Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2005)

3ο Συνέδριο Σύρου στις ΤΠΕ

Υλοποίηση Προτύπων Επικοινωνίας με τη Βιβλιοθήκη java.net στα πλαίσια του Μαθήματος Δίκτυα Υπολογιστών του κύκλου Πληροφορικής του ΤΕΕ

Ευριπίδης Βραχνός

Βιβλιογραφική αναφορά:

Βραχνός Ε. (2024). Υλοποίηση Προτύπων Επικοινωνίας με τη Βιβλιοθήκη java.net στα πλαίσια του Μαθήματος Δίκτυα Υπολογιστών του κύκλου Πληροφορικής του ΤΕΕ. *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, *1*, 145–151. ανακτήθηκε από https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/6185

ΥΛΟΠΟΙΗΣΗ ΠΡΟΤΥΠΩΝ ΕΠΙΚΟΙΝΩΝΙΑΣ ΜΕ ΤΗ ΒΙΒΛΙΟΘΗΚΗ JAVA.NET ΣΤΑ ΠΛΑΙΣΙΑ ΤΟΥ ΜΑΘΗΜΑΤΟΣ ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ ΤΟΥ ΚΥΚΛΟΥ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΟΥ ΤΕΕ

Βραχνός Ευριπίδης
Καθηγητής Πληροφορικής
ΤΕΕ Μήλου
evrachnos@gmail.com
http://www.db-net.aueb.gr/evry/

ΠΕΡΙΛΗΨΗ

Στην εργασία αυτή κάνουμε χρήση της βιβλιοθήκης net της java, για την υλοποίηση μιας δικτυακής εφαρμογής, που έχει σαν σκοπό την καλύτερη κατανόηση κάποιων τοπολογιών και προτύπων που διδάσκονται στο μάθημα Μετάδοση Δεδομένων και Δίκτυα Υπολογιστών του Β' κύκλου της ειδικότητας πληροφορικής του ΤΕΕ. Πιο συγκεκριμένα θα δείξουμε πως μπορούν οι μαθητές με την κατάλληλη καθοδήγηση να υλοποιήσουν αρχικά την τοπολογία του δακτυλίου και στη συνέχεια το πρότυπο δακτυλίου με κουπόνι. Οι μαθητές δούλεψαν ανά ζεύγη. Ένας υλοποίησε τον εξυπηρέτη(server) και ο άλλος τον εξυπηρετούμενο(client). Πιστεύουμε ότι οι μαθητές θα πρέπει να υλοποιούν αντίστοιχες ασκήσεις στα πλαίσια του μαθήματος των δικτύων. Η βιβλιοθήκη net της java μπορεί να χρησιμοποιηθεί έτσι ώστε να αποφύγουμε ανούσιες λεπτομέρειες και να επικεντρωθούμε στις βασικές έννοιες όπως θα δείξουμε στη συνέχεια.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: *java, net, δίκτυα υπολογιστών, ΤΕΕ, δικτυακός προγραμματισμός, τοπολογία δακτυλίου με κουπόνι*

ΕΙΣΑΓΩΓΗ

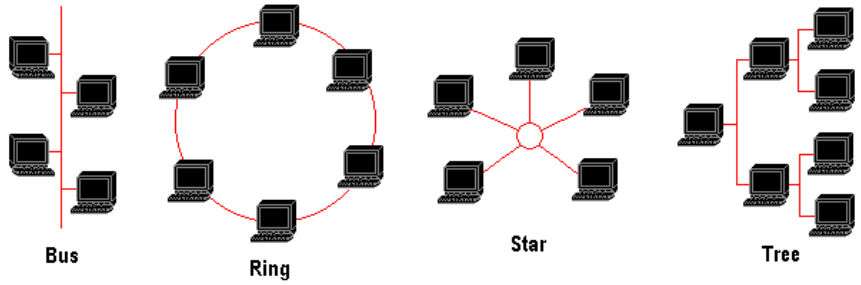
Το μάθημα Δίκτυα Υπολογιστών του Β' κύκλου του ΤΕΕ που εξετάζεται και πανελλαδικά είναι κατά βάση θεωρητικό. Το εργαστήριο του μαθήματος, όσον αφορά τα πρότυπα επικοινωνίας αναλίσκεται σε διάφορα λογισμικά προσομοίωσης που υπάρχουν και με τα οποία οι μαθητές μπορούν να παρακολουθήσουν τη μετάδοση πακέτων μεταξύ υπολογιστών με βάση τα συγκεκριμένα πρότυπα.

Εμείς ακολουθήσαμε μια άλλη προσέγγιση όσον αφορά το θέμα του εργαστηρίου. Αντί οι μαθητές να είναι παθητικοί θεατές ενός λογισμικού προσομοίωσης, ίσως θα ήταν καλύτερα να υλοποιήσουν οι ίδιοι τα πρωτόκολλα αυτά. Αυτό ίσως να ακούγεται λίγο εξωπραγματικό για τα δεδομένα του ΤΕΕ, θα δείξουμε όμως ότι δεν είναι. Το μόνο που μας έλειπε εκτός από την καλή θέληση, ήταν μια γλώσσα προγραμματισμού που να υποστηρίζει προγραμματισμό σε ένα τοπικό δίκτυο. Στην ύλη του μαθήματος *Προγραμματιστικά Εργαλεία στο Διαδίκτυο* περιλαμβάνονται τα γνωστά *java applets*. Έτσι βρήκαμε την ευκαιρία να εξηγήσουμε κάποιες βασικές έννοιες της γλώσσας και να κάνουμε κάποιες ασκήσεις ώστε οι μαθητές να κατανοήσουν τις βασικές δομές της. Όλα αυτά έγιναν με απλά παραδείγματα και με την πρόθεση από μέρους μας να κρύβουμε σε κάθε άσκηση τα δυσνόητα μέρη της γλώσσας όπως αυτό των λειτουργιών εισόδου/εξόδου. Αυτό το καταφέραμε με την υλοποίηση μερικών πολύ απλών μεθόδων για είσοδο και έξοδο τις οποίες χρησιμοποίησαν οι μαθητές.

Οι διδακτικοί μας στόχοι σε αυτή τη δραστηριότητα είναι οι παρακάτω

- Κατανόηση της αρχιτεκτονικής του μοντέλου πελάτη-εξυπηρέτη

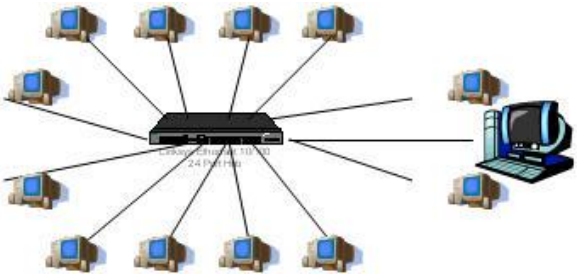
- Κατανόηση των τοπολογιών τοπικών δικτύων



Σχήμα 1. *Οι τέσσερις βασικές τοπολογίες που περιλαμβάνει το μάθημα: δίαυλος, δακτύλιος, αστέρας και δέντρο.*

ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

Η άσκηση έλαβε χώρα στο εργαστήριο του ΤΕΕ το οποίο περιλαμβάνει δώδεκα σταθμούς εργασίας με Windows 2000, και τον κεντρικό υπολογιστή που "τρέχει" Windows 2000 Server. Όλα αυτά συνδέονται σε τοπολογία αστέρα μέσω του κεντρικού διανομέα του εργαστηρίου και έτσι συνιστούν ένα τοπικό δίκτυο. Οι υπολογιστές του τοπικού μας δικτύου μοιράζονται μια σύνδεση ISDN 128-Kbit για πρόσβαση στο διαδίκτυο. Για την υλοποίηση της εργασίας οι μαθητές χρησιμοποίησαν το προγραμματιστικό περιβάλλον *JCreator* το οποίο διατίθεται δωρεάν από την ομώνυμη διεύθυνση.



Σχήμα 2. *Το περιβάλλον του εργαστηρίου.*

Ως γνωστόν κάθε υπολογιστής έχει μια μοναδική διεύθυνση στο δίκτυο γνωστή και ως *IP*. Η διεύθυνση αυτή είναι ένας 32-bit αριθμός στο *IPv4*. Στο Διαδίκτυο νέας γενιάς όπου θα έχουμε το σύστημα διευθυνσιοδότησης *IPv6* η διεύθυνση IP θα είναι ένας 128-bit αριθμός.

Η πρώτη άσκηση την οποία κλήθηκαν να υλοποιήσουν οι μαθητές ήταν μια μέθοδος η οποία ελέγχει αν ένα δοσμένο όνομα και μια διεύθυνση IP αναφέρονται στον ίδιο υπολογιστή στο διαδίκτυο. Προηγουμένως είχε εξηγηθεί η λειτουργία της κλάσης *InetAddress*, το περιεχόμενο ενός αντικειμένου της κλάσης αλλά και οι τρόποι προσπέλασης του. Στη συνέχεια δίνουμε το τμήμα κώδικα που κλήθηκαν να γράψουν οι μαθητές.

```
InetAddress address1 = InetAddress.getByName("www.db-net.aueb.gr");
System.out.println("Η διεύθυνση είναι " + address1.getHostAddress());
InetAddress address2 = InetAddress.getByName("195.251.235.65");
System.out.println("Το όνομα είναι " + address2.getHostName());
if (address1.equals(address2)) {
  System.out.println("Οι διευθύνσεις είναι ίδιες");
}
```

Κάνουμε την παραδοχή ότι οι μαθητές έχουν ήδη παρακολουθήσει ένα μάθημα σε Java και γνωρίζουν βασικές έννοιες όπως η μέθοδος *equals* και η κλήση στατικών μεθόδων. Αυτό με το υπάρχον πρόγραμμα σπουδών θα μπορούσε να γίνει στα πλαίσια των μαθημάτων *Προγραμματισμός Υπολογιστών* και *Προγραμματιστικά Εργαλεία στο Διαδίκτυο*.

Μια άλλη έννοια που θα πρέπει να εξηγηθεί στους μαθητές είναι η έννοια της θύρας (port). Θα πρέπει να καταλάβουν ότι δεν είναι κάτι υλικό αλλά έχει να κάνει σχέση με το πρόγραμμα. Για παράδειγμα θα μπορούσαμε να τους πούμε ότι μια θύρα που αντιστοιχεί σε ένα πρόγραμμα είναι ουσιαστικά ένα αρχείο ή μια περιοχή στη μνήμη η οποία ελέγχεται τακτικά για νέα μηνύματα και χαρακτηρίζεται από έναν μοναδικό αριθμό. Οι θύρες από 1 εώς 1023 είναι δεσμευμένες από το σύστημα, έτσι θα πρέπει να επιλέξουμε έναν αριθμό μεγαλύτερο του 1023, π.χ. 8192.

Επίσης από το μάθημα Λειτουργικά Συστήματα τα παιδιά έχουν έρθει σε επαφή με το μοντέλο *πελάτη-εξυπηρέτη*(client-server). Μια πρακτική εξήγηση για τη συγκεκριμένη περίπτωση χωρίς να μπούμε σε λεπτομέρειες είναι η εξής απλή:

Ο εξυπηρέτης ακούει και ο πελάτης του μιλάει ή αλλιώς ο πελάτης στέλνει πακέτα και ο εξυπηρέτης που περιμένει, τα λαμβάνει. Μπορούμε να χρησιμοποιούμε και την κλισέ φράση "Ο εξυπηρέτης ακούει στη θύρα xx". Εδώ θα μπορούσαμε να ρωτήσουμε τα παιδιά, στην περίπτωση του ταχυδρόμου που φέρνει τα γράμματα στην πόρτα μας ποιος είναι ο πελάτης και ποιος ο εξυπηρέτης. Μια συζήτηση πάνω σε αυτό το παράδειγμα θα είχε ενδιαφέρον.

Η ΥΛΟΠΟΙΗΣΗ

Ο λόγος που επιλέξαμε τη γλώσσα προγραμματισμού Java για τη συγκεκριμένη άσκηση, είναι η πολύ καλή υποστήριξη που παρέχει για την επικοινωνία μεταξύ υπολογιστών, μέσω της βιβλιοθήκη *java.net* . Αν ψάξει κανείς μπορεί να βρει αναρίθμητα παραδείγματα χρήσης αυτής της βιβλιοθήκης στο διαδίκτυο. Εμείς συμβουλευτήκαμε το βιβλίο *Thinking in Java* του *Bruce Eckel* το οποίο διατίθεται δωρεάν σε ηλεκτρονική μορφή από τη διεύθυνση *http://www.EckelObjects.com/Eckel*.

Θα χρειαστεί να υλοποιήσουμε δυο κλάσεις, μια για τον πελάτη και μια για τον εξυπηρέτη.

Ο ΕΞΥΠΗΡΕΤΗΣ (SERVER)

Για να αποκατασταθεί μια σύνδεση μεταξύ δυο υπολογιστών η βιβλιοθήκη αυτή χρησιμοποιεί τη δομή *socket (υποδοχή)*. Η δημιουργία μιας υποδοχής η οποία θα ακούει στη θύρα *port=8192* από την πλευρά του εξυπηρέτη γίνεται με την παρακάτω εντολή:

ServerSocket mySocket = **new ServerSocket**(port);

η οποία χρησιμοποιεί την κλάση *ServerSocket* για να αναπαραστήσει την μια άκρη της υποδοχής με το αντικείμενο *mySocket* που αφορά τον εξυπηρέτη. Το αντικείμενο αυτό ακούει στη θύρα *port* και περιμένει μήνυμα από κάποιον πελάτη. Η μέθοδος που υλοποιεί αυτή τη συμπεριφορά είναι η *accept*:

Socket client = mySocket.accept();

Μόλις κάποιος πελάτης στείλει μήνυμα, η accept επιστρέφει το αντικείμενο *client* το οποίο παριστάνει την υποδοχή που επιτρέπει την επικοινωνία με τον πελάτη, ώστε να μπορεί ο εξυπηρέτης να διαβάζει με τις μεθόδους εισόδου/εξόδου της γλώσσας τα μηνύματα που στέλνει ο πελάτης μέσω της υποδοχής. Εδώ πρέπει να σημειώσουμε μια παρανόηση που γίνεται λόγω του

ονόματος του αντικειμένου *ServerSocket*. Το αντικείμενο αυτό δεν χρησιμοποιείται για την επικοινωνία με τον πελάτη, απλά ακούει στη δεδομένη θύρα. Μόλις κάποιος πελάτης αιτηθεί σύνδεση τότε δημιουργείται το αντικείμενο client τύπου *Socket* για την επικοινωνία μεταξύ του πελάτη και του εξυπηρέτη. Το αντικείμενο τύπου *mySocket* συνεχίζει να ακούει στη θύρα για τυχόν επόμενες αιτήσεις.

Θα πρέπει να τονίσουμε τη διαφορά μεταξύ των δυο τύπων αντικειμένων στους μαθητές:

- Ο *ServerSocket* ακούει ("διαβάζει") αιτήσεις για σύνδεση και επιστρέφει μέσω της *accept* ένα αντικείμενο *Socket* που παριστάνει τη νέα σύνδεση.
- Ο *Socket* ακούει ("διαβάζει") πακέτα δεδομένων και όχι συνδέσεις

Στη συνέχεια δίνουμε το βασικό τμήμα του κώδικα από την υλοποίηση του εξυπηρέτη. Με πλάγια γράμματα είναι ο κώδικας που έγραψαν οι μαθητές. Τους ζητήθηκε να δώσουν τις εντολές έτσι ώστε τα μηνύματα που λαμβάνουμε από τον πελάτη, να εμφανίζονται στην οθόνη. Η σύνδεση τερματίζεται όταν λάβουμε τη λέξη "JAVA". Οι μέθοδοι CreateReader και CreateWriter δημιουργήθηκαν από τον καθηγητή ώστε να "κρύψουμε" από τους μαθητές τις λεπτομέρειες των ρευμάτων εισόδου-εξόδου, και να μην χρειάζεται να εξηγήσουμε πράγματα χωρίς καμία διδακτική αξία.

```
ServerSocket mySocket = new ServerSocket(8192);
System.out.println("Ο εξυπηρέτης ξεκίνησε " + mySocket);

Socket client = mySocket.accept();
System.out.println("Η σύνδεση έγινε "+ socket);
BufferedReader in = CreateReader(socket);

String message = in.readLine();
while (!message.equals("JAVA")) {
    System.out.println(message);
    message = in.readLine();
 }
```

Για λόγους πληρότητας δίνουμε και τις δυο μεθόδους που υλοποιήσαμε, για να απλοποιήσουμε τα πράγματα:

```
BufferedReader CreateReader(Socket socket)
{ return new BufferedReader(new InputStreamReader(socket.getInputStream()));
}

PrintWriter CreateWriter(Socket socket)
{  return new PrintWriter(new BufferedWriter(new OutputStreamWriter(
                                        socket.getOutputStream())),true);
}
```

Ο ΠΕΛΑΤΗΣ (CLIENT)

Με αντίστοιχο τρόπο γίνεται και η δημιουργία της υποδοχής από την πλευρά του πελάτη, μόνο που εδώ ο πελάτης θα πρέπει να δώσει και τη διεύθυνση του εξυπηρέτη στο διαδίκτυο:

```
Socket server = new Socket("www.db-net.aueb.gr", 80);
```

Το αντικείμενο *server* παριστάνει τη σύνδεση στη θύρα *80* του εξυπηρέτη που βρίσκεται στη διεύθυνση *www.db-net.aueb.gr* από την πλευρά του πελάτη.

Όπου *IPAddress* είναι η διεύθυνση του υπολογιστή στον οποίο τρέχει ο εξυπηρέτης. Φυσικά έχουμε συνεννοηθεί όλοι να χρησιμοποιούμε την ίδια θύρα (8192). Παρακάτω δίνεται και το τμήμα κώδικα του πελάτη. Όπως μπορεί να διακρίνει κάποιος είναι σχεδόν ίδιο με αυτό του εξυπηρέτη όσον αφορά την κύρια λειτουργία με τη διαφορά, ότι ενώ ο εξυπηρέτης διαβάζει από την υποδοχή και αντιγράφει το μήνυμα στην προκαθορισμένη έξοδο (οθόνη), ο πελάτης διαβάζει από την προκαθορισμένη είσοδο αυτά που πληκτρολογεί ο χρήστης και τα αντιγράφει στην υποδοχή για να τα λάβει ο εξυπηρέτης.

Να σημειώσουμε ότι η ανάγνωση από το πληκτρολόγιο στη Java δεν είναι κάτι απλό όπως φαίνεται και παρακάτω. Θα μπορούσαμε να είχαμε και εδώ μια μέθοδο που να επιστρέφει ένα ρεύμα εισόδου αλλά θέλαμε να φαίνεται ότι το αυτό το ρεύμα εισόδου αναφέρεται στην προκαθορισμένη είσοδο.

```
BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

Socket socket = new Socket(IPaddress, 8192);
PrintWriter out = CreateWriter(socket);

String message =  in.readLine();
while (!message.equals("JAVA"));
    out.println(message);
    message = in.readLine();
}
```

ΑΠΟ ΤΗ ΘΕΩΡΙΑ ΣΤΗΝ ΠΡΑΞΗ

Οι μαθητές δούλεψαν για τη συγκεκριμένη άσκηση κατά ζεύγη. Ο ένας υλοποίησε τον εξυπηρέτη και ο άλλος τον πελάτη. Στη συνέχεια αφού συνεννοήθηκαν για τη χρήση κοινού port και αφού βρήκε ο κάθε ένας την IP διεύθυνση του άλλου, μέσω της εντολής ping ξεκίνησαν. Ένα πολύ συνηθισμένο πρόβλημα εκτός από τα συντακτικά λάθη ήταν η αδυναμία εύρεσης μιας συνθήκης τερματισμού. Ενώ δηλαδή τα προγράμματα που έφτιαξαν δούλευαν, ελάχιστοι τα τερμάτιζαν σωστά. Το πρόβλημα ήταν ο τρόπος χρήσης της μεθόδου *equals()* για τα *Strings*.

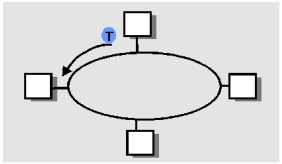
Το επόμενο βήμα ήταν να ανταλλάξουν οι μαθητές αρχεία μεταξύ τους έτσι ώστε ο κάθε ένας να τρέχει στον υπολογιστή του και τον πελάτη και τον εξυπηρέτη. Έτσι με τον πελάτη έστελνε δεδομένα ενώ στην οθόνη του εξυπηρέτη φαίνονταν αυτά που του έστελναν. Ουσιαστικά υλοποίησαν την υπηρεσία *talk* του *UNIX*.

ΥΛΟΠΟΙΗΣΗ ΔΑΚΤΥΛΙΟΥ ΜΕ ΚΟΥΠΟΝΙ

Το επόμενο βήμα ήταν να υλοποιήσουμε τις διάφορες τοπολογίες του βιβλίου. Η πρώτη ήταν η τοπολογία του δακτυλίου. Μετά από συζήτηση οι μαθητές ανακάλυψαν ότι η τοπολογία αυτή ήταν ήδη υλοποιημένη. Το μόνο που έλειπε ήταν να δώσουν στην εφαρμογή-πελάτη που είχαν υλοποιήσει τη διεύθυνση IP του επόμενου στον δακτύλιο και στον εξυπηρέτη τη διεύθυνση IP του προηγούμενου μαθητή.

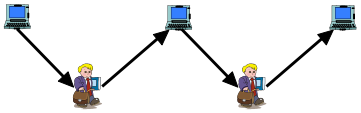
Απλά ο κάθε μαθητής θα έγραφε στο παράθυρο του πελάτη ότι διάβαζε από το παράθυρο του εξυπηρέτη.

Το επόμενο βήμα ήταν η υλοποίηση της τοπολογίας δακτυλίου με κουπόνι.



Σχήμα 3. *Τοπολογία Δακτυλίου με κουπόνι.*

Η μονή αλλαγή που πρέπει να γίνει εδώ δεν είναι στα προγράμματα αλλά στη συμπεριφορά του μαθητή.



Σχήμα 4. *Ο κάθε μαθητής αποφασίζει τι θα στείλει στον επόμενο υπολογιστή.*

Κατ'αρχήν θα πρέπει να συμφωνήσουν όλοι πιο θα είναι το αναγνωριστικό του κουπονιού. Ας πούμε το μήνυμα "###". Όταν ο μαθητής βλέπει στην οθόνη του εξυπηρέτη του αυτό το σύμβολο θα μπορεί να στέλνει ότι μήνυμα θέλει (κατάσταση μετάδοσης). Όταν τελειώσει θα στείλει το κουπόνι στον επόμενο μέσω του πελάτη του. Εάν λάβει ένα απλό μήνυμα θα πρέπει απλά να το αναπαράγει και να το αντιγράψει από την οθόνη του εξυπηρέτη στην οθόνη του πελάτη, ώστε να το στείλει στον επόμενο(κατάσταση ακρόασης).

Ήταν μια πολύ ευχάριστη δραστηριότητα, ειδικά όταν κάποιος μαθητής ήθελε να στείλει ένα μήνυμα σε κάποιον απομακρυσμένο συμμαθητή του.

Θα μπορούσαμε να είχαμε υλοποιήσει μια ακόμα υποδοχή μεταξύ του πελάτη και του εξυπηρέτη του ίδιου υπολογιστή ώστε να αντιγράφονται αυτόματα τα μηνύματα που έρχονται στον επόμενο. Θα χρειαζόταν επίσης να προσθέσουμε και λίγο κώδικα για τη διαχείριση του κουπονιού.

Έτσι όμως ο μαθητής δεν θα συμμετείχε ενεργά ως μέρος της τοπολογίας και ούτε θα καταλάβαινε εύκολα τον τρόπο λειτουργία της. Άλλωστε η πρώτη διαδικασία με ανθρώπινη παρέμβαση είναι πολύ πιο διασκεδαστική.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Σε αυτή την εργασία δείξαμε πως μπορεί να χρησιμοποιηθεί η βιβλιοθήκη java.net για να υλοποιήσουν οι μαθητές την τοπολογία δακτυλίου και πιο συγκεκριμένα το πρότυπο του δακτυλίου με κουπόνι. Δόθηκε στους μαθητές να υλοποιήσουν ένα μικρό κομμάτι του πελάτη και του εξυπηρέτη, εκθέτοντάς τους τα διάφορα στάδια του προβλήματος και δίνοντας τους την ευκαιρία σε κάθε περίπτωση να υλοποιήσουν κάτι δικό τους (ανακαλυπτική μάθηση). Στη συνέχεια οι μαθητές δούλεψαν σε ζευγάρια, όπου ο ένας είχε την ευθύνη της υλοποίησης του πελάτη και ο άλλος του εξυπηρέτη (συνεργατική μάθηση). Τέλος για να υλοποιήσουν το πρότυπο του δακτυλίου με κουπόνι χρειάστηκε να συμμετέχουν και οι ίδιοι στην τοπολογία αφού αποφάσιζαν για το πότε θα στείλουν μηνύματα και αν θα κρατήσουν ή θα προωθήσουν το κουπόνι. Με παρόμοιο τρόπο θα μπορούσαν να υλοποιηθούν και άλλες τοπολογίες. Πιστεύουμε ότι ο προγραμματισμός θα πρέπει να μπει στην ύλη του μαθήματος των Δικτύων του Β' κύκλου διότι μέσα από αυτή τη διαδικασία οι μαθητές θα αφομοιώσουν καλύτερα τις έννοιες του βιβλίου.

ΜΕΛΛΟΝΤΙΚΗ ΔΟΥΛΕΙΑ

Θα μπορούσαμε να κάνουμε πολλά πράγματα με τη βιβλιοθήκη της *net* της *java*. Για παράδειγμα θα μπορούσαμε να υλοποιήσουμε ένα μίνι δωμάτιο συνομιλίας (chat room) με χρήση πολυνηματικού προγραμματισμού στον εξυπηρέτη. Η έννοια του νήματος εξηγείται επιφανειακά στο μάθημα *Λειτουργικά Συστήματα*, ωστόσο μπορεί κανείς εύκολα να χρησιμοποιήσει την κλάση *Thread* της *Java*.

Επίσης θα μπορούσαμε να προσθέσουμε κάποια στοιχεία γραφικών στην εφαρμογή από τη βιβλιοθήκη *swing* της *Java*.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Eckel (1997), Thinking in Java, Thinking in java, http://www.EckelObjects.com/Eckel
1. Τσιλιγκιρίδης Θ. et. al. Μετάδοση Δεδομένων και Δίκτυα Υπολογιστών, τόμος ΙΙ,
2. ΥΠΕΠΘ, Παιδαγωγικό Ινστιτούτο, Αθήνα.