



Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών  
**HY252 – Αντικειμενοστρεφής Προγραμματισμός**  
Εξάμηνο: Χειμερινό 2015-2016  
Διδάσκων: Γιάννης Τζίτζικας  
Βοηθοί: Π. Φαφαλιός, Μ. Μπαριτάκης, Γ. Αθηναίου

## 1<sup>η</sup> Σειρά Ασκήσεων

Ανάθεση: 30 Σεπτεμβρίου 2015

Παράδοση: 20 Οκτωβρίου 2015

### Εκπαιδευτικοί Στόχοι

Η 1<sup>η</sup> 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<sup>η</sup> άσκηση (10 μονάδες) θα σας επιτρέψει να εξοικειωθείτε με τη κλήση στατικών μεθόδων, τυχαίους αριθμούς και στρογγυλεύσεις.

Η 3<sup>η</sup> άσκηση (20 μονάδες) θα σας επιτρέψει να εξοικειωθείτε με τους πίνακες στην Java (ένας πίνακας είναι αντικείμενο) και να χρησιμοποιείτε εξωτερικό κώδικα σε μορφή βιβλιοθήκης jar.

Η 4<sup>η</sup> άσκηση (30 μονάδες) θα σας επιτρέψει να εξοικειωθείτε με τη διαχείριση αρχείων και με τη χρήση κώδικα που σας δίνεται.

Η 5<sup>η</sup> άσκηση (bonus 10 μονάδες) θα σας επιτρέψει να εξοικειωθείτε με τη διαχείριση αρχείων και με τη χρήση κλάσης που σας δίνεται.

### Σημειώσεις

1. **Απορίες** σχετικά με την Α1 θα απαντώνται μόνο μέσω του forum της Α1 στο moodle.
2. Τις πρώτες δύο πρέπει να μπορείτε να τις κάνετε compile και να τις τρέξετε από **γραμμή εντολών**, χωρίς δηλαδή τη χρήση κάποιου **IDE** (έτσι θα γίνει και η εξέταση). Τις δυο τελευταίες (3η και τη 4<sup>η</sup>) μπορείτε να τις κάνετε χρησιμοποιώντας ένα IDE (Eclipse ή NetBeans). Σχετικό φροντιστήριο έχει προγραμματιστεί.
3. **Ημερομηνία παράδοσης** και **bonus**: Όποιος παραδώσει δύο μέρες πριν την προθεσμία παράδοσης θα έχει βαθμολογικό bonus 10%.
4. **Οδηγίες Παράδοσης**. Φτιάξτε ένα φάκελο με όνομα A1\_<<ΑριθμόςΜητρώου>>. Μέσα του δημιουργήστε έναν υποφάκελο για κάθε άσκηση (A11, A12, ...). Συμπιεστέ το φάκελο με όνομα A1\_<<ΑριθμόςΜητρώου>>(**Προσοχή**: όσες ασκήσεις παραδοθούν με διαφορετικό όνομα δεν θα γίνονται δεκτές), και παραδώστε τον μέσω του moodle.  
**ΠΡΟΣΟΧΗ: Πρέπει να παραδώσετε τα .java αρχεία, δηλαδή τα αρχεία που περιέχουν τον κώδικά σας, και όχι τα εκτελέσιμα .class αρχεία!**
5. Η **εξέταση** της Α1 θα γίνει μέσω μίας υποχρεωτικής εργαστηριακής άσκησης (**E1**). 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).

## Άσκηση 1 [40 μονάδες] Το Φ

Πρώτη επαφή με Java, ροή ελέγχου, είσοδος/έξοδος από κονσόλα και παράθυρα διαλόγου.

### (α) [20 μονάδες]

Γράψτε μία κλάση DrawPhi (σε αρχείο DrawPhi.java) της οποίας η μέθοδος main θα διαβάζει την πρώτη παράμετρο από την γραμμή εντολών (command line argument), έστω L η τιμή αυτής της παραμέτρου (που πρέπει να είναι ένας ακέραιος αριθμός), και εν συνεχεία θα εκτυπώνει στην έξοδο με αστεράκια (\*) το ελληνικό γράμμα Φ χρησιμοποιώντας ακριβώς L γραμμές. Αυτό πρέπει να γίνεται μόνο αν η τιμή L είναι μεγαλύτερη ή ίση του 4, αλλιώς δεν πρέπει να κάνει τίποτα.

Για παράδειγμα με java DrawPhi 4 μπορεί να εκτυπώνεται το:

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

ενώ με java DrawPhi 5 μπορεί να εκτυπώνεται το:

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

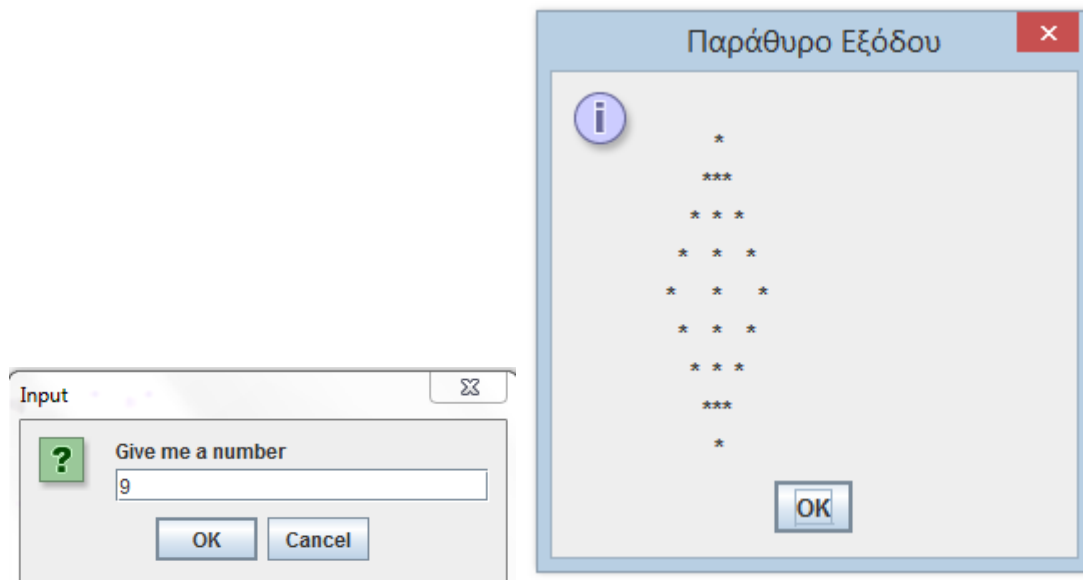
Το βασικό είναι να χρησιμοποιούνται πάντα L γραμμές και το αποτέλεσμα να μοιάζει με το γράμμα Φ (να μην μπορεί δηλαδή κάποιος να το μπερδέψει με κάποιο άλλο γράμμα).

### (β) [5 μονάδες]

Επεκτείνεται το πρόγραμμα σας έτσι ώστε μετά την εκτύπωση του Φ να ζητάει από το χρήστη να δώσει πάλι μια παράμετρο L και εν συνεχεία να εκτυπώνει το αντίστοιχο Φ κ.ο.κ. Αυτό πρέπει να σταματάει μόλις ο χρήστης δώσει μια τιμή μικρότερη του 4 ή μεγαλύτερη του 30.

### (γ) [5 μονάδες]

Δημιουργείστε τώρα μια άλλη κλάση DrawPhiwin (σε αρχείο DrawPhiwin.java), αντιγράφοντας την προηγούμενη. Η νέα κλάση πρέπει να λειτουργεί όπως αυτή του σκέλους (β) με τη μόνη διαφορά ότι θα διαβάζει την παράμετρο L από ένα παράθυρο διαλόγου και θα εκτυπώνει το Φ σε ένα παράθυρο διαλόγου **αφήνοντας διπλάσια κενά** (για αισθητικούς λόγους). Παράδειγμα:



Για τα παράθυρα διαλόγου δείτε τις υποδείξεις. Αφού στο παράθυρο εξόδου θα περνάτε μία συμβολοσειρά, ο αλγόριθμος που είχατε κάνει στο προηγούμενο σκέλος πρέπει να τροποποιηθεί ώστε πλέον να φτιάχνει μία συμβολοσειρά την οποία στο τέλος θα περνάει στο παράθυρο εξόδου.

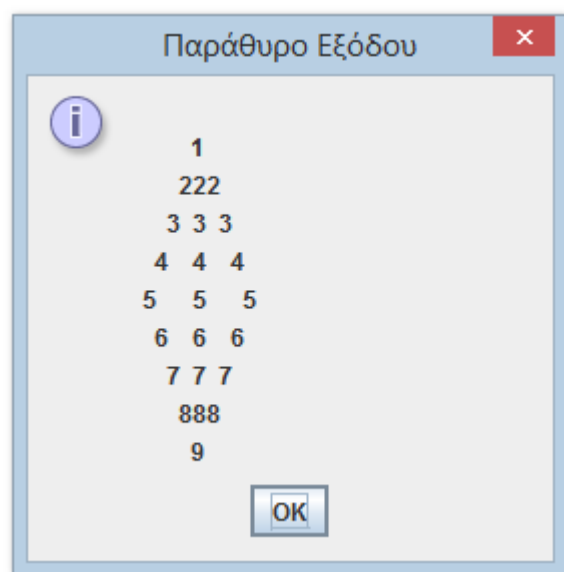
#### (δ) [5 μονάδες]

Κάντε μια παραλλαγή (drawPhiwin2) ώστε η έξοδος αντί για αστεράκια να έχει αριθμούς.

Συγκεκριμένα κάθε αστεράκι μιας γραμμής  $j$  να έχει αντικατασταθεί με τον αριθμό  $j$ .

Ομοίως με πριν θα πρέπει να αφήνει **διπλάσια κενά** σε σχέση με την υλοποίηση του ερωτήματος (α)

Π.χ. για είσοδο ίση με το 9 το παράθυρο εξόδου πρέπει να έχει την εξής μορφή:



### (ε) [5 μονάδες]

Δημιουργήστε τώρα μια άλλη κλάση (drawPhiHtml) της οποίας η κλήση από τη γραμμή εντολών να δημιουργεί ένα html αρχείο με όνομα PhiK.html όπως το ακόλουθο

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

<p style="font-size: K*10%" align="center">Φ</p>

</body>
</html>
```

Το αρχείο αυτό θα μπορείτε να το ανοίξετε με ένα ιστοπλοηγητή. Παρατηρείστε ότι εδώ η τιμή του K πολλαπλασιάζεται επί 10 και γίνεται font size στο HTML αρχείο.

**Παραδοτέα:** Δημιουργήστε ένα φάκελο A11 και μέσα βάλτε τις κλάσεις που υλοποιούν τα παραπάνω.

#### Υποδείξεις

- Μετατροπή μιας συμβολοσειράς str σε ακέραιο: `int x = Integer.parseInt(str);`
- 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 ",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 τη δημιουργία αρχείων και για γράψιμο σε αυτό δείτε το υλικό από το σχετικό φροντιστήριο που θα γίνει.

## Άσκηση 2 [10 μονάδες] Κλήσεις Μεθόδων

### (α) [5 μονάδες]

Γράψτε μία κλάση (CyclicCalls) με μία main η οποία με τη σειρά της θα καλεί μία (static μέθοδο) m1 με όρισμα τον ακέραιο αριθμό 10. Η m1 θα εκτυπώνει στην κονσόλα το μήνυμα “m1 called with argument 10” και στη συνέχεια θα πολλαπλασιάζει το όρισμα που έλαβε με έναν τυχαίο αριθμό από το 0 έως το 1. Κατόπιν θα στρογγυλεύει το γινόμενο στον πλησιέστερο ακέραιο και θα τον περνάει ως παράμετρο στην κλήση της στατικής μεθόδου m2.

Ο κώδικας της m2 θα είναι αντίστοιχος με αυτόν της m1 με τη μόνη διαφορά ότι οι εκτυπώσεις στην κονσόλα πρέπει να αναφέρουν m2 αντί για m1, και ότι στο τέλος θα καλεί την m1 αντί την m2.

### (β) [5 μονάδες]

Για να μην κολλάει η εκτέλεση του παρακάτω κώδικα κάντε την εξής αλλαγή: αν το όρισμα που λαμβάνει μια μέθοδος (η m1 ή η m2) είναι 9 τότε να εμφανίζεται το μήνυμα “Bingo” και τερματίζεται η εκτέλεση του προγράμματος.

**Παραδοτέα:** Δημιουργήστε ένα φάκελο A12 και μέσα βάλτε τις κλάσεις που υλοποιούν τα παραπάνω.

### Υποδείξεις

- Τερματισμός εκτέλεσης προγράμματος: `System.exit(0);`
- Δημιουργία και εκτύπωση ενός τυχαίου αριθμού μεταξύ 0.1 και 1.0:  
`double random = new Random().nextDouble();`  
`System.out.println(random);`

### Άσκηση 3 – [20 μονάδες] Πίνακες

Για αυτήν την άσκηση χρησιμοποιείτε ένα IDE (Eclipse/NetBeans). Σας δίνεται (στο moodle) ένα jar (με όνομα **jfugue-4.0.3.jar**) το οποίο πρέπει να προσθέσετε στη βιβλιοθήκη του project σας.

Στην άσκηση αυτή καλείστε να φτιάξετε μια κλάση με όνομα **ArrayManager** τέτοια ώστε:

1. Η κλάση πρέπει να έχει μια μέθοδο  
**void fillRandom(int min, int max, int[] ar)**  
η οποία να γεμίζει τον πίνακα ar (οσοδήποτε μεγάλος και αν είναι αυτός) με τυχαίους ακραίους από το min έως το max.  
  
Υποδειξη: import java.util.Random;
2. Η κλάση πρέπει να έχει μια μέθοδο  
**void fillFrom(int[] ar, int[] fromArr)**  
η οποία να γεμίζει τα στοιχεία του πίνακα ar αντιγράφοντας τα στοιχεία που έχει ο πίνακας fromArr. Αν οι δύο πίνακες έχουν διαφορετικό μέγεθος να εκτυπώνεται μήνυμα λάθους και να μην γίνεται τίποτα.
3. Η κλάση πρέπει να έχει μια μέθοδο  
**void playMusic(int[] ar)**  
η οποία θα λαμβάνει ως παράμετρο έναν πίνακα ακεραίων και θα παίζει ως μουσική κάθε στοιχείο του πίνακα (δείτε υποδείξεις).  
  
Υποδειξη: import org.jfugue.Player;
4. Η μέθοδος **main** ενδεικτικά θα πρέπει να δημιουργεί δύο πίνακες ακεραίων p1 και p2 έκαστος χωρητικότητας 20 στοιχείων. Εν συνεχεία να χρησιμοποιεί τις μεθόδους της κλάσης που φτιάξατε ώστε να γεμίσει τον p1 με τυχαίους ακραίους από το 50 έως το 120, εν συνεχεία να γεμίζει τον p2 με τα στοιχεία του p1, και στο τέλος να μουσικοποιεί τα στοιχεία του πίνακα p2.

**Παραδοτέα:** Δημιουργήστε ένα φάκελο A13 όπου μέσα θα υπάρχει το αρχείο **ArrayManager.java**, όπου θα περιέχονται η κλάση και οι μέθοδοι των ερωτημάτων 1,2 και 3 καθώς και μια **Main** με τα ζητούμενα του ερωτήματος 4.

#### Υποδείξεις

Για να παίξετε τη 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] ");
```

## Άσκηση 4 – [30 μονάδες] *Text File to Web page*

### (α) [20 μονάδες]

Φτιάξτε ένα πρόγραμμα το οποίο θα μετατρέπει αρχεία κειμένου σε σελίδες διαδικτύου. Συγκεκριμένα, το πρόγραμμα πρέπει να λαμβάνει ως είσοδο το όνομα ενός αρχείου κειμένου και θα δημιουργεί ένα HTML αρχείο (με το ίδιο όνομα και κατάληξη .html) βάση του παρακάτω προτύπου (template):

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title>
      <!-- Εδώ πρέπει να τυπωθεί το όνομα του αρχείου. -->
    </title>
  </head>
  <body>
    <!-- Εδώ πρέπει να τυπωθεί το κείμενο του αρχείου. -->
  </body>
</html>
```

Το πρόγραμμα πρέπει να εφαρμόζει τις παρακάτω μετατροπές:

- Για κάθε αλλαγή γραμμής στο αρχείο κειμένου (χαρακτήρας '\n' ), τυπώστε το αλφαριθμητικό `<br />` στο HTML αρχείο.
- Για κάθε κενό χαρακτήρα ή χαρακτήρα διαστήματος (tab) στο αρχείο κειμένου, τυπώστε το αλφαριθμητικό `&nbsp;` στο HTML αρχείο.
- Για κάθε εμφάνιση του χαρακτήρα '>', τυπώστε το αλφαριθμητικό `&gt;` στο HTML αρχείο.
- Για κάθε εμφάνιση του χαρακτήρα '<', τυπώστε το αλφαριθμητικό `&lt;` στο HTML αρχείο.
- Για κάθε εμφάνιση του χαρακτήρα '&', τυπώστε το αλφαριθμητικό `&amp;` στο HTML αρχείο.

Επίσης, το πρόγραμμά σας στο τέλος πρέπει να τυπώνει στη κονσόλα πόσο χρόνο πήρε η μετατροπή.

### (β) [5 μονάδες]

Επεκτείνετε το πρόγραμμά σας έτσι ώστε τα ελληνικά να μετατρέπονται σε Greeklish (την αντιστοιχία κάθε ελληνικού χαρακτήρα σε αγγλικά μπορείτε να την κάνετε όπως εσείς θέλετε).

### (γ) [5 μονάδες]

Επεκτείνετε το πρόγραμμά σας έτσι ώστε να μπορεί να λαμβάνει ως είσοδο και τη διεύθυνση ενός αρχείου κειμένου στο διαδίκτυο (δηλαδή ένα URL, πχ: <http://users.ics.forth.gr/~fafalios/Lysistrata.txt>). Για ανάγνωση του περιεχομένου μιας σελίδας διαδικτύου δείτε τις υποδείξεις.

**Παραδοτέα:** Δημιουργήστε ένα φάκελο A14 και μέσα βάλτε τη κλάση (ή τις κλάσεις) που υλοποιεί όλα τα ερωτήματα της άσκησης.

## Υποδείξεις

- Για να μετράτε τον χρόνο σε nano seconds χρησιμοποιείστε:

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

- Ο παρακάτω κώδικας διαβάζει και εκτυπώνει το περιεχόμενο ενός URL, δηλαδή μιας σελίδας στο διαδίκτυο:

```
String path = "http://www.csd.uoc.gr/";
URL url = new URL(path);
BufferedReader in = new BufferedReader(
    new InputStreamReader(url.openStream(), "UTF-8"));

int c = in.read();
while (c != -1) {
    System.out.print((char) c);
    c = in.read();
}

in.close();
```

*(πρέπει να έχετε κάνει τα παρακάτω imports)*

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.URL;
```

## **ΟΔΗΓΙΕΣ ΠΑΡΑΔΟΣΗΣ**

Φτιάξτε ένα φάκελο με όνομα A1\_<<ΑριθμόςΜητρώου>>. Μέσα του δημιουργήστε έναν υποφάκελο για κάθε άσκηση (A11, A12, ...). Συμπιεστέ το φάκελο με όνομα A1\_<<ΑριθμόςΜητρώου>> (**ΠΡΟΣΟΧΗ:** όλες ασκήσεις παραδοθούν με διαφορετικό όνομα δεν θα γίνονται δεκτές), και παραδώστε τον μέσω του moodle.

**ΠΡΟΣΟΧΗ:** Πρέπει να παραδώσετε τα .java αρχεία, δηλαδή τα αρχεία που περιέχουν τον κώδικά σας, και όχι τα εκτελέσιμα .class αρχεία!

*Καλή εργασία!*



## ΠΡΟΑΙΡΕΤΙΚΗ ΑΣΚΗΣΗ ΓΙΑ ΔΙΚΗ ΣΑΣ ΕΞΑΣΚΗΣΗ BONUS 10 ΜΟΝΑΔΕΣ

### Άσκηση 5 – [10 μονάδες] Κρυπτογράφηση

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

#### (α) [5 μονάδες]

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 περιεχόμενου 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.java 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`, `.doc`, `.exe`, `.zip`
2. Εάν υπάρχει πρόβλημα εάν ο αριθμός `K` είναι πολύ μεγάλος;
3. Δοκιμάστε να κρυπτογραφήσετε και πολύ μεγάλα αρχεία και σχολιάστε το χρόνο που απαιτήθηκε.

#### (β) [2 μονάδες]

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. Είστε ελεύθεροι να κάνετε όποιο άλλο έλεγχο κρίνεται σκόπιμο. Στη περίπτωση που η μέθοδος επιστρέψει false, η κονσόλα πρέπει να τυπώνει ένα αντίστοιχο μήνυμα.

### (γ) [3 μονάδες]

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. Αναπαραγοντοποιείστε (refactor) ανάλογα την προηγούμενη μέθοδο (του σκέλους (α)) ώστε να μην υπάρχουν στον κώδικα της κλάσης σας επαναλήψεις (κάτι το οποίο είναι πολύ κακή προγραμματιστική πρακτική), αλλά η προηγούμενη μέθοδος να καλεί σε κάποιο σημείο τη νέα. 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).

**Παραδοτέα:** Δημιουργήστε ένα φάκελο A15 και μέσα δημιουργήστε τους εξής υποφακέλους:

A: Έκδοση κώδικα για σκέλος (α) και γραπτή αναφορά

B: Έκδοση κώδικα για το σκέλος (β)

Γ: Έκδοση κώδικα για το σκέλος (γ)

### Προαιρετικά (extra bonus +5 μονάδες)

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

### Υποδείξεις

Δείτε τις διαφάνειες του σχετικού φροντιστηρίου και την τεκμηρίωση (Javadoc) των σχετικών κλάσεων.

- 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 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 a number ", ""));`
- 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, δείτε και την σχετική κλάση (`GraphicalFileLoaderSaver.java`) που έχει ανέβει στο moodle.