

```
// 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;
}

}
```