

```
// 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 Main {

    public static void main(String[] args) {

        //Creation of suspect objects
        Suspect s1 = new Suspect("John Dow", "Sleepy Dog", "Barcelona");
        s1.addNumber("00496955444444");
        s1.addNumber("00496955333333");

        Suspect s2 = new Suspect("Danny Rust", "Rusty Knife", "London");
        s2.addNumber("00446998888888");

        Suspect s3 = new Suspect("Bob Robson", "Frozen Bear", "Oslo");
        s3.addNumber("00478484777777");
        s3.addNumber("00478484666666");
        s3.addNumber("00478484222222");

        Suspect s4 = new Suspect("Nick McGee", "Big Tuna", "San Francisco");
        s4.addNumber("00496955222222");

        Suspect s5 = new Suspect("Luigi Marecchio", "Joe Bananas", "Naples");
        s5.addNumber("00496955111111");

        //Creation of communication objects
        Communication[] comms = new Communication[16];

        comms[0] = new PhoneCall("00496955444444", "00478484777777", 15, 10, 2021,
127);
        comms[1] = new PhoneCall("00496955444444", "00478484777777", 16, 10, 2021,
240);
        comms[2] = new PhoneCall("00446998888888", "00496955333333", 17, 10, 2021,
52);
        comms[3] = new PhoneCall("00446998888888", "00478484777777", 18, 10, 2021,
180);
        comms[4] = new PhoneCall("00478484666666", "00496955333333", 19, 10, 2021,
305);
        comms[5] = new PhoneCall("00496955444444", "00478484222222", 20, 10, 2021,
247);
        comms[6] = new PhoneCall("00478484222222", "00496955333333", 21, 10, 2021,
32);

        comms[7] = new SMS("00496955444444", "00478484777777", 10, 10, 2021,
"fancy a drink tonight?");
        comms[8] = new SMS("00496955333333", "00446998888888", 11, 10, 2021,
"Nitro Bomb prepared");
        comms[9] = new SMS("00446998888888", "00496955444444", 12, 10, 2021,
"flying to Berlin tomorrow");
        comms[10] = new SMS("00478484777777", "00446998888888", 13, 10, 2021, "No
internet connection today");
        comms[11] = new SMS("00478484777777", "00446998888888", 14, 10, 2021, "Gun
Received from Rusty Knife");
        comms[12] = new SMS("00478484777777", "00446998888888", 15, 10, 2021,
"Metro Attack ready");
        comms[13] = new SMS("00478484666666", "00446998888888", 16, 10, 2021,
```

```
"Explosives downtown have been placed");

        comms[14] = new PhoneCall("00478484777777", "00496955222222", 22, 12,
2021, 14);
        comms[15] = new PhoneCall("00496955222222", "00496955111111", 25, 12,
2021, 9);

        //Creation of Registry object
        Registry registry = new Registry();

        registry.addSuspect(s1);
        registry.addSuspect(s2);
        registry.addSuspect(s3);
        registry.addSuspect(s4);
        registry.addSuspect(s5);

        for(int i=0; i<16; i++)
            registry.addCommunication(comms[i]);

        // Έναρξη γραφικής διεπαφής χρήστη. Στέλνω την registry για να μπορεί να
γίνει η αναζήτηση
        new SearchFrame(registry);

    }

}
```

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

public class Communication {

    // Δήλωση ιδιοτήτων
    protected String num1;
    protected String num2;
    protected int day;
    protected int month;
    protected int year;

    //Κατασκευαστής
    public Communication(String num1, String num2, int day, int month, int year){
        this.num1 = num1;
        this.num2 = num2;
        this.day = day;
        this.month = month;
        this.year = year;
    }

    //Μέθοδος εκτύπωσης
    public void printInfo() {
        System.out.println("Between " + num1 + " --- " + num2);
        System.out.println("on " + year + "/" + month + "/" + day);
    }
}
```

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

public class PhoneCall extends Communication{

    // Δήλωση ιδιότητας
    private int seconds;

    // Κατασκευαστής
    public PhoneCall(String num1, String num2, int day, int month, int year, int
seconds){
        super(num1, num2, day, month, year); //Υποκλάση της Communication
        this.seconds = seconds;
    }

    //Μέθοδος εκτύπωσης
    public void printInfo() {
        System.out.println("This phone call has the following info");
        super.printInfo(); //Λαμβάνει την printInfo της υπερκλάσης Communication
        System.out.println("Duration: " +seconds);
    }

    //Μέθοδος getCallDuration για λήψη δευτερολέπτων κλήσης
    public int getCallDuration() {
        return seconds;
    }
}
```

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

public class SMS extends Communication{

    // Δήλωση ιδιότητας
    private String message;

    //Κατασκευαστής
    public SMS(String num1, String num2, int day, int month, int year, String message)
    {
        super(num1, num2, day, month, year); //Υποκλάση της Communication
        this.message = message;
    }

    //Μέθοδος εκτύπωσης
    public void printInfo() {
        System.out.println("This SMS has the following info");
        super.printInfo(); //Λαμβάνει την printInfo της υπερκλάσης Communication
        System.out.println("Text: " +message);
    }

    //Μέθοδος getSMS για λήψη μηνύματος SMS
    public String getSMS() {
        return message;
    }
}
```

```
// 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;  
}  
}
```



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

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class SearchFrame extends JFrame {

    // Δήλωση συστατικών στοιχείων παραθύρου
    private JPanel panel = new JPanel(); // Δημιουργία νέου panel
    private JTextField susName = new JTextField("Please enter a suspect's name"); //
Εισαγωγή ονόματος υπόπτου για εύρεση
    private JButton findButton = new JButton("Find"); //Πλήκτρο Find για εύρεση
    υπόπτου
    private JButton visualizeButton = new JButton("Visualize Network"); //Πλήκτρο Find
για εύρεση υπόπτου

    //Δηλώνω ιδιότητα που δεν δείχνει πουθενά. Θέλω να δείχνει στην registry που πήρα
από την main.
    private Registry newr;
    // Από την registry, θέλω να πάρω ΜΟΝΟ τα ονόματα των υπόπτων. Θα το συνδέσω όλα
στον κατασκευαστή
    private ArrayList<Suspect> someSuspects;

    public SearchFrame (Registry r) {

        newr=r; // Γεφυρώνω την registry που έλαβα από την main με την γραφική
διεπαφή
        someSuspects=r.getSuspects(); // Από την registry, παίρνω μόνο τους
υπόπτους

        // Προσθήκη των συστατικών στο panel
        panel.add(susName);
        panel.add(findButton);
        panel.add(visualizeButton);

        // Δήλωση ButtonListener και προσθήκη του στα πλήκτρα
        ButtonListener listener = new ButtonListener();
        findButton.addActionListener(listener);
        visualizeButton.addActionListener(listener);

        // Προσθήκη του panel στο παράθυρο
        this.setContentPane(panel);

        // Δημιουργία παραθύρου
        this.setSize(350, 150);
        this.setTitle("Find Suspect");
        this.setVisible(true);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }

    // Δημιουργία κλάσης Ακροατή για το πλήκτρο Find
```

```

class ButtonListener implements ActionListener{

    @Override
    public void actionPerformed(ActionEvent e) {

        if(e.getSource().equals(findButton)) {
            String aName= susName.getText();
            boolean exists=false;
            for(int i=0;i<someSuspects.size();i++) {
                if(aName.equals(someSuspects.get(i).getName())) {
                    new SuspectInfoFrame(newr,aName,i);
                    SearchFrame.this.dispose();
                    exists=true;
                }
            }
            if(exists==false) {
                JOptionPane.showMessageDialog(null,"Suspect "
+aName + " Not Found.");
            }
        }
        else {
            SearchFrame.this.dispose();
            new VisualizeFrame(newr);
        }
    }
}

```

```

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

import java.awt.Color;
import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.Collection;
import java.util.HashSet;
import java.util.TreeSet;

import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.border.Border;

public class SuspectInfoFrame extends JFrame {

    // Χρώμα
    private Border grey;

    // Δήλωση συστατικών στοιχείων παραθύρου
    private JPanel MegaPanel , SusInfopanel , SMSpanel , Partnerspanel ,
SuggestedPartpanel;
    private JButton findSMSButton , ReturntoMainB;
    private JTextField SusName,SusCodeName,aNumberField;
    private JTextArea PartnersField , SuggPartField , SMSField , PhoneField;
    private JLabel SusLabel , SuggPartnersLabel;

    // Δηλώνω ιδιότητα τύπου Registry που δεν δείχνει πουθενά. Θέλω να δείχνει στην
registry που πήρα από το SearchFrame.
    private Registry registryC;
    // Και ιδιότητα τύπου int, η θέση του υπόπτου στην Registry.
    int position;

    public SuspectInfoFrame(Registry r, String name, int pos) {

        registryC=r; // Γεφυρώνω τα 2 παράθυρα (την registry που έλαβα από την
SearchFrame)
        this.position=pos; // Γεφυρώνω την θέση που έλαβα από την SearchFrame με
το κυρίως παράθυρο

        MegaPanel=new JPanel(); // Φτιάχνω ένα νέο αντικείμενο, το μεγάλο panel
του παράθυρο

        // Panel υπόπτου (όνομα, alias, τηλέφωνα)
        SusInfopanel = new JPanel();
        SusName = new JTextField(name); // Πεδίο κειμένου (όνομα)
        SusCodeName = new
JTextField(registryC.getSuspects().get(pos).getCodeName()); // Πεδίο κειμένου (alias)

        // Πεδίο με τηλέφωνα
        PhoneField = new JTextArea();

```

```

        String numbersOfSus = "";
        for(int i=0; i<registryC.getSuspects().get(pos).getListOfNumbers().size();
i++) {
            numbersOfSus +=
registryC.getSuspects().get(pos).getListOfNumbers().get(i).toString() + "\n" ;
        }

        PhoneField.append(numbersOfSus); // Προσθήκη των αριθμών στο συστατικό
πεδίο κειμένου

        // Προσθήκη των συστατικών στο panel υπόπτου
        SusInfopanel.add(SusName);
        SusInfopanel.add(SusCodeName);
        SusInfopanel.add(PhoneField);

        // Χρωματισμός, μέγεθος κλπ του panel
        PhoneField.setPreferredSize(new Dimension(130,80));
        grey=BorderFactory.createLineBorder(Color.gray,1);
        SusInfopanel.setBorder(grey);

        //Panel με SMS του υπόπτου με τον αριθμό τηλεφώνου που εισάγουμε
        SMSpanel = new JPanel(); // Δημιουργία Panel
        aNumberField = new JTextField(10); // Πεδίο εισαγωγής αριθμού που
αναζητούμε

        SMSField = new JTextArea(); // Πεδίο κειμένου των μηνυμάτων
        findSMSButton = new JButton("Find SMS"); // Πλήκτρο για αναζήτηση

        // Προσθήκη των συστατικών στοιχείων στο panel
        SMSpanel.add(aNumberField);
        SMSpanel.add(SMSField);
        SMSpanel.add(findSMSButton);

        // Χρωματισμός panel και μέγεθος πεδίου
        SMSpanel.setBorder(grey);
        SMSField.setPreferredSize(new Dimension(200,150));

        // Δήλωση ButtonListener και προσθήκη του στο πλήκτρο findSMSButton
        FindSmsButtonListener findlistener= new FindSmsButtonListener();
        findSMSButton.addActionListener(findlistener);

        //Panel πιθανών συνεργατών του υπόπτου
        Partnerspanel = new JPanel(); // Δημιουργία panel
        SuspLabel = new JLabel("Partners"); // Δημιουργία συστατικού ετικέτας
        PartnersField = new JTextArea(); // Δημιουργία συστατικού πεδίου εμφάνισης
πιθανών συνεργατών

        Collection<String> posPartList = new TreeSet<String>(); // Φτιάχνω ένα
TreeSet για να τους βάλω αλφαβητικά

        for(int
i=0; i<registryC.getSuspects().get(pos).getListOfAssociates().size(); i++) {
            // Προσθέτω όλους τους πιθανούς αλφαβητικά

posPartList.add(registryC.getSuspects().get(pos).getListOfAssociates().get(i).getName()
+" "+ registryC.getSuspects().get(pos).getListOfAssociates().get(i).getCodeName() +
"\n");
        }

        Collection<String> noDups = new HashSet<String>(posPartList); // Φτιάχνω
ένα HashSet για να βγάλω τα διπλά και βάζω το TreeSet
        for(String ps : noDups)

```

```

    {
        PartnersField.append(ps); // Προσθέτω στο πεδίο τα ονόματα
    }

    // Προσθήκη των συστατικών στοιχείων στο panel
    Partnerspanel.add(SusLabel);
    Partnerspanel.add(PartnersField);
    // Χρωματισμός και μέγεθος panel
    Partnerspanel.setBorder(grey);
    PartnersField.setPreferredSize(new Dimension(200,150));

    //Panel προτεινόμενων πιθανών συνεργατών υπόπτου
    SuggestedPartpanel = new JPanel(); // Δημιουργία Panel
    SuggPartnersLabel = new JLabel("Suggested Partners ----->"); // Δημιουργία
    συστατικού ετικέτας
    SuggPartField = new JTextArea(); // Συστατικό για εμφάνιση των ονομάτων

    Collection<String> sugPosPartList = new TreeSet<String>(); // Φτιάχνω ένα
    TreeSet για να τους βάλω αλφαβητικά

    for(int
i=0;i<registryC.getSuspects().get(pos).getSuggestedPossiblePartners().size();i++) {
        // Προσθέτω όλους τους πιθανούς αλφαβητικά

    posPartList.add(registryC.getSuspects().get(pos).getSuggestedPossiblePartners().get(i).getName()
    +"\n");
    }

    Collection<String> noSugDups = new HashSet<String>(sugPosPartList);//
    Φτιάχνω ένα HashSet για να βγάλω τα διπλά και βάζω το TreeSet
    for(String sps : noSugDups)
    {

        SuggPartField.append(sps); // Προσθέτω στο πεδίο τα ονόματα
    }

    // Προσθήκη των συστατικών στοιχείων στο panel
    SuggestedPartpanel.add(SuggPartnersLabel);
    SuggestedPartpanel.add(SuggPartField);

    // Χρωματισμός και μέγεθος panel
    SuggestedPartpanel.setBorder(grey);
    SuggPartField.setPreferredSize(new Dimension(200,150));

    //Προσθήκη όλων panels στο κεντρικό panel
    MegaPanel.add(SusInfopanel);
    MegaPanel.add(SMSpanel);
    MegaPanel.add(Partnerspanel);
    MegaPanel.add(SuggestedPartpanel);

    // Δημιουργία κουμπιού επιστροφής και προσθήκη του στο κεντρικό panel
    ReturntoMainB= new JButton("Back to Search Screen");
    MegaPanel.add(ReturntoMainB);

    // Δήλωση ButtonListener και προσθήκη του στο πλήκτρο επιστροφής στην
    οθόνη αναζήτησης
    ReturnButtonListener listener = new ReturnButtonListener();
    ReturntoMainB.addActionListener(listener);

```

```

        // Προσθήκη του κεντρικού panel (με όλα τα υπο-panel) στο παράθυρο
        this.setContentPane(MegaPanel);

        // Δημιουργία του παραθύρου
        this.setTitle("Suspect Page");
        this.setSize(440,700);
        this.setVisible(true);
        this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    }

    // Δημιουργία κλάσης Ακροατή για το πλήκτρο ReturntoMainB για επιστροφή στην οθόνη
    αναζήτησης
    class ReturnButtonListener implements ActionListener{
        public void actionPerformed(ActionEvent e){
            new SearchFrame(registryC);
            SuspectInfoFrame.this.dispose();
        }
    }

    // Δημιουργία κλάσης Ακροατή για το πλήκτρο findSMSButton για επιστροφή στην οθόνη
    αναζήτησης
    class FindSmsButtonListener implements ActionListener{
        public void actionPerformed(ActionEvent e) {

            String numberGiven = aNumberField.getText(); // Παίρνω τον αριθμό
            που εισήχθη

            ArrayList<SMS> messages = new ArrayList<SMS>(); // Φτιάχνω ένα
            αντικείμενο λιστας τύπου μυνημάτων SMS

            //Προσθέτω στην λίστα τα μηνύματα που ανταλλάξανε ο ύποπτος (και
            ότι αριθμούς έχει) με το συγκεκριμένο νούμερο.
            for(int i=0;
            i<registryC.getSuspects().get(position).getListOfNumbers().size(); i++){

            messages.addAll(registryC.getMessagesBetween(registryC.getSuspects().get(position).getListOfNumbers().get
            numberGiven));

            }

            // Δημιουργία ενός αντικειμένου String με όλα τα μηνύματα
            String allMess = "";
            for(int i=0; i<messages.size(); i++)
                allMess += messages.get(i).getSMS() + "\n";

            // Προσθήκη του αντικειμένου με τα μηνύματα στο panel
            SMSField.setText(allMess);

        }
    }
}

```

```
// 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 Suspect {

    //Δήλωση ιδιοτήτων υπόπτου Suspect
    private String name;
    private String alias;
    private String town;

    //Δημιουργία λιστών
    private ArrayList<String> phoneBook = new ArrayList<String>(); //λίστα αριθμών
    τηλεφώνων που χρησιμοποιεί ο ύποπτος
    private ArrayList<Suspect> possiblePartners = new ArrayList<Suspect>(); //λίστα
    των πιθανών συνεργατών υπόπτου
    private ArrayList<Suspect> commonPartners = new ArrayList<Suspect>(); //λίστα με
    κοινούς πιθανούς συνεργάτες
    private ArrayList<Suspect> suggestedPartners = new ArrayList<Suspect>(); // λίστα
    με προτεινόμενους πιθανούς συνεργάτες

    //Κατασκευαστής
    public Suspect(String name, String alias, String town){
        this.name = name;
        this.alias = alias;
        this.town = town;
    }

    //Μέθοδοι

    //Πρόσθεση αριθμού στην λίστα των αριθμών του υπόπτου.
    public void addNumber(String number) {
        phoneBook.add(number);
    }

    //Προσθέτει τον ύποπτο στην λίστα άλλου υπόπτου των πιθανών συνεργατών
    public void addSuspect(Suspect aSuspect) {

        if(!possiblePartners.contains(aSuspect)){
            possiblePartners.add(aSuspect);
        }
    }

    //Έλεγχος εάν δυο ύποπτοι είναι συνδεδεμένοι.
    public boolean isConnectedTo(Suspect aSuspect) {

        if(possiblePartners.contains(aSuspect)){
            return true;
        }
        else{
            return false;
        }
    }

    //Επιστρέφει μια λίστα με όλους τους κοινούς συνεργάτες των δυο υπόπτων
    public ArrayList<Suspect> getCommonPartners(Suspect aSuspect) {

        for(int i=0;i<possiblePartners.size();i++) {
```

```
        if(possiblePartners.get(i).isConnectedTo(aSuspect) && !
commonPartners.contains(possiblePartners.get(i))) {

            commonPartners.add(possiblePartners.get(i));

        }

    }

    return commonPartners;
}

// Μέθοδος δημιουργίας και λήψης λίστας προτεινόμενων πιθανών συνεργατών
public ArrayList<Suspect> getSuggestedPossiblePartners(){

    Suspect aSuspect = null; // Δήλωση ενός υπόπτου (null)

    // Ανατρέχουμε την λίστα των πιθανών συνεργατών
    for(int i=0; i<possiblePartners.size(); i++)
    {
        if(possiblePartners.get(i).isConnectedTo(aSuspect) && !
possiblePartners.contains(aSuspect))
        {
            suggestedPartners.add(aSuspect);
        }
    }
    return suggestedPartners;
}

// Εκτύπωση της λίστας των πιθανών συνεργατών
public void printInfo(Suspect aSuspect){
    for(int i=0; i<possiblePartners.size(); i++)
        System.out.println(possiblePartners.get(i).name + ", " +
possiblePartners.get(i).alias);
}

// Μέθοδος getName
public String getName() {
    return name;
}

//Μέθοδος getCodeName
public String getCodeName() {
    return alias;
}

public ArrayList<String> getListOfNumbers() {
    return phoneBook;
}

public ArrayList<Suspect> getListOfAssociates(){
    return possiblePartners;
}

}
```



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

import java.awt.Dimension;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JTextField;
import java.awt.BorderLayout;

import edu.uci.ics.jung.algorithms.layout.CircleLayout;
import edu.uci.ics.jung.algorithms.shortestpath.DistanceStatistics;
import edu.uci.ics.jung.graph.Graph;
import edu.uci.ics.jung.graph.SparseGraph;
import edu.uci.ics.jung.visualization.VisualizationImageServer;
import edu.uci.ics.jung.visualization.decorators.ToStringLabeller;

public class VisualizeFrame extends JFrame {

    private Registry nuevaRegistry; // Εισάγω ένα αντικείμενο Registry για να λάβω το
    αντικείμενο που στέλνει από SearchFrame

    private JTextField diameterResult = new JTextField();
    private Double networkDiameterValue;
    private VisualizationImageServer visual;

    private JFrame vnFrame = new JFrame();

    public VisualizeFrame(Registry r) {

        nuevaRegistry = r; // Έχω πρόσβαση στο αντικείμενο Registry που ήρθε από
        το SearchFrame

        // Δημιουργώ αντικείμενο γράφου
        Graph graph = new SparseGraph();

        // Διατρέχω την registry και προσθέτω κορυφές στον γράφο, τα κωδικά
        ονόματα των υπόπτων
        for(Suspect suspect : nuevaRegistry.getSuspects())
        {
            graph.addVertex(suspect.getCodeName());
        }

        int counter=0;
        for(Suspect suspect1 : nuevaRegistry.getSuspects())
            for(Suspect suspect2 : suspect1.getListOfAssociates())
            {
                graph.addEdge(counter, suspect1.getCodeName(),
                suspect2.getCodeName());
                counter++;
            }

        // Σχεδιασμός γράφου
        visual = new VisualizationImageServer(new CircleLayout(graph), new
        Dimension(250,250));
        visual.getRenderContext().setVertexLabelTransformer(new
```

```
ToStringLabeller());

    // Υπολογισμός διαμέτρου γράφου
    networkDiameterValue = DistanceStatistics.diameter(graph);
    diameterResult.setText("Diameter = " + networkDiameterValue);

    // Προσθήκη γραφικών συστατικών στο Frame
    vnFrame.getContentPane().add(visual);
    vnFrame.add(diameterResult, BorderLayout.PAGE_END);
    vnFrame.pack();

    // Δημιουργώ το παράθυρο
    vnFrame.setVisible(true);
    vnFrame.setTitle("Suspects Network");
    vnFrame.setSize(340, 330);
    vnFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

}

}
```