

---

**Εργασία: Ομαδικό Προγραμματιστικό Project**

**Ακαδημαϊκό Έτος : 2021-2022**

**Τίτλος Project:** Ανάπτυξη εφαρμογής κατάταξης υποψηφίων για θέσεις εργασίας με βάση τις δεξιότητές τους

**ID Project:** 49

---

Το υπό ανάπτυξη σύστημα είναι ένα σύστημα καταγραφής θέσεων εργασίας και αντίστοιχα απαιτούμενων δεξιοτήτων για τις θέσεις εργασίας και δεξιοτήτων που έχουν οι υποψήφιοι για μία θέση εργασίας. Το σύστημα θα υποστηρίζεται από μία βάση δεδομένων στην οποία θα αποθηκεύονται πληροφορίες για τις θέσεις εργασίας και τις απαιτούμενες δεξιότητές τους καθώς και για τα προσόντα των υποψηφίων. Το σύστημα θα πρέπει να παρέχει γραφική διασύνδεση χρήστη.

Ένα παράδειγμα θέσης εργασίας με τις απαιτούμενες δεξιότητες είναι το ακόλουθο:

Θέση εργασίας:	Προγραμματιστής Java Enterprise Edition <i>Overqualified = "OXI"</i>	
Δεξιότητα	Απαιτούμενη βαθμολογία	Απαραίτητο;
Γνώση Java Enterprise Edition	8	NAI
Γνώση Hibernate	8	NAI
Γνώση Αντικειμενοστρεφούς Τεχνολογίας Λογισμικού	6	OXI
Γνώση Ανάλυσης Μεγάλου Όγκου Δεδομένων	5	OXI
Γνώση Αγγλικής	4	OXI

Το σύστημα για κάθε θέση εργασίας θα επιτρέπει στον χρήστη να εισάγει τις σχετικές με αυτήν δεξιότητες, την απαιτούμενη βαθμολογία-επίπεδο για κάθε απαιτούμενη δεξιότητα σε μία κλίμακα 1-10, καθώς επίσης και το αν η δεξιότητα αυτή είναι απαραίτητο να καλύπτεται στο απαιτούμενο επίπεδο ή όχι. Επίσης συνολικά για την θέση εργασίας θα δίνεται τιμή "NAI" ή "OXI" στην μεταβλητή "overqualified" η οποία καθορίζει αν θέλουμε ή

όχι άτομα με περισσότερα προσόντα από αυτά που απαιτεί η κάθε δεξιότητα (δηλ. με υψηλότερη βαθμολογία).

Το σύστημα επίσης θα επιτρέπει την καταχώρηση υποψηφίων για μία θέση εργασίας και των βαθμολογιών τους για κάθε δεξιότητα που απαιτείται για τη θέση εργασίας (υποθέστε πως αυτές οι βαθμολογίες είναι επίσης στην κλίμακα 1-10 και εισάγονται με εκτίμηση από κάποια ειδική επιτροπή αξιολόγησης των υποψηφίων με κάποιον τρόπο – π.χ. μέσω εξετάσεων, συνεντεύξεων κ.λπ).

Το σύστημα θα πρέπει να μπορεί να εξαγει την κατάταξη των υποψηφίων για τη θέση με την ακόλουθη διαδικασία:

1. Καταρχήν για κάθε δεξιότητα που έχει επισημανθεί ως απαραίτητη θα αποκλείονται όλοι οι υποψήφιοι που δεν έχουν τουλάχιστον τη βαθμολογία που απαιτείται γι' αυτήν. Στο παράδειγμα που δόθηκε αν κάποιοι υποψήφιοι είχαν στη Java Enterprise Edition ή στο Hibernate βαθμό μικρότερο του 8, αυτοί θα αποκλείονταν σ' αυτό το βήμα.
2. Για όσους υποψήφιους δεν αποκλείστηκαν στο πρώτο βήμα το σύστημα θα κάνει μία προεπεξεργασία ανάλογα με την τιμή της μεταβλητής *overqualified*:
  - a. Αν δεν μας πειράζει κάποιοι υποψήφιοι να έχουν περισσότερα προσόντα από αυτά που απαιτούνται για κάποια θέση (*overqualified* = "ΝΑΙ") τότε θα βρίσκουμε την μεγαλύτερη τιμή υποψηφίου για κάθε δεξιότητα και θα αλλάζει η απαιτούμενη τιμή της δεξιότητας στην μέγιστη τιμή που βρήκαμε. Αν για παράδειγμα η μέγιστη τιμή δεξιότητας στη Java Enterprise Edition είναι ίση με 10, τότε θα αλλάξει η απαιτούμενη τιμή γι' αυτή τη δεξιότητα και θα γίνει ίση με 10.
  - b. Αν προτιμούμε για κάποια θέση οι υποψήφιοι να έχουν όσο περισσότερο κοντινές τιμές γίνεται στις απαιτούμενες δεξιότητες (*overqualified* = "ΟΧΙ") τότε αφήνουμε τις τιμές όλων των δεξιοτήτων ως έχουν.
3. Στη συνέχεια το πρόγραμμα θα υπολογίζει την Ευκλείδεια απόσταση των δεξιοτήτων κάθε υποψηφίου από τις δεξιότητες της θέσης εργασίας, και θα προχωρά στην κατάταξη των υποψηφίων με αύξουσα σειρά απόστασης (από αυτούς που έχουν την μικρότερη απόσταση προς αυτούς που έχουν την μεγαλύτερη). Για υποψήφιους με την ίδια απόσταση δεν έχει σημασία η σειρά εμφάνισής τους. Σε κάθε γραμμή της κατάταξης θα πρέπει να φαίνεται το ονοματεπώνυμο του υποψηφίου και η απόσταση.

Για τη βάση δεδομένων χρησιμοποιείτε την *sqlite3*. Επίσης δημιουργήστε την γραφική διασύνδεση χρήστη με την βιβλιοθήκη *Tkinter*.

### **Ζητούμενα:**

Να γράψετε κώδικα στη γλώσσα προγραμματισμού *python* ο οποίος θα υποστηρίζει τις ακόλουθες περιπτώσεις χρήσης:

1. **Καταχώρηση θέσης εργασίας και δεξιοτήτων:** Σ' αυτή την περίπτωση χρήσης ο χρήστης θα πρέπει να μπορεί να κάνει καταχώρηση θέσης εργασίας, της τιμής της μεταβλητής

overqualified, και των δεξιοτήτων με τον απαιτούμενο βαθμό-επίπεδο κάθε δεξιότητας και το αν είναι απαραίτητο αυτό το επίπεδο ή όχι.

**2. Καταχώρηση υποψηφίων:** Σ' αυτή την περίπτωση χρήσης ο χρήστης θα πρέπει να μπορεί να εισάγει υποψήφιους για μία θέση εργασίας. Για κάθε υποψήφιο θα εισάγεται το όνομά του και προαιρετικά και άλλα στοιχεία (π.χ. τηλέφωνο, διεύθυνση κ.λπ.), ώστε να γίνει η εφαρμογή όσο πιο ρεαλιστική γίνεται. Επίσης θα εισάγεται ο βαθμός γνώσης για κάθε μία από τις δεξιότητες της θέσης εργασίας του υποψηφίου στην κλίμακα 1-10 με βάση την εκτίμηση της επιτροπής αξιολόγησης.

**3. Κατάταξη υποψηφίων:** Σε αυτή την περίπτωση χρήσης θα πρέπει να μπορεί ο χρήστης να εξάγει κατάταξη των υποψηφίων με την σημασία που είπαμε ήδη. Η κατάταξη θα γίνεται με αύξουσα σειρά Ευκλείδειας απόστασης των απαιτούμενων δεξιοτήτων θέσης και των δεξιοτήτων του κάθε υπαλλήλου.

Η ευκλείδεια απόσταση υπολογίζεται ως εξής για δύο διανύσματα  $(x_1, x_2, x_3, x_4)$  &  $(y_1, y_2, y_3, y_4)$ , (υποθέστε πως τα το πρώτο διάνυσμα είναι οι βαθμολογίες των τεσσάρων δεξιοτήτων μιας θέσης και το δεύτερο οι βαθμολογίες κάποιου υποψηφίου στις ίδιες δεξιότητες).

$$\text{Απόσταση} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + (x_4 - y_4)^2}$$

Αν έχετε Python με έκδοση ίση ή μεγαλύτερη από την 3.8 τότε υπάρχει συνάρτηση `dist` στο module `math` που υπολογίζει αυτήν την απόσταση και δέχεται σαν όρισμα δύο λίστες ή πλειάδες, όπως δείχνει το ακόλουθο παράδειγμα:

```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64
bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from math import dist
>>> dist((8,8,6,5,4), (8,8,6,5,4))
0.0
>>> dist((8,8,6,5,4), (8,8,4,3,2))
3.4641016151377544
>>> dist((8,8,6,5,4), (8,8,5,5,2))
2.23606797749979
>>>
```

Στο παράδειγμα οι δεξιότητες της θέσης είναι αυτές που έχουμε στον πίνακα που δόθηκε πριν και υποθέτουμε τρεις υποψήφιους που και οι τρεις έχουν 8 στις δύο πρώτες απαραίτητες δεξιότητες και τις βαθμολογίες που φαίνονται στον ακόλουθο πίνακα στις άλλες δεξιότητες:

Θέση	Υποψ. 1	Υποψ. 2	Υποψ. 3
8	8	8	8
8	8	8	8
6	6	4	5
5	5	3	5
4	4	2	2
	0	3,464102	2,236068

Στο παράδειγμα αυτό πρώτος θα ήταν ο Υποψήφιος 1, δεύτερος ο Υποψήφιος 3 και τρίτος ο Υποψήφιος 2.

**Παραδοτέα:**

Παραδίδετε ένα αρχείο zip που περιέχει:

1. Τον κώδικά σας.
  2. Οδηγίες εγκατάστασης και χρήσης της εφαρμογής σας.
  3. Έκθεση (5-10 σελίδων pdf) που θα εξηγεί: τί κάνατε, πώς το κάνατε, πώς μοιράσατε τη δουλειά, ποια ήταν τα αποτελέσματα. Βάλτε screenshots από την εκτέλεση της εφαρμογής σας (όχι screenshots του κώδικα!!!)
  4. Επιπλέον, κάθε φοιτητής ξεχωριστά θα πρέπει να γράψει μια ατομική έκθεση 1-2 σελίδων pdf που θα εξηγεί τί ακριβώς έκανε ο ίδιος, πόσες ώρες δούλεψε και κατάλογο με βιβλιογραφία (τί μελέτησε, από ποιες πηγές, ακόμα κι αν αυτές δεν είναι βιβλία, αλλά ιστοσελίδες ή video). Μην ξεχάσετε να βάλετε το όνομά σας στο αρχείο.
  5. Αρχείο pdf με τις διαφάνειες της παρουσίασής σας.
  6. Προαιρετικά, μπορείτε να φτιάξετε video με την παρουσίασή σας στο youtube και να αναρτήσετε τον κώδικά σας στο github. Σε αυτήν την περίπτωση βάλτε τα links στο κείμενο της έκθεσής σας.
-