

### Φροντιστηριακή άσκηση #3

#### Θέμα: Αποθήκευση – Ανάκτηση Αντικειμένων (Serialization)

Η φροντιστηριακή αυτή άσκηση αποσκοπεί στην εξοικείωση με τα ρεύματα δεδομένων εισόδου και εξόδου. Μέχρι στιγμής έχετε ήδη χρησιμοποιήσει τα τυπικά ρεύματα δεδομένων εισόδου και εξόδου System.in και System.out της Java. Στο έργο αυτό θα καλυφθεί η αποθήκευση αντικειμένων (serialization), που ουσιαστικά επιτρέπει σε οποιοδήποτε Java αντικείμενο να γραφεί σε ένα ρεύμα εξόδου και να ξαναδημιουργηθεί αργότερα, διαβάζοντας το αντικείμενο από το ρεύμα εισόδου. Υπενθυμίζεται ότι μόνο τα αντικείμενα που είναι υποκλάσεις της διαπροσωπείας java.io.Serializable μπορούν να αποθηκευθούν με τον τρόπο που περιγράφεται στην άσκηση.

Πιο συγκεκριμένα, θα δημιουργήσουμε τις κλάσεις Contact (επαφή: περιγράφει μια εγγραφή ηλεκτρονικής ατζέντας) και ContactDatabase (περιγράφει μία βάση δεδομένων από επαφές). Η υλοποίηση της κλάσης ContactDatabase χρησιμοποιεί ένα πεδίο τύπου ArrayList όπου αποθηκεύει τις επαφές της βάσης. Εκτός από τις προφανείς μεθόδους (addContact, isEmpty, ...) η κλάση δίνει την δυνατότητα, μέσω της μεθόδου saveContactList, να αποθηκευθεί το πεδίο τύπου ArrayList με τις επαφές σε ένα αρχείο και, κατόπιν να επαναφορτωθεί μέσω της μεθόδου loadContactList.

Για την αποθήκευση αντικειμένων στο δίσκο και την μετέπειτα ανάκτηση τους, θα αναπτυχθεί η κλάση FileLoader η οποία περιέχει δύο στατικές μεθόδους αποθήκευσης/ ανάκτησης αντικειμένων κλάσεων που υλοποιούν την διαπροσωπεία Serializable.

- Χρησιμοποιώντας το BlueJ, δημιουργήστε το έργο MySerializableContactDatabase
- Να κατασκευαστεί η κλάση Contact, η οποία αντιστοιχεί σε μια επαφή. Η κλάση αυτή υλοποιεί την διαπροσωπεία java.io.Serializable και περιέχει τρία private πεδία, όπως φαίνεται παρακάτω:

```
private String firstName;
private String lastName;
private int phoneNumber;
```

- Να υλοποιηθούν οι κατασκευαστές της κλάσης Contact (με τις προφανείς υλοποιήσεις):

```
public Contact()
public Contact(String firstName, String lastName, int phoneNumber)
```

- Να υλοποιηθούν οι παρακάτω μέθοδοι της κλάσης Contact:

```
private void setFirstName(String firstName)
public void setLastName(String lastName)
public void setPhoneNumber(int phoneNumber)
public String getFirstName()
public String getLastName()
public int getPhoneNumber()
public String toString()
```

- Να κατασκευαστεί η κλάση ContactDatabase, η οποία αντιστοιχεί σε μία βάση δεδομένων με επαφές η οποία υλοποιείται με το private πεδίο:

```
private ArrayList contactList
```

- Να υλοποιηθούν ο κατασκευαστής και οι παρακάτω μέθοδοι της κλάσης ContactDatabase:

public ContactDatabase ()	Αρχικοποιεί μια κενή λίστα επαφών.
public boolean isEmpty()	Ελέγχει εάν η λίστα των επαφών είναι κενή.
public int getNoOfContacts()	Επιστρέφει τον αριθμό των επαφών.
public void addContact(Contact contact)	Προσθέτει μια επαφή στην λίστα.
public String toString()	Επιστρέφει μια αναπαράσταση της λίστας ως συμβολοσειρά.
public void print()	Τυπώνει το περιεχόμενο της βάσης επαφών

- Να βεβαιωθείτε για την σωστή λειτουργία της υλοποίησής σας δημιουργώντας τουλάχιστον 2 αντικείμενα τύπου Contact και ένα αντικείμενο τύπου ContactDatabase στο οποίο θα εισάγετε τις επαφές που δημιουργήσατε.. Πειραματιστείτε με την λειτουργία των μεθόδων των δύο κλάσεων.

8. Να κατασκευάσετε την κλάση `FileLoader`, η οποία θα περιέχει δύο στατικές μεθόδους ανάκτησης και αποθήκευσης αντικειμένων:

```
public static Object deserializeFromFile(File inputFile)           throws IOException,
                                                                ClassNotFoundException
public static void serializeToFile(File outputFile, Serializable data) throws IOException
```

Να χρησιμοποιήσετε τις υλοποιήσεις που δίνονται παρακάτω. Προσπαθήστε να κατανοήσετε τον τρόπο λειτουργίας τους, ανατρέχοντας εάν είναι αναγκαίο στην βιβλιοθήκη `java.io`.

```
import java.io.*;
public class FileLoader
{
    /**
     * Deserialize an object from the specified file.
     */
    public static Object deserializeFromFile(File inputFile)
    throws IOException, ClassNotFoundException
    {
        Object returnValue = null;
        if (inputFile.exists())
        {
            if (inputFile.isFile())
            {
                ObjectInputStream readIn =
                    new ObjectInputStream(new FileInputStream(inputFile));
                returnValue = readIn.readObject();
                readIn.close();
            }
            else
            {
                System.err.println(inputFile + " is a directory.");
            }
        }
        else
        {
            System.err.println("File " + inputFile + " does not exist.");
        }
        return returnValue;
    } //deserializeFromFile

    /**
     * Serialize an object to the specified file.
     */
    public static void serializeToFile(File outputFile, Serializable data)
    throws IOException
    {
        ObjectOutputStream writeOut =
            new ObjectOutputStream(new FileOutputStream(outputFile));
        writeOut.writeObject(data);
        writeOut.close();
    } // serializeToFile
} //class FileLoader
```

9. Με τη βοήθεια της κλάσης `FileLoader`, να υλοποιηθούν δύο μέθοδοι αποθήκευσης και ανάκτησης της λίστας επαφών (δηλ. του πεδίου `contactList`), όπως φαίνεται παρακάτω:

```
public void saveContactList(String fileName)    Αποθηκεύει την λίστα των επαφών στο δοθέν αρχείο
public void loadContactList(String fileName)    Ανακτά την λίστα των επαφών από το δοθέν αρχείο.
```

Ακολουθεί μια υλοποίηση των παραπάνω μεθόδων την οποία μπορείτε να χρησιμοποιήσετε.

```
import java.util.ArrayList;
import java.io.*;

public class
{
    private ArrayList contactList; //This object implements Serializable

    ....
}
```

```
public void saveContactList(String fileName)
{
    try
    {
        File file = new File(fileName);
        FileLoader.serializeToFile(file, contactList);
    }
    catch (IOException exc)
    {
        System.err.println("An error occured while saving this file.");
    }
} // saveContactList

public void loadContactList(String fileName)
{
    try
    {
        File file = new File(fileName);
        this.contactList = (ArrayList) FileLoader.deserializeFromFile(file);
    }
    catch (IOException exc)
    {
        System.err.println("An error occured while loading this file.");
    }
    catch (ClassNotFoundException exc)
    {
        System.err.println("An error occured while loading this file.");
    }
} // loadContactList

} // class ContactDatabase
```

10. Να πειραματιστείτε με τις μεθόδους `saveContactList` και `loadContactList` εκτελώντας τα ακόλουθα βήματα:
- Δημιουργήστε 2 επαφές
  - Δημιουργήστε μία βάση επαφών (αντικείμενο τύπου `ContactDatabase`)
  - Εισάγετε τις επαφές στη βάση
  - Τυπώστε το περιεχόμενο της βάσης
  - Αποθηκεύστε το περιεχόμενο της βάσης στο αρχείο `contactDB` κάνοντας χρήση της μεθόδου `saveContactList`.
  - Κλείστε το `bluej`.
  - Ανοίξτε το έργο `MySerializableContactDatabase` με το `bluej`.
  - Δημιουργήστε μία βάση επαφών (αντικείμενο τύπου `ContactDatabase`)
  - Φορτώστε το περιεχόμενο της βάσης από το αρχείο `contactDB` κάνοντας χρήση της μεθόδου `loadContactList`.
  - Τυπώστε το περιεχόμενο της βάσης και βεβαιωθείτε ότι το περιεχόμενο της είναι ταυτόσημο με αυτό που δημιουργήσατε αρχικά.