



Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών
HY252 – Αντικειμενοστρεφής Προγραμματισμός
Εξάμηνο: Χειμερινό 2017-2018
Διδάσκων: Γιάννης Τζίτζικας
Βοηθοί: Λ. Δημητράκης, Γ. Καντηλιεράκης

1^η Σειρά Ασκήσεων

Ανάθεση: 4 Οκτωβρίου 2017

Παράδοση: 24 Οκτωβρίου 2017

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 (20 μονάδες) θα σας επιτρέψει να εξοικειωθείτε με τη κλήση στατικών μεθόδων, και να χρησιμοποιείτε εξωτερικό κώδικα σε μορφή βιβλιοθήκης *jar* και κώδικα που σας δίνεται.

Η 3^η άσκηση (20 μονάδες) θα σας επιτρέψει να εξοικειωθείτε με τους πίνακες στην Java (ένας πίνακας είναι αντικείμενο)

Η 4^η άσκηση (20 μονάδες) θα σας επιτρέψει να εξοικειωθείτε με τη διαχείριση αρχείων και με τη χρήση κώδικα που σας δίνεται.

Η 5^η άσκηση (bonus 10 μονάδες) θα σας επιτρέψει να εξοικειωθείτε με τη 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 τις κάνετε compile και να τις τρέξετε από **γραμμή εντολών**, χωρίς δηλαδή τη χρήση κάποιου **IDE** (έτσι θα γίνει και η εξέταση). Τις επόμενες τρεις (2^η, 3^η και 4^η) συστήνεται να τις κάνετε χρησιμοποιώντας ένα IDE (Eclipse ή NetBeans ή IntelliJ). 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**. **ΠΡΟΣΟΧΗ:** (α) Πρέπει να παραδώσετε τα **.java** αρχεία, δηλαδή τα αρχεία που περιέχουν τον κώδικά σας, και όχι τα εκτελέσιμα **.class** αρχεία! (β) Ακολουθήστε επακριβώς τις οδηγίες παράδοσης που αναγράφονται στο τέλος της εκφώνησης. Διαφορετικά η διόρθωση είναι πολύ χρονοβόρα για τους μεταπτυχιακούς οι οποίοι έχουν και αυτοί μαθήματα και εργασία να κάνουν. Αν παραδώστε κάτι που δεν είναι σύμφωνο με τους κανόνες παράδοσης τότε θα έχετε 40% μείωση βαθμού και ίσως δεν βαθμολογηθεί.
5. Η **εξέταση** της Α1 θα γίνει μέσω μίας υποχρεωτικής εργαστηριακής άσκησης (**A1E**). 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] Ζωγραφίζοντας το γράμμα Z

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

(α) [10 μονάδες]

Γράψτε μία κλάση DrawZ (σε αρχείο DrawZ.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 γράμμα του λατινικού αλφαβήτου Z 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 DrawZ 4 μπορεί να εκτυπώνεται το:

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

ενώ με java DrawZ 5 μπορεί να εκτυπώνεται το:

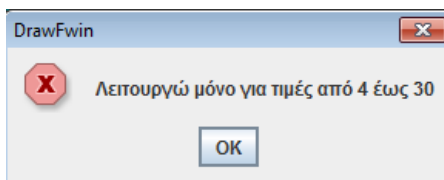
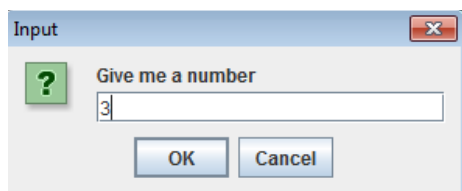
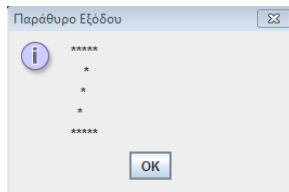
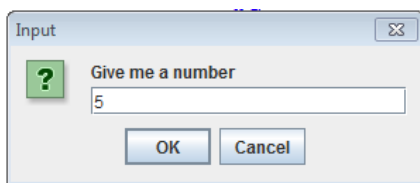
```
*****
 *
 *
 *
*****
```

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

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

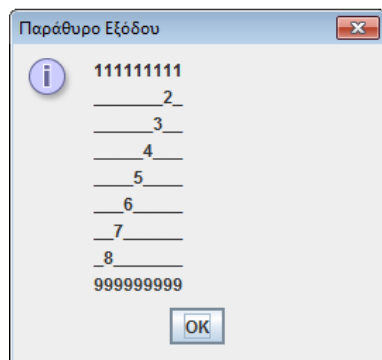
(β) [10 μονάδες]

Δημιουργήστε τώρα μια άλλη κλάση DrawZwin (σε αρχείο DrawZwin.java), αντιγράφοντας την προηγούμενη. Η νέα κλάση πρέπει να λειτουργεί όπως αυτή του σκέλους (α) με τη μόνη διαφορά ότι θα διαβάζει την παράμετρο L από ένα παράθυρο διαλόγου και θα εκτυπώνει το Z σε ένα παράθυρο διαλόγου όπως φαίνεται στο παρακάτω παράδειγμα. Για τα παράθυρα διαλόγου δείτε τις υποδείξεις παρακάτω. 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. Επίσης το πρόγραμμα σας πρέπει να εμφανίζει παράθυρο λάθους όταν η τιμή που δίνει ο χρήστης δεν είναι στο σωστό εύρος.



(γ) [10 μονάδες]

Κάντε μια παραλλαγή (drawZwin2) ώστε η έξοδος αντί για αστεράκια να έχει αριθμούς. Συγκεκριμένα κάθε αστεράκι μιας γραμμής j να έχει αντικατασταθεί με τον αριθμό j, και αντί για κενά να υπάρχουν οι χαρακτήρες «_». Π.χ. για είσοδο ίση με το 9 το παράθυρο εξόδου πρέπει να έχει την εξής μορφή:



(δ) [10 μονάδες]

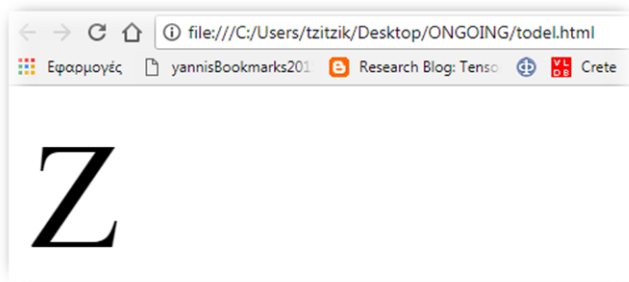
Δημιουργήστε τώρα μια άλλη κλάση (drawZHtml) της οποίας η κλήση να δημιουργεί ένα html αρχείο με όνομα Z.html όπως το ακόλουθο. Στο παράδειγμα που ακολουθεί η τιμή του L που έδωσε ο χρήστης ήταν το 16.

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

<font size="16">Z</font>

</body>
</html>
```

Το αρχείο αυτό θα μπορείτε να το ανοίξετε με ένα ιστοπλοηγτή.



Παραδοτέα: Δείτε τις οδηγίες παράδοσης στο τέλος της εκφώνησης.

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.

Άσκηση 2 [20 μονάδες] Κλήσεις Στατικών Μεθόδων, Χρήση εξωτερικής βιβλιοθήκης

(α) [7 μονάδες] Σε ένα αρχείο **PingPong.java** ορίστε μια κλάση **Ping** (αντίστοιχα και μια κλάση **Pong**) η οποία να έχει μία στατική μέθοδο **ping** (αντίστοιχα η **Pong** να έχει μια στατική μέθοδο **pong**) η οποία να λαμβάνει ως παράμετρο έναν ακέραιο αριθμό **k**.

Η κλήση της **ping(k)** να τυπώνει στην κονσόλα τη συμβολοσειρά "**Ping**" ακολουθούμενη από τιμές **k**, **k+3**, **k+7** και στη συνέχεια αν το **k** είναι μικρότερο του **100** να καλεί την **pong(k+2)**.

Η κλήση της **pong(k)** να τυπώνει στην κονσόλα τη συμβολοσειρά "**Pong**" ακολουθούμενη από τις τιμές **k**, **k+4**, **k+7** και στη συνέχεια αν το **k** είναι μικρότερο του **100** να καλεί την **ping(k+2)**.

Τέλος, η κλάση **Ping** (αντίστοιχα και η κλάση **Pong**) να έχει μια **main** μέθοδο η οποία θα κάνει την κλήση **ping(40)** (αντίστοιχα **pong(40)**). Η έξοδος αν τρέξουμε την **main** της **Ping** να είναι της μορφής:

```
Ping 40 43 47
Pong 42 46 49
Ping 44 47 51
Pong 46 50 53
Ping 48 51 55
Pong 50 54 57
```

...

(β) [7 μονάδες]

Εμπλουτίστε το αρχείο **PingPong.java** με μια κλάση **Utilities** η οποία να έχει μια στατική μέθοδο **void sound(String s)** με τις εξής εντολές:

```
Player p = new Player();
p.play(s);
```

Η κλάση **Player** ορίζεται σε ένα **jar** με όνομα **jfugue-4.0.3.jar** που σας δίνεται στο moodle. Για το λόγο αυτό πρέπει να το προσθέσετε στη βιβλιοθήκη του project σας (για αυτό χρησιμοποιείστε ένα IDE όπως το Eclipse ή NetBeans για αυτή την άσκηση).

Κατόπιν εμπλουτίστε την **Ping** (αντίστοιχα την **Pong**) έτσι ώστε η **ping(k)** (αντίστοιχα η **pong(k)**) αμέσως μετά την εκτύπωση στην κονσόλα, και πριν την κλήση της **pong** (ή **ping**), να φτιάχνει μια συμβολοσειρά **s** που να περιέχει τους αριθμούς που εκτύπωσε στην κονσόλα εγκλεισμένους όμως σε τετραγωνισμένες παρενθέσεις και με ένα κενό μεταξύ τους, π.χ. αν εκτυπώθηκαν οι αριθμοί 50 54 57, τότε η συμβολοσειρά **s** πρέπει να είναι «[50] [54] [57]» και κατόπιν να καλεί την **sound(s)**. Βεβαιωθείτε ότι ακούτε τους ήχους που παράγονται.

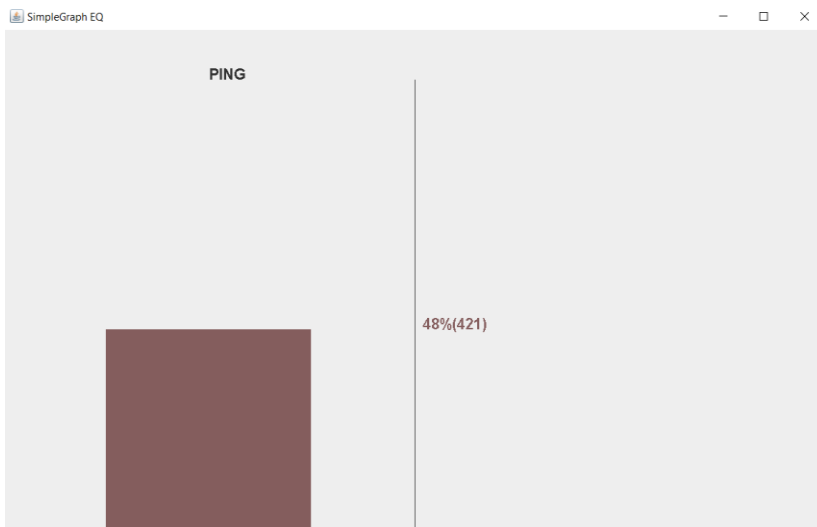
(γ) [6 μονάδες]

Εμπλουτίστε την κλάση **Utilities** με ένα πεδίο **static SimpleEqualizer se = new SimpleEqualizer();** και την εξής μέθοδο



```
public static void setBar(String text, int heightPer);
```

Η κλάση `SimpleEqualizer` σας δίνεται και περιέχεται στο αρχείο **PingPong.java**. Κατόπιν εμπλουτίστε την **Ping** (αντίστοιχα την **Pong**) έτσι ώστε η **ping(k)** (αντίστοιχα η **pong(k)**) αμέσως μετά την κλήση της **sound(s)**, και πριν την κλήση της **pong** (ή **ring**), να καλεί την **Utilities.setBar** με πρώτο όρισμα «ring» (ή «pong» αντίστοιχα) και το επόμενο όρισμα να είναι το **k**. Αυτό θα έχει ως αποτέλεσμα στο γραφικό που θα εμφανιστεί να δημιουργούνται (εναλλάξ) δύο μπάρες (Ping και Pong) οι οποίες θα έχουν κάθε φορά **k%** ύψος (όταν δηλαδή **k=100** θα έχουν το μέγιστο ύψος τους).



Παραδοτέα: Δείτε τις οδηγίες παράδοσης στο τέλος της εκφώνησης.

Υποδείξεις

- Για το (β) σκέλος εκτός της προσθήκης της βιβλιοθήκης στο project θα χρειαστεί στο .java να κάνετε:
 - `import org.jfugue.Player;`
- 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 3 – [20 μονάδες] Πίνακες

Ένα μαγικό τετράγωνο τάξης n είναι μία τετράγωνη συστοιχία $n \times n$ διακριτών ακέραιων αριθμών στην οποία το άθροισμα των n αριθμών σε κάθε **στήλη**, **σειρά** και **διαγώνιο** είναι το ίδιο. Η «μαγεία» βρίσκεται στο γεγονός πως οι αριθμοί σε κάθε στήλη, σειρά και διαγώνιο αθροίζονται στον ίδιο αριθμό, ο οποίος ονομάζεται μαγικό στοιχείο. Παρακάτω παρουσιάζεται ένα 3ης τάξης μαγικό τετράγωνο με μαγικό στοιχείο τον αριθμό 15.

2	7	6	→15
9	5	1	→15
4	3	8	→15
↙15	↓15	↓15	↓15
			↘15

Γράψτε μια κλάση **MagicSquareChecker**. Η main να ζητάει από το χρήστη το n (ο οποίος **πρέπει** να είναι **ακέραιος** από 2 έως 10) και κατόπιν να δίνει επαναληπτικά τα $n \times n$ στοιχεία του πίνακα τα οποία πρέπει να τα αποθηκεύεται σε ένα διδιάστατο πίνακα ακεραίων. Κατόπιν να ελέγχει εάν πρόκειται για «μαγικό τετράγωνο» και αν ναι να εκτυπώνει το αντίστοιχο μήνυμα και το μαγικό στοιχείο. Συγκεκριμένα η κλάση που θα φτιάξετε πρέπει να προσφέρει την εξής **στατική** μέθοδο

```
public static boolean checkIsMagic(int[][] s)
```

η οποία με τη σειρά της να αξιοποιεί τις ακόλουθες **στατικές** μεθόδους:

```
private static int sumOfRow (int[][] s, int k)
private static int sumOfColumn(int[][] s, int k)
private static int sumOfDiagonal1 (int[][] s)
private static int sumOfDiagonal2 (int[][] s)
```

Επιπλέον βοηθητικές **στατικές** μέθοδοι:

```
public static int getMagicNumber(int[][] s)
private static boolean hasDuplicates(int[][] s)
```

Σημείωση: Το **k** που εμφανίζεται σαν όρισμα στις 2 συναρτήσεις (sumOfRow και sumOfColumn) αναπαριστά τον δείκτη της γραμμής και της στήλης αντίστοιχα. Για να είναι ένα τετράγωνο μαγικό **δεν** πρέπει να εμφανίζεται ένας ακέραιος πάνω από μια φορά. Για τέτοιου είδους ελέγχους βολεύει η έννοια του συνόλου (set) και υλοποιήσεις υπάρχουν στις βιβλιοθήκες της java τις οποίες θα δούμε. Εσείς έχετε τις εξής επιλογές για έλεγχο διπλοτύπων: Για κάθε νέο αριθμό που δέχεστε ως είσοδο μπορείτε είτε (α) να κάνετε σειριακή αναζήτηση στον πίνακα για να βεβαιωθείτε ότι ο αριθμός δεν έχει ήδη δοθεί, ή (β) να τον βάζετε στην κλάση MySet που σας δίνεται (η οποία είναι μια απλή υλοποίηση συνόλου με πίνακα) η οποία προσφέρει έλεγχο διπλοτύπων.

Άσκηση 4 – [20 μονάδες] Επιλογή, Διάβασμα, Γράψιμο Αρχείων

Σκοπός αυτής της άσκησης είναι να γράψετε ένα πρόγραμμα το οποίο να μπορεί να διαβάσει ένα τετράγωνο (πίνακα ακεραίων) από ένα αρχείο και να ελέγχει αν είναι μαγικό τετράγωνο, χρησιμοποιώντας και αξιοποιώντας αυτά που έχετε κάνει στην Άσκηση 3. Επίσης θα πρέπει να εμφανίζει τα κατάλληλα μηνύματα και να μπορεί επίσης να γράψει ένα αρχείο.

Συγκεκριμένα:

(α) Το πρόγραμμα πρέπει να ζητάει από το χρήστη να διαλέξει ένα αρχείο (.txt) μέσω παραθύρων διαλόγου. 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.saveFile()`).

(β) Το πρόγραμμά σας πρέπει κατόπιν να ανοίγει το αρχείο, να διαβάζει τους ακεραίους και να τους βάζει σε έναν πίνακα (όπως στην Άσκηση 3). Αν ένα αρχείο έχει προβληματική δομή (π.χ. η πρώτη γραμμή του αρχείου έχει 4 ακεραίους ενώ η δεύτερη γραμμή 3) τότε να βγάζει μήνυμα ότι η μορφοποίηση δεν είναι η κατάλληλη. Αν τα περιεχόμενα του αρχείου διαβάστηκαν επιτυχώς και γέμισε ο πίνακας τότε να γίνεται ο έλεγχος μαγικού τετραγώνου. Εάν είναι, τότε να εμφανίζεται αντίστοιχο μήνυμα και ο μαγικός αριθμός. Αν δεν είναι μαγικό τετράγωνο να εμφανίζεται αντίστοιχο μήνυμα.

(γ) Αν το τετράγωνο ήταν μαγικό να δίνεται η δυνατότητα στο χρήστη να επιλέξει ένα φάκελο (μέσω παραθύρων διαλόγου) όπου να γραφτεί ένα αρχείο το οποίο θα έχει τα περιεχόμενα του πρώτου αρχείου κατόπιν μια κενή γραμμή και μετά μια γραμμή της μορφής «Μαγικό τετράγωνο με μαγικό αριθμό X» όπου X ο μαγικός αριθμός. **Διευκρίνιση:** Ο χρήστης θα πρέπει να διαλέξει μόνο τον φάκελο που θα γραφτεί το αρχείο. Το αρχείο θα πρέπει να δημιουργηθεί μέσω κώδικα και το όνομα που θα του ανατεθεί δεν έχει σημασία π.χ. `MagicSquareSavedFile.txt`

(δ) Επεκτείνεται το πρόγραμμά σας ώστε αντί για αρχείο εισόδου ο χρήστης να έχει τη δυνατότητα να δώσει ένα URL (Universal Resource Locator) και το πρόγραμμά σας να κατεβάζει το αντίστοιχο αρχείο από τον Παγκόσμιο Ιστό. Ενδεικτικά αρχεία για τον έλεγχο των μεθόδων σας μπορείτε να βρείτε στο παρακάτω link: https://www.csd.uoc.gr/~dimitrakis/hy252_A13_testFiles/testX (Όπου X = 1, 2 ή 3) Σας δίνεται ο παρακάτω κώδικας που διαβάζει και εκτυπώνει το περιεχόμενο ενός URL, δηλαδή μιας σελίδας στο διαδίκτυο:




```

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.URL;
import javax.swing.JOptionPane;

public class Downloader {

    static int download(String address) {
        try {
            URL url = new URL(address);
            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();
        } catch (Exception e) {
            System.out.println(e);
            return -1;
        }
        return 0;
    }

    public static void main(String[] a) {
        String toDownload =
            JOptionPane.showInputDialog("Δώστε την διεύθυνση ", "");
        System.out.println(toDownload);

        download(toDownload);
    }
}

```

Προφανώς για να δουλέψει πρέπει να είστε συνδεδεμένοι στο διαδίκτυο.

ΠΡΟΑΙΡΕΤΙΚΗ ΑΣΚΗΣΗ ΓΙΑ ΔΙΚΗ ΣΑΣ ΕΞΑΣΚΗΣΗ

[BONUS 5 ΜΟΝΑΔΕΣ]

Άσκηση 5 – [5 μονάδες] Κρυπτογράφηση

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

(α) [2 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`) τη διεργασία που θέλουμε να εφαρμοστεί (`Encrypt` ή `Decrypt` αντίστοιχα) 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,
String operation)
```

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 Encrypt 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 Decrypt -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).

Στη συνέχεια καλείστε να επεκτείνεται το πρόγραμμά σας, έτσι ώστε κατά τη διαδικασία της κρυπτογράφησης, να δίνεται η δυνατότητα στο χρήστη να δώσει απάντηση σε μία default (fixed size) ερώτηση ασφαλείας με σκοπό να χρησιμοποιηθεί ως ένα επιπλέον επίπεδο ασφαλείας στη διαδικασία της αποκρυπτογράφησης. Με αυτή την επέκταση λοιπόν, το περιεχόμενο του νέου αρχείου θα πρέπει να είναι: η ερώτηση ασφαλείας συνοδευόμενη με την απάντηση (αν έχει οριστεί από τον χρήστη) και στη συνέχεια το περιεχόμενο του αρχείου εισόδου κρυπτογραφημένα.

Σχόλια σχετικά με την ερώτηση-απάντηση ασφαλείας: Πρέπει να ορίσετε μια default ερώτηση η οποία να είναι και συγκεκριμένου μεγέθους π.χ 25 χαρακτήρες. Επίσης πρέπει να επιβάλλεται μέγιστο μέγεθος απάντησης π.χ. 10 χαρακτήρες. Με αυτόν τον τρόπο γνωρίζουμε ότι η ερώτηση μαζί με την απάντηση είναι 35 χαρακτήρες (πληροφορία που θα μας χρειαστεί στη διαδικασία της αποκρυπτογράφησης)

Επιπλέον, στη διαδικασία της αποκρυπτογράφησης βάσει του K, θα πρέπει πλέον να κάνετε και τον απαραίτητο έλεγχο εάν οι πρώτοι x χαρακτήρες περιέχουν την default ερώτηση ασφαλείας ή όχι. **(α)** Εάν την περιέχουν, θα παρουσιάζεται η ερώτηση στο χρήστη. Στη συνέχεια αν δώσει τη σωστή απάντηση, η αποκρυπτογράφηση θα συνεχίζεται κανονικά, ενώ στην αντίθετη περίπτωση θα πρέπει να εμφανίζετε μήνυμα σφάλματος (π.χ. Access Denied) και θα τερματίζεται το πρόγραμμα. **(β)** Εάν δεν την περιέχουν, η αποκρυπτογράφηση θα συνεχίζεται χωρίς περαιτέρω ενέργειες από το χρήστη.

Παράδειγμα

Ας υποθέσουμε ότι έχουμε ορίσει την εξής (default) ερώτηση ασφαλείας: "Who is your best friend?" και έστω ότι έχουμε επιβάλλει περιορισμό μεγέθους της απάντησης 10 χαρακτήρες. Άρα, στη διαδικασία της αποκρυπτογράφησης για να εφαρμόσουμε τον έλεγχο ύπαρξης ερώτησης ασφαλείας θα εξάγουμε τους πρώτους x χαρακτήρες (όπου x = πλήθος χαρακτήρων της default ερώτησης + ένα διαχωριστικό + πλήθος χαρακτήρων περιορισμού απάντησης, στην περίπτωσή μας $x = 24 + 1 + 10 =$

35 χαρακτήρες). Στη συνέχεια, πρέπει πρώτα να αποκωδικοποιήσουμε τους x χαρακτήρες βάσει του K και έπειτα να ελέγξουμε αν στους χαρακτήρες αυτούς περιέχεται η default ερώτηση. **ΠΡΟΣΟΧΗ** Στην περίπτωση που ο χρήστης έχει δώσει απάντηση μικρότερου μεγέθους από τον περιορισμό (10) είναι σημαντικό να καλύψουμε τη διαφορά (σε πλήθος) των χαρακτήρων π.χ. με κάποιο ειδικό σύμβολο έτσι ώστε κατά τη διάρκεια της εξαγωγής να μην αφαιρέσουμε χαρακτήρες από το αρχικό περιεχόμενο του κωδικοποιημένου αρχείου.

Περίπτωση A: Ο χρήστης έχει ορίσει απάντηση στην ερώτηση ασφαλείας (Οι πρώτοι x χαρακτήρες περιέχουν την default ερώτηση).

Έστω ότι ο χρήστης έχει δώσει την εξής απάντηση "Java!". Τότε βάσει του παραδείγματος, οι πρώτοι 35 χαρακτήρες μετά την αποκωδικοποίηση θα είχαν την εξής μορφή "Who is your best friend?=Java!#####". Στη συνέχεια το πρόγραμμα, θα παρουσιάσει στο χρήστη την ερώτηση και θα του ζητηθεί να εισάγει την απάντηση.

Περίπτωση B: Ο χρήστης δεν έχει ορίσει απάντηση στην ερώτηση ασφαλείας (Οι πρώτοι x χαρακτήρες δεν περιέχουν την default ερώτηση).

Στην περίπτωση αυτή η αποκρυπτογράφηση θα συνεχίζεται χωρίς περαιτέρω ενέργειες από το χρήστη.

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 Encrypt 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 επιστρέφει false, η κονσόλα πρέπει να τυπώνει ένα αντίστοιχο μήνυμα.

(γ) [1 μονάδες]

Εμπλουτίστε την κλάση που φτιάξατε και με την εξής μέθοδο:

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

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).

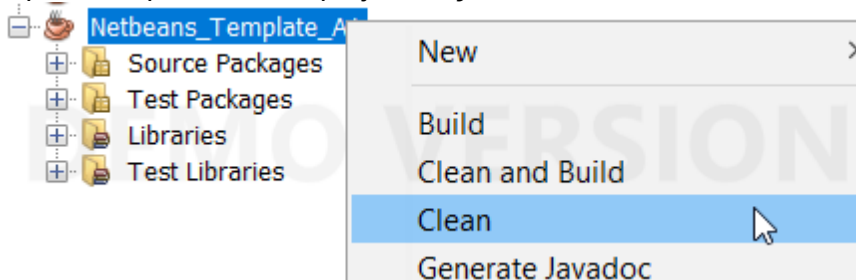
ΟΔΗΓΙΕΣ ΠΑΡΑΔΟΣΗΣ

Βήμα 1

Ανάλογα με το IDE που έχετε, χρησιμοποιήστε ένα από τα τρία project templates που σας παρέχουμε (Netbeans_Template_A1, Eclipse_Template_A1 και IntelliJ_Template_A1). Τα templates αυτά περιέχουν όλα τα απαραίτητα πακέτα (packages) καθώς και τις υπογραφές των κλάσεων που καλείστε να υλοποιήσετε. **ΜΗΝ** αλλάξετε τη δομή αυτή, απλά συμπληρώστε τις μεθόδους των κλάσεων που σας ζητούνται.

Βήμα 2

Προτού παραδώσετε το project σας, κάνετε ένα clean



Βήμα 3

Το όνομα του project σας θα πρέπει να ακολουθεί την εξής μορφή: <IDE_name>_A1_<ΑριθμόςΜητρώου>. Π.χ. στην περίπτωση που χρησιμοποιείτε Netbeans θα είχαμε Netbeans_A1_1111

Βήμα 4

Συμπίεστε το project σας και παραδώστε το μέσω του moodle.

Καλή εργασία!

Τέλος A1