



Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών  
Πολυτεχνείο Κρήτης  
Ακαδημαϊκό Έτος 2022-2023 (Χειμερινό Εξάμηνο)  
Μάθημα: Εργαλεία Ανάπτυξης Λογισμικού &  
Προγραμματισμός Συστημάτων

Διδάσκων: Νίκος Γιατράκος

Εργασία 1 (20% της συνολικής βαθμολογίας του μαθήματος)  
Ομάδες 1 έως και 5 ατόμων

Θέμα: Εργαλεία για την Ανάπτυξη Λογισμικού Βασικού  
Twitter Viewer & Editor

### Σκοπός Εργασίας

Σκοπός της εργασίας είναι η εξοικειώσή σας με τα εργαλεία και τη διαδικασία ανάπτυξης λογισμικού σε Python και η συνεργασία σας σε ομάδες προγραμματιστών. Προκειμένου να επιτευχθεί αυτό, θα αναπτύξετε σε ομάδες, μικρής κλίμακας λογισμικό που θα αφορά έναν Twitter Viewer & Editor - περιορισμένων δυνατοτήτων - με τη χρήση της γλώσσας Python και των εργαλείων Logging, Profiling, Refactoring και Unit Testing που εξετάζουμε στο μάθημα.

### Σενάριο Εφαρμογής και Περιγραφή Συνόλου Δεδομένων (Dataset)

Έχουμε ένα σύνολο δεδομένων από tweets διαφόρων χρηστών, που αφορούν το US Presidential Election 2012. Μέρος από το αντίστοιχο dataset, μεγέθους περίπου 1GB, έχει αναρτηθεί στο ακόλουθο Dropbox link προκειμένου να το χρησιμοποιήσετε στην εργασία σας (βρείτε το download button πάνω αριστερά στο browser):



Download



Save a copy to Dropbox

<https://www.dropbox.com/s/qhxy6tcm16545n4/tweetdhead300000.json?dl=0>

Το αρχείο, το οποίο συγκεντρώθηκε χρησιμοποιώντας παλιότερη έκδοση του [Twitter API](#), περιέχει ~300000 tweets χρηστών σε μορφή json<sup>1</sup>. Κάθε tweet καταλαμβάνει μια γραμμή και μας ενδιαφέρουν ΜΟΝΟ τα πεδία "text" και "created at" από κάθε tweet. Επίσης, για κάθε tweet αγνοούμε οποιοδήποτε προσδιοριστή (id) του δίνεται εντός του json αρχείου και θεωρούμε ότι ο προσδιοριστής του είναι ο αριθμός γραμμής στην οποία εμφανίζεται.

<sup>1</sup> Αλλά θα μπορούσε να περιέχει περισσότερα ή λιγότερα. Ο κώδικας της εργασίας σας θα πρέπει να μπορεί να δουλεύει ακόμη και αν δεν ξέρουμε τον αριθμό των tweets. Ο σχετικά μεγάλος όγκος των tweets (~1GB) είναι για να σας βοηθήσει να καταλάβετε καλύτερα στο profiling τι είναι γρήγορο και τί όχι ή τι είναι memory efficient και τι όχι. Μπορείτε να δουλέψετε και με υποσύνολο του dataset.

Θέλουμε να φτιάξουμε ένα **Twitter Viewer & Editor** που θα δίνει στο χρήστη του τη δυνατότητα να εκτελέσει τις ακόλουθες εντολές:

Εντολή	Περιγραφή
<b>c</b>	Create tweet by giving its "text" (derive "created at" by system date time). Every created tweet is appended at the end of the rest (in memory and, if saved, in the twitter file). Sets current tweet ID to be the created one.
<b>r &lt;number&gt;</b>	Read the tweet at line/with ID <number>. Sets current tweet ID to be the read one.
<b>u &lt;number&gt;</b>	Update the tweet at line/with ID <number>, by giving its new "text" (derive "created at" by system date time). Sets current tweet ID to be the updated one.
<b>d</b>	Delete current tweet (Hint: adjust tweet IDs when needed).
<b>\$</b>	Read the last tweet in the file. Sets current tweet ID to be the last one.
<b>-</b>	Read one tweet up from the current tweet. Updates current tweet ID accordingly.
<b>+</b>	Read one tweet down from your current tweet. Updates current tweet ID accordingly.
<b>=</b>	Print current tweet ID.
<b>q</b>	Quit without save.
<b>w</b>	(Over)write file to disk
<b>x</b>	Exit and save.

Θα πρέπει να κρατάτε με κάποιο τρόπο ποιο είναι το **current tweet ID** και **ο χρήστης θα μπορεί να αλλάζει με βάση τις παραπάνω εντολές το current tweet ID**.

Είναι σημαντικό να σημειωθεί πως όταν θέλουμε να αλλάξουμε (να κάνουμε update) κάποιο tweet, διαγράφουμε την παλιά έκδοση και δημιουργούμε νέα έκδοση της μορφής: `{"text": "...here goes the text...", "created at": "...here goes datetime..."}` εγγράφουμε τη νέα έκδοση στη σωστή γραμμή/θέση βάσει του ID/γραμμή που βρισκόταν το tweet που αλλάξαμε (στη μνήμη ή αν τα σώσουμε τελικά στο αρχείο).

---

### Παραδοτέα και Υποβολή

---

- Ένας από την ομάδα σας, θα κατεβάσει και θα αποθηκεύσει το αρχείο PLH211\_Project1\_2022\_2023.ipynb στον υπολογιστή του από την ενότητα Εργασίες του eclass του μαθήματος: <https://www.eclass.tuc.gr/modules/work/?course=HMMY120>
- Πρώτη του δουλειά μετά από αυτό θα είναι να ανεβάσει το PLH211\_Project1\_2022\_2023.ipynb στο δικό του Colab και **\*\*\*ΝΑ ΤΟ ΜΕΤΟΝΟΜΑΣΕΙ ΟΠΩΣΔΗΠΟΤΕ\*\*\*** με τους Αριθμούς Μητρώου των μελών της ομάδας σας χωρισμένους με underscore. Αν για παράδειγμα η ομάδα σας έχει το Φοιτητή 1 με AM:11111, το Φοιτητή 2 με AM:22222 και το Φοιτητή 3 με AM:33333, το αρχείο σας θα πρέπει να μετονομαστεί σε 11111\_22222\_33333.ipynb
- Στη συνέχεια το μέλος της ομάδας θα κάνει share από το Google Drive του (εκεί αποθηκεύονται έτσι κι αλλιώς by default τα Colab Notebooks) με τα υπόλοιπα μέλη της ομάδας σας. Κάθε μέλος μπορεί να δημιουργήσει copy του .ipynb αρχείου αυτού αν θέλει.
- Αφού συνεννοηθείτε μεταξύ σας, χωρίς διαμεσολάβηση του διδάσκοντα, θα σχηματίσετε ομάδες. Έπειτα, έως την 15/11/2022 θα πρέπει να εγγραφείτε στις κενές

ομάδες που έχω δημιουργήσει στην αντίστοιχη περιοχή του eclass (<https://www.eclass.tuc.gr/modules/group/index.php?course=HMMY120&urlview=1>).

Έως την 15/11/2022 θα μπορείτε να εγγραφείτε και να απεγγραφείτε σε ομάδες. Έπειτα από την 15/11/2022 οι ομάδες θα κλείσουν και θα ΠΡΕΠΕΙ να παραμείνουν έτσι και για τα 2 project του μαθήματος. ΜΗΝ εγγραφείτε σε τυχαίες ομάδες και ΜΗΝ εγγραφείτε σε ομάδα αν δεν έχετε πρώτα επικοινωνήσει και συμφωνήσει με τα υπάρχοντα μέλη της. Αν δε βρίσκετε ομάδα, μπορείτε να εκπονήσετε την εργασία μόνοι σας.

- Όταν τελειώσετε την εργασία σας, κάθε ομάδα θα ανεβάσει ΤΟ ΕΝΑ ΚΑΙ ΤΕΛΙΚΟ .ipynb αρχείο στο eclass. Αυτό είναι το παραδοτέο σας. Όλος ο κώδικας και η τεκμηρίωση της εργασίας σας θα πρέπει να είναι σε code cells και text cells αυτού του .ipynb αρχείου. Το .ipynb αρχείο είναι δομημένο ώστε να λειτουργεί ως template για τη δομή των απαντήσεών σας. Ακολουθήστε αυτό το template.
- Στο Colab θα είστε έτοιμοι να δουλέψετε όλοι μαζί επί της εργασίας σας, αλλά προσοχή, όχι ταυτόχρονα. Το Colab δεν είναι Git ή Github (θα τα γνωρίσουμε στη συνέχεια στο μάθημα). Δεν μπορείτε να δουλεύετε όλοι μαζί ταυτόχρονα στο ίδιο .ipynb αρχείο (θα σώζει τις αλλαγές ενός, στους άλλους θα δίνει warnings ότι κάποιος άλλος άλλαξε το αρχείο).

---

## Υλοποίηση

---

Το πρόγραμμά σας θα ξεκινάει και θα έχει μέσα σε ένα white True loop ένα μήνυμα που θα τυπώνει τον πίνακα με τις πιθανές εντολές που μπορεί να δώσει ο χρήστης.

Όταν ο χρήστης πληκτρολογήσει μια εντολή θα εκτελείται αντίστοιχη μέθοδος στον κώδικα που θα υλοποιεί την αντίστοιχη εντολή. Για παράδειγμα για την εντολή «\$» που φαίνεται στον πίνακα, θα υπάρχει μέθοδος που θα φροντίζει να πηγαίνει στο τελευταίο tweet, να το διαβάσει και να το δείχνει στην οθόνη. Εσωτερικά θα πρέπει να βάζει μια μεταβλητή currentTweet=(ποιος αριθμός) για να ξέρουμε που είμαστε.

Η διαδικασία ανάπτυξης λογισμικού περιλαμβάνει τις εξής φάσεις:

- **Development Phase** (30% του βαθμού της παρούσας εργασίας): Σε 1 code cell όλος ο κώδικας του προγράμματος σας όταν τελειώσετε την πρώτη φάση της ανάπτυξής του, αλλά πριν το logging, profiling, refactoring και unit testing.
- **Logging Phase** (10% του βαθμού της παρούσας εργασίας). Κατά τη χρήση του Twitter Viewer & Editor σας, θα πρέπει να καταγράφονται σε ένα αρχείο logme.txt οι ενέργειες/εντολές που χρησιμοποίησε ο τελευταίος χρήστης που το έτρεξε. Θέλουμε να logάρουμε INFO, κατά βάση, μέσω κατάλληλου conf αρχείου στη μορφή (formatter) `format=%(asctime)s - %(name)s - %(levelname)s - %(message)s` Όπου `levelname` info. Σε ένα code cell θα βάλετε τα περιεχόμενα του .conf αρχείου. Σε 1 άλλο code cell θα βάλετε τον κώδικα του προγράμματος σας, όταν τελειώσετε το Logging Phase, αλλά πριν το profiling, refactoring και unit testing.
- **Profiling Phase** (20% του βαθμού της παρούσας εργασίας): Σε 1 ή περισσότερα code cell θα βάλετε το version του κώδικα στο οποίο βάλατε εντολές για το profiling του προγράμματος σας, πριν το refactoring και το unit testing.
- **Refactoring Phase** (20% του βαθμού της παρούσας εργασίας): Σε 1 code cell όλο τον τελικό κώδικα του προγράμματος σας μετά το refactoring αλλά πριν το unit testing.

Υπόδειξη: Πέραν της *Idiomatic Python* που είδαμε στο μάθημα στην ενότητα του *refactoring*, στο *Refactoring Phase* εξετάστε και βελτιστοποιήστε το πως χρειάζεται να φορτωθούν τα δεδομένα σας. Αποφάσεις:

- ο Θα φορτώνονται σε *dictionary*; Συμφέρει να μετατραπούν σε *lists* ή *tuples*; *Lists of tuples*; *Tuples of lists*;
  - ο Πως φορτώνουμε τα *tweets* ή/και πλοηγούμαστε σε αυτά; Φορτώνουμε τα πάντα στη μνήμη εξ αρχής και χρησιμοποιούμε *for loops*; Ή κάθε φορά που μας ζητείται ένα *tweet* το ψάχνουμε στο αρχείο και μόνο τότε το φορτώνουμε στη μνήμη;
  - ο Χρησιμοποιούμε *comprehensions*; Χρησιμοποιούμε *generators*; ]
- **Unit Testing Phase** (20% του βαθμού της παρούσας εργασίας): Σε 1 code cell θα βάλετε όλον τον test κώδικα του unit testing και σε 1 code cell όλο τον κώδικα του προγράμματος σας μετά το unit testing (π.χ. επειδή το testing έδειξε πράγματα που έπρεπε να διορθωθούν).

Ο κώδικάς σας αρκεί να είναι καλά δομημένος, δε χρειάζεται να είναι απαραίτητα αντικειμενοστρεφής.

---

### Τρόπος Βαθμολόγησης

---

Η εργασία και το μάθημα αφορούν την ανάπτυξη λογισμικού. Η έμφαση στη βαθμολόγηση, λοιπόν, είναι σε σωστό (Development, Logging και Unit Testing Phase) και αποδοτικό (Logging, Profiling και Refactoring) κώδικα. Ως εκ τούτου:

Ο κώδικας των ομάδων ελέγχεται με λογισμικού εντοπισμού αντιγραφών. Ομάδες που φαίνεται να έχουν αντιγράψει οποιοδήποτε μέρος των φάσεων και του κώδικα της εργασίας μηδενίζονται (όλα τα μέλη και των δύο ομάδων - ανεξαρτήτως ποιος αντέγραψε από ποιόν).

Κώδικας που δεν κάνει compile, δίνει runtime errors ή δεν είναι σωστός ως προς τη λειτουργικότητα που ζητείται να αποδοθεί, βαθμολογείται με έως μηδενικό βαθμό ακόμη και αν η τεκμηρίωση και ο σχολιασμός του είναι αkéραια.

Όλες οι άλλες περιπτώσεις (ελλιπής αλλά σωστή λειτουργικότητα, ελλιπής αλλά καταληπτή τεκμηρίωση κλπ) που δεν εμπίπτουν παραπάνω, αξιολογούνται ξεχωριστά ανάλογα με την περίπτωση.

Στο .ipynb που θα παραδώσετε θα αναφέρεται ακριβώς ποια μέλη της ομάδας δούλεψαν σε κάθε φάση (ζητείται μαζί με τα ΑΜ και τα ονόματα των φοιτητών στο .ipynb). Ο στόχος της εργασίας και ο λόγος που η ομάδα έχει περισσότερα μέλη είναι να δουλέψετε συνεργατικά, αλλά ανεξάρτητα στις φάσεις ανάπτυξης. ΔΕΝ θα πρέπει να δουλέψετε όλοι σε όλα, αλλά συγκεκριμένα μέλη της ομάδας θα αναλάβουν κάθε φάση. Το αντίθετο ΔΕΝ προσομοιώνει ένα ρεαλιστικό περιβάλλον software development και επηρεάζει αρνητικά τη βαθμολογία όλης της ομάδας (εκτός αν κάποιος κάνει την εργασία μόνος του). Για παράδειγμα, σε μια ομάδα 5 ατόμων, η διαδικασία ανάπτυξης του λογισμικού σας θα μπορεί να γίνει, ενδεικτικά, κάπως έτσι:

- 2 μέλη της ομάδας θα αναλάβουν από κοινού το development και το logging phase και θα συμπληρώσουν όσα ζητούνται σε αυτές τις φάσεις στο .ipynb αρχείο και θα το παραδώσουν στα άλλα μέλη της ομάδας.
- 1 άλλο μέλος της ομάδας θα πάρει τον κώδικα των προηγούμενων φάσεων και θα

κάνει το profiling. Θα συμπληρώσει όσα ζητούνται σε αυτή τη φάση στο .ipynb αρχείο και θα το παραδώσει στα άλλα μέλη της ομάδας καταγράφοντας τα συμπεράσματα του. Τι αργεί; Τι μπορεί να αλλάξω για να κάνω τον κώδικα πιο γρήγορο και να καταναλώνει λιγότερη μνήμη;

- 1 άλλο μέλος της ομάδας θα πάρει τον κώδικα των προηγούμενων φάσεων και θα κάνει το refactoring βάσει των αποτελεσμάτων του profiling. Θα συμπληρώσει όσα ζητούνται σε αυτή τη φάση στο .ipynb αρχείο και θα το παραδώσει στα άλλα μέλη της ομάδας.
- 1 άλλο μέλος της ομάδας κάνει το τελικό unit testing των μεθόδων. Θα συμπληρώσει όσα ζητούνται σε αυτή τη φάση στο .ipynb αρχείο και θα το παραδώσει σε όλα τα μέλη της ομάδας.
- Όλη η ομάδα ελέγχει και σχολιάζει, βελτιώνει την τελική εργασία. Ακολουθεί η τελική υποβολή.

Κατά την προφορική εξέταση κάθε φοιτητής θα ερωτηθεί επί όσων ανέφερε ότι υλοποίησε ο ίδιος στην εργασία. Μειωμένη συμμετοχή στην ανάπτυξη της εργασίας ή αδυναμία απάντησης στις ερωτήσεις επηρεάζει αρνητικά τη βαθμολόγηση.

---

#### Διαδικαστικά

---

- **Αριθμός Μελών Ομάδας Εργασίας:** 1 έως 5 άτομα και οι ομάδες ΔΕ θα πρέπει να αλλάξουν ανάμεσα στην πρώτη και τη δεύτερη εργασία.
- **Ημερομηνία Παράδοσης Εργασίας:** έως τα μεσάνυχτα της 30/11/2022
- **Τρόπος Παράδοσης:** Ομαδική υποβολή μέσω eclass σε ένα .ipynb αρχείο με κατάλληλο όνομα όπως περιγράφεται παραπάνω.
- **Ημερομηνία Προφορικής Εξέτασης:** Κατόπιν ανακοίνωσης, θα εξεταστείτε ΚΑΙ επί των 2 εργασιών που υλοποιήσατε, εκτελώντας demo λειτουργίας της υλοποίησης των εργασιών σας και του κώδικά σας. Για το λόγο αυτό **οι ομάδες ΔΕ μπορούν να αλλάξουν από την 1<sup>η</sup> στη 2<sup>η</sup> εργασία.**
- Η εργασία ισχύει για την εξεταστική Ιανουαρίου και Σεπτεμβρίου (δηλαδή δεν υπάρχει επαν-υποβολή – βελτίωση για το Σεπτέμβριο). Δεν κρατείται βαθμός εργασιών για το επόμενο ακαδημαϊκό έτος.

---

#### Ερωτήσεις/Απορίες Σχετικά με την Εργασία

---

Για τις ερωτήσεις - απορίες σας μπορείτε

- να χρησιμοποιείτε την αντίστοιχη κατηγορία στην περιοχή συζητήσεων του μαθήματος στο eclass:  
<https://www.eclass.tuc.gr/modules/forum/viewforum.php?course=HMMY120&forum=39414>
- να απευθύνεστε στο διδάσκοντα μέσω email:
  - Θέμα Email : ProjectPLH211 – ONOMATEΠΩΝΥΜΟ & ΑΜ φοιτητή
  - Παραλήπτης: **ngiatrakos@tuc.gr**

---

Καλή Δουλειά!