

HY-252 Αντικειμενοστραφής Προγραμματισμός

Χειμερινό Εξάμηνο 2012
Διδάσκων: Χριστοφίδης Βασίλης

1^η Σειρά Ασκήσεων

Ημερομηνία Παράδοσης: 19/10/2012

Άσκηση 1 – Loops, IO (20%)

Όνομα αρχείου : Assi gn1_1 o gi n. j ava

Γράψτε ένα πρόγραμμα που παίρνει ως παράμετρο κατά την εκτέλεσή του έναν ακέραιο $n > 1$ και τυπώνει ένα καράβι με τα παρακάτω χαρακτηριστικά

- Μήκος και ύψος πανιού n
- Κατάρτι ύψους $\lfloor n/2 \rfloor$
- Κατάστρωμα μήκους $2 \times n + 1$ και ύψους $\lfloor n/2 \rfloor + 1$.

Παράδειγμα εκτέλεσης:

java PrintStarShip 4

```
n = 4
  *
  **
 ***
****
  *
  *
*****
*****
  ****
  } Πανί
  } Κατάρτι
  } Κατάστρωμα
```

Βοήθεια (Hints):

- Μπορείτε να μετατρέψετε ένα `String` σε `int` με την παρακάτω εντολή:
`int n = Integer.parseInt(someString);`
- Βεβαιωθείτε ότι η παράμετρος που δίνεται από τον χρήστη είναι ορθή.

Άσκηση 2 – String, File, I/O (25%)

Όνομα αρχείου : Assi gn2_1 o gi n. j ava

α) Γράψτε μία μέθοδο `maxOcc(String str)` η οποία παίρνει ως παράμετρο ένα αντικείμενο τύπου `String` και επιστρέφει τον αριθμό του μέγιστου “run” μέσα σε αυτό το string. Ως “run” ορίζεται μια σειρά από 0 ή περισσότερους χαρακτήρες οι οποίοι είναι συνεχόμενοι και ίδιοι μεταξύ τους. Έτσι το μέγιστο “run” του αλφαριθμητικού “abbbcc” είναι 3 (τρεις συνεχόμενες εμφανίσεις του χαρακτήρα ‘b’). Το μέγιστο “run” του αλφαριθμητικού “abc” 1.

Παραδείγματα:

```
maxOcc("abc") → 1
maxOcc("") → 0
maxOcc("avvvvddd") → 4
```

β) Γράψτε ένα πρόγραμμα Java το οποίο θα ζητάει από τον χρήστη να δώσει από την 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 εκτυπώνει στο αρχείο εξόδου τις λέξεις του αρχείου εισόδου μαζί με τον αριθμό του μέγιστου "run" τους χρησιμοποιώντας τη μέθοδο `maxOcc()` του ερωτήματος α).

Παράδειγμα:

```
Enter the input filename:
input.txt
Enter the output filename:
output.txt
```

Παράδειγμα αρχείου εισόδου (input.txt)

```
aaab bbc abc
fffddr f
ererww
```

Έξοδος του παραδείγματος (output.txt)

```
aaab:      3
bbc:       2
abc:       1
fffddr:    4
f:         1
ererww:    2
```

Βοήθεια (Hints):

- Θα σας φανούν χρήσιμες οι μέθοδοι: `charAt` και `length` από την κλάση `String`. Για περισσότερες λεπτομέρειες πάνω στην κλάση `String` ανατρέξτε στο Java API <http://docs.oracle.com/javase/6/docs/api/java/lang/String.html>.
- Αν έχετε ένα `String` από το οποίο θέλετε να πάρετε τα κομμάτια του αγνοώντας τα κενά (δηλαδή από ένα string "cat dog elephant" να πάρετε "cat", "dog", "elephant"), μπορείτε χρησιμοποιήσετε τη μέθοδο `split` του `String`.

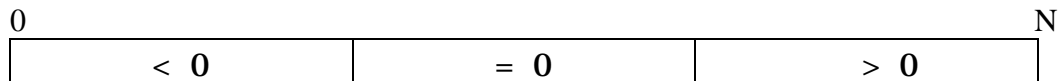
```
String s = "cat dog elephant";
String[] parts = s.split(" "); //υπάρχει ακριβώς ένα
space στην παράμετρο
//parts[0] είναι "cat"
//parts[1] είναι "dog"
//parts[2] είναι "elephant"
```

Αν θέλετε να χωρίσετε το `String` με κάποιο άλλο διαχωριστικό (delimiter), τότε κάνετε το ίδιο, πχ: "cat###dog###elephant".split("###") → Δίνει ["cat", "dog", "elephant"]

Άσκηση 3 – Arrays (25%)

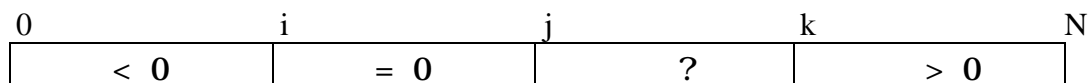
Όνομα αρχείου : Assi gn3_1 o g i n. j a v a

Γράψτε μία μέθοδο `rearrangeArray(int[] b)` η οποία δέχεται σαν είσοδο έναν πίνακα N ακεραίων και αναδιοργανώνει τα στοιχεία του με τέτοιο τρόπο, ώστε όλοι οι αρνητικοί αριθμοί να προηγούνται των μηδενικών, και όλοι οι θετικοί αριθμοί να έπονται αυτών. Στην παρακάτω εικόνα μπορείτε να δείτε σχηματικά την μορφή που θα πρέπει να έχει ο πίνακας b .



Βοήθεια (Hints):

Προκειμένου να πετύχετε το επιθυμητό αποτέλεσμα θα πρέπει να αντιμετωπίσετε τα στοιχεία του πίνακα με τέτοιο τρόπο ώστε σε οποιοδήποτε ενδιάμεσο στάδιο πριν ολοκληρωθεί η διαδικασία τα στοιχεία του πίνακα να έχουν την μορφή:



Πιο συγκεκριμένα τα στοιχεία στο $b[0 \dots i-1]$ είναι αρνητικά, στο $b[i \dots j-1]$ μηδέν, στο $b[j \dots k-1]$ δεν έχουν ακόμα προσπελασθεί και στο $b[k \dots N-1]$ θετικά. Για την αντιμετάθεση των τιμών δύο στοιχείων του πίνακα πρέπει να υλοποιήσετε μία βοηθητική μέθοδο `swap()`.

Άσκηση 4 – Recursion (30%)

Όνομα αρχείου : Assi gn4_1 o g i n. j a v a

Υλοποιήστε ένα πρόγραμμα το οποίο δέχεται σαν είσοδο ένα αλφαριθμητικό και κάνοντας χρήση αναδρομής θα τυπώνει όλες τις πιθανές μεταθέσεις των χαρακτήρων του αλφαριθμητικού. Για παράδειγμα δεδομένης της εισόδου “tag” το πρόγραμμα εμφανίζει τις λέξεις tag, tga, gta, gat, atg, agt. Κάθε χαρακτήρας του αλφαριθμητικού θα πρέπει να αντιμετωπίζεται σαν ξεχωριστός χαρακτήρας ακόμα και αν επαναλαμβάνεται, δηλ για είσοδο “bbb” το πρόγραμμα θα εμφανίζει 6 φορές “bbb” (οι πιθανές διαφορετικές μεταθέσεις k αντικειμένων είναι $k!$).

Βοήθεια (Hints):

- Θα σας φανούν χρήσιμες οι μέθοδοι: `charAt`, `length` από την κλάση `String`.
- Για να υπολογίσετε τις μεταθέσεις μιας συμβολοσειράς, ‘abcd’ κρατήστε κάθε φορά σταθερό έναν χαρακτήρα, αρχίζοντας από αριστερά και προχωρώντας προς τα δεξιά, και υπολογίστε όλες τις μεταθέσεις που αρχίζουν με τον χαρακτήρα αυτόν. Για παράδειγμα αρχικά κρατάμε σταθερό τον χαρακτήρα ‘a’ και υπολογίζουμε αναδρομικά τις μεταθέσεις που αρχίζουν με ‘a’: abcd, abdc, acbd, acdb, adbc, adcb. Στη συνέχεια κρατάμε σταθερό τον χαρακτήρα ‘b’ και υπολογίζουμε αναδρομικά τις μεταθέσεις που αρχίζουν με ‘b’: bacd, badc, bcad, bcda, bdac, bdca κοκ.
- Μη αναδρομικές λύσεις θα θεωρούνται λάθος.

Οδηγίες

Όλες οι ασκήσεις σας θα πρέπει να μπορούν να γίνουν compile και να τρέξουν από command line, με τον τρόπο που έχετε διδαχθεί στις διαλέξεις του μαθήματος.

Το login στην ονομασία των ασκήσεων πρέπει να αντικατασταθεί από το login σας.
Για παράδειγμα εάν το login σας ήταν christop η άσκηση 4 θα έπρεπε να ονομάζεται Assign4_christop.java