

**Αντικειμενοστρεφής Προγραμματισμός II**  
**1<sup>ο</sup> ατομικό φυλλάδιο εργαστηριακών ασκήσεων**  
**Ημερομηνία 01/11/2022**

**Οδηγίες για παράδοση Ασκήσεων:**

- Η παράδοση των εργασιών θα γίνει **ηλεκτρονικά** μέσω του **e-class** την **Τετάρτη 23/11/2022** στις **23.59**.
- Θα πρέπει να σταλεί ένα αρχείο zip με όνομα `LoginName_Lab01` (π.χ. `icsd21001_Lab01.zip`). Το αρχείο zip θα περιέχει τα εξής:
  - Project των ασκήσεων (Θα πρέπει να συμπεριλαμβάνονται όλα τα αρχεία του project ανά άσκηση. Στην αρχή του πηγαίου κώδικα θα αναγράφεται ο αριθμός μητρώου, το ονοματεπώνυμο σας και ο αριθμός της άσκησης). **Επιβάλλεται η χρήση σχολίων στον κώδικα σας.**
  - Ένα **αρχείο .pdf** με οθόνες εκτέλεσης (screenshots) των προγραμμάτων σας **που να φαίνεται ξεκάθαρα το αποτέλεσμα εκτέλεσης του κάθε προγράμματος.** Στην αρχή του αρχείου θα αναγράφεται ο αριθμός μητρώου και το ονοματεπώνυμο σας.
- **Καμία εργασία δεν θα διορθωθεί εάν δεν έχει ακριβώς αυτή τη μορφή.**

**1η άσκηση (10%)**

Καλείστε να δημιουργήσετε μια κλάση που διαχειρίζεται έναν πίνακα ακεραίων τιμών (**NumbersTable**). Η κλάση που θα ορίσετε θα πρέπει να υλοποιεί τις ακόλουθες στατικές μεθόδους:

- ***void fillTableRandomly(int [])***  
Γεμίζει τον πίνακα που δέχεται ως παράμετρο με τυχαίους ακέραιους αριθμούς στο διάστημα [1, 50].
- ***void printTableH(int [])***  
Τυπώνει όλα τα στοιχεία του πίνακα σε μια γραμμή.
- ***void printTableV(int [])***  
Τυπώνει όλα τα στοιχεία του πίνακα σε μια στήλη.
- ***void swapValues(int [], int, int)***  
Εναλλάσσει τις τιμές των κελιών του πίνακα που ορίζονται από τις 2 τελευταίες παραμέτρους.
- ***int minLocationFrom(int [], int)***  
Επιστρέφει τη θέση του μικρότερου στοιχείου του πίνακα. Η δεύτερη παράμετρος ορίζει την θέση του πίνακα από όπου θα ξεκινήσει η αναζήτηση του μικρότερου αριθμού.

- ***int maxLocationFrom(int[], int)***  
Επιστρέφει τη θέση του μεγαλύτερου στοιχείου του πίνακα. Η δεύτερη παράμετρος ορίζει την θέση του πίνακα από όπου θα ξεκινήσει η αναζήτηση του μεγαλύτερου αριθμού.
- ***int LocationNumber(int [], int)***  
Επιστρέφει την πρώτη εμφάνιση του αριθμού της δεύτερης παραμέτρου στον πίνακα. Αν δεν υπάρχει ο αριθμός επιστρέφει -1.
- ***int[] cloneTable (int [])***  
Δημιουργία ενός νέου πίνακα, αντιγράφου του πίνακα που δέχεται ως παράμετρο.
- ***int[] mergeTables (int[], int[])***  
Δημιουργία ενός νέου πίνακα, που προκύπτει από την συγχώνευση των 2 πινάκων που δέχεται ως παράμετρο.
- ***String ConvertToString (int [])***  
Δημιουργία ενός αλφαριθμητικού με όλους τους αριθμούς του πίνακα.
- ***String ConvertToString (int [], int, int)***  
Δημιουργία ενός αλφαριθμητικού με όλους τους αριθμούς του πίνακα από την θέση που καθορίζεται από την 1<sup>η</sup> παράμετρο έως την θέση που καθορίζεται στην 2<sup>η</sup> παράμετρο.

Υλοποιήστε ξεχωριστή κλάση που θα περιέχει την main μέθοδο στην οποία θα πρέπει να ορίσετε κατάλληλους πίνακες ακεραίων και να καλέσετε τις στατικές μεθόδους που υλοποιήσατε.

## 2η άσκηση (25%)

Τα electronic sports αποτελούν μία ακόμα νέα τάση της εποχής. Παίζονται από παίκτες οι οποίοι συναγωνίζονται μεταξύ τους σε δημοφιλή παιχνίδια μέσω υπολογιστών. Ένα δημοφιλές παιχνίδι είναι και το «League of Legends II». Στη φάση των ημιτελικών αυτού του παιχνιδιού που πραγματοποιείται στην Ιταλία, διαγωνίζονται 50 ομάδες. Στο τελικό προκρίνονται εκείνες που θα συγκεντρώσουν βαθμολογία μεγαλύτερη από τα 4/5 του μέσου όρου. Να γράψετε κώδικα Java, ο οποίος:

1. Να διαβάζει και να αποθηκεύει σε κατάλληλη δομή το όνομα, τη βαθμολογία, καθώς και την χώρα που εκπροσωπεί καθεμιά από τις 50 ομάδες. Θεωρήστε πως οι βαθμολογίες είναι ακέραιες τιμές στο διάστημα [1, 100] χωρίς να απαιτείται σχετικός έλεγχος.
2. Να υπολογίζει και να εμφανίζει τον μέσο όρο βαθμολογίας όλων των ομάδων.
3. Να εμφανίζει ποια ομάδα είχε βαθμολογία πιο κοντά στον μέσο όρο. Θεωρήστε πως είναι μόνο μία.
4. Να εμφανίζει ταξινομημένα (φθίνουσα ή αύξουσα) τα ονόματα των διαφορετικών χωρών που συμμετείχαν στον τελικό. Θεωρήστε πως υπάρχουν σίγουρα ομάδες που προκρίνονται στον τελικό και ότι μπορούν να υπάρχουν ομάδες από την ίδια χώρα.

5. Να εμφανίζει το πλήθος των ομάδων που προκρίνονται. Μόνο για αυτές τις ομάδες θα εμφανίζει στην ολοκλήρωση του προγράμματος τα στοιχεία τους, δηλαδή θα εμφανίζει ανά γραμμή το όνομα, τη βαθμολογία και την χώρα της κάθε ομάδας χωρισμένα με κενό.

Η εφαρμογή θα πρέπει να υλοποιηθεί εφαρμόζοντας τις βασικές αρχές του αντικειμενοστρεφούς προγραμματισμού (κατάλληλος ορισμός κλάσεων, ενθυλάκωση, συσχέτιση κλάσεων, κλπ.).

### 3η άσκηση (30%)

Υλοποιήστε πρόγραμμα το οποίο θα διαβάζει μια συμβολοσειρά και θα αποθηκεύει τους χαρακτήρες της με κατάλληλο τρόπο. Πιο συγκεκριμένα θα πρέπει να υλοποιήσετε τρεις βασικές λειτουργίες για τη διαχείριση της συμβολοσειράς:

- Η πρώτη θα αποθηκεύει σε κατάλληλη δομή τους χαρακτήρες που τη συνθέτουν καθώς και τη θέση τους σε αυτή. Στους χαρακτήρες συμπεριλαμβάνονται όλοι, όπως ειδικά σύμβολα (., !, κλπ.) εκτός από τα κενά. Έτσι για παράδειγμα αν η συμβολοσειρά είναι η «HELLO SAMOS!» η λειτουργία αυτή θα πρέπει να αποθηκεύει στη δομή τις ακόλουθες πληροφορίες:

[H: {0} E: {1} L: {2, 3} O: {4, 9} S: {6, 10} A: {7} M: {8} !: {11}]

Σημειώνεται ότι δεν είναι απαραίτητο να αποθηκεύονται με αυτή την σειρά στη δομή.

- Η δεύτερη λειτουργία θα διαβάζει τους χαρακτήρες με τις πληροφορίες θέσης, από τη δομή που φυλάσσονται, θα συνθέτει ξανά τη συμβολοσειρά στην αρχική της κατάσταση τοποθετώντας και τα απαραίτητα κενά και ακολούθως θα την τυπώνει.
- Η τρίτη μέθοδος θα εμφανίζει πληροφορίες για τους χαρακτήρες της συμβολοσειράς, δηλαδή ποιος (ποιοι) χαρακτήρες εμφανίζονται με μεγαλύτερη συχνότητα στη συμβολοσειρά και ποιος (ποιοι) με μικρότερη.

Σκόπιμο είναι στην υλοποίησή σας να επιλέξετε τη χρήση των πιο αποδοτικών δομών δεδομένων (Collections) για βέλτιστα αποτελέσματα.

### 4η άσκηση (35%)

Ένας τύπος σταυρολέξου είναι το **Κρυπτόλεξο**. Αυτό αποτελείται από ένα δισδιάστατο πίνακα με γράμματα τοποθετημένα σε τυχαία σειρά. Στον πίνακα υπάρχουν συγκεκριμένες λέξεις σχηματισμένες κάθετα και οριζόντια σε διάφορες θέσεις. Ο παίκτης καλείται να εντοπίσει τις δοσμένες λέξεις.

A. Ορίστε την κλάση κρυπτόλεξο (**WordSearch**) για τη διαχείριση ενός πίνακα **15x25**. Η κλάση θα πρέπει να παρέχει τις ακόλουθες λειτουργίες:

1. Τη μέθοδο **addWord** που προσθέτει μια λέξη σε συγκεκριμένη θέση στον πίνακα (οριζόντια ή κάθετα), εφόσον αυτό είναι δυνατό.

2. Τη μέθοδο **fillBlanks**, η οποία αφού τοποθετηθούν όλες οι λέξεις, γεμίζει τις υπόλοιπες θέσεις του πίνακα με τυχαία γράμματα.
3. Τη μέθοδο **search**, η οποία αναζητά μια συγκεκριμένη (δοσμένη) λέξη στο κρυπτόλεξο, και αν η αναζήτηση είναι επιτυχής, επιστρέφει τη θέση της. Η θέση της ορίζεται από τρεις τιμές. Αν η λέξη είναι οριζόντια από την «Γραμμή», «Στήλη Αρχής» και «Στήλη Τέλους» ενώ αν είναι κάθετα από την «Στήλη», «Γραμμή Αρχής», «Γραμμή\_τέλους».
4. Τη μέθοδο **display**, η οποία θα τυπώνει τον πίνακα του κρυπτόλεξου.

**B.** Ορίστε την κλάση **Word**. Κάθε αντικείμενο **Word** αποτελείται από μια λέξη προς αναζήτηση στο κρυπτόλεξο μαζί με τις οδηγίες τοποθέτησής της στον πίνακα του παιχνιδιού. Για παράδειγμα αν έχουμε τη λέξη **DREAM** και τις τιμές **0, 3, 5**, ορίζουμε ότι η λέξη θα πρέπει να τοποθετηθεί, οριζόντια (0:οριζόντια, 1:κάθετα), στην 3<sup>η</sup> γραμμή (ή στήλη), με τον πρώτο χαρακτήρα στην 5<sup>η</sup> θέση (γραμμή ή στήλη).

**Γ.** Ορίστε την κλάση **PlayGame**.

1. Αρχικοποιεί ένα παιχνίδι **WordSearch**.

a. Παρέχει τη μέθοδο **ReadWords**, η οποία ορίζει μια ArrayList με όνομα **PoolWords** με 50 αντικείμενα τύπου **Word**.

b. Δημιουργεί ένα αντικείμενο **WordSearch** που μπορεί να περιέχει 12 (το μέγιστο) τυχαίες λέξεις από την λίστα **PoolWords**. Το πλήθος των λέξεων δίνεται από το χρήστη. Σε περίπτωση που μια λέξη δεν μπορεί να τοποθετηθεί λόγω του ότι υπάρχει ήδη άλλη λέξη στο κρυπτόλεξο στην ίδια θέση, συνεχίζεται η τοποθέτηση των λέξεων με την επόμενη τυχαία.

2. Παρέχει τη μέθοδο **Play**. Αυτή ρωτάει τον χρήστη για μια λέξη και την αναζητάει στον κρυπτόλεξο. Υποθέστε ότι χρήστης έχει στη διάθεσή του μια λίστα όλων των λέξεων (χωρίς τις συντεταγμένες τους) που υπάρχουν στην αρχική λίστα των πενήντα λέξεων. Το παιχνίδι συνεχίζεται μέχρι ο χρήστης να επιλέξει να σταματήσει.

Μπορείτε να ορίσετε και όποιες άλλες πρόσθετες μεθόδους επιθυμείτε στην υλοποίησή σας.

Η υλοποίηση της εφαρμογής σας θα πρέπει να ακολουθεί τις βασικές αρχές του αντικειμενοστραφούς προγραμματισμού (υψηλή ενθουσίαση, υψηλή συνοχή, κλπ.).