



Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών

HY252 – Αντικειμενοστρεφής Προγραμματισμός

Εξάμηνο: Χειμερινό 2013-2014

Διδάσκων: Γιάννης Τζίτζικας

Βοηθοί: Γιάννης Καργάκης, Βαγγέλης Λαδάκης, Παύλος Φαφαλιός

1^η Σειρά Ασκήσεων

Ανάθεση: 17 Οκτωβρίου 2013

Παράδοση: 4 Νοεμβρίου 2013

Οι εκπαιδευτικοί στόχοι αυτής της σειράς ασκήσεων είναι: βασικοί τύποι, ροή ελέγχου, πίνακες, αναδρομή, είσοδος και έξοδος από κονσόλα, αρχεία, απλές κλάσεις, χρήση κλάσεων από τη βιβλιοθήκη της Java.

Άσκηση 1 (20%) Παλινδρομικές Φράσεις

Input/Output, Πίνακες

Καλείστε να γράψετε πρόγραμμα το οποίο θα αποφασίζει αν ένα αλφαριθμητικό είναι *παλινδρομικό* δηλαδή αν διαβάζοντας το ανάποδα παίρνουμε το ίδιο αλφαριθμητικό Παραδείγματα παλινδρομικών αναγνωριστικών ακολουθούν: «*abcba*», «*A man, a plan, a canal: Panama*», 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. Για να το επιτύχετε μετατρέψτε το αλφαριθμητικό που δίνεται σαν είσοδος από το χρήστη σε δύο πίνακες από χαρακτήρες. Ο πρώτος θα περιεχί 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 θα πρέπει να αγνοούνται.

Υποδείξεις:

- Για να διαβάσετε την είσοδο stdin :

```
Scanner in = new Scanner(System.in);  
String nextLine = in.next();  
int aNumber = in.nextInt();
```

- Για να μετράτε τον χρόνο σε nano seconds χρησιμοποιήστε :

```
long time = System.nanoTime();
```

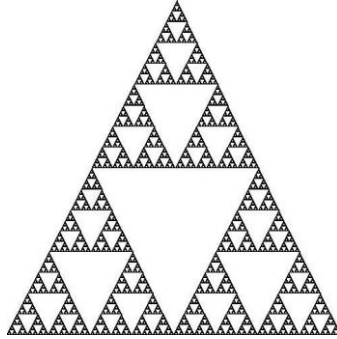
- Για τον έλεγχο των κεφαλαίων γραμμάτων και των ειδικών χαρακτήρων, χρήσιμες μεθόδους θα βρείτε στο API της κλάσης String

(<http://docs.oracle.com/javase/6/docs/api/java/lang/String.html>)

Άσκηση 2 (25%) Fractals

Classes, Arrays, Strings, Recursion, Files

Καλείστε να γράψετε ένα αναδρομικό πρόγραμμα Java το οποίο θα φτιάχνει ένα fractal (μορφόκλασμα) με τρίγωνα όπως το εξής:



α) Δημιουργία Στοιχειώδους Κλάσης (5%)

Στο πρώτο βήμα θα πρέπει να υλοποιήσετε μια κλάση "*Pixel*" η οποία θα πρέπει να υλοποιεί τις παρακάτω συναρτήσεις.

```
/*
 * Pixel constructor
 * @x,y pixel coordinates
 * @color the color of the pixel
 */
public Pixel(int x, int y, Color color);

/*
 * modifiers
 */
public void setX(int x);
public void setY(int y);
public void setColor(Color color);

/*
 * accessors
 */
public int getX();
public int getY();
public Color getColor();
```

(you will have to import java.awt.Color;)

Δηλαδή να έχει απλά τα απαραίτητα ιδιωτικά μέλη τα οποία διαχειρίζονται οι παραπάνω μέθοδοι.

β) Χρήση Κλάσης που μας δίδεται, Strings, Αρχεία (10%)

Για το δεύτερο βήμα σας δίδεται η κλάση "*Picture*" την οποία θα χρησιμοποιήσετε για να εμφανίσετε τις εικόνες που θα δημιουργεί το πρόγραμμά σας. Ένα αντικείμενο τύπου "*Picture*" δημιουργεί ένα

καινούργιο γραφικό παράθυρο, και οι μέθοδοι της κλάσης αυτής που θα χρειαστεί να χρησιμοποιήσετε είναι οι ακόλουθες:

- `Picture(int w, int h)` //make a new Canvas w pixels width and h pixels height
- `set(Pixel pixel)` // draw a pixel into the Canvas. Uses the class you defined in (a)
- `show()` //show the result on the window
- `setOriginLowerLeft()` //the 0,0 pixel is on the bottom left
- `setOriginUpperLeft()` //the 0,0 pixel is on the top left

Σας δίδεται επίσης ένα αρχείο με ονομασία `fractal.txt` του οποίου κάθε γραμμή κρατάει πληροφορία για ένα pixel στην εξής μορφή:

(cordinates) (color of the pixel as triples of Red Green Blue values)
x y Red Green Blue

όπου όλες οι τιμές είναι ακέραιες. Τώρα δημιουργήστε μία δική σας κλάση με όνομα **ReadImageFromTxtFile** της οποίας η main θα διαβάζει το αρχείο και θα αναπαριστά (ζωγραφίζει) το περιεχόμενό του χρησιμοποιώντας την κλάση "*Pixel*" του ερωτήματος (α) και την κλάση "*Picture*".

Προσοχή: για να αναπαρασταθεί σωστά το περιεχόμενο του `fractal.txt` θα πρέπει να έχετε θέσει σωστά το σύστημα συντεταγμένων με τη `setOriginUpperLeft()`. Επίσης ο Canvas πρέπει να είναι $N \times N$ όπου $N=512$ pixels.



γ) Πίνακες, αντικείμενα, αναδρομικές συναρτήσεις (10%)

Τώρα δημιουργήστε μία δική σας κλάση με όνομα **RecursiveTriangle**. Στη main αυτής της κλάσης ορίστε ένα αντικείμενο τύπου *Picture*. Επίσης ορίστε έναν διδιάστατο πίνακα από αντικείμενα τύπου "*Pixel*" με τα οποία θα αναπαραστήσετε αυτό που θα ζωγραφίσετε (θέτοντας το χρώμα τους). Για να ορίσετε το χρώμα των pixels της ζωγραφιάς σας θα χρειαστεί να φτιάξετε μία αναδρομική (recursive) μέθοδο που θα υλοποιεί τον ακόλουθο αναδρομικό αλγόριθμο.

Η ιδέα είναι απλή. Αρχικά έχετε 3 σημεία (p_1, p_2, p_3) που ορίζουν τις κορυφές ενός τριγώνου (του μεγαλύτερου στη ζωγραφιά μας). Μπορούμε μέσα του να ζωγραφίσουμε άλλα τρία μικρότερα τρίγωνα με τον εξής απλό τρόπο: για κάθε ζευγάρι κορυφών βρίσκουμε το μέσο τους, ήτοι υπολογίζουμε τις συντεταγμένες των σημείων $(p_1+p_2)/2$, $(p_2+p_3)/2$ και $(p_1+p_3)/2$. Με αυτά τα σημεία και τις κορυφές του τριγώνου μπορείτε να ζωγραφίσετε 3 νέα μικρότερα τρίγωνα (δείτε το παραπάνω σχήμα). Συνεχίζετε (αναδρομικά) την ίδια διαδικασία για τα 3 μικρά τρίγωνα, κ.ο.κ, έως ότου τα τρίγωνα προς σχεδίαση έχουν γίνει πολύ μικρά (π.χ. οι 3 κορυφές έχουν τις ίδιες συντεταγμένες).

Προσοχή: Ο canvas εξαρχής (by default) είναι μαύρος.

Υποδείξεις: `Integer.parseInt(String str);` (static function)
`File` (class)

FileReader (class)
BufferedReader (class)

Άσκηση 3 – (55%) Text Indexing for Information Retrieval

String, File, I/O, Arrays (55%)

Στόχος της άσκησης είναι να φτιάξετε ένα απλό Σύστημα Ανάκτησης Πληροφοριών (Information Retrieval System) από αρχεία κειμένου.

α) (15%) Ο Ευρετηριαστής

Γράψτε ένα πρόγραμμα Java το οποίο θα ζητάει από το χρήστη να δώσει από τη γραμμή εντολών το όνομα ενός αρχείου εισόδου και το όνομα ενός αρχείου εξόδου. Στη συνέχεια αφού διαβάσει το αρχείο εισόδου το πρόγραμμα θα πρέπει να δημιουργεί στο αρχείο εξόδου ένα «ευρετήριο λέξεων». Ένα «ευρετήριο λέξεων» περιέχει όλες τις διαφορετικές λέξεις του αρχείου εισόδου (κάθε λέξη σε διαφορετική γραμμή) και για κάθε λέξη εμφανίζει τις θέσεις στις οποίες βρέθηκε στη μορφή «γραμμή;θέση». Οι λέξεις στο ευρετήριο πρέπει να είναι σε πεζά γράμματα (lowercase) και να μη λαμβάνουν υπόψη ειδικούς χαρακτήρες στην αρχή ή στο τέλος της κάθε λέξης.

Για παράδειγμα, για το αρχείο εισόδου:

```
my name is George,  
your name is Nick  
and we love Java.
```

Θα πρέπει να δημιουργείτε στο αρχείο εξόδου το παρακάτω ευρετήριο:

```
my 1;1  
name 1;4,2;6  
is 1;9,2;11  
george 1;12    (προσέξτε, όχι "george,")  
your 2;1  
nick 2;14  
and 3;1  
we 3;5  
love 3;8  
java 3;13    (προσέξτε, όχι "java.")
```

β) (10%) Χρήση Ευρετηρίου

Γράψτε ένα πρόγραμμα Java το οποίο θα ζητάει από το χρήστη να δώσει από την γραμμή εντολών τη διεύθυνση ενός αρχείου κειμένου, τη διεύθυνση του αρχείου που έχει το ευρετήριο λέξεων, ένα String αναζήτησης και ένα ακέραιο N.

Κάνοντας χρήση του ευρετηρίου θα επιστρέφει ένα ή περισσότερα αποσπάσματα μήκους το πολύ N που περιέχουν αυτήν την λέξη (αν φυσικά υπάρχει η λέξη στο ευρετήριο). Ένα απόσπασμα (snippet)

είναι ένα κομμάτι του κειμένου το οποίο περιέχει τη λέξη που δόθηκε. Για παράδειγμα, στη περίπτωση που ο χρήστης δώσει τη λέξη «Nick», ένα απόσπασμα (με μήκος 24 χαρακτήρες) είναι το παρακάτω:

```
«name is Nick and we love»
```

Θεωρήστε ότι το String αναζήτησης που δίνει ο χρήστης πρέπει να περιέχει μία μόνο λέξη.

Τέλος, στη περίπτωση που η λέξη δε βρεθεί, το πρόγραμμα πρέπει να εμφανίζει το παρακάτω κείμενο:

```
«No results for the given query!»
```

γ) (10%) Γενίκευση του Ευρετηριαστή

Επεκτείνετε τα ερώτημα α) ώστε το πρόγραμμα να ζητάει από το χρήστη να δώσει από τη γραμμή εντολών το όνομα ενός φακέλου (αντί ενός αρχείου εισόδου). Στη συνέχεια αφού διαβάσει όλα τα αρχεία κειμένου που περιέχονται στον φάκελο, το πρόγραμμα θα δημιουργεί στο αρχείο εξόδου ένα «ευρετήριο λέξεων» που όμως πλέον θα περιέχει για κάθε λέξη και τα αρχεία στα οποία βρέθηκε στη μορφή «αρχείο;γραμμή;θέση» όπως παρακάτω:

```
and example1.txt;3;1
george example2.txt;1;12
is example1.txt;1;9,example2.txt;2;11
```

δ) (10%) Χρήση Ευρετηρίου

Επεκτείνετε το ερώτημα β) ώστε να επιστρέφει μαζί με τα αποσπάσματα και τα αρχεία στα οποία βρέθηκε η λέξη, όπως παρακάτω:

```
example1.txt
«name is Nick and we love»
```

```
example2.txt
«called Nick and ask to»
```

Το πρόγραμμα θα ζητάει από το χρήστη να δώσει από την γραμμή εντολών τη διεύθυνση ενός ευρετηρίου λέξεων (όπως αυτό που παράγεται από το ερώτημα γ), ένα αντικείμενο τύπου String και ένα ακέραιο N (για το μέγιστο μήκος του snippet). Θεωρήστε πάλι ότι το String αναζήτησης που δίνει ο χρήστης πρέπει να περιέχει μία μόνο λέξη.

ε) (10%) Κατέβασμα και Ευρετηρίαση κειμένων από το διαδίκτυο

Επεκτείνετε τα ερώτημα α) ώστε το πρόγραμμα να ζητάει από τον χρήστη να δώσει από την γραμμή εντολών το URL μιας ιστοσελίδας (αντί ένα αρχείο κειμένου). Το πρόγραμμα θα διαβάζει το περιεχόμενο της σελίδας (το οποίο συνήθως είναι σε HTML μορφή) και θα δημιουργεί στο αρχείο εξόδου ένα «ευρετήριο λέξεων» για τη συγκεκριμένη σελίδα.

Το παρακάτω τμήμα Java κώδικα τυπώνει το περιεχόμενο μιας ιστοσελίδας:

```
String urlStr = "http://en.wikipedia.org/wiki/Yellowfin_tuna";
URL url = new URL(urlStr);
```

```

URLConnection con = url.openConnection();
BufferedReader in = new BufferedReader(
    new InputStreamReader(con.getInputStream(), "UTF-8"));

String pageContents = "";
String input;
while ((input = in.readLine()) != null) {
    pageContents = pageContents + input + "\n";
}
in.close();
System.out.println(pageContents);

```

Πριν την ευρετηρίαση, προσπαθήστε να αφαιρέσετε HTML κώδικα που δε χρησιμεύει σε μία αναζήτηση (δηλαδή κείμενο που περιέχεται σε "<" και ">"). Πχ για το κείμενο "my name is George", θέλουμε να ευρετηριάσουμε μόνο τις λέξεις του κειμένου "my name is George", δηλαδή όχι τις λέξεις font, color, blue, κτλ.

Υποδείξεις:

- Μπορείτε να κάνετε την υλοποίηση χρησιμοποιώντας πίνακες με max μέγεθος N, π.χ. N = 100000.
- Μπορείτε να θεωρήσετε 2 πίνακες, ένα για την αποθήκευση των λέξεων και ένα για την αποθήκευση των θέσεων που βρέθηκε η κάθε λέξη (πχ αν η λέξη «fish» υπάρχει στη θέση 5 στον πρώτο πίνακα, μπορούμε να βρούμε τις εμφανίσεις της στη θέση 5 του δεύτερου πίνακα). Για την αποθήκευση των θέσεων μιας λέξης μπορείτε να χρησιμοποιήσετε String, πχ String positions = "1;9,2;11"; το οποίο θα χτίζετε σταδιακά (δηλαδή αρχικά θα έχει τιμή "1:9", μετά θα γίνει "1;9,2;11", κτλ.)
- Την εμφάνιση του snippet μπορείτε να την υλοποιήσετε όπως θέλετε. Ο μόνος περιορισμός είναι το μέγιστο μέγεθος που πρέπει να έχει.
- Μπορείτε να μετατρέψετε ένα String σε int με την παρακάτω εντολή:
int n = Integer.parseInt(someString);
- Μπορείτε να διαχωρίσετε ένα String με βάση ένα χαρακτήρα με την παρακάτω εντολή:
String[] words = line.split(delimiterString);
Για παράδειγμα, για το String "my name is George" και καλώντας την split(" ") θα επιστραφεί ο πίνακας: ["my", "name", "is", "george"]. Αντίστοιχα, για το String "1;9,2;1" καλώντας την split(";") θα επιστραφεί ο πίνακας: ["1;9", "2;1"].
- Μπορείτε να αφαιρέσετε όλους τους ειδικούς χαρακτήρες από την αρχή και το τέλος ενός String με την παρακάτω εντολή:
String wordNoSpecialChars = word.replaceAll("(^[^a-zA-z0-9]*|[^a-zA-Z0-9]*\$)", "");
Για παράδειγμα, το String "George." θα γίνει "George"
- Μπορείτε να μετατρέψετε ένα String σε μικρούς χαρακτήρες με την παρακάτω εντολή:
String wordLowerCase = word.toLowerCase();
- Μπορείτε να αφαιρέσετε κενά διαστήματα, πχ κενό ή \t (tab), από την αρχή και το τέλος ενός String με την παρακάτω εντολή:
String wordTrim = word.trim();
- Μπορείτε να βρείτε τη θέση ενός String μέσα σε ένα άλλο String με την παρακάτω εντολή:
int wordPosition = someString.indexOf(someOtherString);
Για παράδειγμα, για το someString = "my name is George and your name is Nick" και someOtherString = "name" θα επιστρέψει 3,
- Μπορείτε επίσης να βρείτε τη θέση ενός String μέσα σε ένα άλλο String αρχίζοντας όμως από μία θέση και μετά με την παρακάτω εντολή:
int wordPosition = someString.indexOf(someOtherString, startFrom);

Για παράδειγμα, για το someString = "my name is George and your name is Nick", someOtherString = "name" και startFrom=4, θα επιστρέψει 27

- Μπορείτε να ελέγξετε αν δύο Strings είναι ίδια με την εντολή
someString.equals(someOtherString)
η οποία επιστρέφει true αν τα 2 string είναι ακριβώς ίδια.
- Μπορείτε να βρείτε όλες τις συναρτήσεις που υποστηρίζει η κλάση String εδώ:
<http://docs.oracle.com/javase/6/docs/api/java/lang/String.html>
- Το παρακάτω τμήμα Java κώδικα τυπώνει όλα τα αρχεία ενός φακέλου:
File folderDir = new File(folderPath); // e.g. folderPath = "C:/myfolder"
File[] listOfFiles = folderDir.listFiles();
for (int i = 0; i < listOfFiles.length; i++) {
 if (listOfFiles[i].isFile()) {
 System.out.println(listOfFiles[i]);
 }
}

ΟΔΗΓΙΕΣ ΠΑΡΑΔΟΣΗΣ

Φτιάξτε ένα φάκελο με όνομα A1_<<ΑριθμόςΜητρώου>>. Μέσα του δημιουργήστε έναν υποφάκελο για κάθε άσκηση (A1.1, A1.2, A1.3). Συμπίεστε το φάκελο με όνομα A1_<<ΑριθμόςΜητρώου>>(Προσοχή_όσες ασκήσεις παραδοθούν με διαφορετικό όνομα δεν θα γίνονται δεκτές), και παραδώστε τον μέσω του moodle

Απορίες σχετικά με την Α1 θα απαντώνται μόνο μέσω του forum της Α1 στο moodle.

Καλή εργασία