Θ.Ε. ΠΛΣ50 (2022-23) – ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ Ε1

Ημερομηνία ανάρτησης	**10.10.2022**
Ημερομηνία αποστολής	**Βάσει χρονοδιαγράμματος, μέχρι την Τετάρτη, 16 Νοεμβρίου 2022, 11:59 μμ.** (προσοχή, το σύστημα υποβολής θα κλείσει αυτόματα μόλις παρέλθει η παραπάνω χρονική στιγμή, σύμφωνα με το ρολόι του συστήματος, που μπορεί να διαφέρει ελαφρά από το δικό σας)
Ανακοίνωση ενδεικτικής επίλυσης	**23.11.2022**

Θεματολογία-στόχος

Στην εργασία αυτή θα εξασκηθείτε σε μερικές από τις απαραίτητες γνώσεις της γλώσσας προγραμματισμού Java, που θα χρησιμοποιήσουμε καθ' όλη τη χρονιά σε πολλές από τις εργασίες της ΘΕ. Ειδικότερα, θα ασχοληθείτε με τύπους και μεταβλητές, τελεστές, εκφράσεις, είσοδο από το πληκτρολόγιο και έξοδο στην οθόνη, εντολές ελέγχου ροής και ανακύκλωσης, μονοδιάστατους πίνακες, ορισμό κλάσεων, μεθόδους και πέρασμα παραμέτρων, κατασκευαστές και αντικείμενα.

Παρατηρήσεις

Περιμένουμε όλες οι εργασίες να υποβληθούν στο study.eap.gr. Ο πηγαίος κώδικας Java (**ένα ή περισσότερα αρχεία .java για κάθε θέμα ή υποερώτημα**) που θα συνοδεύει την εργασία θα πρέπει να βρίσκεται σε ξεχωριστούς υποκαταλόγους (ένας υποκατάλογος για κάθε θέμα ή υποερώτημα). **Μην συμπεριλάβετε αρχεία .class, ή άλλα αρχεία που τυχόν δημιουργούνται από το Ολοκληρωμένο Περιβάλλον Ανάπτυξης (IDE) που χρησιμοποιείτε.** Συνιστάται να συμπεριλάβετε επεξηγήσεις για τον τρόπο που λύσατε την εργασία, μαζί με μερικές εικόνες (screenshots) από την εκτέλεση των προγραμμάτων σας, **σε συνοδευτικό αρχείο κειμένου μορφής *.doc ή *.odt** (αρχεία *.pdf γίνονται δεκτά μόνο όταν συνοδεύονται από το αντίστοιχο doc/odt). Στο study.eap.gr, σε κάθε περίπτωση, ανεβάζετε **ένα μόνο συμπιεσμένο αρχείο** (*.zip ή *.rar) που θα περιέχει όλους τους επιμέρους υποκαταλόγους και αρχεία.

Συγκεκριμένα, το συμπιεσμένο αρχείο της 1ης Εργασίας θα πρέπει να περιέχει τους υποκαταλόγους Thema1, Thema2, Thema3 και Thema4, που θα περιέχουν τα αρχεία Java (και όχι άλλα αρχεία) των Θεμάτων 1, 2, 3 και 4, αντίστοιχα, και προαιρετικά το συνοδευτικό αρχείο explain.doc ή explain.odt εκτός καταλόγων.

Μπορείτε να χρησιμοποιήσετε το Ολοκληρωμένο Περιβάλλον Ανάπτυξης (IDE) της αρεσκείας σας, π.χ. BlueJ, Eclipse, Netbeans. Εκτός από ενδεχομένως άλλο IDE, συνιστάται να εγκαταστήσετε και το BlueJ και να δοκιμάσετε ότι ο κώδικάς σας λειτουργεί σωστά και σε αυτό.

Ανεξάρτητα από το αν θα υποβάλετε συνοδευτικό αρχείο κειμένου, ο κώδικας σας πρέπει να περιλαμβάνει επεξηγηματικά και ταυτόχρονα περιεκτικά, κατά την κρίση σας, σχόλια.

Εισαγωγή

Δε χρειάζεται να κάνετε τίποτε περισσότερο από όσα σας ζητούνται. Συνιστάται, πριν ασχοληθείτε με την εργασία, να μελετήσετε τα παραδείγματα, τις ασκήσεις αυτοαξιολόγησης και τις δραστηριότητες από τα κεφάλαια του Τόμου Β', Γλώσσες Προγραμματισμού, ΕΑΠ, Πάτρα 2015, που αντιστοιχούν στις εβδομάδες μελέτης μέχρι την ημερομηνία αποστολής της εργασίας βάσει χρονοδιαγράμματος. Στο τέλος της Εργασίας περιλαμβάνεται **ενδεικτικός** πίνακας με τα βασικά γνωστικά αντικείμενα της γλώσσας προγραμματισμού Java που πρέπει να έχετε διαβάσει για να μπορείτε να αντιμετωπίσετε κάθε θέμα. Τονίζεται ότι ο πίνακας έχει σκοπό να σας βοηθήσει και όχι να σας υποχρεώσει στον ακριβή τρόπο υλοποίησης κάθε θέματος. Παρόμοια, τυχόν υποδείξεις που δίνονται δεν είναι υποχρεωτικές.

Τα μηνύματα που τυπώνουν τα προγράμματά σας στην οθόνη συνιστάται να είναι γραμμένα με λατινικούς χαρακτήρες (σε greeklish ή αγγλικά), προς αποφυγή προβλημάτων που σχετίζονται με την κωδικοποίηση των ελληνικών στο λειτουργικό σύστημα του υπολογιστή σας.

Θέμα 1: Απόσταση Χάμινγκ (Hamming)

Η απόσταση Χάμινγκ[1] (Hamming distance) μεταξύ δύο αλφαριθμητικών (ακολουθίες χαρακτήρων) με ίδιο μήκος είναι το πλήθος των θέσεων στις οποίες τα αντίστοιχα σύμβολα είναι διαφορετικά. Για παράδειγμα, η απόσταση Χάμινγκ ανάμεσα στα αλφαριθμητικά “10111101” και “10010001” είναι 2, αφού διαφέρουν στην τρίτη και στην πέμπτη θέση, και μόνον εκεί, ενώ η απόσταση ανάμεσα στα αλφαριθμητικά “cats” και “dogs” είναι 3.

Να γράψετε ένα πρόγραμμα Java που θα περιέχει μόνο μια κλάση με όνομα Thema1. Η κλάση θα περιέχει μόνο μια μέθοδο, την public static void main(). Το πρόγραμμα θα διαβάζει από το πληκτρολόγιο δύο αλφαριθμητικά ίδιου μήκους και θα υπολογίζει την απόσταση Χάμινγκ ανάμεσά τους. Ειδικότερα, το πρόγραμμα:

- Ζητά από το χρήστη (με κατάλληλο μήνυμα) να εισάγει το πρώτο αλφαριθμητικό, το διαβάζει και το αποθηκεύει στη μεταβλητή target.
- Ζητά από το χρήστη (με κατάλληλο μήνυμα) να εισάγει το δεύτερο αλφαριθμητικό, το διαβάζει και το αποθηκεύει στη μεταβλητή other, ελέγχοντας αν έχει το ίδιο μήκος με το πρώτο. Εάν δεν έχει το ίδιο μήκος, εμφανίζει κατάλληλο μήνυμα λάθους και επαναλαμβάνει την ανάγνωση του δεύτερου αλφαριθμητικού από τον χρήστη, μέχρις ότου το δεύτερο αλφαριθμητικό να έχει ίδιο μήκος με το πρώτο.
- Με μια επαναληπτική διαδικασία διατρέχει τα δύο αλφαριθμητικά και μετρά το πλήθος των αντίστοιχων θέσεων στις οποίες διαφέρουν (Υπόδειξη: μπορείτε να χρησιμοποιήσετε τη μέθοδο charAt της κλάσης String για να συγκρίνετε τους αντίστοιχους χαρακτήρες).
- Εμφανίζει το αποτέλεσμα στο χρήστη.

Η έξοδος του προγράμματός σας στην οθόνη θα μοιάζει σύμφωνα με το ακόλουθο δείγμα εκτέλεσης:

Δείγμα εκτέλεσης:

```
Enter target string: qwerty
Enter second string (length = 6):qerty
Error in length
Enter second string (length = 6):qserty
The Hamming distance of qwerty and qserty is 1
```

Θέμα 2: Αλφαριθμητικά και Λέξεις

Για τους σκοπούς αυτού του θέματος, θεωρούμε ότι ένα αλφαριθμητικό είναι **λέξη** όταν δεν περιέχει κανένα χαρακτήρα διαστήματος (space character). Να γράψετε πρόγραμμα Java που θα αποτελείται από μία κλάση με όνομα Thema2 και θα περιέχει μόνο μια μέθοδο, την public static void main(). Το πρόγραμμά σας θα διαβάζει από το πληκτρολόγιο αλφαριθμητικά που θα εισάγει ο χρήστης, θα ελέγχει αν καθένα από αυτά είναι λέξη ή όχι και στο τέλος θα εμφανίζει στατιστικά στοιχεία:

α. Πόσα αλφαριθμητικά πληκτρολογήθηκαν.

β. Το ποσοστό των λέξεων ως προς το σύνολο των αλφαριθμητικών που δόθηκαν.

γ. Από τις λέξεις, το ποσοστό αυτών που αρχίζουν με κεφαλαίο γράμμα.

Ειδικότερα, το πρόγραμμά σας:

- Θα ρωτά τον χρήστη πόσα αλφαριθμητικά πρόκειται να εισαγάγει και θα διαβάζει την απάντηση του (ακέραιος θετικός αριθμός, δεν θα κάνετε έλεγχο, θεωρούμε ότι ο χρήστης δίνει σωστή είσοδο).
- Με μια επαναληπτική διαδικασία, θα διαβάζει τα αλφαριθμητικά, ένα κάθε φορά, και θα μετρά αυτά που είναι λέξη, σύμφωνα με την παραδοχή που κάναμε: Θα ελέγχει αν το αλφαριθμητικό περιέχει ή όχι τον χαρακτήρα διαστήματος (Υπόδειξη: μπορείτε να χρησιμοποιήσετε τη μέθοδο contains της κλάσης String).

[1] https://el.wikipedia.org/wiki/Απόσταση_Χάμινγκ

- Από τις λέξεις που εντοπίζει, θα μετρά πόσες από αυτές ξεκινούν με πρώτο γράμμα κεφαλαίο (Υπόδειξη: μπορείτε να χρησιμοποιήσετε συνδυαστικά τις μεθόδους charAt της κλάσης String και isUpperCase της κλάσης Character).
- Στο τέλος θα εμφανίζει το πλήθος των αλφαριθμητικών που διαβάστηκαν, το επί τοις εκατό ποσοστό των λέξεων που εντοπίστηκαν σε αυτά τα αλφαριθμητικά και τέλος το επί τοις εκατό ποσοστό των λέξεων που ξεκινούν με κεφαλαίο γράμμα. Θα ελέγχετε τυχόν περίπτωση που προκύπτει διαίρεση με το μηδέν και θα προσαρμόζετε κατάλληλα τα εμφανιζόμενα αποτελέσματα.

Η έξοδος του προγράμματός σας στην οθόνη θα μοιάζει σύμφωνα με τα ακόλουθα δείγματα εκτέλεσης:

Δείγμα εκτέλεσης 1:

```
How many strings?:5
Enter string no 1:Window
Enter string no 2:We are Greeks
Not a word
Enter string no 3:WeAreGreeks!!
Enter string no 4:it's show time
Not a word
Enter string no 5:showtime

Results
Total number of strings: 5
Percentage of words: 60.0%
Words starting with capital letter: 66.666664%
```

Δείγμα εκτέλεσης 2:

```
How many strings?:3
Enter string no 1:this is a test
Not a word
Enter string no 2:another test
Not a word
Enter string no 3:one more test
Not a word

Results
Total number of strings: 3
Percentage of words: 0.0%
Words starting with capital letter: no words found
```

Θέμα 3: Επεξεργασία δεδομένων σε πίνακες

Να γράψετε ένα πρόγραμμα Java το οποίο θα περιλαμβάνει μια μόνο κλάση με όνομα Thema3. Η κλάση θα περιέχει δύο μεθόδους, την public static void main και την public static int hamming.
Η μέθοδος hamming θα δέχεται ως παραμέτρους δύο αλφαριθμητικά και θα επιστρέφει έναν ακέραιο αριθμό που είναι η απόσταση Χάμινγκ των δύο αυτών αλφαριθμητικών. Για παράδειγμα, αν ως ορίσματα δοθούν τα “dog” και “dig” η μέθοδος θα επιστρέφει τον ακέραιο 1, που είναι η απόσταση Χάμινγκ ανάμεσα σε αυτά. Στην περίπτωση που τα ορίσματα δεν έχουν το ίδιο μήκος, η μέθοδος θα επιστρέφει -1.
Η μέθοδος main θα περιλαμβάνει ως τοπικές μεταβλητές έναν πίνακα αλφαριθμητικών με 5 στοιχεία και όνομα stringList και έναν πίνακα ακεραίων με 5 στοιχεία και όνομα distances. Η main θα εκτελεί τις ακόλουθες ενέργειες:

- Θα διαβάζει πέντε αλφαριθμητικά, ένα κάθε φορά, που θα τα πληκτρολογεί ο χρήστης και θα τα αποθηκεύει στον πίνακα stringList.

- Θα διαβάζει ένα επιπλέον αλφαριθμητικό, που θα το αποθηκεύει σε μια τοπική μεταβλητή με όνομα target.
- Σε επαναληπτική διαδικασία, θα υπολογίζει (καλώντας τη μέθοδο hamming) τις αποστάσεις των αλφαριθμητικών του πίνακα stringList από το target και θα τις αποθηκεύει στις αντίστοιχες θέσεις του πίνακα distances (θα είναι είτε ακέραιος >=0 ή -1 αν δεν έχει εφαρμογή η απόσταση λόγω διαφορετικού μήκους).
- Μετά τους υπολογισμούς θα διατρέχει τον πίνακα distances και θα εντοπίζει τη μικρότερη απόσταση που αυτός περιλαμβάνει. Εννοείται ότι δεν θα υπολογίζεται η τιμή -1.
- Θα εμφανίζει (όλα) τα περιεχόμενα του πίνακα distances στην οθόνη, καθώς και το αλφαριθμητικό του πίνακα stringList με τη μικρότερη απόσταση από το target. Αν υπάρχουν περισσότερες από μια περιπτώσεις, θα εμφανίζει μια.

Η έξοδος του προγράμματός σας στην οθόνη θα μοιάζει σύμφωνα με τα ακόλουθα δείγματα εκτέλεσης:

Δείγμα εκτέλεσης 1:

```
1. Enter string: dog
2. Enter string: cat
3. Enter string: jim
4. Enter string: bed
5. Enter string: toe

Enter target: house
Contents of array distances
0 -1
1 -1
2 -1
3 -1
4 -1
No Hamming distance found
```

Δείγμα εκτέλεσης 2:

```
1. Enter string: dog
2. Enter string: cat
3. Enter string: jim
4. Enter string: bed
5. Enter string: blackboard

Enter target: bid
Contents of array distances
0 3
1 3
2 2
3 1
4 -1
String with min Hamming distance: bed
```

Θέμα 4: Περισσότερες από μια κλάσεις

Να ορίσετε μια κλάση με όνομα StringUtils η οποία θα περιλαμβάνει:

- Ένα αλφαριθμητικό πεδίο με όνομα myString.
- Ένα ακέραιο πεδίο με όνομα len που θα κρατάει το μήκος του myString.
- Έναν κατασκευαστή, ο οποίος θα δέχεται ένα αλφαριθμητικό και θα το αποθηκεύει στο myString, ενώ στο len θα αποθηκεύει το μήκος του αλφαριθμητικού.

- Μια μέθοδο με όνομα hamming, η οποία θα δέχεται ένα αλφαριθμητικό με όνομα tmp και θα επιστρέφει την απόσταση Χάμινγκ του tmp από το myString (ή -1 αν δεν ορίζεται, λόγω διαφοράς στο μήκος).
- Μια μέθοδο με όνομα reverseDistance, η οποία θα επιστρέφει την απόσταση Χάμινγκ του myString από το αντίστροφό του, χρησιμοποιώντας την παραπάνω μέθοδο hamming. Για παράδειγμα, το αντίστροφο του “dog” είναι το “god”, δηλαδή οι χαρακτήρες σε αντίστροφη σειρά. Σε αυτό το παράδειγμα, η απόσταση θα είναι 2 (Υπόδειξη: Ένας τρόπος για να δημιουργήσετε το αντίστροφο ενός αλφαριθμητικού είναι να χρησιμοποιήσετε τη μέθοδο toCharArray της κλάσης String για να το μετατρέψετε σε πίνακα από χαρακτήρες και στη συνέχεια να διατρέξετε τον πίνακα αυτό από το τέλος προς την αρχή προσθέτοντας τους χαρακτήρες σε ένα αρχικά κενό String, με τον τελεστή +).

Να ορίσετε μια δεύτερη κλάση με όνομα Thema4, η οποία θα περιλαμβάνει τη μέθοδο main. Η main:

- Θα διαβάζει ένα αλφαριθμητικό από το πληκτρολόγιο (εμφανίζοντας σχετικό μήνυμα).
- Θα δημιουργεί ένα αντικείμενο τύπου StringUtils, καλώντας τον κατασκευαστή με όρισμα το αλφαριθμητικό.
- Θα καλεί τη μέθοδο reverseDistance για να υπολογίσει την απόσταση Χάμινγκ ανάμεσα στο αλφαριθμητικό και το αντίστροφό του και θα εμφανίζει το αποτέλεσμα στην οθόνη.
- Θα ρωτά τον χρήστη αν θέλει να δώσει άλλο αλφαριθμητικό και σε περίπτωση θετικής απάντησης θα επαναλαμβάνει αυτή τη διαδικασία, διαφορετικά θα τερματίζει.

Η έξοδος του προγράμματός σας στην οθόνη θα μοιάζει σύμφωνα με το ακόλουθο δείγμα εκτέλεσης:

Δείγμα εκτέλεσης:

```
Give a string:
ΑΘΗΝΑ
Hamming distance between your string and the reverse is 2
Type S to Stop, other to repeat:

Give a string:
DOG
Hamming distance between your string and the reverse is 2
Type S to Stop, other to repeat:

Give a string:
ANNA
Hamming distance between your string and the reverse is 0
Type S to Stop, other to repeat:
S
```

Σημειώσεις:

Σε όλα τα θέματα δεν χρειάζεται να κάνετε ελέγχους εκτός από εκείνες τις περιπτώσεις που αναφέρεται στην εκφώνηση.

Τα προγράμματα της 1ης εργασίας δε χρειάζεται να κάνουν χειρισμό εξαιρέσεων.

Γνωστικά Αντικείμενα ανά θέμα	Θ1	Θ2	Θ3	Θ4
Τύποι και μεταβλητές, τελεστές, εκφράσεις	X	X	X	X
Είσοδος πληκτρολογίου και έξοδος οθόνης	X	X	X	X
Εντολές ελέγχου ροής	X	X	X	X
Επανάληψη	X	X	X	X
Μονοδιάστατοι πίνακες			X	X

Μέθοδοι και πέρασμα παραμέτρων			X	X
Κατασκευαστές				X
Περισσότερες από μία κλάσεις και αντικείμενα				X

ΚΡΙΤΗΡΙΑ ΑΞΙΟΛΟΓΗΣΗΣ	
Θέμα 1: Απόσταση Χάμινγκ	10
Θέμα 2: Αλφαριθμητικά και Λέξεις	30
Θέμα 3: Επεξεργασία δεδομένων σε πίνακες	30
Θέμα 4: Περισσότερες από μια κλάσεις	30
Εικόνα εργασίας - σχολιασμός	10
ΣΥΝΟΛΟ	**110**
Ο συνολικός βαθμός θα διαιρεθεί δια 10, ώστε να προκύψει ο τελικός βαθμός της εργασίας.	

Καλή Επιτυχία