Οδηγός Εργασίας

Project Python Scrabble

ΕΡΓΑΣΙΑ Ε2

ΘΕΩΡΙΕΣ ΜΑΘΗΣΗΣ & ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ 2024

ΤΜ. ΠΛΗΡΟΦΟΡΙΚΗΣ ΑΠΘ ΣΤ. ΔΗΜΗΤΡΙΑΔΗΣ

Ελλάδα / Το επιτραπέζιο παιχνίδι Scrabble μπαίνει στα σχολεία ως εργαλείο εκμάθησης των ελληνικών

Ο πρόεδρο του Κέντρου Ελληνικής Γλώσσας μιλά για τις ενέργειες του επιστημονικού φορέα του υπουργείου Παιδείας να εντάξει το δημοφιλές επιτραπέζιο παιχνίδι Scrabble στο σχολικό περιβάλλον

NEWSROOM, 4.8.2019 | 11:08

Περιεχόμενα

- (1) Γενική περιγραφή
- (2) Κλάσεις
- (3) Αναλυτικά: Η ροή και οι κανόνες του παιχνιδιού
- (4) Δομή δεδομένων για τις αποδεκτές λέξεις
- (5) Αλγόριθμος play για την κλάση Computer
- (6) Βιβλιοθήκη & import
- (7) Τεκμηρίωση & docstring
- (8) Αξιολόγηση εργασίας
- (9) Παραδοτέο
- (10) Υπενθύμιση για τις εξετάσεις του μαθήματος

(1) Γενική περιγραφή

Τι θα χρειαστείτε για την εργασία σας

- Για να ολοκληρώσετε σωστά την εργασία σας θα χρειαστείτε:
- (α) Αυτόν τον **Οδηγό Εργασίας**
- (β) Τα **Πακέτα διαφανειών** για την Python και ειδικά αυτό που αναφέρεται ως «07-Python-ScriptsForScrabble»
- (γ) Το αρχείο **greek7.txt** που περιέχει τις λέξεις της ελληνικής γλώσσας που έχουν μέχρι και 7 γράμματα *(διαθέσιμο στη σελίδα μαθήματος, ενότητα 'Scripts For Scrabble)*

- *Όλα τα παραπάνω βρίσκονται στη σελίδα του μαθήματος*

Γενική περιγραφή της εργασίας 1/4

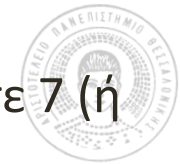
- Ο γενικός στόχος της εργασίας είναι να αναπτύξετε μια **εφαρμογή** σε κώδικα Python η οποία να υλοποιεί την απλοποιημένη εκδοχή του παιχνιδιού Scrabble στον υπολογιστή.

- Ειδικότερος στόχος της εργασίας είναι να δημιουργηθεί μια βασική **αρχιτεκτονική** και **αλγόριθμος** που να καθοδηγεί το λογισμικό (τον παίκτη Η/Υ) (ή και τον παίκτη-άνθρωπο σε κάποιες περιπτώσεις) να παίξει το παιχνίδι

- Η εφαρμογή σας θα πρέπει να ακολουθεί σε γενικές γραμμές την ανάλυση που περιγράφεται στις επόμενες διαφάνειες

Γενικά: Ροή και Κανόνες του παιχνιδιού 2/4

- Η **απλοποιημένη εκδοχή του Scrabble** θα παιχτεί ως εξής:
- 1) Υπάρχει ένα αρχικό «σακκουλάκι» που περιέχει συγκεκριμένο πλήθος γραμμάτων
- 2) Στην αρχή του παιχνιδού δίνονται (αφαιρούνται από το σακκουλάκι) με τυχαίο τρόπο 7 γράμματα στον παίκτη και 7 γράμματα στον Η/Υ.
  - Τα γράμματα που θα έχει κάθε παίκτης θα είναι πάντοτε 7. Προς το τέλος του παιχνιδιού όταν δεν θα υπάρχουν άλλα γράμματα διαθέσιμα τότε μπορούν να είναι και λιγότερα από 7.
- 3) Ο παίκτης σκέφτεται μια λέξη που μπορεί να φτιάξει με τα γράμματα και την πληκτρολογεί. Το λογισμικό εκτιμά αν η λέξη είναι αποδεκτή και αν <u>ναι</u> τότε:
  - Α) του δίνει τους αντίστοιχους πόντους της λέξης (που είναι το άθροισμα των πόντων που δίνει κάθε γράμμα)
  - Β) συμπληρώνει τα γράμματα του παίκτη ώστε να έχει πάντοτε 7 (ή λιγότερα αν δεν υπάρχουν διαθέσιμα)



Γενικά: Ροή και Κανόνες του παιχνιδιού 3/4

- *συνέχεια…*

- 4) Στη συνέχεια «παίζει» ο Η/Υ και γίνονται οι κατάλληλες ενέργειες από το λογισμικό.
- 5) Αν ο παίκτης ή ο Η/Υ δεν βρίσκουν λέξη (ή δεν θέλουν να παίξουν κάποια λέξη) μπορούν να ζητήσουν αλλαγή γραμμάτων (αλλάζουν και τα 7 ή όσα γράμματα έχουν διαθέσιμα) και χάνουν τη σειρά τους.
- 6) Το παιχνίδι συνεχίζεται με παρόμοιες κινήσεις από παίκτη και Η/Υ μέχρις ότου να τελειώσουν τα γράμματα στο σακκουλάκι και κανείς να μην μπορεί να σχηματίσει αποδεκτή λέξη ή να παραιτηθεί κάποιος από το παιχνίδι (παίκτης ή Η/Υ).
- 7) Τότε το λογισμικό ανακοινώνει τη λήξη του παιχνιδιού και τον νικητή και παρουσιάζει το τελικό σκορ.

Γενικά: Ροή και Κανόνες του παιχνιδιού 4/4

- Μπορείτε να επιλέξετε αν θα εργαστείτε **ατομικά ή ομαδικά** (N_{max} = 2, δηλ. μέχρι 2 άτομα στην ομάδα)

- Σε περίπτωση ομαδικής εργασίας τότε ο **φόρτος εργασίας αυξάνει** ώστε κάθε μέλος της ομάδας να συνεισφέρει επαρκώς στην ανάπτυξη της εργασίας. Περισσότερες οδηγίες για την ομάδα 2 ατόμων δίνονται στην επόμενη διαφάνεια**.**

- 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 και να μου κοινοποιηθεί έγκαιρα ώστε να πάρει την έγκρισή μου.

Για την ομαδική εργασία

- Σε περίπτωση ομαδικής εργασίας τότε:
- Α) Το λογισμικό-παιχνίδι της ομάδας θα πρέπει να υλοποιεί σενάριο **τριών παικτών**.
- Β) Θα πρέπει να αφορά **1 Human Player + 2 Computer Players** και για κάθε Computer Player θα πρέπει να υλοποιεί **και ένα διαφορετικό αλγόριθμο** για το πώς μπορεί να παίξει ο κάθε Computer Player (οι αλγόριθμοι για την κλάση Computer αναφέρονται στην ενότητα «Αλγόριθμος» αλλά μπορεί η ομάδα να προτείνει και να υλοποιήσει και δικό της)
- Γ) Το κάθε μέλος της ομάδας θα πρέπει να **γνωρίζει την τεκμηρίωση ολόκληρου του κώδικα** που αποτελεί την εργασία.

Περιβάλλον ανάπτυξης

- Η εφαρμογή σε κώδικα Python θα πρέπει να μπορεί να εκτελείται είτε σε περιβάλλον **IDLE** είτε σε περιβάλλον **Jupyter Notebook** και να εφαρμόζει τις αρχές **αντικειμενοστρεφούς προγραμματισμού** (Α.Π.) που διδάχθηκαν στο μάθημα.
- Όλες οι πληροφορίες κατά την εξέλιξη του παιχνιδιού θα εμφανίζονται με μηνύματα χαρακτήρων (character mode), δηλ. η εφαρμογή δεν θα είναι παραθυρική (δεν θα χρησιμοποιεί γραφικά)
- Πχ. μια πιθανή εμφάνιση μηνυμάτων από το παιχνίδι και πληκτρολόγησης λέξης από τον παίκτη, μπορεί να είναι όπως παρακάτω:

```
**************************************************
     *** Παίκτης: Stavros  *** Σκορ: 0
     >>> Γράμματα:  ['A', 'N', 'Ψ', 'H', 'P', 'Y', 'T']


     ΛΕΞΗ: ΑΥΤΗ
```

Παράδειγμα μηνυμάτων εξέλιξης παιχνιδιού

- Εδώ βλέπετε ένα παράδειγμα για το πώς μπορεί να εξελίσσεται το παιχνίδι στην οθόνη
- Δεν είστε υποχρεωμένοι να ακολουθήσετε αυτό το παράδειγμα
- Μπορείτε να διαμορφώσετε τα μηνύματα όπως θέλετε εσείς αρκεί να είναι κατανοητά από τους παίκτες

```
************************************************************
     *** Παίκτης: Stavros  *** Σκορ: 0
     >>> Γράμματα:  ['Α', 'Ν', 'Ψ', 'Η', 'Ρ', 'Υ', 'Τ']

     ΛΕΞΗ: ΑΥΤΗ
     Πόντοι Λέξης: 5
     *** Παίκτης: Stavros  *** Σκορ: 5
     >>> Γράμματα:  ['Ν', 'Ψ', 'Ρ', 'Β', 'Ν', 'Η', 'Ι']

************************************************************
************************************************************
     *** Παίκτης: PC  *** Σκορ: 0
     >>> Γράμματα:  ['Ρ', 'Ε', 'Α', 'Ε', 'Τ', 'Α', 'Λ']

     Παίζω τη Λέξη: ΑΕΡΑΤΕ
     Πόντοι Λέξης: 7
     *** Παίκτης: PC  *** Σκορ: 7
     >>> Γράμματα:  ['Λ', 'Ο', 'Ρ', 'Ο', 'Σ', 'Θ', 'Ι']

************************************************************
************************************************************
     *** Παίκτης: Stavros  *** Σκορ: 5
     >>> Γράμματα:  ['Ν', 'Ψ', 'Ρ', 'Β', 'Ν', 'Η', 'Ι']


     ΛΕΞΗ:
```

(2) Κλάσεις

Οι 5 κλάσεις του παιχνιδιού

- Προγραμματιστικά η εφαρμογή θα πρέπει:
- 1) Να χρησιμοποιεί **τουλάχιστον τις παρακάτω 5 Κλάσεις:**
- → **SakClass**: κλάση που καθορίζει πώς λειτουργεί το αντικείμενο «σακουλάκι» με τα γράμματα του παιχνιδιού
- → **Player**: βασική κλάση από την οποία παράγονται οι Human και Computer
  - → **Human**: κλάση παράγωγος της Player που καθορίζει λειτουργίες του παιχνιδιού για τον άνθρωπο-παίκτη
  - → **Computer**: κλάση παράγωγος της Player που καθορίζει λειτουργίες του παιχνιδιού για τον υπολογιστή-παίκτη
- → **Game**: κλάση που περιγράφει το πώς εξελίσσεται μια παρτίδα (game)

- Πέρα από τις παραπάνω μπορείτε να υλοποιήσετε και άλλες κλάσεις αν θέλετε εσείς
- Περισσότερες πληροφορίες για τις κλάσεις δίνονται στις επόμενες διαφάνειες

Κλάση SakClass – Βασικές λειτουργίες

sak
(αντικείμενο τύπου SakClass)

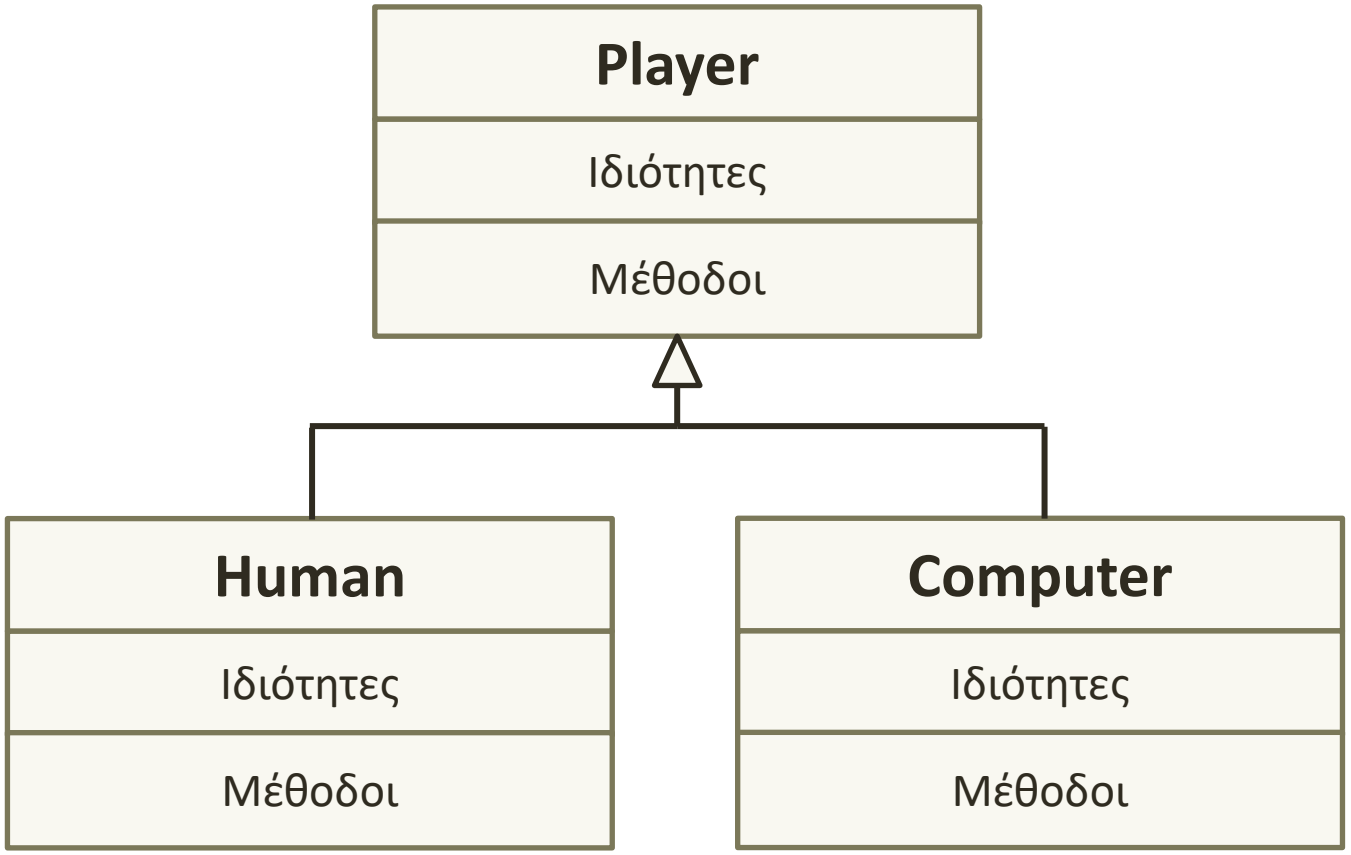
getletters()
(βγάζει από το σακουλάκι για τον παίκτη Ν γράμματα)

ΓΡΑΜΜΑΤΑ ΠΑΙΚΤΗ

putbackletters()
(επιστρέφει τα γράμματα του παίκτη στο σακουλάκι)

randomize_sak()
(«ετοιμάζει» δηλ. «τυχαιοποιεί» το σακουλάκι με τα γράμματα)

Κλάσεις: Player, Human & Computer

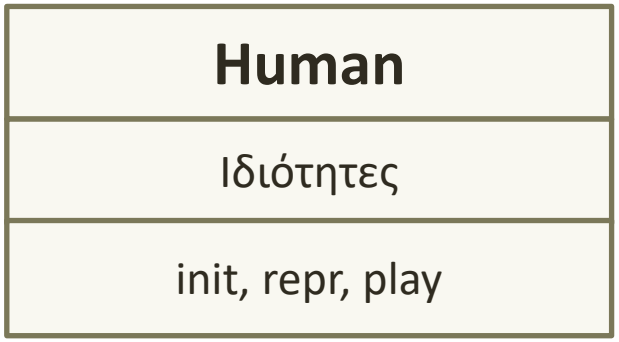


- Player: Βασική
- Human, Computer: Παράγωγοι της Player

Κλάση: **Player**

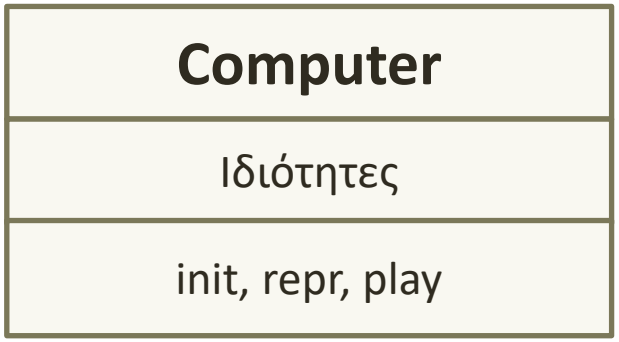
- **Player**: Βασική
- Μέθοδοι: init, repr και όποιες άλλες κρίνετε εσείς
- Ιδιότητες: όποιες κρίνετε εσείς

Κλάση: **Human**



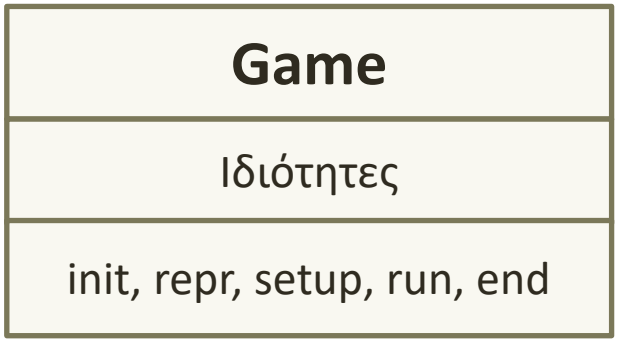
- **Human**: Παράγωγη της Player
- Μέθοδοι: init, repr, **play** και όποιες άλλες κρίνετε εσείς
- Η μέθοδος **play** θα υλοποιεί τον αλγόριθμο σύμφωνα με τον οποίο θα παίζει ο παίκτης
- Ιδιότητες: όποιες κρίνετε εσείς

Κλάση: **Computer**



- **Computer**: Παράγωγη της Player
- Μέθοδοι: init, repr, **play** και όποιες άλλες κρίνετε εσείς
- Η μέθοδος **play** θα υλοποιεί τον αλγόριθμο σύμφωνα με τον οποίο θα παίζει ο Η/Υ
- Ιδιότητες: όποιες κρίνετε εσείς

Κλάση: **Game**



- **Game**: Βασική
- Μέθοδοι: init, repr, **setup, run, end** και όποιες άλλες κρίνετε εσείς
- Η μέθοδος **setup** κάνει τις απαραίτητες ενέργειες κατά το ξεκίνημα του παιχνιδιού
- Η μέθοδος **run** 'τρέχει' το παιχνίδι
- Η μέθοδος **end** κάνει τις απαραίτητες ενέργειες κατά το κλείσιμο του παιχνιδιού
- Ιδιότητες: όποιες κρίνετε εσείς

Πώς θα διαχειριστείτε τον κώδικα των κλάσεων και του κύριου προγράμματος

- Ο κώδικας των κλάσεων θα βρίσκεται σε **ένα ξεχωριστό αρχείο** το οποίο θα ονομάσετε **classes.py**
- Ο κώδικας του κύριο προγράμματος θα βρίσκεται σε ένα αρχείο με όνομα **main-AEM.py**
  - Όπου ΑΕΜ θα γράψετε τον αριθμό του ΑΕΜ σας, πχ. main-1234.py

- Αν εργάζεστε σε ομάδα το όνομα αρχείου θα περιλαμβάνει και τα δύο ΑΕΜ το ένα μετά το άλλο **main-AEM1AEM2.py**
  - Πχ. main-12343001.py

(3) Αναλυτικά: Ροή και Κανόνες του παιχνιδιού

Ροή και Κανόνες του παιχνιδιού 1/7

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

- **1)** Το παιχνίδι μπορεί να ξεκινά με μία εισαγωγική οθόνη που παρουσιάζει κατάλογο επιλογών στον παίκτη. Παράδειγμα μιας τέτοιας οθόνης βλέπετε δίπλα:

- Επίσης αρχικά το πρόγραμμα θα πρέπει να δημιουργεί ένα **αντικείμενο –στιγμιότυπο** (πχ. **sak**) τύπου **SakClass**
- 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»
- Πληροφορίες για το πόσα είναι αυτά τα γράμματα υπάρχουν εδώ

sak
(αντικείμενο τύπου SakClass)

Ροή και Κανόνες του παιχνιδιού 2/7

- **2)** Η εξέλιξη του παιχνιδιού έχει ως εξής:
- Το πρόγραμμα κληρώνει 7 γράμματα από το «σακουλάκι» για τον παίκτη και 7 για τον υπολογιστή. 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.
- 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 χρειάζεται να σχεδιάσετε έτσι ακριβώς και τη δική σας εφαρμογή)

```
**********************************************************
     *** Παίκτης: Stavros  *** Σκορ: 0
     >>> Γράμματα:  ['A', 'N', 'Ψ', 'H', 'P', 'Y', 'T']

     ΛΕΞΗ: ΑΥΤΗ
```

Ροή και Κανόνες του παιχνιδιού 3/7

- **3)** Όταν ο παίκτης πληκτρολογήσει μια λέξη τότε το πρόγραμμά σας πρέπει να κάνει τουλάχιστον τους εξής ελέγχους:
- 3-Α) Ελέγχει αν η λέξη αποτελείται από γράμματα που όντως διαθέτει ο παίκτης. Αν **όχι**, εμφανίζει ένα σχετικό μήνυμα και περιμένει νέα λέξη
- 3-Β) Αν **ναι**, τότε ελέγχει αν η λέξη που δόθηκε περιλαμβάνεται στον κατάλογο αποδεκτών λέξεων που έχει προκύψει από το αρχείο **greek7.txt**.
- 3-Γ) Αντί για λέξη ο παίκτης αν θέλει **μπορεί να πληκτρολογήσει 'p'** (δηλ. pass) οπότε το πρόγραμμα θα πρέπει: α) να κληρώσει νέα γράμματα για τον παίκτη και β) μετά να επιστρέψει τα προηγούμενα γράμματα του παίκτη στο «σακουλάκι». Στην περίπτωση αυτή ο παίκτης χάνει τη σειρά του
- Στη συνέχεια περνά στο βήμα 5 που περιγράφεται σε επόμενη διαφάνεια.

- **Πού βρίσκεται το αρχείο greek7.txt;**
- Είναι διαθέσιμο στη σελίδα του μαθήματος (ενότητα '*Scripts For Scrabble*'). Το πώς κατασκευάζεται και τι περιλαμβάνει το εξηγούμε στο εργαστήριο

Ροή και Κανόνες του παιχνιδιού 4/7

- **4)** Αν η λέξη που **πληκτρολόγησε** ο παίκτης **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 των μηνυμάτων παρακάτω, *η μορφή του είναι μόνον ενδεικτική και όχι υποχρεωτική*)

```
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
```

Ροή και Κανόνες του παιχνιδιού 5/7

- **5)** Εφόσον δοθεί ‘Enter’ (ή ‘p’) από τον παίκτη στη συνέχεια το πρόγραμμα:
- 5-1) Συμπληρώνει με νέα γράμματα τα διαθέσιμα του παίκτη (ώστε να είναι πάντοτε 7, θυμηθείτε ότι ταυτόχρονα πρέπει να τα αφαιρεί από το «σακουλάκι»).
- 5-2) 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:
```

Ροή και Κανόνες του παιχνιδιού 6/7

- **6)** Το παιχνίδι συνεχίζεται έτσι μέχρις ότου συμβεί κάτι από τα παρακάτω:
- 6-1) 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, ώστε να τερματίσει το παιχνίδι
- 6-2) Δεν υπάρχουν πλέον γράμματα αρκετά στο «σακουλάκι» ώστε να αντικατασταθούν όσα λείπουν, είτε του παίκτη είτε του Η/Υ
- 6-3) Ο Η/Υ δεν βρίσκει κάποια αποδεκτή λέξη να παίξει και δεν υπάρχουν γράμματα στο «σακουλάκι» ώστε να αλλάξει

Ροή και Κανόνες του παιχνιδιού 7/7

- **7)** Σε οποιαδήποτε από τις προηγούμενες περιπτώσεις το πρόγραμμα σταματά και:
- 7-1) Ανακοινώνει τις βαθμολογίες Παίκτη και Η/Υ ανακηρύσσοντας τον νικητή
- 7-2) 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.
- Για να αποθηκεύσετε δεδομένα σε αρχείο θα χρησιμοποιήσετε την **βιβλιοθήκη pickle ή json (προτιμήστε json)** για την οποία μπορείτε να διαβάσετε στο πακέτο διαφανειών 07-Python-ScriptsForScrabble- στη σελίδα του μαθήματος.

(4) Δομή δεδομένων για τις αποδεκτές λέξεις

Δομή δεδομένων για τις λέξεις: σε Λεξικό (dictionary) ή σε Λίστα (list);

- Το πρόγραμμά σας στο ξεκίνημα θα πρέπει να κάνει μια ακόμη σημαντική εργασία:
- Να φορτώνει τις λέξεις της γλώσσας από το αρχείο **greek7.txt** και να τις μεταφέρει σε μια κατάλληλη δομή στον κώδικα ώστε να μπορούν να γίνουν οι απαραίτητοι έλεγχοι στο παιχνίδι
- Θα πρέπει να αποφασίσετε για τη **δομή δεδομένων** που θα περιέχει τις λέξεις της γλώσσας που θα συμβουλεύεται το πρόγραμμά σας.
- Από τη είδος της δομής θα εξαρτηθεί η **αποδοτικότητα του κώδικά** σας.
- Πρσέξτε το σημείο αυτό, το παρουσιάζουμε στο εργαστήριο και έχουμε δώσει την απάντηση.

(5) Αλγόριθμος play για την κλάση Computer

Επιλέξτε τον αλγόριθμο με τον οποίο θα παίζει ο Η/Υ (κλάση Computer)

- Μια σημαντική απόφαση και εργασία που πρέπει να κάνετε είναι να επιλέξετε με ποιό αλγόριθμο θα παίζει ο Η/Υ (κλάση Computer)
- Μπορείτε:
- Α) Να επιλέξετε έναν από τους αλγορίθμους που προτείνονται στη συνέχεια
- Β) Να καθορίσετε έναν δικό σας (στην περίπτωση αυτή ενημερώστε με έγκαιρα ώστε να εξετάσω τον αλγόριθμο που προτείνετε και να σας δώσω έγκριση)

ΑΛΓΟΡΙΘΜΟΣ 1 Min-Max-Smart

- Ο αλγόριθμος 1 του Η/Υ είναι ο **MIN-MAX-SMART** (προσοχή: πρόκειται για ομάδα τριών αλγορίθμων που πρέπει όλοι να υλοποιηθούν αν επιλέξετε αυτή την περίπτωση)
- Δηλ. μπορεί να έχει κατ' επιλογή του παίκτη από τις ρυθμίσεις, μία από 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.
- 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.

- Οι μεταθέσεις μιας λίστας αντικειμένων μπορούν εύκολα να είναι διαθέσιμες στον κώδικά σας μέσω της συνάρτησης itertools.permutations()

ΑΛΓΟΡΙΘΜΟΣ 2 Smart-Fail

- Ο αλγόριθμος παιχνιδιού του Η/Υ έχει την μορφή: **SMART-FAIL**

- **SMART**: Καλείται πρώτα ο αλγόριθμος SMART (όπως εξηγήθηκε στο σενάριο 1) και δημιουργεί μια λίστα με πιθανές λέξεις που μπορούν να παιχτούν
- **FAIL**: Στη συνέχεια καλείται ο FAIL. Όπως ένα άνθρωπος δεν βρίσκει πάντοτε τη βέλτιστη λέξη ο αλγόριθμος FAIL εισάγει ένα βαθμό αποτυχίας στο να παίξει τη βέλτιστη λέξη που έχει βρει ο SMART.
- Ο αλγόριθμος FAIL παίρνει ως είσοδο τη λίστα πιθανών λέξεων που παράγει ο SMART και επιλέγει να παίξει πχ. τη $2^{η}$ καλύτερη ή την $3^{η}$ καλύτερη λέξη στη λίστα (αντί της βέλτιστης πρώτης) ανάλογα πώς θα τον γράψετε.
- Μπορείτε να **αποφασίσετε εσείς τις λεπτομέρειες του αλγορίθμου FAIL** έτσι ώστε ο FAIL να προσομοιώνει τη μνήμη/ικανότητα ενός ανθρώπου που δεν γνωρίζει όλες τις λέξεις ή δεν μπορεί να βρει πάντοτε την καλύτερη δυνατή λέξη.

ΑΛΓΟΡΙΘΜΟΣ 3 Smart-Learn

- Ο αλγόριθμος παιχνιδιού του Η/Υ έχει την μορφή: **SMART-LEARN**

- **SMART**: Ο αλγόριθμος όπως εξηγήθηκε στο σενάριο 1
- **LEARN**: Ο υπολογιστής «μαθαίνει» νέες λέξεις, δηλ. ο αλγόριθμος προσομοιώνει έναν παίκτη ο οποίος μαθαίνει καθώς παίζει.
- Δηλ. αν ο παίκτης-άνθρωπος σε κάποια κίνηση εισάγει μια **λέξη που δεν υπάρχει** στο αρχείο greek7.txt (αλλά είναι μέχρι 7 γράμματα) τότε αντί να απορριφθεί ερωτάται ο παίκτης αν θέλει **να συμπεριληφθεί στο αρχείο λέξεων** για τη συνέχεια.
- Εφόσον ο παίκτης επιλέξει 'ΝΑΙ' η λέξη συμπεριλαμβάνεται στη δομή λέξεων του τρέχοντος παιχνιδιού ώστε να αναγνωρίζεται στη συνέχεια.
- Στο σενάριο αυτό θα πρέπει στο τέλος του παιχνιδιού να ενημερώσετε το αρχείο greek7.txt με τις νέες λέξεις ώστε να είναι διαθέσιμες σε επόμενη παρτίδα.
- Μπορείτε να σχεδιάσετε εσείς τις τεχνικές-προγραμματιστικές λεπτομέρειες του σεναρίου

ΑΛΓΟΡΙΘΜΟΣ 4 Smart-Teach

- Ο αλγόριθμος παιχνιδιού του Η/Υ έχει την μορφή: **SMART-TEACH**

- **SMART**: Ο αλγόριθμος όπως εξηγήθηκε στο σενάριο 1
- **TEACH**: Ο υπολογιστής «διδάσκει» τον παίκτη, δηλ. τον ενημερώνει ποια θα ήταν η καλύτερη λέξη να παίξει.
- Όταν είναι σειρά του παίκτη-ανθρώπου να παίξει, ο υπολογιστής εκτελεί επίσης τον αλγόριθμο SMART.
- Αφού παίξει ο παίκτη κάποια λέξη ο υπολογιστής ενημερώνει τον παίκτη αν η λέξη που έπαιξε είναι η καλύτερη δυνατή. Αν δεν είναι, τότε τον ενημερώνει ποια θα ήταν η καλύτερη ή και η $2^{η}$ καλύτερη λέξη που θα μπορούσε να παίξει.
- Μπορείτε να σχεδιάσετε εσείς τις τεχνικές-προγραμματιστικές λεπτομέρειες του σεναρίου.

Αλγόριθμοι και Ρυθμίσεις Παιχνιδιού

- Κατά ΄το ξεκίνημα του παιχνιδιού θα πρέπει να μπορεί ο παίκτης να επιλέξει με ποιόν αλγόριθμο θα παίξει ο υπολογιστής, εφόσον φυσικά το σενάριό σας δίνει αυτή τη δυνατότητα.

- Πχ. αν εφαρμόσατε τον Min-Max-Smart τότε από τις ρυθμίσεις θα πρέπει ο παίκτης να μπορεί να καθορίσει αν ο Η/Υ θα παίξει με την επιλογή Min ή Max ή Smart

- Αυτό μπορεί να γίνει π.χ. μέσα από την επιλογή '**Ρυθμίσεις**' στην αρχική οθόνη του παιχνιδιού.

(6) Βιβλιοθήκη & import

Module (Βιβλιοθήκη, άρθρωμα)

- Ένα ‘module’ (‘**βιβλιοθήκη**’ ή ‘άρθρωμα’) είναι ένα αρχείο .py στο οποίο έχετε αποθηκεύσει κώδικα που κάνει συγκεκριμένες εργασίες, πχ. κλάσεις ή και απλές 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 classes.py
- Τότε στο κύριο πρόγραμμά σας μπορείτε να γράψετε:
- **import classes**
- Και να δημιουργήσετε το αντικείμενο sak τύπου SakClass γράφοντας:
- **sak = classes.SakClass()**

(7) Τεκμηρίωση & docstring

Τεκμηρίωση (με μορφή docstring)

- Συμπεριλάβετε τεκμηρίωση του κώδικά σας σε μορφή **docstring** στην αρχή του κύριου προγράμματος ενσωματωμένο σε μια συνάρτηση με όνομα '**guidelines**'
- Δείτε την επόμενη ενότητα ('Αξιολόγηση') για να ξέρετε το τι πληροφορίες θα συμπεριλάβετε στην τεκμηρίωση (docstring)
- Επίσης συμπεριλάβετε κάθε άλλη χρήσιμη πληροφορία για τον τρόπο που εκτελείτε το πρόγραμμά σας και που πρέπει να ξέρει κάποιος που θέλει να παίξει το παιχνίδι σας.

- Το docstring τεκμηρίωσης θα πρέπει να εμφανίζεται όταν πληκτρολογηθεί:
- >>> **help(guidelines)**

- **Δοκιμάστε ότι το παραπάνω δουλεύει σωστά πριν υποβάλετε την εργασία σας.**

(8) Αξιολόγηση εργασίας

Πριν υποβάλετε την εργασία σας ελέγξτε πρώτα τα εξής:

- Στο docstring της guideline να εμφανίζονται οι εξής πληροφορίες:

1. Ποιές κλάσεις έχετε υλοποιήσει στον κώδικά σας
2. Ποιά κληρονομικότητα έχετε υλοποιήσει (Π.χ. Ποιά η βασική κλάση και ποιές οι παράγωγες που έχετε υλοποιήσει)
3. Ποιά επέκταση μεθόδων έχετε υλοποιήσει (π.χ. Ποιές μέθοδοι επεκτείνουν την ανώτερη στις κλάσεις που δημιουργήσατε)
4. Αν έχετε εφαρμόσει υπερφόρτωση τελεστών ή χρησιμοποιήσατε decorators και σε ποιό σημείο του κώδικα *(θα ληφθεί θετικά στην αξιολόγηση της συνολικής επίδοσής σας στο μάθημα)*
5. Σε ποιά δομή (λίστα ή λεξικό-dictionary) οργανώνει η εφαρμογή σας τις λέξεις της γλώσσας στη διάρκεια του παιχνιδιού
6. Ποιόν αλγόριθμο (ή αλγόριθμους) υλοποιήσατε για να παίξει ο Η/Υ

Κριτήρια Αξιολόγησης 1/2

- *Η αξιολόγηση της εργασίας θα γίνει με βάση τα παρακάτω κριτήρια*

ΚΡΙΤΗ-ΡΙΟ	ΧΑΡΑΚΤΗΡΙ-ΣΤΙΚΟ	ΣΧΟΛΙΑ - ΕΞΗΓΗΣΕΙΣ
1	ΚΛΑΣΕΙΣ	Αν έχουν υλοποιηθεί κλάσεις, κληρονομικότητα, επέκταση μεθόδων. Αν οι κλάσεις Game, Human & Computer συνεργάζονται σωστά ώστε να παιχτεί το παιχνίδι. *Η υπερφόρτωση τελεστών και η χρήση decorators δεν είναι υποχρεωτική αλλά αν υλοποιηθεί θα ληφθεί υπόψην θετικά στη συνολική βαθμολογία.*
2	ΔΟΜΗ Λέξεων της γλώσσας	Εξηγήστε ποια είναι η δομή που χρησιμοποιήσατε για να διαχειριστείτε τις λέξεις της γλώσσας στον κώδικά σας *- (εξηγήστε το στο docstring)*
3	MODULE	Ο κώδικας των κλάσεων να βρίσκεται σε **ένα** ιδιαίτερο αρχείο βιβλιοθήκη-module (ξεχωριστό αρχείο) όπως έχει περιγραφεί.
4	ΑΛΓΟΡΙΘΜΟΣ Η/Υ	Ποιος ή ποιοί αλγόριθμοι έχουν υλοποιηθεί για το παίξιμο (play) του Η/Υ *- (εξηγήστε το στο docstring)*

- ***Συνέχεια κριτηρίων αξιολόγησης …*** →

Κριτήρια Αξιολόγησης 2/2

ΚΡΙΤΗ-ΡΙΟ	ΧΑΡΑΚΤΗΡΙ-ΣΤΙΚΟ	ΣΧΟΛΙΑ - ΕΞΗΓΗΣΕΙΣ
5	**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 προβλεπόμενων λειτουργιών.
6	**ΤΕΚΜΗΡΙΩΣΗ (docstring)**	Να περιλαμβάνεται στο docstring κάθε απαραίτητη πληροφορία όπως περιγράφηκε σε προηγουμένη διαφάνεια *(ή και οτιδήποτε άλλο είναι σημαντικό για την κατανόηση της λειτουργίας του κώδικά σας)*
7	**ΑΡΧΕΙΑ**	Αν έχουν υποβληθεί **τα 2 αρχεία** που είναι απαραίτητα για να εκτελείται σωστά ο κώδικάς σας Δηλαδή, ο κώδικας των κλάσεων και ο κώδικας του κύριου προγράμματος (ή και άλλα αρχεία αν χρειάζεται από τη λογική της εφαρμογής σας, π.χ. αν χρειάζεται αρχείο με αρχικές ρυθμίσεις, κλπ.) *>>> <u>ΔΕΝ</u> χρειάζεται να υποβάλλετε το αρχείο greek7.txt*

(9) Παραδοτέο

Μορφή Παραδοτέου

- Παραδοτέο της εργασίας σας είναι **1 zip ή rar αρχείο που περιλαμβάνει:**

- (α) Το αρχείο **classes.py** που περιλαμβάνει τις **κλάσεις** της εργασίας σας
- (β) Το αρχείο **main-AEM.py** που περιλαμβάνει το **κύριο πρόγραμμα**
  - Πχ. main1234.py ή main12343005.py σε περίπτωση 2 συνεργατών με ΑΕΜ 1234 και 3005 (τα ΑΕΜ συνεχόμενα)
- (γ) Κάθε άλλο **απαραίτητο αρχείο** όπως εσείς θεωρείτε σωστό για την εκτέλεση του κώδικα (π.χ. αν χρειάζεται κάποιο αρχείο τύπου json) *(προσοχή: ΟΧΙ το greek7.txt)*

- >>> Τα παραπάνω αρχεία τα **συμπιέζετε σε αρχείο zip ή rar**
- >>> Ονομάζετε το συμπιεσμένο αρχείο: **PythonScrabbleAEM** (ή βάζετε ΑΕΜ1ΑΕΜ2 αν είστε 2 συνεργάτες)
- **Υποβάλετε μέσω elearning** μέχρι την **προθεσμία** που θα ανακοινωθεί στην εξεταστική που θέλετε να παραδώσετε την εργασία

Υπενθύμιση για τις εξετάσεις του μαθήματος

- Το μάθημα έχει **3 εξετάσεις**:
- **E1**: Εργασία ChatGPT 20%
- **E2**: Εργασία (Project) Python Scrabble 30%
- **Γραπτά** 50%

- Μπορείτε να "**μοιράσετε**" τις εξετάσεις όπως θέλετε στις εξεταστικές: Ιουνίου & Σεπτεμβρίου 2024 (και Φεβρουαρίου 2025 αν είστε επί πτυχίω)

- *Προσοχή*: Εφόσον χρωστάτε το μάθημα και ξαναδιδαχθεί το εαρινό 2025 υποχρεούστε σε συνολική επανεξέταση (δεν κρατούνται βαθμοί μετά τον Φεβρουάριο 2025)

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)
- Βιβλιοθήκη random για διαχείριση ψευδοτυχαίων αριθμών