

Διεπαφή Java για χαμηλού επιπέδου διαδικτυακό προγραμματισμό.

Η γλώσσα προγραμματισμού Java παρέχει το πακέτο `java.net` που περιέχει βοηθητικές κλάσεις και διασυνδέσεις (interfaces) για την δημιουργία διαδικτυακών εφαρμογών.

Λεπτομέρειες για την τεκμηρίωση του πακέτου είναι προσβάσιμες στην ακόλουθη σελίδα.

<https://docs.oracle.com/javase/8/docs/api/java/net/package-summary.html>

Στην συνέχεια θα κάνουμε μια σύντομη περιγραφή των βασικότερων συστατικών που απαιτούνται για μια χαμηλού επιπέδου διαχείριση μιας σύνδεσης πελάτη εξυπηρετητή.

Η κλάση **`java.net.ServerSocket`** μπορεί να χρησιμοποιηθεί από έναν εξυπηρετητή (server) για την δημιουργία μια υποδοχής (socket) η οποία μπορεί να αναμένει συνδέσεις από πελάτες σε κάποια προκαθορισμένη θύρα (port).

Ένα βασικό βήμα σε κάποιον εξυπηρετητή είναι η δημιουργία ενός αντικειμένου `ServerSocket` με μια κλήση της μορφής:

```
ServerSocket listener = new ServerSocket(port);
```

όπου `port` είναι ο αριθμός θύρας ο οποίος θα χρησιμοποιηθεί για συνδέσεις TCP

Στην συνέχεια ο εξυπηρετητής μπορεί να κάνει μια κλήση της μεθόδου **`accept()`**. Η κλήση αυτή θα μπλοκάρει την εκτέλεση του προγράμματος μέχρι κάποιος πελάτης να προσπαθήσει να συνδεθεί στον server και στην συγκεκριμένη θύρα.

Για παράδειγμα η κλήση

```
Socket server;
```

```
server = listener.accept();
```

θα μπλοκάρει μέχρι την απαίτηση για σύνδεση από κάποιον πελάτη και όταν η σύνδεση ολοκληρωθεί θα επιστρέψει ένα νέο αντικείμενο τύπου `Socket` που θα είναι υπεύθυνο για την περαιτέρω επικοινωνία με τον πελάτη, ώστε το αρχικό αντικείμενο `ServerSocket` να εξακολουθήσει να αναμένει συνδέσεις με άλλους πελάτες.

Στη συνέχεια ο εξυπηρετητής μπορεί να χρησιμοποιήσει τις μεθόδους `getInputStream()` και `getOutputStream` του αντικειμένου `Socket` για να παραλάβει δύο αντικείμενα ροής δεδομένων ένα για την ροή εισόδου από την υποδοχή και ένα για την ροή εξόδου προς την υποδοχή. Ότι στέλνει ο πελάτης μπορεί ο εξυπηρετητής να το παραλαμβάνει στην ροή εισόδου και ότι ο εξυπηρετητής τοποθετεί στην ροή εξόδου θα αποστέλλεται στον πελάτη. Αυτές οι διεπαφές αποκρύπτουν τις λεπτομέρειες αποστολής και παραλαβής πακέτων που υλοποιούνται από ένα επίπεδο μεταφοράς όπως το TCP.

Η Java εν γένει παρέχει δυο βασικές οικογένειες ροών δεδομένων. Αυτές που είναι προσανατολισμένες σε bytes (δηλαδή οκτάδες από bits) και οι οποίες είναι γνωστές σαν ροές εισόδου και εξόδου (`InputStreams` και `OutputStreams`) και αυτές που είναι προσανατολισμένες σε χαρακτήρες και είναι γνωστές σαν Αναγνώστες (`Readers`) και Εγγραφείς (`Writers`). Οι αναγνώστες και εγγραφείς θεωρούνται ανωτέρου επιπέδου ροές και η συμπεριφοράς τους (δηλαδή τι ακριβώς πληροφορίες θα

σταλούν ή θα παραληφθούν) εξαρτάται από το είδος της κωδικοποίησης (character encoding) που χρησιμοποιείται. Η συμπεριφορά των ροών bytes είναι περισσότερο απόλυτη μιας και τα byte που στέλνονται ή παραλαμβάνονται δεν διερμηνεύονται ανάλογα με κάποια κωδικοποίηση (π.χ. Ascii, UTF-8, Unicode κλπ)

Τις περισσότερες φορές στον προγραμματισμό εφαρμογών είναι πιο κατάλληλο να θέλουμε να στείλουμε ή να παραλάβουμε χαρακτήρες και για τον λόγο αυτό παρέχονται δύο κλάσεις γέφυρα (InputStreamReader & OutputStreamWriter) που χρησιμοποιούνται για να μετατρέπουν ροές χαρακτήρων σε ροές byte.

Επίσης υπάρχουν διάφορες κατηγορίες Αναγνωστών (Reader) και Εγγραφέων (Writers) οι οποίες παρέχουν επιπλέον χρήσιμη λειτουργικότητα. Δύο βασικά παραδείγματα είναι οι κλάσεις `BufferedReader` και `PrintWriter`. Η `BufferedReader` δημιουργείται σαν μια ενδιάμεση ροή και σκοπό έχει την ανάκτηση μιας μεγάλης ποσότητας χαρακτήρων από την πηγή χωρίς να έχουν ζητηθεί ακόμη από τον χρήστη. Οι επιπλέον χαρακτήρες τοποθετούνται σε μια προσωρινή μνήμη (buffer) και είναι διαθέσιμοι να επιστραφούν σε μεταγενέστερες κλήσεις. Η `PrintWriter` είναι ένας Εγγραφέας (Writer) που παρέχει μεθόδους όπως η `println()` η οποία τοποθετεί τους απαραίτητους χαρακτήρες τερματισμού γραφής όπως \n σε Unix/Linux ή \r\n σε Windows. Επίσης η `PrintWriter` δίνει την δυνατότητα άμεσης αποστολής των δεδομένων στον προορισμό (flushing) όταν στέλνονται χαρακτήρες νέας γραμμής (\n). Η προεπιλογή είναι η συσσώρευση των χαρακτήρων αποστολής σε μια προσωρινή μνήμη και η πραγματική αποστολή τους όταν αυτή γεμίσει ή όταν γίνει ρητή κλήση της μεθόδου `flush`.

Τα παρακάτω δύο παραδείγματα συναρτήσεων επιδεικνύουν την δημιουργία `BufferedReader` & `PrintWriter` για κάποια υποδοχή (socket)

```
public static BufferedReader getReader(Socket s)
    throws IOException {
    return(new BufferedReader(
        new InputStreamReader(s.getInputStream())));
}

public static PrintWriter getWriter(Socket s)
    throws IOException {
    // Second argument of true means autoflush.
    return(new PrintWriter(s.getOutputStream(), true));
}
```

Μια άλλη μέθοδος της κλάσης `Socket` είναι `getInetAddress` που επιστρέφει ένα αντικείμενο της κλάσης `InetAddress` το οποίο αναπαριστά μια IP διεύθυνση. Η κλάση αυτή περιέχει μια μέθοδο `getHostName()` που επιστρέφει το όνομα τομέα (DNS name) της IP διεύθυνσης.

Παράδειγμα ενός απλού Εξυπηρετητή

Ο συμπιεσμένος κατάλογος `echoserver.zip` περιέχει τις κλάσεις που υλοποιούν έναν Εξυπηρετητή (`EchoServer`) ο οποίος επιστρέφει το κείμενο κάποιας HTTP αίτησης στον πελάτη. Περιγράφονται στο Κεφ 19 του βιβλίου *Java Servlets και Σελίδες Διακομιστή Java*.

Η κλάση EchoServer είναι μια υπο-κλάση της NetworkServer ή οποία αντικαθιστά την μέθοδο handleConnection με νέα υλοποίηση (overriding) που καθορίζει τι θα επιστρέφει ο εξυπηρετητής σαν απάντηση σε μια αίτηση.

Η κλάση NetworkServer υλοποιεί ένα γενικό Server ο οποίος μπορεί να περιμένει να δεχθεί συνδέσεις σε κάποια θύρα. Επίσης περιέχει μια παράμετρο που καθορίζει τον μέγιστο αριθμό συνδέσεων που μπορεί να εξυπηρετήσει.

Η κλάση SocketUtil περιέχει τον ορισμό δύο βοηθητικών μεθόδων getReader και getWriter που μετατρέπουν τις ροές bytes εισόδου/εξόδου κάποιας υποδοχής σε ροές χαρακτήρων.

Η κλάση ThreadedEchoServer.java είναι μια νέα έκδοση της EchoServer που την επεκτείνει με δυνατότητα πολυνημάτων ώστε κάθε νέα αίτηση να χειρίζεται από ένα διαφορετικό νήμα Java.

Για να πειραματιστείτε με την κλάση EchoServer μπορείτε να αποσυμπιέσετε το αρχείο zip σε ένα φάκελο εργασίας

Στην συνέχεια μπορείτε να ανοίξετε ένα εργαλείο γραμμής εντολών και να μεταβείτε (χρησιμοποιώντας την εντολή cd στο φάκελο εργασίας σας).

Χρησιμοποιείτε την εντολή

```
Javac *.java
```

για να μεταφράσετε τα αρχεία Java

Χρησιμοποιήστε την εντολή

```
Java EchoServer
```

Για να εκινήσετε τον EchoServer. Η προεπιλογή είναι να ακούει στην θύρα 8088.

Αφήστε τον EchoServer να τρέχει στο παράθυρο και ξεκινήστε ένα φυλλομετρητή (browser)

Δώστε το ακόλουθο URL

```
localhost:8088
```

και πατήστε Enter. Θα παρατηρήσετε ότι θα εμφανιστεί μια σελίδα της μορφής που στην ουσία είναι το κείμενο της HTTP αίτησης που στάλθηκε στον server.

