το επίπεδο εφαρμογών. Το επίπεδο υπολογιστή υπηρεσίας-δικτύου μεταφέρει τα δεδομένα χρησιμοποιώντας σύρματα, καλώδια οπτικών



Εικόνα 1-2. Τα επίπεδα ενός δικτύου

ινών ή κάποιο άλλο φυσικό μέσο, με προορισμό το αντίστοιχο επίπεδο στο απομακρυσμένο σύστημα, το οποίο στη συνέχεια μεταφέρει τα δεδομένα προς τα πάνω έως το επίπεδο εφαρμογών του απομακρυσμένου συστήματος.

Για παράδειγμα, όταν ένας φυλλομετρητής Ιστού στέλνει μια αίτηση σε έναν διακομιστή Ιστού για την ανάκτηση μιας σελίδας, ο φυλλομετρητής «συνομιλεί» στην πραγματικότητα με το επίπεδο μεταφοράς του τοπικού μηχανήματος-πελάτη. Το επίπεδο μεταφοράς «σπάει» την αίτηση σε τμήματα TCP (TCP segments), προσθέτει μερικούς αριθμούς ακολουθίας (sequence numbers) και αθροίσματα ελέγχου στα δεδομένα, και μεταβιβάζει την αίτηση στο τοπικό επίπεδο διαδικτύου. Το τελευταίο «τεμαχίζει» τα τμήματα σε αυτοδύναμα πακέτα IP (IP datagrams) με το απαιτούμενο μέγεθος για το τοπικό δίκτυο, και στη συνέχεια τα μεταβιβάζει στο επίπεδο υπολογιστή υπηρεσίας-δικτύου για μετάδοση μέσω του σύρματος. Το επίπεδο υπολογιστή υπηρεσίας-δικτύου κωδικοποιεί τα ψηφιακά δεδομένα ως αναλογικά σήματα κατάλληλα για το συγκεκριμένο φυσικό μέσο μετάδοσης και στέλνει την αίτηση μέσω του σύρματος, όπου θα διαβαστεί από το επίπεδο υπολογιστή υπηρεσίας-δικτύου του απομακρυσμένου συστήματος για το οποίο προορίζεται.

Το επίπεδο υπολογιστή υπηρεσίας-δικτύου στο απομακρυσμένο σύστημα αποκωδικοποιεί τα αναλογικά σήματα και τα μετατρέπει σε ψηφιακά δεδομένα, και κατόπιν μεταβιβάζει τα αυτοδύναμα πακέτα IP που προκύπτουν στο επίπεδο διαδικτύου του διακομιστή. Το τελευταίο εκτελεί μερικούς απλούς ελέγχους για να βεβαιωθεί ότι τα αυτοδύναμα πακέτα IP δεν έχουν αλλοιωθεί, τα ανασυναρμολογεί αν έχουν «τεμαχιστεί», και τα μεταβιβάζει στο επίπεδο μεταφοράς του διακομιστή· αυτό, με τη σειρά του, ελέγχει ότι έχει παραλάβει όλα τα δεδομένα και αιτείται, αν χρειαστεί, την αναμετάδοση τυχόν τμημάτων που έχουν χαθεί ή αλλοιωθεί. (Στην πραγματικότητα, η συγκεκριμένη αίτηση διοχετεύεται ξανά προς τα κάτω μέσω του επιπέδου διαδικτύου και του επιπέδου υπολογιστή υπηρεσίας-δικτύου του διακομιστή, και επιστρέφει στο σύστημα-πελάτη· εκεί, κινείται ξανά ανοδικά έως το επίπεδο μεταφοράς του πελάτη, το οποίο αναμεταδίδει τα χαμένα δεδομένα στέλνοντας τα ξανά προς τα κάτω για να διέλθουν από τα διάφορα επίπεδα. Όσον αφορά το επίπεδο εφαρμογών, η όλη διαδικασία είναι διαφανής.) Μόλις το επίπεδο μεταφοράς του διακομιστή παραλάβει αρκετά συνεχόμενα, διαδοχικά αυτοδύναμα πακέτα, τα ανασυναρμολογεί και τα εγγράφει σε ένα ρεύμα (stream) το οποίο διαβάζεται από το πρόγραμμα-διακομιστή Ιστού που εκτελείται στο επίπεδο εφαρμογών του διακομιστή. Ο διακομιστής αποκρίνεται στην αίτηση και στέλνει την απάντηση προς τα κάτω (μέσα από τα διάφορα επίπεδα) στο σύστημά του, προκειμένου να μεταδοθεί μέσω του Διαδικτύου και να παραδοθεί στον πελάτη Ιστού.

Όπως ίσως έχετε ήδη υποπτευθεί, η πραγματική διαδικασία είναι πολύ πιο περίπλοκη. Το επίπεδο υπολογιστή υπηρεσίας-δικτύου είναι μακράν το πιο σύνθετο, και πολλές από τις σχετικές λεπτομέρειες έχουν αποκρυβεί εσκεμμένα. Για παράδειγμα, είναι απολύτως πιθανό το ενδεχόμενο τα δεδομένα που αποστέλλονται μέσω του Διαδικτύου να περάσουν από πολλούς δρομολογητές και τα αντίστοιχα επίπεδά τους προτού φτάσουν στον τελικό προορισμό. Μπορεί να πρέπει να μετατραπούν από ραδιοκύματα που ταξιδεύουν στον αέρα σε ηλεκτρικά σήματα που μεταφέρονται μέσω χάλκινων συρμάτων και μετά σε παλμούς φωτός που διέρχονται από καλώδια οπτικών ινών, και το αντίστροφο, πιθανώς περισσότερες από μία φορές. Παρ' όλα αυτά, στο 90% των περιπτώσεων, ο κώδικας της Java που θα έχετε γράψει θα λειτουργεί στο επίπεδο εφαρμογών και θα πρέπει να «συνομιλεί» μόνο με το επίπεδο μεταφοράς. Στο υπόλοιπο 10% των περιπτώσεων θα βρίσκεται στο επίπεδο μεταφοράς και θα συνομιλεί με το επίπεδο εφαρμογών ή το επίπεδο διαδικτύου. Η πολυπλοκότητα του επιπέδου υπολογιστή υπηρεσίας-δικτύου δεν σας αφορά· αυτό είναι και το νόημα του μοντέλου επιπέδων.



Αν ανατρέξετε στη βιβλιογραφία των δικτύων, είναι πιθανό να συναντήσετε ένα εναλλακτικό μοντέλο επτά επιπέδων που ονομάζεται Μοντέλο Αναφοράς της Ανοικτής Διασύνδεσης Συστημάτων (Open Systems Interconnection, OSI). Στην περίπτωση δικτυακών προγραμμάτων που έχουν αναπτυχθεί σε Java, η χρήση του μοντέλου OSI αποτελεί υπερβολή. Η μεγαλύτερη διαφορά μεταξύ του μοντέλου OSI και του μοντέλου TCP/IP το οποίο χρησιμοποιείται σε αυτό το βιβλίο είναι ότι το πρώτο χωρίζει το επίπεδο υπολογιστή υπηρεσίας-δικτύου στο επίπεδο συνδέσμου μετάδοσης δεδομένων (data link layer) και το φυσικό επίπεδο (physical layer), ενώ παράλληλα εισάγει το επίπεδο παρουσίασης (presentation layer) και το επίπεδο συνόδου (session layer), ή επίπεδο συνδιάλεξης, ανάμεσα στα επίπεδα εφαρμογών και μεταφοράς. Το μοντέλο OSI είναι πιο γενικό και περισσότερο κατάλληλο για δίκτυα που δεν στηρίζονται στο TCP/IP, αν και συχνά εξακολουθεί να είναι υπερβολικά πολύπλοκο. Σε κάθε περίπτωση, οι σχετικές κλάσεις (classes) της Java λειτουργούν σωστά μόνο σε δίκτυα TCP/IP και πάντα στο επίπεδο εφαρμογών ή μεταφοράς· συνεπώς, για τους σκοπούς του βιβλίου, δεν προκύπτει κανένα κέρδος από τη χρήση του πιο περίπλοκου μοντέλου OSI.

Όσον αφορά το επίπεδο εφαρμογών, αυτό φαίνεται να «συνομιλεί» απευθείας με το επίπεδο εφαρμογών του άλλου συστήματος· το δίκτυο δημιουργεί μια λογική διαδρομή που ενώνει τα δύο επίπεδα εφαρμογών. Είναι εύκολο να κατανοήσετε τη λογική διαδρομή αν σκεφθείτε τι συμβαίνει σε μια σύνοδο συνομιλίας (chat session) του IRC. Οι περισσότεροι από τους συμμετέχοντες σε μια συνομιλία IRC θα σας πουν ότι συνομιλούν με κάποιον άλλον. Αν τους στριμώξετε πολύ, μπορεί να σας πουν ότι συνομιλούν με τον υπολογιστή τους (στην πραγματικότητα, με το επίπεδο εφαρμογών), που συνομιλεί με τον υπολογιστή του άλλου ατόμου, ο οποίος συνομιλεί με το άλλο άτομο. Οτιδήποτε έχει βάθος που εκτείνεται σε περισσότερα από ένα επίπεδα είναι ουσιαστικά αόρατο, και αυτό ακριβώς πρέπει να ισχύει. Στη συνέχεια θα περιγράψουμε το κάθε επίπεδο με περισσότερες λεπτομέρειες.

Το επίπεδο υπολογιστή υπηρεσίας-δικτύου

Ως προγραμματιστής της Java, βρίσκεστε κοντά στην κορυφή της «τροφικής αλυσίδας» των δικτύων. Πολλά πράγματα συμβαίνουν χαμηλότερα, εκεί όπου δεν μπορεί να τα εντοπίσει το ραντάρ σας. Στο καθιερωμένο μοντέλο αναφοράς για διαδίκτυα που βασίζονται στο πρωτόκολλο IP (το μόνο είδος δικτύων που κατανοεί πραγματικά η Java), τα κρυμμένα τμήματα του δικτύου ανήκουν στο *επίπεδο υπολογιστή υπηρεσίας-δικτύου* (host-to-network layer), που είναι επίσης γνωστό ως επίπεδο συνδέσμου μετάδοσης (link layer), επίπεδο συνδέσμου μετάδοσης δεδομένων (data link layer), ή επίπεδο διασύνδεσης δικτύου (network interface layer). Το επίπεδο υπολογιστή υπηρεσίας-δικτύου ορίζει πώς μια συγκεκριμένη διασύνδεση δικτύου –π.χ. μια κάρτα Ethernet ή μια κεραία WiFi– στέλνει αυτοδύναμα πακέτα IP μέσω της φυσικής σύνδεσής της με το τοπικό δίκτυο και τον κόσμο.

Εκείνο το κομμάτι του επιπέδου υπολογιστή υπηρεσίας-δικτύου που αποτελείται από το υλικό το οποίο συνδέει διαφορετικούς υπολογιστές (σύρματα, καλώδια οπτικών ινών, ραδιοκύματα ή σήματα καπνού) είναι συχνά γνωστό ως το φυσικό επίπεδο του δικτύου. Ως προγραμματιστής της Java, δεν χρειάζεται να ανησυχείτε ποτέ για το συγκεκριμένο επίπεδο, παρά μόνο σε περιπτώσεις που κάτι πάει στραβά –π.χ. όταν το βύσμα σύνδεσης που βρίσκεται στο πίσω μέρος του

υπολογιστή σας χαλαρώσει ή κάποιος εργάτης κόψει στην κυριολεξία το καλώδιο της γραμμής T-1 που σας συνδέει με τον υπόλοιπο κόσμο. Με άλλα λόγια, η Java δεν «βλέπει» ποτέ το φυσικό επίπεδο.

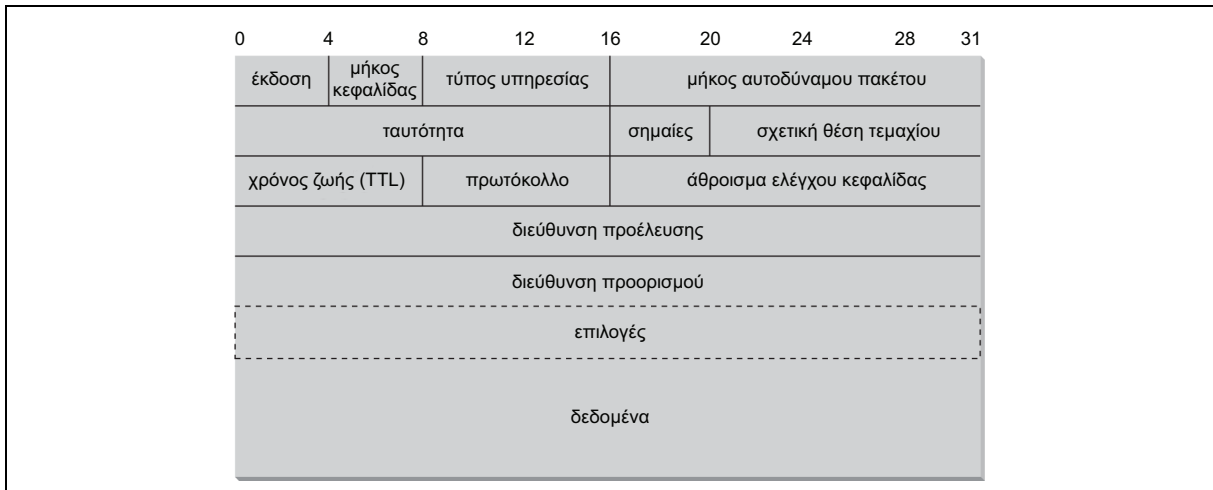
Ο βασικός λόγος για τον οποίο μπορεί να πρέπει να λάβετε υπόψη το επίπεδο υπολογιστή υπηρεσίας-δικτύου και το φυσικό επίπεδο –στη σπάνια περίπτωση που θα χρειαστεί να ασχοληθείτε με αυτά έστω και στο ελάχιστο– είναι η απόδοση. Για παράδειγμα, αν οι πελάτες σας στηρίζονται σε γρήγορες, αξιόπιστες συνδέσεις οπτικών ινών, θα σχεδιάσετε το πρωτόκολλο και τις εφαρμογές σας με διαφορετικό τρόπο από ό,τι αν βρίσκονται σε μια εξέδρα άντλησης πετρελαίου στη Βόρεια θάλασσα και χρησιμοποιούν δορυφορικές συνδέσεις με υψηλό λανθάνοντα χρόνο (latency). Οι επιλογές σας θα είναι ακόμα πιο διαφορετικές αν οι πελάτες σας εξαρτώνται από κάποιο πρόγραμμα δεδομένων 3G στο οποίο χρεώνονται ανά byte για σχετικά χαμηλό εύρος ζώνης (bandwidth). Και φυσικά, αν γράφετε κώδικα για μια καταναλωτική εφαρμογή γενικής χρήσης την οποία θα μπορούσε να χρησιμοποιήσει οποιοσδήποτε από τους εν λόγω πελάτες, κατά πάσα πιθανότητα θα προσπαθήσετε να βρείτε τη χρυσή τομή· εναλλακτικά, μπορεί να αναπτύξετε την εφαρμογή έτσι ώστε να ανιχνεύει τις δυνατότητες του εκάστοτε πελάτη και να προσαρμόζεται με δυναμικό τρόπο. Ωστόσο, ανεξάρτητα από τον τύπο των φυσικών συνδέσεων με τους οποίους ενδέχεται να έρθει αντιμέτωπη η εφαρμογή σας, οι διασυνδέσεις προγραμματισμού εφαρμογών (Application Programming Interfaces, API) που θα χρησιμοποιηθούν για την επικοινωνία μέσω των διάφορων δικτύων θα είναι οι ίδιες. Αυτό καθίσταται εφικτό λόγω του επιπέδου διαδικτύου.

Το επίπεδο διαδικτύου

Το επόμενο επίπεδο του δικτύου –και το πρώτο που θα πρέπει να σας απασχολήσει– είναι το *επίπεδο διαδικτύου* (internet layer). Στο μοντέλο OSI, το επίπεδο διαδικτύου είναι γνωστό με τον πιο γενικό όρο *επίπεδο δικτύου* (network layer). Ένα πρωτόκολλο επιπέδου δικτύου ορίζει τον τρόπο με τον οποίο τα bit και τα byte είναι οργανωμένα σε μεγαλύτερες ομάδες που ονομάζονται πακέτα, καθώς και τη μέθοδο διευθυνσιοδότησης με βάση την οποία τα διαφορετικά μηχανήματα εντοπίζουν το ένα το άλλο. Το Πρωτόκολλο Διαδικτύου (Internet Protocol, IP), που χρησιμοποιείται περισσότερο από οποιοδήποτε άλλο πρωτόκολλο επιπέδου δικτύου στον κόσμο, είναι το μόνο τέτοιο πρωτόκολλο που κατανοεί η Java.

Στην πραγματικότητα, πρόκειται για δύο πρωτόκολλα: το IPv4, που χρησιμοποιεί διευθύνσεις των 32 bit, και το IPv6, το οποίο χρησιμοποιεί διευθύνσεις των 128 bit και διαθέτει μερικά πρόσθετα τεχνικά χαρακτηριστικά που διευκολύνουν τη δρομολόγηση. Κατά τον χρόνο συγγραφής του βιβλίου, πάνω από το 90% της κυκλοφορίας στο Διαδίκτυο οφειλόταν στο IPv4· όμως, το IPv6 έχει αρχίσει να καλύπτει τη διαφορά και μπορεί κάλλιστα να έχει ξεπεράσει το IPv4 μέχρι να κυκλοφορήσει η επόμενη έκδοση του βιβλίου. Αν και μιλάμε για δύο πολύ διαφορετικά δικτυακά πρωτόκολλα τα οποία δεν μπορούν να διαλειτουργήσουν στο ίδιο δίκτυο χωρίς ειδικές πύλες και/ή πρωτόκολλα διοχέτευσης σε σήραγγα (tunneling protocols), η Java αποκρύπτει σχεδόν όλες τις διαφορές από τον προγραμματιστή.

Τόσο στο IPv4 όσο και στο IPv6, τα δεδομένα αποστέλλονται μέσω του επιπέδου διαδικτύου οργανωμένα σε πακέτα τα οποία ονομάζονται *αυτοδύναμα πακέτα* (datagrams). Κάθε αυτοδύναμο πακέτο IPv4 περιέχει μια κεφαλίδα με μήκος από 20 έως 60 byte και ένα φορτίο (payload) το οποίο περιέχει 65.515 byte δεδομένων. (Στην πράξη, η πλειοψηφία των αυτοδύναμων πακέτων IPv4 έχουν πολύ μικρότερο μέγεθος, που κυμαίνεται από μερικές δεκάδες byte έως λίγο παραπάνω από οκτώ kilobyte.) Ένα αυτοδύναμο πακέτο IPv6 περιλαμβάνει μεγαλύτερη κεφαλίδα και έως τέσσερα gigabyte δεδομένων.



Εικόνα 1-3. Η δομή ενός αυτοδύναμου πακέτου IPv4

Στην Εικόνα 1-3 μπορείτε να δείτε πώς αυτές οι διάφορες ποσότητες είναι οργανωμένες σε ένα αυτοδύναμο πακέτο IPv4. Όλα τα bit και τα byte έχουν διάταξη μεγάλου άκρου (big endian): το περισσότερο και το λιγότερο σημαντικό bit ή byte βρίσκονται στα αριστερά και στα δεξιά, αντίστοιχα.

Πέρα από τη δρομολόγηση και τη διευθυνσιοδότηση, ο δεύτερος σκοπός του επιπέδου διαδικτύου είναι να δίνει τη δυνατότητα σε διαφορετικούς τύπους επιπέδων υπολογιστή υπηρεσίας-δικτύου να «συνομιλούν» μεταξύ τους. Οι δρομολογητές του Διαδικτύου λειτουργούν ως μεταφραστές μεταξύ WiFi και Ethernet, Ethernet και DSL (Digital Subscriber Line, Ψηφιακή Συνδρομητική Γραμμή), DSL και πρωτοκόλλων οπισθοζεύξης (backhaul) οπτικών ινών κ.ο.κ. Χωρίς το επίπεδο διαδικτύου ή κάτι παρόμοιο, κάθε υπολογιστής θα μπορούσε να συνομιλήσει μόνο με άλλους υπολογιστές που ανήκουν σε δίκτυα του ίδιου τύπου. Το επίπεδο διαδικτύου είναι υπεύθυνο για την αλληλοσύνδεση ετερογενών δικτύων με χρήση ομοιογενών πρωτοκόλλων.

Το επίπεδο μεταφοράς

Τα ανεπεξέργαστα αυτοδύναμα πακέτα (raw datagrams) εμφανίζουν μερικά μειονεκτήματα. Το πιο σημαντικό από αυτά είναι ότι δεν παρέχεται εγγύηση για την παράδοσή τους. Ακόμα και αν παραδοθούν, ενδέχεται να έχουν αλλοιωθεί κατά τη μεταφορά. Το άθροισμα ελέγχου κεφαλίδας (header checksum) μπορεί να ανιχνεύσει αλλοιώσεις στην κεφαλίδα, αλλά όχι και στο κομμάτι του αυτοδύναμου πακέτου που περιέχει τα δεδομένα. Τέλος, ακόμα και στην περίπτωση που τα αυτοδύναμα πακέτα παραδοθούν χωρίς αλλοιώσεις, δεν είναι σίγουρο ότι η σειρά με την οποία παραλαμβάνονται δεν θα είναι διαφορετική από τη σειρά με την οποία αποστέλλονται. Τα μεμονωμένα αυτοδύναμα πακέτα μπορεί να ακολουθήσουν διαφορετικές διαδρομές από την προέλευση προς τον προορισμό. Το γεγονός ότι το αυτοδύναμο πακέτο Α αποστέλλεται πριν από το αυτοδύναμο πακέτο Β δεν σημαίνει υποχρεωτικά ότι το Α θα φτάσει στον προορισμό πριν από το Β.

Το *επίπεδο μεταφοράς* (transport layer) είναι εκείνο που εξασφαλίζει ότι τα πακέτα θα παραληφθούν με την ίδια σειρά όπως αποστέλλονται, καθώς και ότι δεν θα υπάρξει απώλεια ή αλλοίωση δεδομένων. Αν ένα πακέτο χαθεί, το επίπεδο μεταφοράς μπορεί να ζητήσει από τον αποστολέα να προχωρήσει σε επαναμετάδοση του πακέτου. Τα δίκτυα IP υλοποιούν αυτή τη δυνατότητα προσθέτοντας σε κάθε αυτοδύναμο πακέτο μία επιπλέον κεφαλίδα που περιέχει περισσότερες πληροφορίες. Υπάρχουν δύο κύρια πρωτόκολλα στο συγκεκριμένο επίπεδο. Το Πρωτόκολλο Ελέγχου Μετάδοσης (Transmission Control Protocol, TCP) είναι ένα πρωτόκολλο με υψηλή επι-

Προγραμματισμός δικτυακών εφαρμογών με Java

Αυτός ο πρακτικός οδηγός αποτελεί μια πλήρη εισαγωγή στην ανάπτυξη δικτυακών εφαρμογών με τη γλώσσα προγραμματισμού Java. Θα σας βοηθήσει να μάθετε να αξιοποιείτε τη βιβλιοθήκη δικτυακών κλάσεων της Java για να εκτελείτε γρήγορα και εύκολα συνήθεις δικτυακές εργασίες, όπως η συγγραφή κώδικα για πολυνηματικούς διακομιστές, η κρυπτογράφηση επικοινωνιών, η μετάδοση δεδομένων σε τοπικό δίκτυο μέσω πολυεκπομπής, και η δημοσίευση δεδομένων σε προγράμματα που εκτελούνται στην πλευρά του διακομιστή.

Με τη βοήθεια των ολοκληρωμένων και λειτουργικών προγραμμάτων που έχει συμπεριλάβει, ο συγγραφέας σας δείχνει στην πράξη πώς χρησιμοποιούνται οι μέθοδοι και οι κλάσεις που περιγράφονται στο βιβλίο. Αυτή η διεξοδικά αναθεωρημένη τέταρτη αμερικανική έκδοση καλύπτει την αρχιτεκτονική REST, το πρωτόκολλο SPDY, την ασύγχρονη είσοδο/έξοδο, και πολλές άλλες νέες τεχνολογίες.

Το βιβλίο θα σας δώσει τη δυνατότητα:

- Να εξερευνήσετε τα πρωτόκολλα στα οποία βασίζεται το Διαδίκτυο, όπως το TCP/IP και το UDP/IP
- Να μάθετε με τι τρόπο η βασική διασύνδεση API εισόδου/εξόδου της Java χειρίζεται τα δεδομένα εισόδου και εξόδου των δικτύων
- Να ανακαλύψετε τον τρόπο με τον οποίο η κλάση `InetAddress` βοηθά τα προγράμματα της Java να αλληλεπιδρούν με το DNS
- Να εντοπίσετε, να προσδιορίσετε και να «κατεβάσετε» δικτυακούς πόρους με τις κλάσεις `URI` και `URL` της Java
- Να εμβαθύνετε στο πρωτόκολλο HTTP και να μάθετε τα πάντα για την αρχιτεκτονική REST, τις κεφαλίδες HTTP και τα cookies
- Να γράψετε κώδικα για δικτυακούς διακομιστές και πελάτες χρησιμοποιώντας τις κλάσεις υποδοχών χαμηλού επιπέδου της Java
- Να βασιστείτε στην είσοδο/έξοδο χωρίς μπλοκάρισμα για να διαχειριστείτε πολλές συνδέσεις ταυτόχρονα

Ο **Elliott Rusty Harold** έχει αφιερώσει τις δύο τελευταίες δεκαετίες της ζωής του στην ανάπτυξη δικτυακών εφαρμογών με την Java. Έχει γράψει αρκετά βιβλία για την Java, την XML και την HTML, και είναι τακτικός ομιλητής σε γνωστά επαγγελματικά συνέδρια. Ταυτόχρονα, συνεχίζει να συνεισφέρει σε αρκετά έργα ανοικτού πηγαίου κώδικα, μεταξύ των οποίων οι βιβλιοθήκες Jaxen XPath και XOM.



ΕΚΔΟΣΕΙΣ
ΚΛΕΙΔΑΡΙΘΜΟΣ

Δουκίου 4, Σταθμός Λαρίσης, 10440 ΑΘΗΝΑ, Τηλ. 210-5237635
info@klidarithmos.gr www.klidarithmos.gr
www.facebook.com/klidarithmos.gr

ISBN 978-960-461-713-5



9 789604 617135