



Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών

HY252 – Αντικειμενοστρεφής Προγραμματισμός

Εξάμηνο: Χειμερινό 2024-2025

Διδάσκων: Γιάννης Τζίτζικας

Βοηθοί: Σ. Γερονικολάκης [p1192] (1^η), Χ Πολυματίδης [5209] (2^η), Γ. Μέλλιος [4416] (3^η), Β. Καραβάς [4893] (4^η)

1^η Σειρά Ασκήσεων

Ανάθεση: 1 Οκτωβρίου 2024

Παράδοση: 21 Οκτωβρίου 2024

Εκπαιδευτικοί Στόχοι

Η 1^η άσκηση (40 μονάδες) θα σας επιτρέψει να έχετε μια πρώτη επαφή με τη Java, τη ροή ελέγχου, την είσοδο/έξοδο από κονσόλα και από απλά παράθυρα διαλόγου.

Η 2^η άσκηση (30 μονάδες) θα σας επιτρέψει να εξοικειωθείτε με την είσοδο από κονσόλα, τις συμβολοσειρές, τους πίνακες, την ανάγνωση αρχείων και χρήση κώδικα από εξωτερική βιβλιοθήκη

Η 3^η άσκηση (30 μονάδες) θα σας επιτρέψει να εξοικειωθείτε με τη χρήση και εμπλουτισμό εξωτερικού κώδικα, τους πίνακες, την υλοποίηση αλγορίθμων και τη χρονομέτρηση της εκτέλεσής τους.

Η 4^η άσκηση, που δίνει bonus 5%, θα σας επιτρέψει να εξοικειωθείτε με την διαχείριση αρχείων.

Σημειώσεις

1. **Απορίες** σχετικά με την Α1 θα απαντώνται μόνο μέσω του forum της Α1 στο moodle.
2. Την πρώτη άσκηση πρέπει να μπορείτε να την κάνετε compile και να την τρέξετε και από **γραμμή εντολών**, χωρίς δηλαδή τη χρήση κάποιου **IDE** (σχετικές οδηγίες θα συζητηθούν και στο 1^ο φροντιστήριο). Τις επόμενες ασκήσεις συστήνεται να τις κάνετε χρησιμοποιώντας ένα IDE (Eclipse ή NetBeans ή IntelliJ). Σχετικό υλικό για τα IDEs υπάρχει στην ιστοσελίδα.
3. **Ημερομηνία παράδοσης** και **bonus**: Όποιος παραδώσει δύο μέρες πριν την προθεσμία παράδοσης θα έχει βαθμολογικό bonus 10%.
4. **Οδηγίες Παράδοσης**. **ΠΡΟΣΟΧΗ: Ακολουθήστε επακριβώς τις οδηγίες παράδοσης που αναγράφονται στο έγγραφο «Οδηγίες Παράδοσης Ασκήσεων».** Διαφορετικά η διόρθωση είναι πολύ χρονοβόρα για τους μεταπτυχιακούς οι οποίοι έχουν και αυτοί μαθήματα και εργασία να κάνουν. **Αν παραδώσετε κάτι που δεν είναι σύμφωνο με τους κανόνες παράδοσης τότε ίσως υπάρξει μείωση βαθμού και ίσως δεν βαθμολογηθεί.**
5. Η **εξέταση** της Α1 θα γίνει μέσω μίας υποχρεωτικής εργαστηριακής άσκησης (**Α1Ε**) (θα δούμε πως θα την κάνουμε εφέτος). Θα είναι μια απλή και μικρή άσκηση που θα πρέπει να κάνετε στο εργαστήριο/αναγνωστήριο υπό την εποπτεία των βοηθών του μαθήματος (σε περίπτωση απουσίας ή αποτυχίας δεν θα προσμετρηθούν οι βαθμοί της Α1). Επίσης γίνεται αυτοματοποιημένος έλεγχος αντιγραφών, **οποιαδήποτε αντιγραφή** συνεπάγεται μηδενισμό σε όλη τη σειρά ασκήσεων.
6. **Η εργασία είναι ατομική και απαγορεύεται η χρήση κώδικα που δεν έχετε γράψει οι ίδιοι, είτε από το διαδίκτυο είτε από κάποιο συμφοιτητή σας. Σε περίπτωση εντοπισμού αντιγραφής η εργασία θα μηδενίζεται. Αν κάτι αντιγράψετε από κάπου (όσο μικρό ή μεγάλο είναι), πρέπει να το αναφέρετε στην αναφορά.**

Άσκηση 1 [40 μονάδες] Ζωγραφίζοντας το γράμμα Δ

Εκπαιδευτικοί Στόχοι: Πρώτη επαφή με Java, ροή ελέγχου, είσοδος/έξοδος από κονσόλα, διαχείριση συμβολοσειρών, παράθυρα διαλόγου, ζωγραφική.

Στόχος είναι να γράψετε μια κλάση **DrawDelta** (σε αρχείο `DrawDelta.java`) η οποία θα μπορεί να ζωγραφίσει το γράμμα Δ με διάφορους τρόπους και σε διάφορα μεγέθη.

(i) [5 μονάδες] – Διαχείριση Εισόδου

Η μέθοδος `main` πρέπει να μπορεί να διαβάσει δύο παραμέτρους από την γραμμή εντολών (command line arguments), ας τις ονομάσουμε `M` (από το `mode`) και `L` (από το `length`). Οι αποδεκτές τιμές για τη μεταβλητή `M` πρέπει να είναι οι `'c'` (από το `console`), `'w'` (από το `window`), `'f'` (από το `file`), `'g'` (από το `graphics`). Οι αποδεκτές τιμές για τη μεταβλητή `L` είναι οποιοσδήποτε θετικός ακέραιος από το 3 έως το 20.

Το πρόγραμμα σας πρέπει να εκτυπώνει ένα μήνυμα λάθους και να τερματίζει η λειτουργία του αν δεν έχουν δοθεί δύο παράμετροι ή αν οι τιμές αυτών των παραμέτρων δεν είναι οι αποδεκτές. Δείτε τις σχετικές υποδείξεις.

(ii) [5 μονάδες] - Ζωγραφική σε Κονσόλα

Αν `M=c` τότε το πρόγραμμα σας πρέπει να τυπώνει το γράμμα Δ στην κονσόλα με αστεράκια και το ύψος του να είναι `L`. Για παράδειγμα με `java DrawDelta c 6` πρέπει να εκτυπώνεται το:

```
      *
     * *
    *   *
   *     *
  *       *
 *         *
*****
```

Το βασικό είναι να χρησιμοποιούνται πάντα `L` γραμμές και το αποτέλεσμα να μοιάζει με το γράμμα Δ (να μην μπορεί δηλαδή κάποιος να το μπερδέψει με κάποιο άλλο γράμμα). Μπορεί η `main` να καλεί μια στατική μέθοδο της κλάσης σας για να το κάνει.

(iii) [5 μονάδες] - Ζωγραφική σε Παράθυρο Διαλόγου

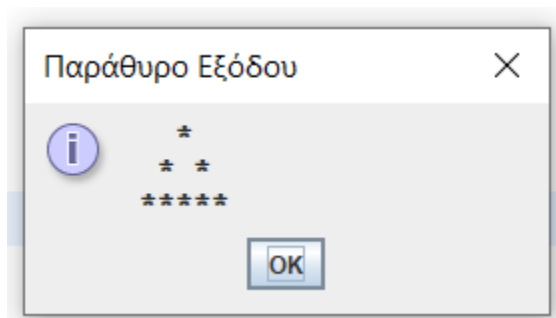
Αν `M=w` τότε το πρόγραμμα σας πρέπει να τυπώνει το γράμμα Δ σε ένα παράθυρο εξόδου. Για να εκτυπώσετε μια συμβολοσειρά σε παράθυρο εξόδου μπορείτε να χρησιμοποιήσετε την εντολή

```
OptionPane.showMessageDialog(null,
    "Ακολουθούν 2 αστεράκια τυπωμένα σε δύο γραμμές \n*\n* ",
    "Παράθυρο Εξόδου",
    JOptionPane.INFORMATION_MESSAGE);
```

Έχοντας κάνει

```
import javax.swing.JOptionPane;
```

Για παράδειγμα με `java DrawDelta w 3` το γράμμα Δ πρέπει να εμφανίζεται σε ένα παράθυρο διαλόγου όπως φαίνεται στο παρακάτω παράδειγμα.



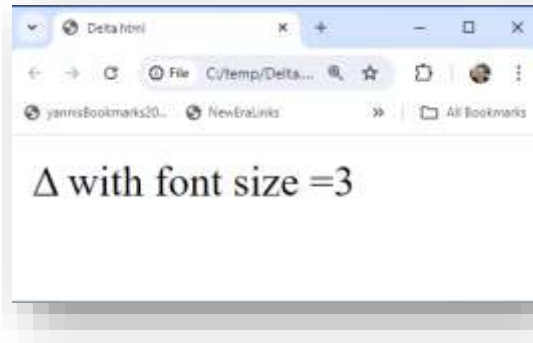
Αν η στοίχιση δεν είναι σωστή, λόγω της γραμματοσειράς, προσθέστε στον κώδικα σας την εντολή `UIManager.put("OptionPane.messageFont", new Font("Lucida Console", Font.BOLD, 14));` έχοντας κάνει `import java.awt.Font;`
`import javax.swing.JOptionPane;`
`import javax.swing.UIManager;`

(iv) [5 μονάδες] - Εγγραφή σε Αρχείο

Αν $M=f$ τότε το πρόγραμμα σας πρέπει να δημιουργεί ένα αρχείο και μέσα του να γράψει HTML κώδικα. Συγκεκριμένα με `java DrawDelta f 3` το πρόγραμμά σας πρέπει να δημιουργεί ένα html αρχείο με όνομα `D.html` όπως το ακόλουθο. Στο παράδειγμα που ακολουθεί η τιμή του `L` που έδωσε ο χρήστης ήταν το 3.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
</head>
<body><font size="3">Δ with font size = 3</font></body>
</html>
```

Το αρχείο αυτό θα μπορείτε να το ανοίξετε με ένα ιστοπλοηγητή, και θα εμφανιστεί κάτι της μορφής:



Για τη δημιουργία αρχείων και για γράψιμο σε αρχεία δείτε το υλικό από το σχετικό φροντιστήριο που θα γίνει. Ένα απλό παράδειγμα που δημιουργεί ένα αρχείο και γράφει σε αυτό, ακολουθεί:

```
PrintWriter writer;  
try {  
    writer = new PrintWriter("C:\\temp\\AAA.txt", "UTF-8");  
    writer.println("The first line");  
    writer.println("The second line");  
    writer.close();  
} catch (Exception e) {  
    System.out.println("πρόβλημα: "+e);  
}
```

(v) [5 μονάδες] - Ζωγραφική σε Παράθυρο Γραφικών

Αν $M=g$ τότε το πρόγραμμα σας πρέπει να τυπώνει το γράμμα Δ σε ένα παράθυρο τύπου γραφικών με κάπως .. καλλιγραφικό τρόπο (όπως θέλετε, απλά να μοιάζει με Δ). Συγκεκριμένα προσθέστε στην κλάση σας την εξής μέθοδο:

```
static void drawDgraphics(int L) {  
    Frame f = new Frame("Ζωγραφίζοντας το Δ") {  
        public void paint(Graphics g) {  
            Graphics2D g2 = (Graphics2D) g;  
            g2.draw(new Line2D.Double(50, 300, 200, 50)); // a line  
            // συμπληρώστε  
        }  
    };  
    f.setSize(400,400);  
    f.setVisible(true);  
}
```

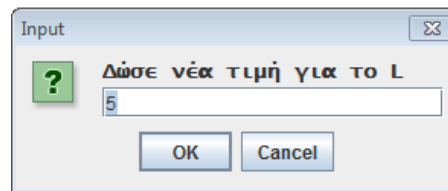
Έχοντας προσθέσει τα εξής:

```
import java.awt.Frame;  
import java.awt.Graphics;  
import java.awt.Graphics2D;  
import java.awt.geom.Line2D;  
import java.awt.geom.QuadCurve2D;
```

Εσείς πρέπει να την καλείτε από τη main και να συμπληρώσετε την paint. Δείτε τι επιλογές δίνονται στο <https://docs.oracle.com/javase/tutorial/2d/geometry/primitives.html> και συμπληρώστε τη μέθοδο paint ώστε να ζωγραφιστεί ένα κάπως.. καλλιγραφικό «Δ» της αρεσκείας σας.

(vi) [5 μονάδες] - Ροή Ελέγχου

Εμπλουτίστε το πρόγραμμά σας ώστε το τέλος να ζητάει μια νέα είσοδο για το L. Αυτό να το κάνει από την κονσόλα, εκτός αν η M=w όπου τότε πρέπει να ζητάει την επιλογή μέσω ενός παράθυρου διαλόγου, όπως το ακόλουθο:



- Για να διαβάσετε την είσοδο stdin :

```
Scanner in = new Scanner(System.in);  
int aNumber = in.nextInt();  
// String nextLine = in.next(); // Αν θέλατε να διαβάσετε String
```

Στην αρχή του αρχείου θα πρέπει να έχετε κάνει:

```
import java.util.Scanner;
```

- Για να πάρετε είσοδο από παράθυρο διαλόγου χρειάζεται να έχετε κάνει

```
import javax.swing.JOptionPane;
```


και να χρησιμοποιήσετε μια εντολή της μορφής:

```
int inputk = Integer.parseInt(JOptionPane.showInputDialog(  
"Give me a number ",5)); // το 5 είναι η default τιμή
```

Αφού διαβάσει την νέα τιμή L θα ζωγραφίζει το Δ και αυτό θα γίνεται έως ότου ο χρήστης δώσει ένα L που δεν είναι στο επιτρεπτό εύρος τιμών.

(vii) [10 μονάδες] - Αναδρομική Κλήση της main

Η main μιας κλάσης είναι μια μέθοδος που μπορεί να κληθεί όπως όλες οι μέθοδοι (δηλαδή δεν έχει κάτι το ιδιαίτερο). Για να το κατανοήσουμε, και να δούμε μια πρώτη μορφή αναδρομικής κλήσης, εμπλουτίστε το πρόγραμμά σας ως εξής: Λίγο πριν ζητήσει νέα τιμή του L (σκέλος vii) να καλείται η main αλλά με διαφορετικές παραμέτρους: συγκεκριμένα με ένα L μειωμένο κατά ένα. Με αυτόν τον τρόπο αν η αρχική κλήση ήταν με L=10, τότε το πρόγραμμα θα ζωγραφίσει το Δ με μέγεθος 10, κατόπιν με 9, κ.ο.κ. έως 3 (το ελάχιστο αποδεκτό) και μετά θα ζητήσει νέα τιμή από το χρήστη (στην ουσία λόγω της αναδρομής θα δείτε ότι θα ζητήσει νέα τιμή από το χρήστη τόσες φορές όσες οι κλήσεις της main που έγιναν).

ΣΗΜΕΙΩΣΗ.

Μάθετε να εκτελείτε το πρόγραμμα σας (και να περνάτε τις παραμέτρους που χρειάζονται) με διάφορους τρόπους: (α) από το IDE που χρησιμοποιείτε, (β) από command line, (γ) φτιάχνοντας ένα jar αρχείο (οδηγίες για εκτέλεση από command line και για δημιουργία jar αρχείων υπάρχουν στις διαφάνειες του φροντιστηρίου).

Υποδείξεις

- Μετατροπή μιας συμβολοσειράς str σε ακέραιο: `int x = Integer.parseInt(str);`
- Για να διαβάζετε την είσοδο stdin :
`Scanner in = new Scanner(System.in);`
`int aNumber = in.nextInt();`
`// String nextLine = in.next(); // Αν θέλατε να διαβάσετε String`

Στην αρχή του αρχείου θα πρέπει να έχετε κάνει:

```
import java.util.Scanner;
```

- Για να πάρετε είσοδο από παράθυρο διαλόγου χρειάζεται να έχετε κάνει
`import javax.swing.JOptionPane;`
και να χρησιμοποιήσετε μια εντολή της μορφής:
`int inputk = Integer.parseInt(JOptionPane.showInputDialog("Give me a number ",4)); // το 4 είναι η default τιμή`
- Για να εκτυπώσετε μια συμβολοσειρά σε παράθυρο εξόδου
`JOptionPane.showMessageDialog(null,`
`"Ακολουθούν 2 αστεράκια τυπωμένα σε δύο γραμμές \n*\n* ",`
`"Παράθυρο Εξόδου",`
`JOptionPane.INFORMATION_MESSAGE);`

Για παράθυρα άλλου τύπου (προειδοποίησης, λάθους) δείτε τι άλλες τιμές μπορούν να περάσουν στην τελευταία παράμετρο (αντί του `JOptionPane.INFORMATION_MESSAGE`)

- Για τη δημιουργία αρχείων και για γράψιμο σε αρχεία δείτε το υλικό από το σχετικό φροντιστήριο που θα γίνει. Ένα απλό παράδειγμα ακολουθεί:

```
PrintWriter writer;  
try {  
    writer = new PrintWriter("C:\\temp\\AAA.txt", "UTF-8");  
    writer.println("The first line");  
    writer.println("The second line");  
    writer.close();  
} catch (Exception e) {  
    System.out.println("Πρόβλημα: "+e);  
}
```

Άσκηση 2 – [30 μονάδες] Είσοδος/Εξοδος, Συμβολοσειρές, Πίνακες, Αρχεία

(α) [10 μονάδες]

Καλείστε να γράψετε μια κλάση με όνομα **PalindromikesFrasesis** η οποία θα έχει κώδικα ο οποίος θα μπορεί να κρίνει αν μία συμβολοσειρά είναι *παλινδρομική* δηλαδή αν διαβάζοντας την ανάποδα παίρνουμε την ίδια συμβολοσειρά. Παραδείγματα παλινδρομικών συμβολοσειρών ακολουθούν: «**abcba**», «**A man, a plan, a canal: Panama**», «**Άννα**» ή το πιο γνωστό «**Νίψον ανομήματα μη μόναν όψιν**».

Συγκεκριμένα, το πρόγραμμα θα ζητάει από το χρήστη να εισάγει μια φράση και αυτό θα του επιστρέφει ανάλογο μήνυμα για το εάν η φράση είναι παλινδρομική ή όχι. Η `main` θα καλεί μια στατική μέθοδο της κλάσης με υπογραφή

```
static boolean isPalindromikiFrash(String s)
```

Εκείνη η μέθοδος πρέπει να μετατρέπει το αλφαριθμητικό `s` σε δύο πίνακες από χαρακτήρες. Ο πρώτος θα περιέχει όλους τους χαρακτήρες της φράσης με τη σειρά που τους εισήγαγε ο χρήστης ενώ ο δεύτερος τους χαρακτήρες αυτούς ανάποδα.

Επίσης το πρόγραμμα σας πρέπει να μετράει πόσο χρόνο του πήρε για να κάνει αυτόν τον έλεγχο

Προσοχή: οι ειδικοί χαρακτήρες και τυχόν κεφάλαια γράμματα ή τόνοι θα πρέπει να αγνοούνται (αυτό βολεύει να γίνει πριν γεμίσετε τους δύο πίνακες), άρα το πρόγραμμα σας να μπορεί να εντοπίσει ότι είναι παλινδρομικές οι φράσεις που αναγράφονται στην εκφώνηση

Υποδείξεις:

- Για να διαβάσετε την είσοδο `stdin` :

```
Scanner in = new Scanner(System.in);
String nextLine = in.nextLine();
```
- Για να μετράτε τον χρόνο σε `nano seconds` χρησιμοποιείτε :

```
long time = System.nanoTime();
```
- Για τον έλεγχο των κεφαλαίων γραμμάτων και των ειδικών χαρακτήρων, χρήσιμες μεθόδους θα βρείτε στο API της κλάσης `String`
(<http://docs.oracle.com/javase/6/docs/api/java/lang/String.html>)
- Για να βοηθηθείτε δείτε τι προσφέρει έτοιμο η κλάση `String` της Java. Επίσης μπορεί να σας φανεί χρήσιμη η ακόλουθη εντολή η οποία βγάζει τόνους από τα γράμματα του
 - `Normalizer.normalize(text, Form.NFD).replaceAll("\\p{InCombiningDiacriticalMarks}+", "")`. Για παράδειγμα η:

```
System.out.println(Normalizer.normalize("Γιάννης",
Form.NFD).replaceAll("\\p{InCombiningDiacriticalMarks}+", ""));
```

θα εκτυπώσει «Γιαννης»

(β) [10 μονάδες]

Κατεβάστε το αρχείο **gr.dic**¹ το οποίο περιέχει πάνω από μισό εκατομμύριο διαφορετικές ελληνικές λέξεις (και ονόματα), και αποθηκεύστε το στο project σας (π.χ. σε ένα φάκελο Resources).

Κατόπιν φτιάξτε μια κλάση με όνομα **PalindromikesLexikou** η οποία θα έχει κώδικα που θα ανοίγει και θα διαβάζει το αρχείο, και σε ένα πέρασμα θα τυπώνει:

- (α) το πλήθος των λέξεων του λεξικού,
- (β) το μέσο όρο (average) του μήκους (σε γράμματα) των λέξεων,
- (γ) το πλήθος των παλινδρομικών λέξεων (καλώντας την μέθοδο που κάνατε στην Άσκηση 2(α)),
- (δ) τις παλινδρομικές λέξεις που βρήκε,
- (ε) το ποσοστό των παλινδρομικών λέξεων επί του συνόλου,
- (στ) το χρόνο εκτέλεσης του προγράμματός σας, και
- (ζ) στο τέλος να τυπώνει τις παλινδρομικές λέξεις που βρήκε σε φθίνουσα σειρά ως το μήκος τους (για το σκοπό αυτό είναι καλό το πρόγραμμα σας να έχει έναν πίνακα από συμβολοσειρές και όποτε βρίσκει μία να την προσθέτει σε αυτόν), άρα να τυπώνει κάτι της μορφής:

εξέλεξε ενέμενε ανέμενα ανάσσανα όρθρο ωθηθώ ρεβέρ νώτων νώμων νότον νόμον νωπών
νωδών ανανά αιτία αιτία αίσια έσωσε ένωνε έδιδε άρθρα άπαπα άμωμα άλαλα άκακα Νά
θαν Λασάλ Λαβάλ Αιαία ορρό αλλά άλλα όψο όσο όρο όνο όλο όζο ωθώ τετ ποπ ορό οπό
οδό νυν κοκ ιοί εφέ ατά αρά ανά αλά αγά αβά ήθη ήδη ήβη έχε άψα άρα άμα άβα TNT
ΠΑΠ ΝΙΝ ΑΣά Αία Ότο Έσε

Υποδείξεις:

Ανοιγμα και ανάγνωση αρχείου γραμμή γραμμή

```
try {
    FileInputStream fis=new FileInputStream("Resources/gr.dic");    //file to be scanned
    Scanner sc=new Scanner(fis);
    while(sc.hasNextLine()) {    //returns true if there is another line to read
        System.out.println(sc.nextLine());    //returns the line that was skipped
    }
    sc.close();    //closes the scanner
} catch(IOException e) {
    System.out.println(e);
}
```

Χρονομέτρηση

// finding the time before the operation is executed

long start = System.currentTimeMillis();

<κώδικας που θέλουμε να χρονομετρήσουμε>

// finding the time after the operation is executed

long end = System.currentTimeMillis();

//finding the time difference and converting it into seconds

¹ από το φάκελο στη διεύθυνση

<https://github.com/YannisTzitzikas/SoundexGR/tree/master/SoundexGR/Resources/dictionaries/EN-winedt>


```
float sec = (end - start) / 1000F; System.out.println(sec + " seconds");
```

(γ) [10 μονάδες]

Πληροφορίες για τη βιβλιοθήκη *jfugue*. Σας δίνεται (στα *templates* στο φάκελο *Library*) ένα *jar* (με όνομα *jfugue-4.0.3.jar*) το οποίο πρέπει να προσθέσετε στη βιβλιοθήκη του *project* σας. Για να το χρησιμοποιήσετε στον κώδικά σας, πρέπει να κάνετε το σχετικό *import* (*import org.jfugue.Player;*). Για να παίξετε τη μουσική νότα που αντιστοιχεί σε έναν ακέραιο (από το 0 έως το 127), ας πούμε για τον ακέραιο 80, αρκούν τα παρακάτω:

```
Player p = new Player();
```

```
p.play("[80] ");
```

(Δείτε αν θέλετε και το σχετικό κεφάλαιο για αυτήν την βιβλιοθήκη²).

Στην άσκηση αυτή καλείστε να εμπλουτίσετε τον κώδικά σας, ώστε να ... «μελοποιήσετε» τις παλινδρομικές λέξεις που βρήκατε (στο βήμα ζ). Συγκεκριμένα αντιστοιχείστε κάθε γράμμα του ελληνικού αλφαβήτου μιας λέξης σε έναν αριθμό από το 0..127 (κατά προτίμηση από 60 και μετά). Με αυτόν τον τρόπο κάθε λέξη αντιστοιχίζεται στην ακολουθία ακεραίων των γραμμάτων της. Πριν κάνετε την αντιστοίχιση μιας λέξης, να έχετε αφαιρέσει τους τόνους και να τα έχετε κάνει μικρά. Ένα παράδειγμα που δείχνει λέξεις και τις νότες ακολουθεί (το *i* είναι η διάρκεια της νότας, μπορείτε εκτός από *i* να βάλετε *q,h,w* ό,τι υποστηρίζει η βιβλιοθήκη³).

```
word=Εσε, normalized word=εσε, music score= [64]i [78]i [64]i  
word=Οτο, normalized word=οτο, music score= [74]i [79]i [74]i
```

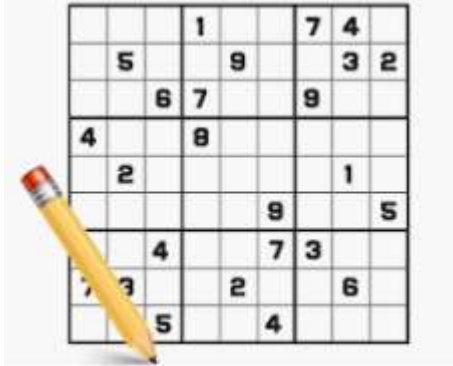
Το πρόγραμμα σας θα στέλνει κάθε τέτοια λέξη στον *player* ώστε να παράγεται ο σχετικός ήχος. (είστε ελεύθεροι να κάνετε ό,τι αντιστοίχιση σε νότες θέλετε).

² ² Πληροφορίες για το τι υποστηρίζει η *jfugue* μπορείτε να βρείτε εδώ:
<https://www.cs.utexas.edu/ftp/novak/cs315/jfugue-chapter2.pdf>

³ Πληροφορίες για το τι υποστηρίζει η *jfugue* μπορείτε να βρείτε εδώ:
<https://www.cs.utexas.edu/ftp/novak/cs315/jfugue-chapter2.pdf>

Άσκηση 3 – [30 μονάδες] Sudoku

Κατανόηση και εμπλουτισμός κώδικα, πίνακες, χρονομέτρηση.



Η άσκηση αυτή σχετίζεται με το γνωστό παιχνίδι Sudoku, το διασκεδαστικό Γιαπωνέζικο παιχνίδι παζλ που χρησιμοποιεί αριθμούς, αλλά δεν χρειάζεται μαθηματικά.

Αρχικά διαβάστε το άρθρο

<https://www.baeldung.com/java-sudoku>.

Εν συνεχεία κατεβάστε και τρέξτε την έκδοση με **backtracking** από το αρχείο

<https://github.com/eugenp/tutorials/blob/master/algorithms-modules/algorithms-miscellaneous-2/src/main/java/com/baeldung/algorithms/sudoku/BacktrackingAlgorithm.java>

Η οποία μπορεί να λύσει ένα παζλ Sudoku.

ΠΡΟΣΟΧΗ: Για να κάνετε αυτήν την άσκηση **δεν χρειάζεται να καταλάβετε ακριβώς τον αλγόριθμο και τον κώδικα**, απλά τι κάνει η κάθε μέθοδος γενικά (δηλαδή ποιος είναι ο ρόλος της).

Εν συνεχεία εμπλουτίστε τον κώδικα ώστε:

(α) [5 μονάδες] Ο αλγόριθμος προϋποθέτει ότι το αρχικό πάζλ (board) που του δίδεται είναι σωστό, και δεν κάνει κάποιο έλεγχο. Αυτό μπορείτε να το καταλάβετε μόνοι σας αν αλλάξετε την πρώτη γραμμή του παζλ που βρίσκεται μέσα στον κώδικα ως εξής:

από

```
{8, 0, 0, 0, 0, 0, 0, 0, 0}
```

σε

```
{12, 0, 0, 0, 0, 0, 0, 0, 0} // θα λάβουμε runtime exception
```

ή σε

```
{8, 8, 0, 0, 0, 0, 0, 0, 0} // θα μας επιστρέψει το board άλυτο
```

Ή σε

```
{8, 6, 0, 0, 0, 0, 0, 0, 0} // θα μας επιστρέψει το board άλυτο χωρίς να μας ενημερώσει για κάτι
```

Για το λόγο αυτό προσθέστε μια μέθοδο **boolean isValidBoard(int[][] brd)** που να ελέγχει αν πράγματι το παζλ είναι σωστό (δηλαδή αν δεν παραβιάζεται κάποιος περιορισμός του sudoku):

- Τα κελιά κάθε γραμμής έχουν κενά (δηλαδή 0) ή τιμές από το 1 έως το 9 και όχι διπλότυπα

- Τα κελιά κάθε γραμμής στήλης έχουν κενά ή τιμές από το 1 έως το 9 και όχι διπλότυπα
- Κάθε ένα από τα 9 πλέγματα 3x3 δεν έχει διπλότυπα

(β) [5 μονάδες] Προσθέστε μια μέθοδο **isSolvableBoard(board)** η οποία να επιστρέφει true αν το board είναι ορθό και επιλύεται (αξιοποιώντας τον κώδικα που κατεβάσατε και αυτά που κάνατε στο προηγούμενο σκέλος).

(γ) [10 μονάδες] Προσθέστε μια μέθοδο που να μπορεί να φτιάξει ένα board το οποίο να έχει **X** κενά κελιά (π.χ. **X=75**) και τα υπόλοιπα να τα γεμίζει τυχαία με αριθμούς.

(δ) [10 μονάδες] Γράψτε κώδικα που να μπορεί να δημιουργεί **N** **έγκυρα** ζευγάρια Sudoku (άσκηση-λύση). Ο κώδικας σας μπορεί πρώτα να παράγει ένα τυχαίο παζλ (μέσω αυτού που κάνατε στο σκέλος γ) και μετά να ελέγχει αν είναι έγκυρο και επιλύσιμο (σκέλος β). Στο τέλος να τυπώνει τα έγκυρα boards που κατάφερε να φτιάξει και επίσης πόσο χρόνος δαπανήθηκε, και πόσα άκυρα boards φτιάχτηκαν. Προσπαθήστε ώστε ο κώδικας σας να μπορεί να παράξει τουλάχιστον 10 έγκυρα ζευγάρια σε λιγότερο από 10 λεπτά.

Η εκτέλεση του προγράμματος σας θα τυπώνει κάτι της μορφής (εδώ):

N=100 X=75	N=10 X=65	N=10 X=54
Board #1 0 0 0 0 0 0 2 0 0 0 0 2 6 0 7 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 Solution of the Board #1 1 4 3 5 7 8 2 6 9 5 7 2 6 1 9 3 4 8 6 8 9 2 3 4 1 5 7 2 1 4 3 8 5 7 9 6 3 5 7 9 2 6 4 8 1 8 9 6 7 4 1 5 2 3 4 2 1 8 6 3 9 7 5 9 3 8 4 5 7 6 1 2 7 6 5 1 9 2 8 3 4 ... Board #100 0 0 0 0 0 6 0 0 0 7 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 3 0 0 0 0 0 0 0 Solution of the Board #100 1 2 3 4 5 6 7 8 9 7 4 6 2 8 9 1 3 5	Board #1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 4 0 0 0 9 0 5 0 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0 0 7 0 6 0 0 3 0 0 0 8 0 0 0 7 4 0 0 0 0 0 0 7 0 0 0 5 0 0 0 4 0 0 0 0 0 0 0 0 1 2 0 Solution of the Board #1 2 3 4 6 9 7 5 1 8 1 5 6 2 3 8 7 4 9 7 8 9 1 5 4 3 6 2 5 1 7 9 6 3 2 8 4 9 4 8 5 7 2 6 3 1 3 6 2 4 8 1 9 5 7 4 9 1 3 2 5 8 7 6 8 2 5 7 1 6 4 9 3 6 7 3 8 4 9 1 2 5 ... Board #10 5 0 0 0 0 8 0 0 0 2 0 0 0 4 6 0 0 1 0 0 6 0 0 0 0 0 0 0 0 0 0 0 3 0 9 0 7 0 0 0 0 0 0 0 0 0 0 0 9 0 2 0 0 0 0 0 0 0 8 0 0 0 0 0 0 0 0 0 7 0 8 0 0 0 3 0 0 0 0 0 0 Solution of the Board #10 5 1 4 2 3 8 6 7 9	Board #1 0 0 0 0 0 0 8 0 5 0 0 0 0 9 4 0 0 0 0 3 0 0 1 8 0 0 0 0 0 5 0 0 0 0 0 0 8 0 0 0 0 9 7 2 0 9 6 7 8 0 0 0 5 1 0 0 1 0 0 3 0 0 0 2 0 3 0 0 5 0 9 8 0 0 6 7 0 0 0 0 0 Solution of the Board #1 1 4 9 2 3 7 8 6 5 7 5 8 6 9 4 2 1 3 6 3 2 5 1 8 4 7 9 3 2 5 1 7 6 9 8 4 8 1 4 3 5 9 7 2 6 9 6 7 8 4 2 3 5 1 5 8 1 9 2 3 6 4 7 2 7 3 4 6 5 1 9 8 4 9 6 7 8 1 5 3 2 ... Board #10 5 4 2 0 0 0 0 9 0 7 0 0 0 0 0 0 4 0 0 0 8 5 6 0 0 0 0 4 2 0 0 8 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 7 3 2 0 0 0 0 3 0 9 2 7 0 0 1 5 0 0 5 0 0 3 0 8 7 0 0 0 6 0 0 0 0 0 Solution of the Board #10 5 4 2 1 3 7 8 9 6

5 8 9 3 1 7 2 4 6 2 6 1 5 3 4 8 9 7 3 5 4 7 9 8 6 1 2 8 9 7 1 6 2 3 5 4 4 1 2 8 7 5 9 6 3 6 7 8 9 4 3 5 2 1 9 3 5 6 2 1 4 7 8 Empty cells per board : 75 Valid boards created : 100 Invalid boards created : 42 Unsolvable boards created: 0 Elapsed time in seconds : 0.426	2 3 9 7 4 6 8 5 1 8 7 6 1 5 9 2 3 4 1 4 2 5 6 3 7 9 8 7 9 5 8 1 4 3 2 6 3 6 8 9 7 2 1 4 5 4 2 7 6 8 5 9 1 3 6 5 1 3 9 7 4 8 2 9 8 3 4 2 1 5 6 7 Empty cells per board : 65 Valid boards created : 10 Invalid boards created : 506 Unsolvable boards created: 1 Elapsed time in seconds : 10.71	7 3 6 8 9 2 5 4 1 1 9 8 5 6 4 3 7 2 4 2 1 7 8 5 9 6 3 6 5 3 4 1 9 7 2 8 9 8 7 3 2 6 1 5 4 3 6 9 2 7 8 4 1 5 2 1 5 9 4 3 6 8 7 8 7 4 6 5 1 2 3 9 Empty cells per board : 54 Valid boards created : 10 Invalid boards created : 64614 Unsolvable boards created: 62 Elapsed time in seconds : 21.555
---	--	--

ΠΡΟΑΙΡΕΤΙΚΟ: Προφανώς αν κάποιος θέλει να παράγει πολλά παζλ, θα μπορούσε επίσης από τη λύση του κάθε παζλ να παράγει πολλά άλλα, σβήνοντας τυχαία αριθμούς από τη λύση του παζλ. Δεν χρειάζεται εσείς να κάνετε κάτι τέτοιο, απλά το σημειώνουμε για τη δική σας ενημέρωση. Αν κάποιος όμως το επιθυμεί μπορεί προαιρετικά να κάνει και αυτό.

Υποδείξεις

Παραγωγή τυχαίου ακεραίου από MIN_VALUE έως MAX_VALUE:

```
Random randomGenerator = new Random();
int randomInt = randomGenerator.nextInt(MAX_VALUE-MIN_VALUE+1) + MIN_VALUE;
```

BONUS ΑΣΚΗΣΗ

Άσκηση 4 – Κρυπτογράφηση Αρχείων [5% bonus]

File, I/O, Χρήση κλάσης της οποίας ο κώδικας δίδεται

(α)

Γράψτε μια κλάση Java με όνομα `encrypt` (σε αρχείο `encrypt.java`). Η `main` της θα λαμβάνει ως παραμέτρους από την γραμμή εντολών το όνομα ενός αρχείου εισόδου (π.χ. `myMessage.txt`) και έναν φυσικό αριθμό, ας τον ονομάσουμε `K`, και θα δημιουργεί ένα νέο αρχείο του οποίου το όνομα θα περιέχει τη λέξη `Encrypted` στην αρχή (π.χ. `EncryptedmyMessage.txt`). Το περιεχόμενο του νέου αρχείου θα πρέπει να είναι το περιεχόμενο του αρχείου εισόδου κρυπτογραφημένο.

Συγκεκριμένα η κλάση `encrypt` πρέπει να έχει την εξής στατική μέθοδο:

```
(i) static void Transform(String inputFileName, String outputFileName, int inputK)
```

η οποία θα καλείται από την `main` της κλάσης με τα κατάλληλα ορίσματα. Θα διαβάζει/γράφει το αρχείο εισόδου/εξόδου ανά byte. Εκ τούτου πρέπει να χρησιμοποιεί την `FileInputStream` και `FileOutputStream`. Η κρυπτογράφηση που θα προσφέρεται θα γίνεται αντικαθιστώντας το κάθε byte με ένα byte στο οποίο έχει προστεθεί η τιμή `inputK` (που αντιστοιχεί στην παράμετρο `K` από τη γραμμή εντολών).

Προφανώς η ίδια κλάση επαρκεί και για την αποκρυπτογράφηση ενός κρυπτογραφημένου αρχείου. Για παράδειγμα αν έχουμε ένα αρχείο `myMessage.txt` η εντολή

```
java encrypt myMessage.txt 2
```

θα δημιουργήσει το αρχείο `EncryptedmyMessage.txt` του οποίου το περιεχόμενο θα είναι κρυπτογραφημένο.

Για να το αποκρυπτογραφήσουμε αρκεί η εντολή

```
java encrypt EncryptedmyMessage.txt -2
```

το οποίο θα δημιουργήσει το αρχείο `EncryptedEncryptedmyMessage.txt` του οποίου το περιεχόμενο πρέπει (αν η υλοποίηση είναι σωστή) να είναι αυτό του αρχικού (ήτοι αυτό του `myMessage.txt`).

Το πρόγραμμά σας στο τέλος πρέπει να τυπώνει στη κονσόλα πόσο χρόνο πήρε η κρυπτογράφηση.

Δοκιμάστε αυτό που κατασκευάσατε και δείτε:

1. Αν δουλεύει σωστά σε οποιοδήποτε αρχείο, π.χ. `.txt`, `.pdf`, `.doc`, `.zip`
2. Υπάρχει πρόβλημα εάν ο αριθμός `K` είναι πολύ μεγάλος;
3. Δοκιμάστε να κρυπτογραφήσετε και πολύ μεγάλα αρχεία και σχολιάστε το χρόνο που απαιτήθηκε.

(β)

Για να βεβαιωθείτε ότι δουλεύει σωστά αυτό που κάνατε, προσθέστε στην κλάση μια στατική μέθοδο `boolean test()` η οποία θα κάνει διάφορους ελέγχους ορθότητας. Η μέθοδος θα επιστρέφει `true` αν όλοι οι έλεγχοι εκτελεστούν επιτυχώς. Για παράδειγμα μετά από μια κλήση του χρήστη:

```
java encrypt.java LaLa.exe 7
```

πρέπει η `main` να καλεί την `test()`, η οποία θα μπορούσε να καλεί την `transform` με `K = 7` και μετά να μετατρέπει το αποτέλεσμα με `K = -7`, και στο τέλος να ελέγχει ότι το αρχείο που προκύπτει έχει τα ίδια

ακριβώς περιεχόμενα με το αρχικό. Στην περίπτωση που η μέθοδος επιστρέψει false, η κονσόλα πρέπει να τυπώνει ένα αντίστοιχο μήνυμα. Είστε ελεύθεροι να κάνετε όποιο άλλο έλεγχο κρίνετε σκόπιμο.

(γ)

Εμπλουτίστε την κλάση που φτιάξατε και με την εξής μέθοδο:

```
static void Transform(File inputFile, File outputFile, int inputK)
```

Στην ουσία η παραπάνω απλά λαμβάνει διαφορετικού τύπου παραμέτρους. Αναπαραγοντοποιήστε (refactor) ανάλογα την προηγούμενη μέθοδο (του σκέλους α)) ώστε να μην υπάρχουν στον κώδικα της κλάσης σας επαναλήψεις (κάτι το οποίο είναι κακή προγραμματιστική πρακτική), αλλά η προηγούμενη μέθοδος (του σκέλους α) να καλεί σε κάποιο σημείο τη νέα (του σκέλους γ). Εν συνεχεία χρησιμοποιήστε τις εντολές που θα βρείτε στις υποδείξεις ώστε η επιλογή του αρχείου προς κρυπτογράφηση (ή αποκρυπτογράφηση) να γίνεται με γραφικό (παραθυρικό) τρόπο.

Εμπλουτίστε τη ροή ελέγχου της main έτσι ώστε η επιλογή του γραφικού τρόπου να ενεργοποιείται εάν ο χρήστης καλέσει τη κλάση encrypt χωρίς καμία παράμετρο εισόδου. Αν ο χρήστης έχει δώσει παραμέτρους εισόδου τότε το πρόγραμμα πρέπει να λειτουργεί όπως στο σκέλος (α).

Προαιρετικά

1/ Να λαμβάνει ως είσοδο το όνομα ενός φακέλου και να κρυπτογραφεί όλα τα περιεχόμενα του.

Υποδείξεις

- Για να μετράτε τον χρόνο σε nano seconds χρησιμοποιήστε:

```
long startTime = System.nanoTime();
... (code)
... (code)
long endTime = System.nanoTime();
long timeInNanosecs = endTime - startTime;
```

- Για να επιλέξετε ένα αρχείο από παράθυρο διαλόγου χρειάζεται να έχετε κάνει:

```
import javax.swing.JFileChooser;

και να χρησιμοποιήσετε τις παρακάτω εντολές:
JFileChooser fileChooser = new JFileChooser();
fileChooser.setDialogTitle("Select a file");
int userSelection = fileChooser.showSaveDialog(null);
if (userSelection == JFileChooser.APPROVE_OPTION) {
    File file = fileChooser.getSelectedFile();    String filepath =
file.getAbsolutePath();
    System.out.println("The path of the selected file is: " + filepath);
}
```

Εναλλακτικά, μπορείτε να χρησιμοποιήσετε τη σχετική κλάση (GraphicalFileLoaderSaver.java) που έχει ανέβει στο moodle (συγκεκριμένα την μέθοδο GraphicalFileLoaderSaver.openFile()).

Καλή εργασία