

Οδηγός Εργασίας

Project Python Scrabble

ΕΡΓΑΣΙΑ Θ2

ΘΕΩΡΙΕΣ ΜΑΘΗΣΗΣ & ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ 2017



Περιεχόμενα

- (1) Τι θα χρειαστείτε
- (2) Γενική περιγραφή της εργασίας
- (3) Ανάλυση και Περιορισμοί
- (4) Σενάριο Παιχνιδιού
- (5) Αλγόριθμος Παιχνιδιού Υπολογιστή
- (6) Βιβλιοθήκη
- (7) Τεκμηρίωση
- (9) Αξιολόγηση
- (9) Παραδοτέο
- (10) Γενικά για τις εξετάσεις του μαθήματος



(1) Τι θα χρειαστείτε για την εργασία σας

- Για να ολοκληρώσετε σωστά την εργασία σας θα χρειαστείτε:
- (α) Αυτόν τον **Οδηγό Εργασίας** (που προφανώς τον έχετε ήδη)
- (β) Τα **Πακέτα διαφανειών** για την Python
- – Αν θέλετε να παρακολουθήσετε διαλέξεις μου μπορείτε να εγγραφείτε δωρεάν στο online μάθημα Python: «[Εισαγωγή στον Προγραμματισμό με Python](#)»
- (γ) Το αρχείο **greek.txt** που περιέχει ~350 K λέξεις της ελληνικής γλώσσας και με βάση το οποίο θα δημιουργήσετε το **greek7.txt** που περιέχει τις λέξεις της ελληνικής που έχουν μέχρι και 7 γράμματα. Όσοι παρακολουθήσατε το εργαστήριο έχετε ήδη το greek7.txt
- Όλα τα παραπάνω βρίσκονται στη σελίδα του μαθήματος
- **Πηγές στο διαδίκτυο:**
- [Σκράμπλ – Βικιπαίδεια](#): Τι είναι το Scrabble και γενικοί κανόνες του παιχνιδιού
- Python **itertools** (ενσωματωμένη βιβλιοθήκη) – ειδικά τη συνάρτηση [itertools.permutations\(\)](#)
- [Built in types](#) (ειδικά τις μεθόδους συμβολοσειρών: [strip](#), [split](#))



(2) Γενική περιγραφή της εργασίας

- Ο στόχος της εργασίας είναι να αναπτύξετε μια εφαρμογή σε κώδικα Python η οποία να **υλοποιεί μια απλοποιημένη εκδοχή του παιχνιδιού Scrabble στον υπολογιστή**
- Στην εκδοχή αυτή ένας παίκτης παίζει Scrabble με αντίπαλο τον υπολογιστή
- Η εφαρμογή σας θα πρέπει να ακολουθεί σε γενικές γραμμές την ανάλυση που περιγράφεται στις επόμενες διαφάνειες
- Μπορείτε να επιλέξετε αν θα εργαστείτε ατομικά ή ομαδικά (μέχρι 2 άτομα)
- Σε περίπτωση ομαδικής εργασίας κάθε μέλος της ομάδας θα πρέπει να γνωρίζει την τεκμηρίωση ολόκληρου του κώδικα που αποτελεί την εργασία
- Κάθε **βελτίωση ή επέκταση** του Σεναρίου / Αλγορίθμου του Παιχνιδιού (σε σχέση με αυτά που προτείνω στον παρόντα Οδηγό) είναι απολύτως δεκτή αρκεί να είναι ικανοποιητικά **τεκμηριωμένη**

(3) Ανάλυση και Περιορισμοί

- Η εφαρμογή θα τρέχει στο περιβάλλον IDLE της βασικής Python και θα βασίζεται σε κώδικα Python έκδοσης 3.4 ή μεγαλύτερης
- Όλες οι πληροφορίες κατά την εξέλιξη του παιχνιδιού θα εμφανίζονται στη γραμμή εντολών '>>>' του περιβάλλοντος IDLE (δηλ. η εφαρμογή δεν θα είναι παραθυρική και δεν θα χρησιμοποιεί γραφικά)
- Προγραμματιστικά η εφαρμογή θα πρέπει:
- Α) Να χρησιμοποιεί τουλάχιστον **2 Κλάσεις** και τουλάχιστον **1 Αντικείμενο από κάθε Κλάση**
 - Η μία κλάση θα είναι **ΟΠΩΣΔΗΠΟΤΕ η SakClass** (την αναπτύξαμε σε σημαντικό βαθμό στο εργαστήριο και περιμένω να τη δω στον κώδικά σας).
 - Η άλλη μπορεί να είναι **όποια θέλετε**. Προτείνω την **Player**, δηλ. μια κλάση που 'περιγράφει' τους Παίκτες που εμπλέκονται στο παιχνίδι.
- Β) Να χρησιμοποιεί **Συναρτήσεις** (περισσότερες από 1)
- Γ) Να χρησιμοποιεί **Λίστα και Λεξικό** (1 απ το καθένα ή και περισσότερα)
- Δ) Να χρησιμοποιεί δική σας **βιβλιοθήκη** κώδικα Python την οποία να εισάγει με `import` στο αρχείο του κύριου προγράμματος

(4) Σενάριο Παιχνιδιού

1/6

- **1)** Στο ξεκίνημα το πρόγραμμα θα πρέπει να δημιουργεί τον τύπο δεδομένου **SakClass** και ένα **αντικείμενο –στιγμιότυπο** (πχ. **sak**).
- Το αντικείμενο θα υλοποιεί το «**σακουλάκι**» του φυσικού παιχνιδιού. Δηλαδή, θα είναι δομή δεδομένων η οποία περιέχει τα γράμματα και τις αξίες τους («βαθμοί» που δίνει το κάθε γράμμα) και τις μεθόδους που σχετίζονται με το «σακουλάκι» (πχ. εξαγωγή γραμμάτων με τυχαίο τρόπο για να δοθούν στον παίκτη ή τον Η/Υ).
- Πληροφορίες για το πόσα είναι αυτά τα γράμματα υπάρχουν [εδώ](#)
- **2)** Το πρόγραμμα κληρώνει 7 γράμματα από το «σακουλάκι» και 7 για τον Η/Υ. Παρουσιάζει στον παίκτη τα γράμματά του μαζί με την αξία τους (η αξία μπορεί να είναι ένας ακέραιος αριθμός που εμφανίζεται δίπλα στο γράμμα). Ταυτόχρονα το πρόγραμμα θα αφαιρεί από το «σακουλάκι» τα γράμματα που κληρώθηκαν. Η πληροφορία για το πόσα γράμματα παραμένουν στο σακουλάκι θα πρέπει να είναι γνωστή στον παίκτη.
- **3)** Το πρόγραμμα στη συνέχεια περιμένει τον παίκτη να πληκτρολογήσει μια λέξη με τα γράμματα που διαθέτει (ένα παράδειγμα οθόνης βλέπετε παρακάτω – είναι απολύτως ενδεικτικό – δεν εννοώ να μοιάζει έτσι και η δική σας εφαρμογή)

>>>

Στο σακουλάκι: 75 γράμματα – Παίζεις:

Διαθέσιμα Γράμματα: Ε,1 – Β,8 – Α,1 – Χ,8 – Μ,3 – Η,1 – Θ,10

Λέξη:|

- **4)** Όταν ο παίκτης πληκτρολογήσει μια λέξη τότε το πρόγραμμά σας πρέπει να κάνει 2 ελέγχους:
- 4-A) Ελέγχει αν η λέξη αποτελείται από γράμματα που όντως διαθέτει ο παίκτης. Αν όχι, εμφανίζει ένα σχετικό μήνυμα και περιμένει νέα λέξη
- 4-B) Αν ναι, τότε ελέγχει αν η λέξη που δόθηκε περιλαμβάνεται στον κατάλογο αποδεκτών λέξεων που έχει προκύψει από το αρχείο **greek7.txt**. Το αρχείο greek7.txt είναι το αρχείο με τις λέξεις μέχρι και 7 γράμματα που θα πρέπει να δημιουργήσετε με βάση το αρχείο greek.txt που περιλαμβάνει όλες τις λέξεις ανεξαρτήτως μήκους. Το δημιουργήσαμε ήδη στο εργαστήριο.
- 4-Γ) Σε περίπτωση που ο παίκτης δεν βρίσκει λέξη με τα γράμματα που διαθέτει μπορεί να πληκτρολογήσει **‘p’** (δηλ. pass) οπότε το πρόγραμμα θα πρέπει: α) να κληρώσει νέα γράμματα για τον παίκτη και β) να επιστρέψει τα διαθέσιμα γράμματα στο «σακουλάκι». Στη συνέχεια περνά στο βήμα 6 που περιγράφεται παρακάτω.

- **5)** Αν η λέξη είναι αποδεκτή το πρόγραμμα υπολογίζει τους πόντους της λέξης και εμφανίζει στον παίκτη το νέο του σκορ, μαζί με μια προτροπή της μορφής 'Enter για συνέχεια' (πχ. δείτε το παράδειγμα της οθόνης παρακάτω)

>>>

Στο σακουλάκι: 75 γράμματα - Παίζεις:

Διαθέσιμα Γράμματα: Ε,1 - Β,8 - Α,1 - Χ,8 - Μ,3 - Η,1 - Θ,10

Λέξη: ΜΕΘΗ

Αποδεκτή Λέξη - Βαθμοί: 15 - Σκορ: 55

Enter για Συνέχεια



- **6)** Εφόσον δοθεί 'Enter' (ή 'ρ') από τον παίκτη στη συνέχεια το πρόγραμμα:
- 6-1) Συμπληρώνει με νέα γράμματα τα διαθέσιμα του παίκτη (ώστε να είναι πάντοτε 7, θυμηθείτε ότι ταυτόχρονα πρέπει να τα αφαιρεί από το «σακουλάκι»).
- 6-2) Εμφανίζει τα γράμματα του υπολογιστή και την λέξη που παίζει, μαζί με τη βαθμολογία λέξης και το συνολικό σκορ του υπολογιστή. Συμπληρώνει κρυφά τα γράμματα του Η/Υ και τέλος επανέρχεται στο βήμα 3.
- Παράδειγμα οθόνης με τα βήματα αυτά βλέπετε παρακάτω.

Enter για Συνέχεια

Διαθέσιμα Γράμματα: Γ,4 - Β,8 - Α,1 - Χ,8 - Κ,2 - Α,1 - Ο,1

Στο σακουλάκι: 71 γράμματα - Παίζει ο Η/Υ:

Γράμματα Η/Υ: Σ,1 - Ν,1 - Ο,1 - Φ,8 - Ζ,10 - Η,1 - Ε,1

Λέξη Η/Υ: ΝΕΦΟΣ, Βαθμοί: 12 - Σκορ Η/Υ: 42

Στο σακουλάκι: 66 γράμματα - Παίζεις:

Διαθέσιμα Γράμματα: Γ,4 - Β,8 - Α,1 - Χ,8 - Κ,2 - Α,1 - Ο,1

Λέξη: |

- **7)** Το παιχνίδι συνεχίζεται έτσι μέχρις ότου συμβεί κάτι από τα παρακάτω:
 - 7-1) Ο παίκτης επιθυμεί να σταματήσει (ή δεν βρίσκει αποδεκτή λέξη να παίξει **και** δεν υπάρχουν γράμματα για να αλλάξει) οπότε εισάγει τον χαρακτήρα **‘q’** όταν είναι η σειρά του να πληκτρολογήσει λέξη.
 - 7-2) Δεν υπάρχουν πλέον γράμματα αρκετά στο «σακουλάκι» ώστε να αντικατασταθούν όσα λείπουν, είτε του παίκτη είτε του Η/Υ
 - 7-3) Ο Η/Υ δεν βρίσκει κάποια αποδεκτή λέξη να παίξει
- **8)** Σε οποιαδήποτε από τις παραπάνω περιπτώσεις το πρόγραμμα σταματά και:
 - 8-1) Ανακοινώνει τις βαθμολογίες Παίκτη και Η/Υ ανακηρύσσοντας τον νικητή
 - 8-2) Καταχωρεί σε **κατάλληλη δομή δεδομένων που αποθηκεύεται σε αρχείο** μια νέα καταχώρηση με όποια στοιχεία του παιχνιδιού κρίνονται σημαντικά (πχ. πόσες κινήσεις παίχτηκαν, ποιο ήταν το σκορ παίκτη και Η/Υ). Την δομή αυτή το πρόγραμμα την «φορτώνει» όταν ξεκινά το παιχνίδι και μπορεί να δίνει στον παίκτη σχετική ενημέρωση.

-- Σενάριο Παιχνιδιού

6/6

- **9)** Το παιχνίδι μπορεί να ξεκινά με μία εισαγωγική οθόνη η οποία να δίνει κάποιες δυνατότητες στον παίκτη πριν ξεκινήσει το παιχνίδι.
- Παράδειγμα μιας τέτοιας οθόνης βλέπετε δίπλα:

```
>>>
***** SCRABBLE *****
-----
1: Σκορ
2: Ρυθμίσεις
3: Παιχνίδι
q: Έξοδος
-----
```

- **10)** Θα πρέπει να σκεφθείτε προσεκτικά για τη δομή που θα περιέχει το λεξικό που θα συμβουλευέται το πρόγραμμά σας. Από τη είδος της δομής και τη μορφή της θα εξαρτηθεί η αποδοτικότητα του κώδικά σας. Το συζητήσαμε και αυτό στο εργαστήριο και έχουμε δώσει την απάντηση.
- **11)** Για την καλύτερη ανάπτυξη του κώδικά σας λάβετε υπόψη σας τον κώδικα για τις ασκήσεις 3 και 4 της ενότητας 'Κλάσεις-Αντικείμενα' οι εκφωνήσεις περιλαμβάνονται στο αρχείο:

08.1-Python-Classes-OOP-Basics-1.pdf

(5) Αλγόριθμος Παιχνιδιού Υπολογιστή

- Ο αλγόριθμος παιχνιδιού του Η/Υ μπορεί να έχει τις **τρεις** παρακάτω μορφές:
- Α) **MIN Letters**: Το πρόγραμμα δημιουργεί όλες τις δυνατές **μεταθέσεις** (**permutations**) των γραμμάτων που διαθέτει ο Η/Υ ξεκινώντας από 2 και ανεβαίνοντας μέχρι τα 7 γράμματα. Για κάθε μετάθεση ελέγχει αν είναι αποδεκτή λέξη και παίζει την πρώτη αποδεκτή λέξη που θα εντοπίσει (δηλ. με τα λιγότερα γράμματα).
- Β) **MAX Letters**: Όπως και στο Α αλλά το πρόγραμμα ξεκινά από τις μεταθέσεις των γραμμάτων ανά 7 και **κατεβαίνει** προς το 2. Παίζει πάλι την πρώτη αποδεκτή λέξη αλλά τώρα αυτή με τα περισσότερα γράμματα.
- Γ) **SMART**: Όπως και στο Α αλλά **εξαντλεί** όλες τις μεταθέσεις 2 ως και 7 γραμμάτων χωρίς να σταματά. Βρίσκει τις αποδεκτές λέξεις και στο τέλος παίζει τη λέξη που δίνει τους περισσότερους βαθμούς.
- Ο παίκτης να μπορεί να καθορίζει το επίπεδο/αλγόριθμο παιχνιδιού του Η/Υ από τις **ρυθμίσεις** (δες προηγούμενη διαφάνεια). Περιμένω να δω στον κώδικά σας **τουλάχιστον 2** από τους παραπάνω αλγορίθμους ώστε να μπορούμε να καθορίζουμε με ποιο τρόπο θα παίζει ο Η/Υ.
- Οι μεταθέσεις μιας λίστας αντικειμένων μπορούν εύκολα να είναι διαθέσιμες στον κώδικά σας μέσω της συνάρτησης [itertools.permutations\(\)](#)

(6) Βιβλιοθήκη (module, άρθρωμα)

- Μια **‘βιβλιοθήκη’** (module = άρθρωμα) είναι ένα αρχείο .py στο οποίο έχετε αποθηκεύσει κώδικα που κάνει συγκεκριμένες εργασίες, πχ. κλάσεις ή συναρτήσεις
- Μια τέτοια βιβλιοθήκη συνδέεται με το κύριο πρόγραμμά σας με εντολή **import**
- Πχ. ας υποθέσουμε πως έχετε γράψει την κλάση SakClass και την έχετε αποθηκεύσει σε ένα αρχείο mylib.py
- Τότε στο κύριο πρόγραμμά σας μπορείτε να γράψετε:
- **import mylib**
- Και να δημιουργήσετε το αντικείμενο sak τύπου SakClass γράφοντας:
- **sak = mylib.SakClass()**



(7) Τεκμηρίωση (με μορφή **docstring**)

- Συμπεριλάβετε τεκμηρίωση του κώδικά σας σε μορφή **docstring** στην αρχή του κύριου προγράμματος.
- Κυρίως με ενδιαφέρει να τεκμηριώσετε τις **κλάσεις** που χρησιμοποιείτε εξηγώντας **ιδιότητες** και τις **μεθόδους** που χρησιμοποιούν
- Επίσης συμπεριλάβετε κάθε χρήσιμη πληροφορία για τον τρόπο που εκτελείτε το πρόγραμμά σας και που πρέπει να ξέρει κάποιος που θέλει να παίξει το παιχνίδι σας.



(8) Αξιολόγηση 1/2

- Η αξιολόγηση της εργασίας θα γίνει με βάση τα παρακάτω

ΚΡΙΤΗ-ΡΙΟ	ΧΑΡΑΚΤΗΡΙ-ΣΤΙΚΟ	ΣΧΟΛΙΑ - ΕΞΗΓΗΣΕΙΣ
A	ΚΛΑΣΕΙΣ – ΑΝΤΙΚΕΙ-ΜΕΝΑ	Να υλοποιηθεί η κλάση SakClass (υποχρεωτικά) και το αντικείμενο sak , καθώς και η κλάση Player και το αντικείμενο pl (προσοχή: απλά προτείνω την Player, μπορείτε να γράψετε μίαν άλλη δική σας). Υλοποιήστε τουλάχιστον 3 μεθόδους για κάθε κλάση (περιλαμβάνεται και η μέθοδος-κατασκευαστής). Αξιολογείται η πληρότητα κατασκευής των κλάσεων, δηλ. η ενσωμάτωση μεθόδων που υλοποιούν όσο το δυνατόν περισσότερες λειτουργίες.
B	ΣΥΝΑΡΤΗΣΕΙΣ	Να χρησιμοποιούνται Συναρτήσεις (περισσότερες από 1). Αξιολογείται η ορθότητα του κώδικα των συναρτήσεων και η αποδοτική χρήση τους.
Γ	ΛΙΣΤΑ, ΛΕΞΙΚΟ	Να χρησιμοποιούνται Λίστα και Λεξικό (1 από το καθένα ή και περισσότερα). Αξιολογείται ο τρόπος χρήσης τους ώστε να υλοποιείται ο κώδικας αποδοτικά (δηλ. συντομότερος σε πλήθος εντολών και ταχύτερος σε εκτέλεση).
Δ	ΒΙΒΛΙΟΘΗΚΗ	Οι κλάσεις θα βρίσκονται σε ιδιαίτερη βιβλιοθήκη (ξεχωριστό αρχείο που θα ονομαστεί σύμφωνα με τις οδηγίες που δίνονται στην επόμενη υποενότητα 8).
E	ΑΛΓΟΡΙΘΜΟΣ Η/Υ	Υλοποιήστε τουλάχιστον 2 από τους προτεινόμενους αλγόριθμους . Είναι απόλυτα αποδεκτό να αλλάξετε κάποιον από τους προτεινόμενους με έναν άλλο αποδοτικότερο κάτι που μπορεί να σας δώσει υψηλότερη βαθμολογία.

- Συνέχεια κριτηρίων αξιολόγησης ... →

(8) Αξιολόγηση 2/2

ΚΡΙΤΗ- ΡΙΟ	ΧΑΡΑΚΤΗΡΙ- ΣΤΙΚΟ	ΣΧΟΛΙΑ - ΕΞΗΓΗΣΕΙΣ
ΣΤ	ΟΡΘΟΤΗΤΑ ΚΩΔΙΚΑ - ΜΗΝΥΜΑΤΑ	Αξιολογείται γενικά η ορθότητα του κώδικα ώστε να παίζεται σωστά το παιχνίδι καθώς και η κατανοητή εμφάνιση των μηνυμάτων και υλοποίηση των διάφορων προβλεπόμενων λειτουργιών. Μπορείτε να σχεδιάσετε την εμφάνιση των μηνυμάτων όπως θέλετε και όχι οπωσδήποτε σύμφωνα με τα προηγούμενα παραδείγματα που δίνω.
Z	ΤΕΚΜΗΡΙΩΣΗ	Γράψτε μια ικανοποιητική τεκμηρίωση με βάση τις οδηγίες που δίνονται στη σχετική διαφάνεια. Έλλειψη τεκμηρίωσης μειώνει τη βαθμολογία σας.
Η	ΑΡΧΕΙΑ	Υποβάλετε όλα τα αρχεία που είναι απαραίτητα για να εκτελείται σωστά ο κώδικάς σας. Λάθος εκτέλεση του κώδικα λόγω έλλειψης κάποιου αρχείου μειώνει τη βαθμολογία σας. ΠΡΟΣΟΧΗ: <u>μην μου στείλετε το αρχείο greek7.txt</u> με τις λέξεις που χρησιμοποιεί το παιχνίδι. Αυτό το έχω ήδη, ΑΛΛΑ όπως εξήγησα και στο εργαστήριο περιμένω να δω να χρησιμοποιείτε μια κατάλληλη δομή που προκύπτει από το greek7.txt και ΔΕΝ είναι κατ' ανάγκη λίστα.

(9) Παραδοτέο

- Παραδοτέο της εργασίας σας είναι **1 zip ή rar αρχείο που περιλαμβάνει:**
- (α) Το αρχείο **mylibAEM.py** που περιλαμβάνει τις **κλάσεις** της εργασίας σας
 - Πχ. mylib1234.py ή mylib12341245.py σε περίπτωση 2 συνεργατών με AEM 1234 & 1245 (προσοχή: τα AEM συνεχόμενα χωρίς ενδιάμεσες παύλες κλπ.)
- (β) Το αρχείο **mainAEM.py** που περιλαμβάνει το **κύριο πρόγραμμα**
 - Πχ. main1234.py ή main12341245.py σε περίπτωση 2 συνεργατών (τα AEM επίσης συνεχόμενα)
- (γ) Κάθε άλλο **απαραίτητο αρχείο** όπως εσείς νομίζετε (**ΑΛΛΑ ΌΧΙ το greek7.txt**)
- >>> Τα παραπάνω αρχεία (α,β,γ) τα **συμπιέζετε σε αρχείο zip ή rar**
- >>> Ονομάζετε το συμπιεσμένο αρχείο: **PythonScrabbleAEM** (ή βάζετε AEM1AEM2 αν είστε 2 συνεργάτες – Προσοχή μην μπερδευτείτε: όπου AEM θα βάλετε τον πραγματικό **αριθμό** του AEM σας) και το **υποβάλετε μέσω της σελίδας στο elearning** μέχρι την **προθεσμία** που θα ανακοινωθεί στην εξεταστική που θέλετε να παραδώσετε την εργασία

(10) Γενικά για τις εξετάσεις του μαθήματος

- Το μάθημα έχει **3 δραστηριότητες αξιολόγησης (εξετάσεις)**:
 - **Θ1**: Τεστ Python (εξέταση εργαστηρίου) 20%
 - **Θ2**: Εργασία (Project) Python Scrabble 40%
 - **Θ3**: Γραπτά 40%
-
- Μπορείτε να “μοιράσετε” τις εξετάσεις όπως θέλετε στις 3 εξεταστικές Ιουνίου-Σεπτεμβρίου 2017 & Φεβρουαρίου 2018 (κρατώ βαθμούς)
 - *Προσοχή*: Εφόσον χρωστάτε το μάθημα και ξαναδιδαχθεί το εαρινό 2018 υποχρεούστε σε συνολική επανεξέταση (δεν κρατούνται βαθμοί μετά τον Φεβρουάριο 2018)