



Πανεπιστήμιο Δυτικής Αττικής
Τμήμα Μηχανικών Πληροφορικής και
Υπολογιστών

Ηλεκτρονική Μάθηση

Εργασία : Τεχνικό εγχειρίδιο εκπαιδευτικής εφαρμογής

Ονοματεπώνυμα μελών της ομάδας : Κωνσταντακόπουλος
Χαράλαμπος , Μηνά Λοϊζος , Γκόφας Δημήτριος , Καψής
Στυλιανός

Αριθμοί Μητρώου : 71347452 , 71347850 , 71347123 , 71347427

Εξάμηνο σπουδών : 12ο

Πρόγραμμα σπουδών : Π.Α.Δ.Α.

Ακαδημαϊκό έτος : 2021-2022

Περιεχόμενα

1. Εξώφυλλο	Σελ.1
2. Περιεχόμενα	Σελ.2
3. Setup/Εργαλεία εφαρμογής	Σελ.3
4. Πληροφορίες εφαρμογής	Σελ.4
5. Αρχικό παράθυρο – Main class	Σελ.5-6
6. Το παράθυρο των κεφαλαίων	Σελ.7-8
7. Το παράθυρο των διαγωνισμάτων	Σελ.9-10
8. Το παράθυρο των διαγωνισμάτων/Αποτελέσματα	Σελ.11-12
9. Αποτελέσματα	Σελ.13-15

Setup/Εργαλεία Εφαρμογής

Η συγκεκριμένη εφαρμογή δημιουργήθηκε με γλώσσα προγραμματισμού java 8 και με το IDE πρόγραμμα NetBeans (version 8.2).

```
C:\Users\babis>javac -version  
javac 1.8.0_281
```

About



NetBeans IDE and NetBeans Platform are based on software from netbeans.org, which has been dual licensed under the Common Development and Distribution License (CDDL) and the GNU General Public License version 2 with Classpath exception. For more information, please visit www.netbeans.org.

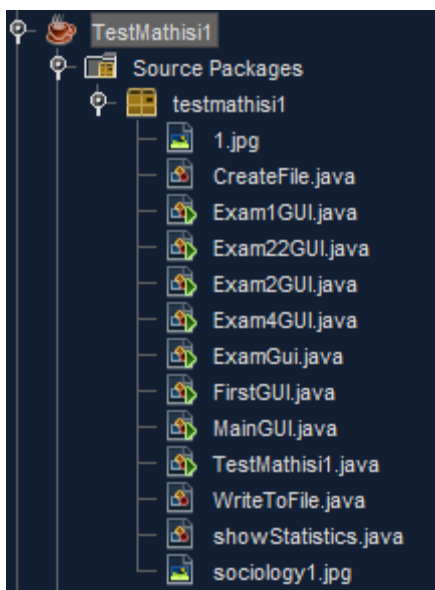
Product Version: NetBeans IDE 8.2 (Build 201609300101)
Updates: NetBeans IDE is updated to version [NetBeans 8.2 Patch 2](#)
Java: 1.8.0_281; Java HotSpot(TM) 64-Bit Server VM 25.281-b09
Runtime: Java(TM) SE Runtime Environment 1.8.0_281-b09
System: Windows 10 version 10.0 running on amd64; Cp1252; en_US (nb)
User directory: C:\Users\babis\AppData\Roaming\NetBeans\8.2
Cache directory: C:\Users\babis\AppData\Local\NetBeans\Cache\8.2

Close

Πληροφορίες εφαρμογής

Η εφαρμογή έχει συνολικά 11 java κλάσεις και η κάθε μία έχει προγραμματιστεί με τέτοιον τρόπο ξεχωριστά για την καλύτερη λειτουργία του κώδικα. Πιο συγκεκριμένα:

- 1) **TestMathisi1** - Main κλάση της εφαρμογής
- 2) **FirstGUI** - Το αρχικό παράθυρο (μενού) της εφαρμογής
- 3) **MainGUI** – Το παράθυρο των κεφαλαίων της ύλης του μαθήματος
- 4) **ExamGUI** – Το παράθυρο των διαγωνισμάτων του μαθήματος
- 5) **Exam1GUI** - Το παράθυρο του διαγωνίσματος 1
- 6) **Exam2GUI** - Το παράθυρο του διαγωνίσματος 2-1
- 7) **Exam22GUI** - Το παράθυρο του διαγωνίσματος 2-2
- 8) **Exam4GUI** - Το παράθυρο του διαγωνίσματος 4
- 9) **CreateFile** – Δημιουργία αρχείου .txt
- 10) **WriteToFile** – Εγγραφή στο αρχείο .txt
- 11) **showStatistics** – Αποθήκευση και εμφάνιση αποτελεσμάτων διαγωνισμάτων



(Σημείωση)

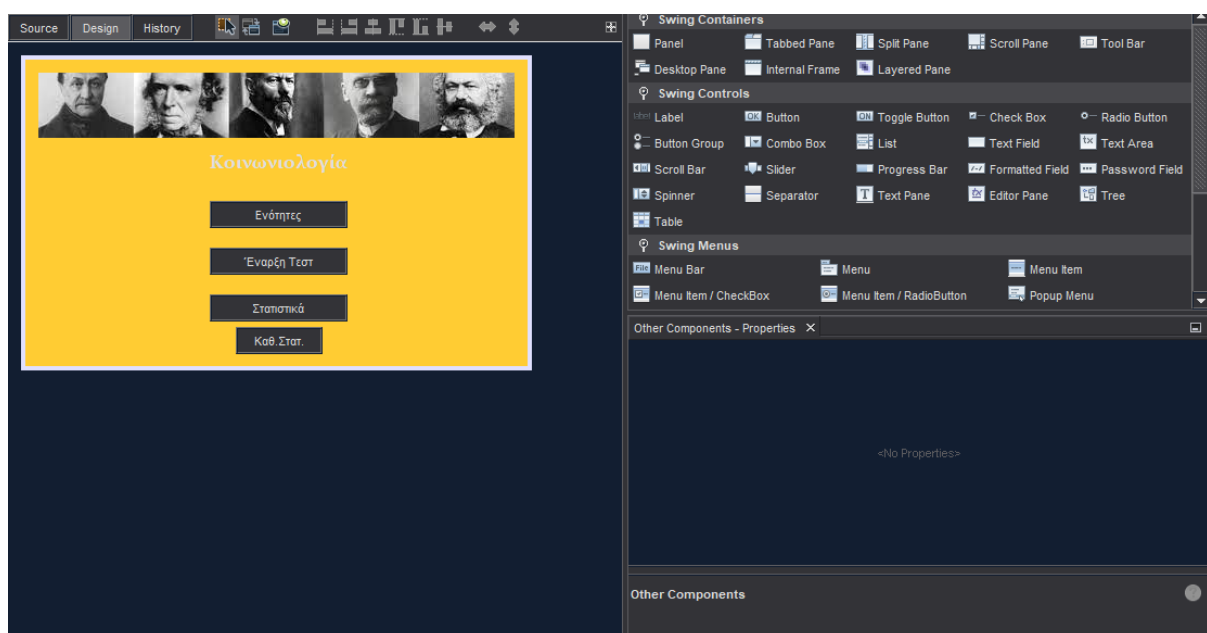
Για να τρέξει η εφαρμογή σε .jar μορφή χρειάζεται να είναι εγκατεστημένη η java στον υπολογιστή, παρόλα αυτά έχουμε συμπεριλάβει την εφαρμογή μας σε και .exe μορφή που για να τρέξει χρειάζεται να είναι στο ίδιο directory το jre 1.8.0

Αρχικό Παράθυρο – Main κλάση εφαρμογής

Το αρχικό παράθυρο της εφαρμογής:



Το **GUI** του αρχικού παραθύρου αλλά και όλης της εφαρμογής **κατασκευάστηκε** με τον ενσωματωμένο **GUI editor του NetBeans** (με swing components), ο οποίος έχει λειτουργίες drag n drop για οποιοδήποτε component θέλουμε (κουμπί, label, panel) παράγοντας κώδικα στο εκτελέσιμο αρχείο:



Αρχικό Παράθυρο – Main κλάση εφαρμογής

Αφού δημιουργήσουμε το αρχικό παράθυρο, συμπληρώνουμε τον παρακάτω κώδικα στην Main class της εφαρμογής μας προκειμένου να μας εμφανιστεί:

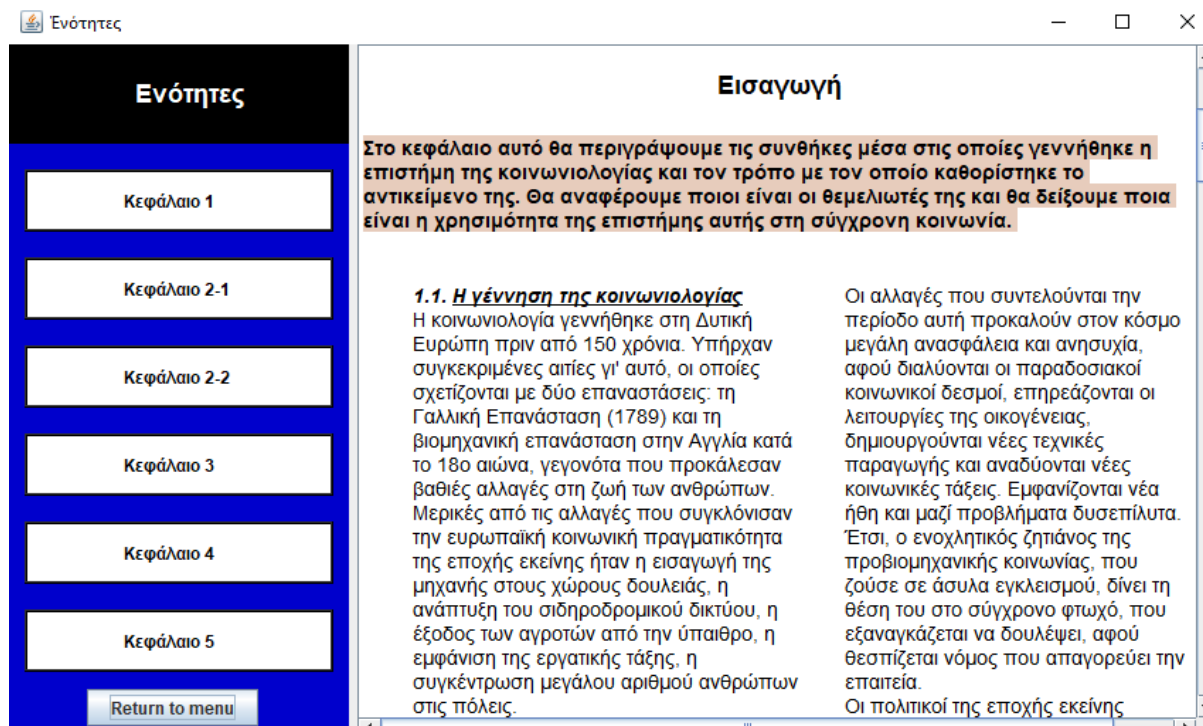
```
public class TestMathisil {  
  
    public static void main(String[] args)  
    {  
  
        FirstGUI frame = new FirstGUI();  
        frame.setTitle("Αρχικό μενού");  
        frame.setVisible(true);  
        frame.setResizable(true);  
        frame.setLocationRelativeTo(null);  
    }  
  
}
```

Στην ουσία φτιάχνουμε ένα **αντικείμενο (frame)** της **κλάσης** του αρχικού παραθύρου (**FirstGUI**) και το κάνουμε να **εμφανίζεται** με τη **μέθοδο setVisible()**.

Το παράθυρο των κεφαλαίων

Παρόλα αυτά επειδή θέλαμε το **περιεχόμενο** να παρουσιάζεται με **σχετικά ωραίο τρόπο** (στοίχιση, χρωματισμένα γράμματα, εικόνες) και επειδή η **java δεν υποστηρίζει τέτοιο περιεχόμενο natively**, μέσα στα εισαγωγικά προσθέσαμε κώδικα HTML.

Το **component JEditorPane υποστηρίζει κώδικα HTML + CSS** (δηλαδή αντί για απλό string, δέχεται κώδικα HTML και το εμφανίζει σαν web browser) και έτσι μπορούμε να **επιτύχουμε** ένα σχετικά **εμφανίσιμο αποτέλεσμα**:



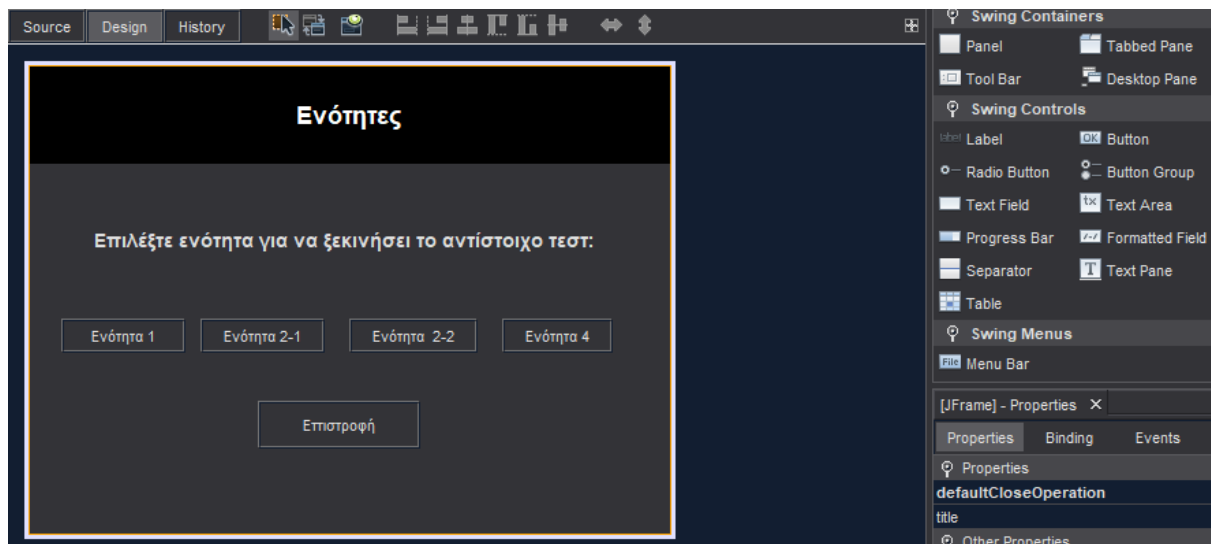
Τέλος, για να κάνουμε το παράθυρο των κεφαλαίων να εμφανιστεί θα πρέπει να συνδέσουμε τη κλάση αυτή (MainGUI) με το αντίστοιχο κουμπί στο αρχικό παράθυρο (ενότητες). Αυτό επιτυγχάνεται ως εξής:

```
private void ChaptersBTNActionPerformed(java.awt.event.ActionEvent evt) {  
  
    MainGUI frame = new MainGUI();  
    frame.setTitle("Ενότητες");  
    frame.setVisible(true);  
    frame.setResizable(true);  
    frame.setLocationRelativeTo(null);  
    super.dispose();  
  
}
```

Χρησιμοποιούμε ίδια τεχνική όπως την εμφάνιση του αρχικού παραθύρου μόνο που ο **κώδικας τοποθετείται μέσα στην μέθοδο actionPerformed()** του κουμπιού ενότητες.

Το παράθυρο των διαγωνισμάτων

Το παράθυρο:



Επίσης το συγκεκριμένο panel κατασκευάστηκε με το GUI editor του netbeans. Πατώντας το κάθε κουμπί (Ενότητα 1,2-1,2-2,4) εμφανίζεται στο χρήστη το διαγώνισμα που αντιστοιχεί στην ύλη του αντίστοιχου κεφαλαίου.

Το κάθε παράθυρο διαγωνίσματος υλοποιήθηκε με **ξεχωριστή κλάση** για καλύτερη ευκολία και **αντιστοιχήθηκε με το κάθε κουμπί** ακριβώς όπως είδαμε και σε προηγούμενα παράθυρα, με τη **μέθοδο actionPerformed()**.

Παρακάτω βλέπουμε το κώδικα του κουμπιού (διαγωνίσματος) ενότητας 1:

```
private void EnotitalBtnActionPerformed(java.awt.event.ActionEvent evt) {  
  
    ExamlGUI frameT1 = new ExamlGUI();  
    frameT1.setTitle("Διαγώνισμα Ενότητας 1");  
    frameT1.setVisible(true);  
    frameT1.setResizable(true);  
    frameT1.setLocationRelativeTo(null);  
    super.dispose();  
  
}
```

Το παράθυρο των διαγωνισμάτων

Παρακάτω βλέπουμε το διαγώνισμα που εμφανίζεται:

The screenshot shows a Java Swing window titled "Exam1GUI.java". It contains a quiz with two sections: A) Ερωτήσεις πολλαπλής επιλογής (Multiple Choice Questions) and B) Ερωτήσεις σωστού-λάθους (True/False Questions). Section A has five questions about history and social changes in the 18th century. Section B has one question about the Enlightenment. Each question has radio button options.

A) Ερωτήσεις πολλαπλής επιλογής:

1) Πότε γεννήθηκε η κοινωνιολογία στην Δυτική Ευρώπη;

☐ 1789 ☐ 1689 ☐ 1768 ☐ 1881

2) Πότε έγινε η Γαλλική Επανάσταση;

☐ 1679 ☐ 1789 ☐ 1668 ☐ 1778

3) Ποιές αλλαγές συγκλόνισαν την ευρωπαϊκή κοινωνική πραγματικότητα του 18ου αιώνα;

☐ Εισαγωγή της μηχανής στην εργασία ☐ Ανάπτυξη του σιδηροδρομικού δικτύου ☐ Έξοδος των αγροτών από την ύπαιθρο ☐ Όλα τα παραπάνω

4) Ποιόν αιώνα ξεκίνησε η βιομηχανική επανάσταση στην Αγγλία;

☐ 18ο ☐ 16ο ☐ 17ο ☐ 19ο

5) Τι προκάλεσαν οι αλλαγές στην ευρωπαϊκή κοινωνική πραγματικότητα του 18ου αιώνα;

☐ Ανοσφάλεια και ανησυχία ☐ Ηρεμία και τάξη ☐ Απαλοφή κοινωνικών προβλημάτων ☐ Εδραίωση παλαιών κοινωνικών θεσμών

B) Ερωτήσεις σωστού-λάθους:

1) Η κοινωνιολογία έχει ως αντικείμενο μελέτης τα κοινωνικά φαινόμενα, τη δράση των κοινωνικών ομάδων, τις σχέσεις μεταξύ ατόμου και ομάδων, τις σχέσεις μεταξύ κοινωνικών ομάδων και τις κοινωνικές διαδικασίες.

☐ Σωστό ☐ Λάθος

The screenshot shows a window titled "Διαγώνισμα Ενότητας 1". It contains a quiz with two sections: A) Ερωτήσεις πολλαπλής επιλογής (Multiple Choice Questions) and B) Ερωτήσεις σωστού-λάθους (True/False Questions). Section A has five questions about history and social changes in the 18th century. Section B has three questions about the Enlightenment. Each question has radio button options. A "Submit" button is at the bottom right.

A) Ερωτήσεις πολλαπλής επιλογής:

1) Πότε γεννήθηκε η κοινωνιολογία στην Δυτική Ευρώπη;

☐ 1789 ☐ 1689 ☐ 1768 ☐ 1881

2) Πότε έγινε η Γαλλική Επανάσταση;

☐ 1679 ☐ 1789 ☐ 1668 ☐ 1778

3) Ποιές αλλαγές συγκλόνισαν την ευρωπαϊκή κοινωνική πραγματικότητα του 18ου αιώνα;

☐ Εισαγωγή της μηχανής στην εργασία ☐ Ανάπτυξη του σιδηροδρομικού δικτύου ☐ Έξοδος των αγροτών από την ύπαιθρο ☐ Όλα τα παραπάνω

4) Ποιόν αιώνα ξεκίνησε η βιομηχανική επανάσταση στην Αγγλία;

☐ 18ο ☐ 16ο ☐ 17ο ☐ 19ο

5) Τι προκάλεσαν οι αλλαγές στην ευρωπαϊκή κοινωνική πραγματικότητα του 18ου αιώνα;

☐ Ανοσφάλεια και ανησυχία ☐ Ηρεμία και τάξη ☐ Απαλοφή κοινωνικών προβλημάτων ☐ Εδραίωση παλαιών κοινωνικών θεσμών

B) Ερωτήσεις σωστού-λάθους:

1) Η κοινωνιολογία έχει ως αντικείμενο μελέτης τα κοινωνικά φαινόμενα, τη δράση των κοινωνικών ομάδων, τις σχέσεις μεταξύ ατόμου και ομάδων, τις σχέσεις μεταξύ κοινωνικών ομάδων και τις κοινωνικές διαδικασίες.

☐ Σωστό ☐ Λάθος

2) Ιστορικές μελέτες έδειξαν πως στην Ευρώπη του Μεσαίωνα σχεδόν κανείς δεν παντρευόταν από έρωτα.

☐ Σωστό ☐ Λάθος

3) Κατά τον 18ο αιώνα δημιουργήθηκαν νέοι κοινωνικοί δεσμοί, ήθη και έθιμα λόγω των κοινωνικών αλλαγών της πε...

☐ Σωστό ☐ Λάθος

Πατώντας το κουμπί **Submit** στο συγκεκριμένο αλλά και στο κάθε διαγώνισμα, το **παράθυρο κλείνει** και ο **χρήστης επιστρέφει πίσω** στο κεντρικό παράθυρο διαγωνισμάτων. Επίσης τα οι ερωτήσεις είναι multiple choice και σωστού-λάθους με τη μορφή JRadioButtons και τα αποτελέσματα είναι διαθέσιμα μετά την ολοκλήρωση του διαγωνίσματος.

Το παράθυρο των διαγωνισμάτων/Αποτελέσματα

Για να καταλάβουμε πως προκύπτουν τα αποτελέσματα πρέπει να ρίξουμε μια ματιά στο κώδικα του κουμπιού Submit όταν ο χρήστης ολοκληρώνει ένα διαγώνισμα:

```
private void Submit1BtnActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    ExamGui frameT = new ExamGui();  
    frameT.setTitle("Διαγώνισμα");  
    frameT.setVisible(true);  
    frameT.setResizable(true);  
    frameT.setLocationRelativeTo(null);  
    super.dispose();  
    System.out.print("Exam1GUI: " + ans1 + ans2 + ans3 + ans4 + ans5 + ans6 + ans7 + ans8);  
    CreateFile.create();  
    WriteToFile.write("Exam1GUI: ");  
    WriteToFile.write(ans1 + ans2 + ans3 + ans4 + ans5 + ans6 + ans7 + ans8);  
}
```

Μόλις πατηθεί το κουμπί **Submit**, δημιουργούμε ένα καινούργιο αρχείο **.txt** με τη μέθοδο **create()** της κλάσης **WriteFile**, το οποίο χρησιμοποιείται για την αποθήκευση των αποτελεσμάτων. Έπειτα γράφουμε σε αυτό το αρχείο τα αποτελέσματα με τη μέθοδο **write()** της κλάσης **WriteToFile**. Στην ουσία αποθηκεύουμε τα αποτελέσματα σε μορφή String (True, false).

Κλάση **CreateFile**:

```
import java.io.File; // Import the File class  
import java.io.IOException; // Import the IOException class to handle errors  
  
public class CreateFile {  
    public static void create() {  
        try {  
            File myObj = new File("statistics.txt");  
            if (myObj.createNewFile()) {  
                System.out.println("File created: " + myObj.getName());  
            } else {  
                System.out.println("File already exists.");  
            }  
        } catch (IOException e) {  
            System.out.println("An error occurred.");  
            e.printStackTrace();  
        }  
    }  
}
```

Το παράθυρο των διαγωνισμάτων/Αποτελέσματα

Κλάση WriteToFile:

```
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter; // Import the FileWriter class
import java.io.IOException; // Import the IOException class to handle errors

public class WriteToFile {
    public static void write(String line) {
        try (BufferedWriter writer = new BufferedWriter(new FileWriter("statistics.txt", true))) {

            writer.write(line);
            writer.newLine();
            writer.close();
        } catch (IOException ex) {

        }
    }
}
```

Για την **διαλογή της σωστής από τη λάθος απάντηση** χρησιμοποιήσαμε τον εξής τρόπο: Κάθε φορά που **πατιέται η κάθε επιλογή** από το χρήστη, έχουμε ορίσει με τη **μέθοδο ActionPerformed()**, να αποθηκεύεται η αντίστοιχη απάντηση (σωστή ή λάθος) στο κάθε string.

Πχ. Στη **1^η ερώτηση έχουμε 4 επιλογές**, εκ των οποίων η **1^η είναι σωστή** και όλες οι υπόλοιπες λάθος. Ανάλογα με τι **επιλέξει ο χρήστης**, θα **αποθηκευτεί** στο **String** του **αποτελέσματος ερωτήματος 1**, το **αντίστοιχο αποτέλεσμα** (True or false), το οποίο με τη σειρά του θα αποθηκευτεί στο αρχείο .txt:

```
private void ElBtnln1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ans1="true1,";
}

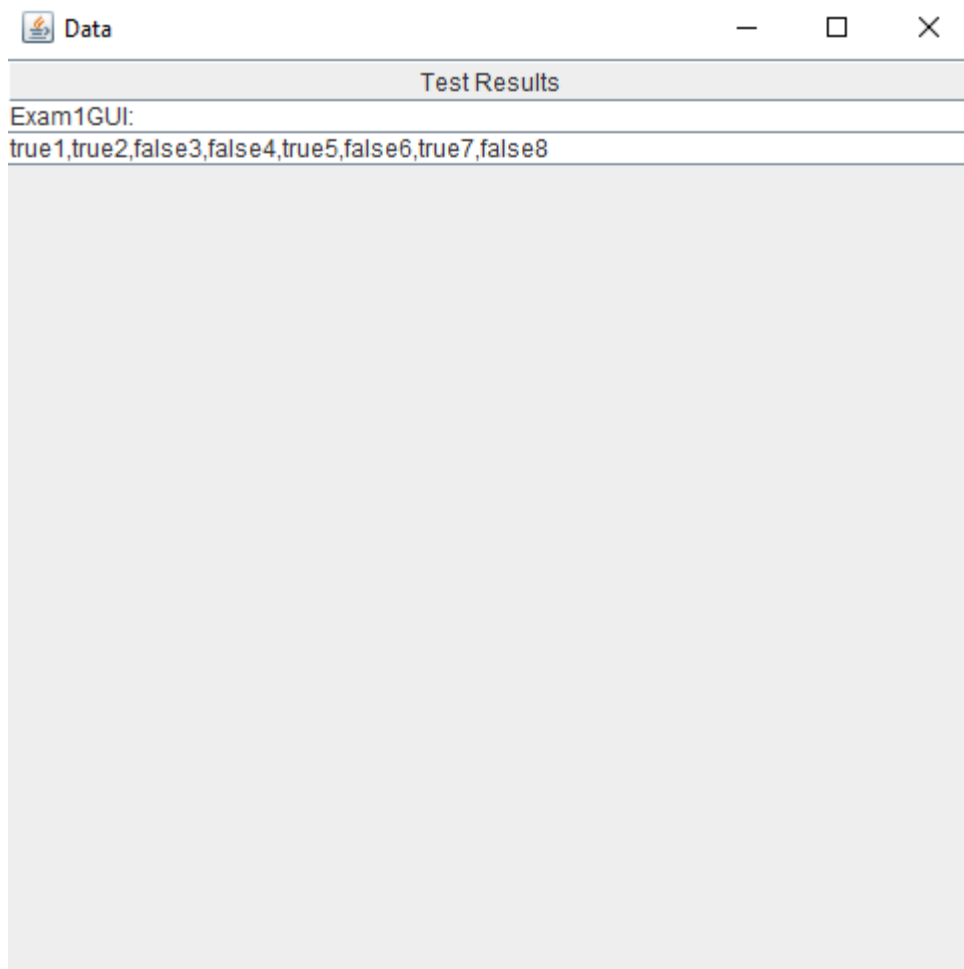
private void ElBtnln2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ans1="false1,";
}

private void ElBtnln3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ans1="false1,";
}

private void ElBtnln4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ans1="false1,";
}
```

Αποτελέσματα

Αφού ο χρήστης έχει ολοκληρώσει κάποιο ή όλα τα διαγωνίσματα τότε πατώντας το κουμπί Στατιστικά στο αρχικό μενού μπορεί να δει τα αποτελέσματα του κάθε διαγωνίσματος. Αν έχει **απαντήσει σωστά** τότε θα **εμφανίζεται** δίπλα από το κάθε ερώτημα η **λέξη true** και αν έχει **απαντήσει λάθος** θα **εμφανίζεται** δίπλα από κάθε ερώτημα η **λέξη false**:



Ο κώδικας που υλοποιείται όταν πατιέται το κουμπί Στατιστικά:

```
private void StatsBTNActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    ArrayList<String> fdata=new ArrayList<>();  
    fdata=showStatistics.read();  
    showStatistics.Open(fdata);  
}
```

Στην ουσία δημιουργούμε μια ArrayList από Strings και χρησιμοποιούμε τις **μεθόδους read()** και **Open()** της κλάσης **showStatistics**.

Αποτελέσματα

Η κλάση showStatistics:

```
import java.io.*;
import java.util.ArrayList; // import the ArrayList class
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JScrollPane;
import javax.swing.JTable;

public class showStatistics {
    public static ArrayList<String> read() {
        ArrayList<String> fdata=new ArrayList<>();
        try
        {
            File file=new File("statistics.txt");
            FileReader fr=new FileReader(file);
            BufferedReader br=new BufferedReader(fr);
            String line;
            while ((line=br.readLine()) !=null)
            {
                fdata.add(line);
                System.out.println(line);
            }
            fr.close();
        }
        catch(IOException e)
        {
            e.printStackTrace();
        }
    }
    finally {
        return fdata;
    }
}

public static void Open(ArrayList<String> fdata) {
    JFrame frame;
    String [] columnNames={"Test Results"};

    frame=new JFrame("Data");

    String [][] data=new String [fdata.size()][1];
    int i=0;
    for(String s : fdata) {
        data[i][0]=s;
        i++;
    }
    JTable table=new JTable(data,columnNames);
    frame.setVisible(true);
    frame.setSize(500,500);
    frame.add(new JScrollPane(table));

}
}
```

Στην ουσία η μέθοδος **read()** διαβάζει τα περιεχόμενα του αρχείου statistics.txt και τα “φορτώνει” στη μεταβλητή fdata, ενώ η μέθοδος **Open()** εμφανίζει τα αποτελέσματα στο παράθυρο των αποτελεσμάτων Data.

Αποτελέσματα

Τέλος, πατώντας ο χρήσης το κουμπί Καθ. Στατ. στο αρχικό μενού, καθαρίζει το παράθυρο των αποτελεσμάτων από προηγούμενα αποτελέσματα που έχουν γίνει.

Ο κώδικας υλοποίησης του κουμπιού:

```
private void clearActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    CreateFile.create();  
    try {  
        FileWriter myWriter = new FileWriter("statistics.txt");  
  
        myWriter.close();  
        System.out.println("Successfully wrote to the file.");  
    } catch (IOException e) {  
        System.out.println("An error occurred.");  
        e.printStackTrace();  
    }  
}
```