



ARISTOTLE  
UNIVERSITY OF  
THESSALONIKI

*Πολυτεχνική Σχολή  
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών  
Υπολογιστών*

## ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ II

ΥΠΟΧΡΕΩΤΙΚΗ ΕΡΓΑΣΙΑ

**Ομάδα ΑJ**

Μπέκου Βασιλική ΑΕΜ: 10524

Τάνια Καραγιαννίδη ΑΕΜ: 10573

# Ανάπτυξη Εφαρμογής Chat and VoIP

## Εισαγωγή

Σε αυτή την εργασία μας ζητείται να υλοποιήσουμε μια End-to-end Chat and VoIP εφαρμογή χρησιμοποιώντας γλώσσα προγραμματισμού Java. Στόχος είναι να επιτευχθεί η επικοινωνία μεταξύ δύο υπολογιστών γνωρίζοντας μόνο τις IPs των συσκευών των χρηστών μέσω των οποίων θα επιχειρούν να συνομιλήσουν, αλλά τις λογικές πόρτες που θα χρησιμοποιούν τοπικά η εφαρμογές τους. Συγκεκριμένα η εφαρμογή θα πρέπει να υλοποιεί δύο βασικές λειτουργίες επικοινωνίας:

**1)Μηνύματα κειμένου και 2)Φωνητικές κλήσεις.**

## 1<sup>ο</sup> Ζητούμενο: Μηνύματα κειμένου

Στο πρώτο ζητούμενο της εργασίας καλούμαστε να υλοποιήσουμε τον τρόπο με τον οποίο δυο χρήστες διαφορετικών συσκευών θα μπορέσουν να ανταλλάξουν μηνύματα κειμένου και να συνομιλήσουν μέσω της εφαρμογής μας.

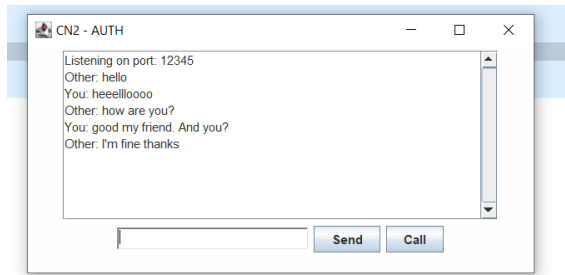
Στα πλαίσια υλοποίησης των μηνυμάτων κειμένου επειδή δεν καταφέραμε να υλοποιήσουμε σωστά τις κλήσεις έχουμε αλλάξει τη λειτουργικότητα του κουμπιού CALL. Συγκεκριμένα για να επιτευχθεί η σύνδεση ο αρχικός χρήστης θα πατήσει το κουμπί CALL και έπειτα θα πληκτρολογήσει την IP του υπολογιστή του δεύτερου χρήστη(υποθέτουμε ότι δεν την γνωρίζει εξ αρχής). Στη συνέχεια γράφει το μήνυμα που θέλει να στείλει και πατάει SEND.

## **Αρχιτεκτονική**

- Πρωτόκολλο: UDP
- Λογική Πύλη: 12345(default)

## **GUI Elements**

- Main Window: Java AWT Frame  
Τίτλος: "CN2 - AUTH"  
Μέγεθος: 500x250 pixels
- callButton: Ένα κουμπί που ανοίγει το παράθυρο διαλόγου για εισαγωγή της διεύθυνσης IP του συνομιλητή.
- sendButton: Κουμπί για επίτευξη σύνδεσης μεταξύ δυο υπολογιστών
- textArea: Περιοχή μόνο για ανάγνωση όπου εμφανίζονται τα μηνύματα που αποστέλλονται και λαμβάνονται.
- inputTextField: Μήνυμα/IP input
- dialDialog: Ένα παράθυρο διαλόγου για εισαγωγή της IP του συνομιλητή.



## Ανάλυση λειτουργικότητας συναρτήσεων του κώδικα

**Κύρια κλάση App:** Η κλάση App είναι η κύρια εφαρμογή που υλοποιεί τα WindowListener και ActionListener. Χρησιμοποιεί πρωτόκολλο UDP για επικοινωνία, επιτρέποντας σε δύο ομότιμα (peers) να ανταλλάσσουν μηνύματα μέσω της ίδιας θύρας (12345). Μέσα στην κλάση ορίζονται οι μεταβλητές μας οι οποίες έχουν αναλυθεί παραπάνω ως προς τη λειτουργικότητα τους (GUI Elements).

**Συνάρτηση App:** Αρχικοποιεί τα γραφικά στοιχεία, ορίζει τους ακροατές συμβάντων (event listeners) και ρυθμίζει το διάλογο για εισαγωγή της IP του συνομιλητή. Επιπλέον Καλεί τη μέθοδο **initNetwork()** για να ρυθμίσει την υποδοχή UDP και να ξεκινήσει το νήμα λήψης.

**Συνάρτηση createDialog:** Δημιουργεί το παράθυρο διαλόγου για εισαγωγή της διεύθυνσης IP του συνομιλητή. Ελέγχεται αν είναι έγκυρη η IP και αν είναι αποθηκεύεται και εμφανίζεται μήνυμα επιβεβαίωσης στο textArea.

**Συνάρτηση initNetwork:** Δημιουργεί ένα παράθυρο διαλόγου για εισαγωγή της διεύθυνσης IP του συνομιλητή. Εμφανίζει μήνυμα στο textArea ότι η εφαρμογή είναι έτοιμη να λαμβάνει μηνύματα. Στη συνέχεια ξεκινά το νήμα λήψης μέσω της startReceiving().

**Συνάρτηση startReceiving:** Εκτελεί ένα νήμα παρασκηνίου (receiveThread) που «ακούει» συνεχώς για εισερχόμενα UDP πακέτα. Όταν λαμβάνεται ένα πακέτο: Εξάγεται το μήνυμα και το εμφανίζει στο textArea και η διεύθυνση του αποστολέα αποθηκεύεται για μελλοντική επικοινωνία.

**Συνάρτηση actionPerformed:** Διαχειρίζεται τα κουμπιά SEND και CALL. (έχουν επεξηγηθεί παραπάνω)

**Συνάρτηση windowClosing:** Διαχειρίζεται το κλείσιμο του παραθύρου. Συγκεκριμένα σταματά το νήμα λήψης (running = false), κλείνει την υποδοχή UDP και στο τέλος τερματίζει την εφαρμογή.

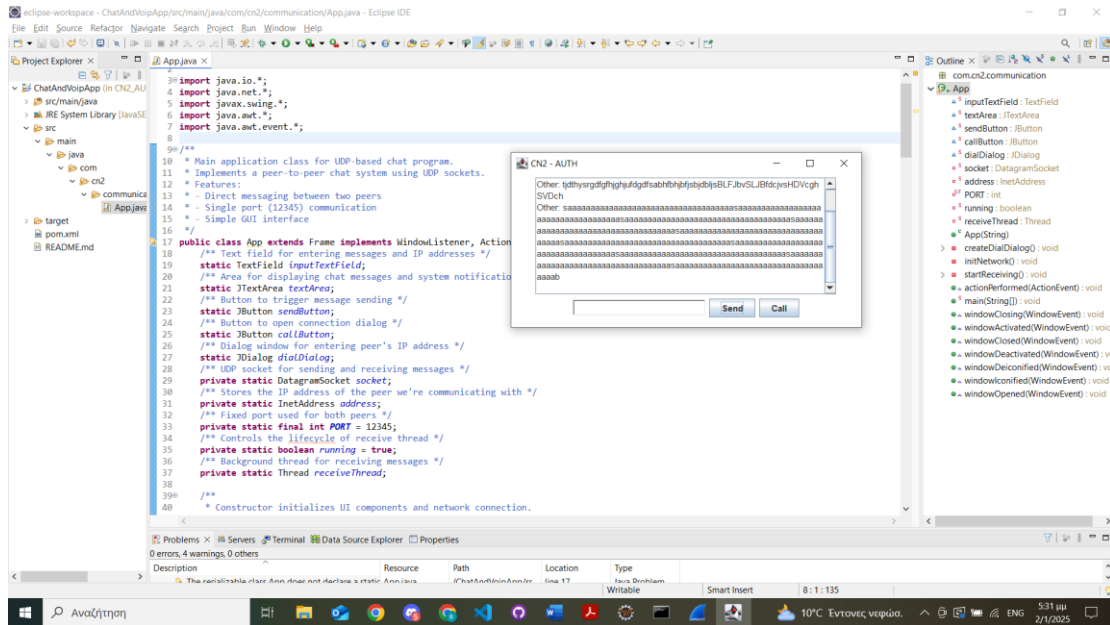
**Συνάρτηση main:** Αποτελεί το σημείο εκκίνησης του προγράμματος, δημιουργεί αρχικά ένα αντικείμενο της κλάσης App και καθιστά ορατό το παράθυρο αφού πρώτα ορίζει το μέγεθος του.

*Στον κώδικα που παραδίδουμε υπάρχουν και τα αντίστοιχα σχόλια για την επεξήγηση των βασικών λειτουργιών που περιγράφονται*

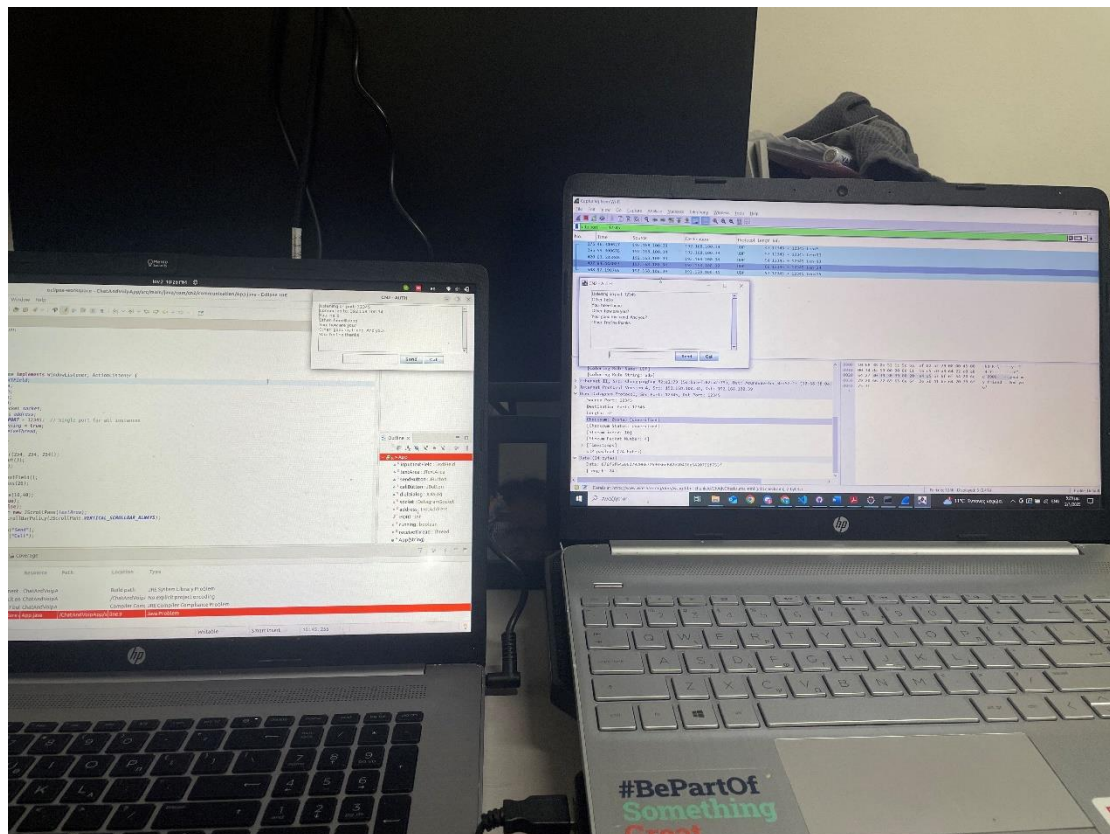
Παρακάτω βλέπουμε κάποια screenshot από το Wireshark όπου φαινονται τα πακέτα μέσω του port 12345 και το παράθυρο επικοινωνίας:

The screenshot shows the Wireshark interface with a packet capture filter set to 'udp port == 12345'. The packet list shows five packets. The selected packet (No. 5) is a UDP packet from 192.168.100.39 to 192.168.100.34, length 15. The packet details pane shows the Ethernet II, Internet Protocol Version 4, and User Datagram Protocol (Source Port: 12345, Destination Port: 12345) layers. The UDP payload (11 bytes) is shown as data. The packet bytes pane displays the raw data in hexadecimal and ASCII. A chat window titled 'CN2 - AUTH' is overlaid on the right, showing a conversation: 'Listening on port: 12345', 'Other: hello', 'You: heeelllllllll', 'Other: how are you?', 'You: good my friend. And you?', 'Other: I'm fine thanks'.

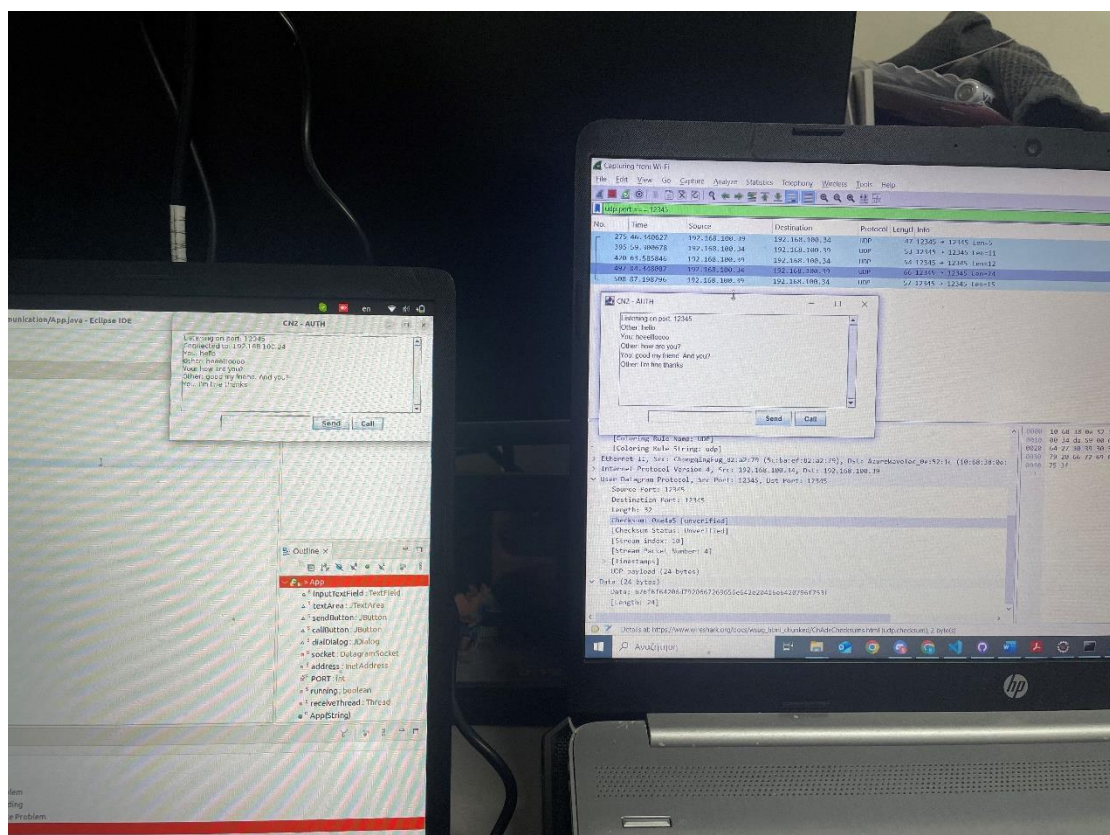
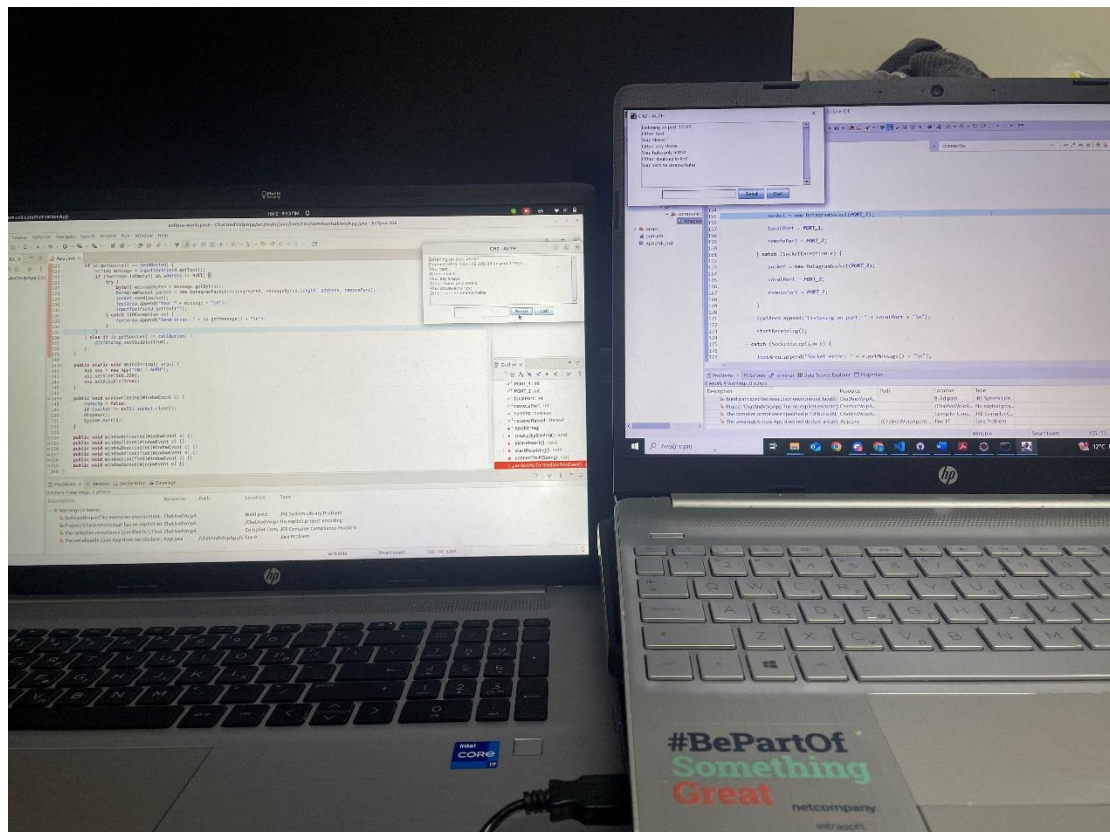
The screenshot shows the Wireshark interface with a packet capture filter set to 'udp port == 12345'. The packet list shows five packets. The selected packet (No. 5) is a UDP packet from 192.168.100.39 to 192.168.100.34, length 15. The packet details pane shows the Ethernet II, Internet Protocol Version 4, and User Datagram Protocol (Source Port: 12345, Destination Port: 12345) layers. The UDP payload (15 bytes) is shown as data. The packet bytes pane displays the raw data in hexadecimal and ASCII. A chat window titled 'CN2 - AUTH' is overlaid on the right, showing a conversation: 'Listening on port: 12345', 'Other: hello', 'You: heeelllllllll', 'Other: how are you?', 'You: good my friend. And you?', 'Other: I'm fine thanks'.



Παρακάτω δίνονται και φωτογραφίες μεταξύ της επικοινωνίας των δύο υπολογιστών μας και κάποια κομμάτια κώδικα:







Ο κώδικας και η αναφορά μας βρίσκονται και στο παρακάτω GitHub repository:  
<https://github.com/bekouvas/ComputerNetwork-Project.git>