

```
// Φοιτητής: Ευστάθιος Ιωσηφίδης
// AM: iis21027
// Άδεια χρήσης: GNU General Public License v3.0
// Use UTF-8 encoding to view the comments

import java.util.ArrayList;

public class Registry {

    //Δημιουργία λιστών

    private ArrayList<Communication> allCommunications = new
ArrayList<Communication>(); // Λίστα με όλες τις επικοινωνίες (τηλέφωνο/SMS)
    private ArrayList<Suspect> allSuspects = new ArrayList<Suspect>(); //Λίστα με
όλους τους υπόπτους

    //Μέθοδος προσθήκης υπόπτου στην λίστα όλων των υπόπτων
    public void addSuspect(Suspect aSuspect) {
        allSuspects.add(aSuspect);
    }

    //Μέθοδος προσθήκης εγγραφής επικοινωνίας
    public void addCommunication(Communication aCommunication) {
        allCommunications.add(aCommunication);

        for(int i=0;i<allSuspects.size();i++) {
            //Αναζήτηση εάν ποιος έχει στην λίστα του το πρώτο νούμερο (num1)

            if(allSuspects.get(i).getListOfNumbers().contains(aCommunication.num1)) {

                // Να ανατρέξει την λίστα υπόπτων
                for(int j=0;j<allSuspects.size();j++) {
                    // Αναζήτηση ποιος έχει στην λίστα του το δεύτερο
                    νούμερο (num2)

                    if(allSuspects.get(j).getListOfNumbers().contains(aCommunication.num2)) {
                        //Ενημέρωση της λίστας των συνεργατών

                        allSuspects.get(j).getListOfAssociates().add(allSuspects.get(i));
                        allSuspects.get(i).getListOfAssociates().add(allSuspects.get(j));
                    }
                }
            }
        }
    }

    //Μέθοδος για λήψη υπόπτου με τους περισσότερους πιθανούς συνεργάτες
    public Suspect getSuspectWithMostPartners() {
        // Δήλωση μεταβλητών εντός μεθόδου
        int tempMax=0;
        int top=0;

        //Ανατρέχω την λίστα όλων των υπόπτων
        for(int i=0;i<allSuspects.size();i++)

            //Όσοι είναι οι πιθανοί συνεργάτες του allSuspects.get(i)
            for(int j=0;j<allSuspects.get(i).getListOfAssociates().size();j++)
            {

                //Έλεγχος με τον προσωρινό μέγα ύποπτο
```

```
        if(allSuspects.get(i).getListOfAssociates().size() >
tempMax) {
                                //Ανάθεση του νέου μέγα υπόπτου
                                tempMax =
allSuspects.get(i).getListOfAssociates().size();
                                top=i;
        }
    }
    //Επιστροφή υπόπτου με τους περισσότερους πιθανούς συνεργάτες
    return allSuspects.get(top);
}

//Μέθοδος επιστροφής κλήσης με την μεγαλύτερη διάρκεια
public PhoneCall getLongestPhoneCallBetween(String number1, String number2) {

    // Δήλωση μεταβλητών-αναφορών-αντικειμένων εντός μεθόδου
    int tempMax=0;
    PhoneCall maxDuration = null;

    //Ανατρέχουμε την allCommunications
    for(Communication c : allCommunications)
    {
        //Έλεγχος αν είναι κλήση
        if(c instanceof PhoneCall)
        {
            PhoneCall phoneCall = (PhoneCall) c; //Ρητή μετατροπή σε
κλήση

            if(phoneCall.num1.equals(number1) &&
phoneCall.num2.equals(number2)) {
                                //Έλεγχος αν η διάρκεια της επικοινωνίας είναι
                                //μεγαλύτερη από την τρέχουσα μέγιστη
                                if(phoneCall.getCallDuration() > tempMax) {
                                    //Να αποθηκευτεί η μέγιστη διάρκεια ως
                                    //τρέχουσα μέγιστη
                                    tempMax = phoneCall.getCallDuration();
                                    maxDuration = phoneCall;
                                }
                            }
            }
        }
    }
    // Επιστροφή της διάρκειας
    return maxDuration;
}

//Μέθοδος που επιστρέφει μηνύματα που μεταξύ τους που περιέχουν κακές λέξεις
public ArrayList<SMS> getMessagesBetween(String number1, String number2) {

    // Δήλωση μεταβλητών-αναφορών-αντικειμένων-δομών εντός μεθόδου
    String tempMessage = "";
    ArrayList<SMS> sms = new ArrayList<SMS>();

    //Ανατρέχουμε την allCommunications
    for(Communication c : allCommunications)
    {
        // Ελέγχουμε αν είναι SMS
        if(c instanceof SMS) {
            SMS Sms = (SMS) c; //Ρητή μετατροπή σε SMS
        }
    }
}
```

```
        if(Sms.num1.equals(number1) && Sms.num2.equals(number2)) {  
            //Έλεγχος ύπαρξης κακών λέξεων  
            if(Sms.getSMS().contains("Bomb") ||  
                Sms.getSMS().contains("Attack") ||  
Sms.getSMS().contains("Explosives")  
                || Sms.getSMS().contains("Gun")) {  
                //Να αποθηκευτεί το ελεχθέν SMS ως τρέχον  
                tempMessage = Sms.getSMS();  
                //Προσθήκη στην ArrayList sms  
                sms.add(Sms);  
            }  
        }  
    }  
    //Επιστροφή του SMS  
    return sms;  
}  
  
public ArrayList<Suspect> getSuspects() {  
    return allSuspects;  
}  
}
```