

Ε.Α.Π./ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

3η ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ

ΑΚΑΔΗΜΑΪΚΟΥ ΕΤΟΥΣ 2022-2023

28/01/2023

Ημερομηνία παράδοσης εργασίας: Κυριακή 05/03/2023

Καταληκτική ημερομηνία παραλαβής: Τετάρτη ¹ 08/03/2023

Ημερομηνία ανάρτησης ενδεικτικών λύσεων: Σάββατο 11/03/2023

Καταληκτική ημερομηνία αποστολής σχολίων στον φοιτητή: Κυριακή 26/03/2023

ΥΠΟΕΡΓΑΣΙΑ 1.

Δημιουργία κλάσεων

(βαθμοί 25)

ΥΠΟΕΡΓΑΣΙΑ 2.

Μέθοδοι οριζόμενες από τον χρήστη

(βαθμοί 25)

ΥΠΟΕΡΓΑΣΙΑ 3.

Δημιουργία αντικειμένων από κλάσεις βιβλιοθήκης

(βαθμοί 25)

ΥΠΟΕΡΓΑΣΙΑ 4.

Κληρονομικότητα

(βαθμοί 25)

ΣΥΝΟΛΟ

(βαθμοί 100)

¹ Σύμφωνα με τον Κανονισμό Σπουδών, η καταληκτική ημερομηνία για την παραλαβή της Γ.Ε. από το μέλος ΣΕΠ είναι η επόμενη Τετάρτη από το τέλος της εβδομάδας παράδοσης Γ.Ε.

ΥΠΟΕΡΓΑΣΙΑ 1.

(βαθμοί 25)

Βιβλιοπωλείο: Ζητείται να συμπληρώσετε το πρόγραμμα του αρχείου `l_code_template.py` το οποίο περιέχει δύο κλάσεις, την κλάση `Book()` που περιέχει αντικείμενα με τα στοιχεία των βιβλίων και την κλάση `BookStore()` που διαχειρίζεται τη διεπαφή με τον χρήστη η οποία επιτρέπει την εισαγωγή / αναζήτηση / διαγραφή / παρουσίαση των στοιχείων των βιβλίων.

α) Ζητείται να συμπληρωθεί η μέθοδος αρχικοποίησης της κλάσης `Book()`, καθώς και η μέθοδος εκτύπωσης του κάθε αντικειμένου (βιβλίου) με όλα τα στοιχεία του αξιοποιώντας την ειδική συνάρτηση `__repr__()` ή `__str__()`.

β) Ζητείται να συμπληρωθεί η μέθοδος αρχικοποίησης της κλάσης `BookStore()` καθώς και η μέθοδος εκτύπωσης του κάθε αντικειμένου (βιβλιοπωλείου) ώστε να παρουσιάζονται: *i)* το πλήθος των βιβλίων, *ii)* όλα τα βιβλία με τα στοιχεία τους, αξιοποιώντας την ειδική συνάρτηση `__repr__()` ή `__str__()`.

γ) Ζητείται να συμπληρωθεί η μέθοδος προσθήκης νέου βιβλίου (`addBook()`) και η μέθοδος διαγραφής βιβλίου (`deleteBook()`) της κλάσης `BookStore()`. Κάθε βιβλίο θα είναι ένα στοιχείο μιας λίστας. Για τη διαγραφή του βιβλίου το πρόγραμμα θα ζητάει από τον χρήστη το ISBN του βιβλίου.

δ) Ζητείται να συμπληρωθεί η μέθοδος αναζήτησης βιβλίων (`searchBooksByAuthor()`) της κλάσης `BookStore()`. Η μέθοδος `searchBooksByAuthor()` θα επιστρέφει μια λίστα με βιβλία που έχουν στη συγγραφική ομάδα τον συγκεκριμένο συγγραφέα.

Προσοχή! Να εφαρμοστεί αμυντικός προγραμματισμός σε όλα τα επίπεδα του προγράμματος. Για την εισαγωγή ενός νέου βιβλίου θα πρέπει να χρησιμοποιείται η μέθοδος `addBook()` της κλάσης `BookStore()`, στην οποία θα υπάρχουν 5 ορίσματα που θα περιγράφουν το κάθε βιβλίο. Ο τίτλος του βιβλίου, το όνομα του συγγραφέα και ο αριθμός ISBN θα είναι συμβολοσειρές, το έτος κυκλοφορίας θα είναι ακέραιος αριθμός και το κόστος του βιβλίου θα είναι πραγματικός αριθμός. Για την καταμέτρηση του πλήθους των βιβλίων θα πρέπει να χρησιμοποιηθεί μεταβλητή κλάσης.

Το πρόγραμμα θα πρέπει να ακολουθεί τα βήματα του template. Ακολουθεί παράδειγμα εκτέλεσης.

0) Καταχώρηση Βιβλίων

Βιβλίο: Python Crash Course, Συγγραφέας: Eric Matthews, Έτος: 2016, Τιμή: 27.95€, ISBN: 1593279280 . (Επιτυχής προσθήκη !)

Βιβλίο: Learning Python , Συγγραφέας: Mark Lutz, Έτος: 2021, Τιμή: 40.29€, ISBN: 1449355730 . (Επιτυχής προσθήκη !)

Βιβλίο: Head First Python, Συγγραφέας: Paul Barry, Έτος: 2017, Τιμή: 36.25€, ISBN: 7519813630 . (Επιτυχής προσθήκη !)

Βιβλίο: Introduction to Machine Learning with Python, Συγγραφέας: Andreas C. Mulle, Έτος: 2020, Τιμή: 31.99€, ISBN: 1449369413 . (Επιτυχής προσθήκη !)

Βιβλίο: Python for Data Analysis, Συγγραφέας: Wes McKinney, Έτος: 2022, Τιμή: 38.38€, ISBN: 1098104032 . (Επιτυχής προσθήκη !)

Βιβλίο: Deep Learning with Python, Συγγραφέας: Francois Chollet, Έτος: 2017, Τιμή: 30.2€, ISBN: 1617284433 . (Επιτυχής προσθήκη !)

1) Αναζήτηση βιβλίων με το όνομα και επώνυμο συγγραφέα

Πληκτρολογήστε Όνομα και Επώνυμο συγγραφέα: **Mark Lutz**

Τα βιβλία που βρέθηκαν είναι:

Βιβλίο: Learning Python , Συγγραφέας: Mark Lutz, Έτος: 2021, Τιμή: 40.29€, ISBN: 1449355730

2) Διαγραφή βιβλίου με βάση το ISBN

Πληκτρολογήστε ISBN: **1098104032**

Το Βιβλίο Βιβλίο: Python for Data Analysis, Συγγραφέας: Wes McKinney, Έτος: 2022, Τιμή: 38.38€, ISBN: 1098104032 διαγράφηκε.

3) Εκτύπωση όλων των διαθέσιμων βιβλίων με όλη την σχετική πληροφορία

Πλήθος βιβλίων: 5.

Βιβλίο: Python Crash Course, Συγγραφέας: Eric Matthews, Έτος: 2016, Τιμή: 27.95€, ISBN: 1593279280

Βιβλίο: Learning Python , Συγγραφέας: Mark Lutz, Έτος: 2021, Τιμή: 40.29€, ISBN: 1449355730

Βιβλίο: Head First Python, Συγγραφέας: Paul Barry, Έτος: 2017, Τιμή: 36.25€, ISBN: 7519813630

Βιβλίο: Introduction to Machine Learning with Python, Συγγραφέας: Andreas C. Mulle, Έτος: 2020, Τιμή: 31.99€, ISBN: 1449369413

Βιβλίο: Deep Learning with Python, Συγγραφέας: Francois Chollet, Έτος: 2017, Τιμή: 30.2€, ISBN: 1617284433

ΥΠΟΕΡΓΑΣΙΑ 2.

(βαθμοί 25)

CCTV Camera: Το πρόγραμμα του αρχείου `2_code_template.py` περιγράφει τον τρόπο χειρισμού του εξοπλισμού επιτήρησης (Cameras ασφαλείας) που υπάρχουν σε ένα Control Room. Συγκεκριμένα περιλαμβάνονται:

1. Η κλάση `Camera()` η οποία με τη δημιουργία μεθόδων διαχειρίζεται και παραμετροποιεί τη λειτουργία της Camera, με σκοπό την διεπαφή με τον χρήστη. Αναλυτικότερα:

- a. Η μέθοδος `def __init__(self, location):` αρχικοποιεί το αντικείμενο Camera.
- b. Η μέθοδος `def power(self):` ενεργοποιεί ή απενεργοποιεί την Camera.
- c. Η μέθοδος `def zoomUp(self):` αυξάνει το zoom της Camera.
- d. Η μέθοδος `def zoomDown(self):` μειώνει το zoom της Camera.
- e. Η μέθοδος `def nightVision(self):` ενεργοποιεί ή απενεργοποιεί τη λειτουργία νυχτερινής όρασης της Camera.
- f. Η μέθοδος `def __str__(self):` αποτυπώνει τις συγκεντρωτικές πληροφορίες για όλες τις λειτουργίες της Camera.

2. Η κλάση `Panel()` η οποία με τις αντίστοιχες μεθόδους δημιουργεί και διαχειρίζεται ως πίνακας ελέγχου τα αντικείμενα τύπου Camera που εντάσσονται στο Control Room. Αναλυτικότερα:

- a. Η μέθοδος `def __init__(self):` δημιουργεί τα αντικείμενα τύπου Camera.
- b. Η μέθοδος `def select_camera(self):` επιτρέπει στον χρήστη να επιλέξει την Camera την οποία θέλει να ελέγξει.
- c. Η μέθοδος `def control_panel(self):` επιτρέπει στον χρήστη να παραμετροποιήσει όλες τις προσφερόμενες λειτουργίες της επιλεγμένης Camera (`power`, `zoom`, `night_vision`).

ΖΗΤΕΙΤΑΙ:

α) Να συμπληρώσετε τη μέθοδο αρχικοποίησης `def __init__(self, location):` της κλάσης `Camera()` με όλα τα απαραίτητα στοιχεία για την λειτουργία της.

β) Να συμπληρώσετε τις παρακάτω μεθόδους:

1. `def power(self):` Μέθοδος που ενεργοποιεί ή απενεργοποιεί την Camera.

2. `def zoomUp(self)` : Μέθοδος που αυξάνει το zoom της Camera.
3. `def zoomDown(self)` : Μέθοδος που μειώνει το zoom της Camera.
4. `def nightVision(self)` : Μέθοδος που ενεργοποιεί ή απενεργοποιεί τη λειτουργία νυχτερινής όρασης της Camera.

γ) Να συμπληρώσετε τη μέθοδο αρχικοποίησης `def __init__(self)` : της κλάσης `Panel()` δημιουργώντας μια λίστα με 3 αντικείμενα τύπου `Camera` (πχ Ταμείο, Είσοδος, Αποθήκη).

δ) Να συμπληρώσετε τη μέθοδο `def select_camera(self)` : η οποία επιτρέπει στον χρήστη να επιλέξει την Camera την οποία θα ελέγξει.

Σημαντικό! Για την διευκόλυνση σας στο template περιλαμβάνονται οι παρακάτω μέθοδοι:

- 1) `def __str__(self)` : για την αποτύπωση των συγκεντρωτικών πληροφοριών για όλες τις λειτουργίες της Camera.
- 2) `def control_panel(self)` : για τον έλεγχο της επιλεγμένης Camera (power, zoom, night_vision)

Προσοχή! Το πρόγραμμα θα πρέπει να ακολουθεί τις κατευθύνσεις του template.

Υπόδειξη! Μπορείτε να χρησιμοποιήσετε για τον ορισμό των ορίων (μέγιστο - ελάχιστο zoom) διαβάθμιση από 0 έως 10.

ΠΑΡΑΔΕΙΓΜΑ ΕΚΤΕΛΕΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ:

Οι κάμερες ασφαλείας είναι:

1. Ταμείο
2. Είσοδος
3. Αποθήκη

Επιλέξτε κάμερα, <return> για έξοδο:1

Πανελ ελέγχου κάμερας Ταμείο

Στοιχεία Κάμερας:

Τοποθεσία: Ταμείο

Κατάσταση κάμερας: απενεργοποιημένη

o(on/off), (z)oom (+/-), (n)(yes/no), <enter>: exit:o on

Πανελ ελέγχου κάμερας Ταμείο

Στοιχεία Κάμερας:

Τοποθεσία: Ταμείο

Κατάσταση κάμερας: σε λειτουργία

zoom: 5

o(on/off), (z)oom (+/-), (n)(yes/no), <enter>: exit:z +

Πανελ ελέγχου κάμερας Ταμείο
Στοιχεία Κάμερας:
Τοποθεσία: Ταμείο
Κατάσταση κάμερας: σε λειτουργία
zoom: 6

o(on/off), (z)oom (+/-), (n)(yes/no), <enter>: exit:n yes

Πανελ ελέγχου κάμερας Ταμείο
Στοιχεία Κάμερας:
Τοποθεσία: Ταμείο
Κατάσταση κάμερας: σε λειτουργία
zoom: 6
(νυχτερινή όραση)

o(on/off), (z)oom (+/-), (n)(yes/no), <enter>: exit:enter

Οι κάμερες ασφαλείας είναι:
1. Ταμείο
2. Είσοδος
3. Αποθήκη
Επιλέξτε κάμερα, <return> για έξοδο:3

Πανελ ελέγχου κάμερας Αποθήκη
Στοιχεία Κάμερας:
Τοποθεσία: Αποθήκη
Κατάσταση κάμερας: απενεργοποιημένη

o(on/off), (z)oom (+/-), (n)(yes/no), <enter>: exit:o on

Πανελ ελέγχου κάμερας Αποθήκη
Στοιχεία Κάμερας:
Τοποθεσία: Αποθήκη
Κατάσταση κάμερας: σε λειτουργία
zoom: 5

o(on/off), (z)oom (+/-), (n)(yes/no), <enter>: exit:z -

Πανελ ελέγχου κάμερας Αποθήκη
Στοιχεία Κάμερας:
Τοποθεσία: Αποθήκη
Κατάσταση κάμερας: σε λειτουργία
zoom: 4

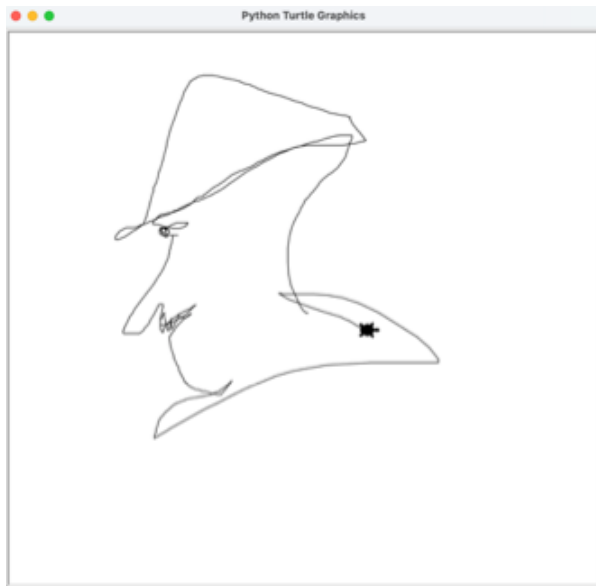
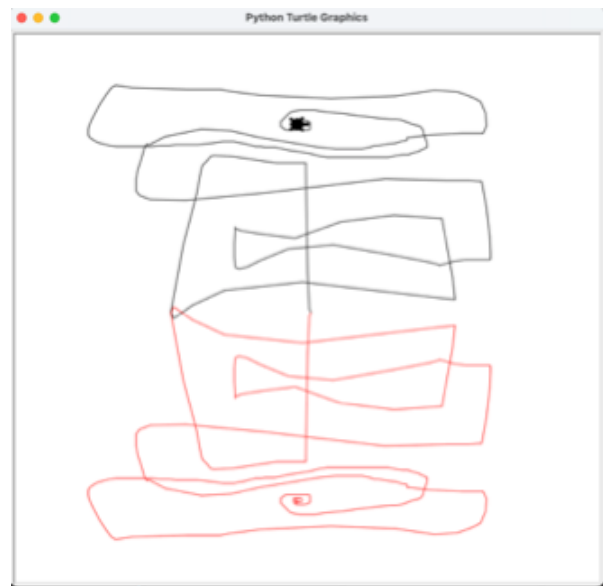
o(on/off), (z)oom (+/-), (n)(yes/no), <enter>: exit:n yes

Πανελ ελέγχου κάμερας Αποθήκη
Στοιχεία Κάμερας:
Τοποθεσία: Αποθήκη
Κατάσταση κάμερας: σε λειτουργία
zoom: 4
(νυχτερινή όραση)

ΥΠΟΕΡΓΑΣΙΑ 3.

(βαθμοί 25)

Δημιουργία αντικειμένων από κλάσεις βιβλιοθήκης.

**Σχήμα (α)****Σχήμα (β)**

Η βιβλιοθήκη `turtle` της Python μας επιτρέπει να σχεδιάσουμε γραφικά πάνω σε ένα καμβά με χρήση μιας γραφίδας (μέθοδος `Turtle()`).

Για παράδειγμα, αν τρέξετε τον κώδικα το αρχείου `3_code_template.py` θα ανοίξει ένα παράθυρο γραφικών με μια γραφίδα (χελωνίτσα) που αν την σύρετε με το ποντίκι, μπορείτε να σχεδιάσετε όπως στο Σχήμα (α). Η γραφίδα υλοποιείται ως κλάση `Pen`.

Όπως θα δείτε στον κώδικα, έχει οριστεί επίσης μια κλάση `Tracer` (Ιχνηλάτης), η οποία παρακολουθεί με κάποια παραμόρφωση την κίνηση της γραφίδας.

Ζητείται να αλλάξετε τον κώδικα, ώστε όταν ο χρήστης κινεί τη γραφίδα να κινούνται οι αντίστοιχοι ιχνηλάτες. Χρησιμοποιήσετε τη δομή `traces` της `Pen` για σύνδεση της γραφίδας με τους ιχνηλάτες.

Πειραματιστείτε με τις παραμέτρους των αντικειμένων τύπου `Tracer`, μελετήστε το ρόλο που παίζουν οι παράμετροι `dx`, `dy`, `a`, `b`, `width`, `color`.

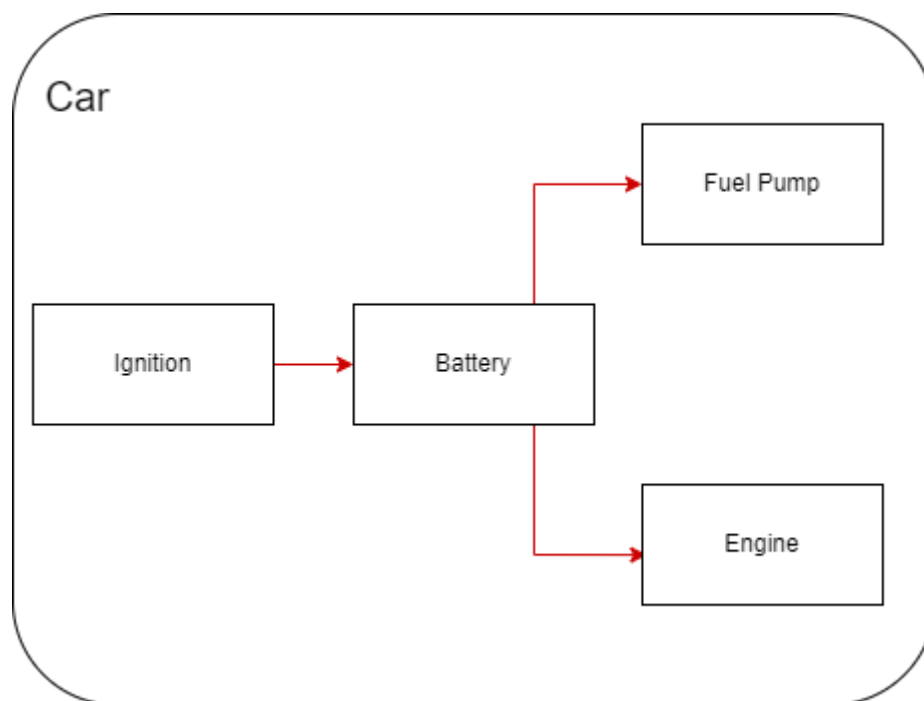
(α). Κάνετε τις κατάλληλες τροποποιήσεις στον κώδικα ώστε η κίνηση της γραφίδας να παράγει συμμετρικά σχέδια προς τον άξονα των `x` - ως προς το σχέδιο της γραφίδας, όπως στο σχήμα (β), περιλάβετε στην απάντησή σας ένα screenshot λειτουργίας του κώδικά σας σε αυτή τη μορφή.

(β). Μετατρέψτε ως σχόλιο την απάντηση στο ερώτημα (α). Αλλάξτε τον κώδικα, ώστε να γίνει χρήση περισσότερων ιχνηλατών με διαφορετικά χαρακτηριστικά, και προσθέστε το screenshot στην εργασία σας.

ΥΠΟΕΡΓΑΣΙΑ 4.

(βαθμοί 25)

Αυτοκίνητο: Το πρόγραμμα του αρχείου `4_code_template.py` περιέχει μία ιεραρχία κλάσεων που περιγράφουν ένα αυτοκίνητο και τα εξαρτήματά του (Components). Για λόγους απλοποίησης, θα επικεντρωθούμε στο σύστημα εκκίνησης του κινητήρα του αυτοκινήτου. Υποθέτουμε ότι ένα αυτοκίνητο αποτελείται από τη μίζα εκκίνησης (Ignition), την μπαταρία (Battery), την αντλία βενζίνης (FuelPump) και τον κινητήρα (Engine). Κάθε εξάρτημα έχει μία τρέχουσα κατάσταση (state) και μπορεί να είναι ενεργό ή ανενεργό. Κάθε εξάρτημα μπορεί να συνδεθεί (μέθοδος `connect()`) με άλλα εξαρτήματα. Όταν ένα εξάρτημα ενεργοποιείται (μέθοδος `start()`) ενεργοποιεί όλα τα εξαρτήματα στα οποία είναι συνδεδεμένο. Όταν ένα εξάρτημα απενεργοποιείται (μέθοδος `stop()`) απενεργοποιεί όλα τα εξαρτήματα στα οποία είναι συνδεδεμένο και στη συνέχεια απενεργοποιείται το ίδιο. Επιπλέον, κάθε εξάρτημα έχει μια μέθοδο (`status()`) που εκτυπώνει την κατάστασή του και καλεί την ίδια μέθοδο για τα συνδεδεμένα εξαρτήματα. Προσοχή! Οι συνδέσεις δεν είναι αμφίδρομες. Για παράδειγμα αν συνδέσουμε τη μίζα στην μπαταρία δεν σημαίνει ότι συνδέεται και η μπαταρία με τη μίζα.



Για το απλοποιημένο σχήμα συνδέσεων του παραπάνω σχήματος, η μπαταρία συνδέεται στη μίζα και η αντλία βενζίνης και ο κινητήρας στην μπαταρία.

ΖΗΤΕΙΤΑΙ:

α) Να συμπληρώσετε τη μέθοδο αρχικοποίησης της κλάσης Car που θα δημιουργεί όλα τα απαραίτητα εξαρτήματα και τις συνδέσεις μεταξύ τους ώστε να μπορεί να λειτουργήσει ο κινητήρας του όταν ενεργοποιηθεί η μίζα.

β) Να συμπληρώσετε τις παρακάτω μεθόδους της κλάσης Component:

- 1) Την μέθοδο `__init__(self)` η οποία πραγματοποιεί αρχικοποίηση της κατάστασης και της δομής αποθήκευσης συνδεδεμένων εξαρτημάτων.
- 2) Την μέθοδο `connect(self, other)` η οποία προσθέτει το εξάρτημα `other` στη δομή αποθήκευσης συνδεδεμένων εξαρτημάτων.
- 3) Την μέθοδο `start(self)` η οποία ενεργοποιεί το εξάρτημα, πραγματοποιεί εκκίνηση των συνδεδεμένων εξαρτημάτων και εκτυπώνει μήνυμα έναρξης του κάθε εξαρτήματος.
- 4) Την μέθοδο `stop(self)` η οποία πραγματοποιεί απενεργοποίηση των συνδεδεμένων εξαρτημάτων και τέλος απενεργοποιεί το εξάρτημα, εμφανίζοντας αντίστοιχο μήνυμα.

γ) Να συμπληρώσετε τις μεθόδους `init` των κλάσεων `Ignition`, `Battery`, `FuelPump` και `Engine` ώστε να έχουν σωστή τιμή τα ονόματα στα διάφορα εξαρτήματα.

Προσοχή! Το πρόγραμμα θα πρέπει να ακολουθεί τις κατευθύνσεις του template.

ΑΠΟΤΕΛΕΣΜΑ ΕΚΤΕΛΕΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ:

```
=====
Κατάσταση: Μίζα εκτός λειτουργίας
Κατάσταση: Μπαταρία εκτός λειτουργίας
Κατάσταση: Αντλία Καυσίμου εκτός λειτουργίας
Κατάσταση: Κινητήρας εκτός λειτουργίας
=====
Εξάρτημα Μίζα: έναρξη λειτουργίας
Εξάρτημα Μπαταρία: έναρξη λειτουργίας
Εξάρτημα Αντλία Καυσίμου: έναρξη λειτουργίας
Εξάρτημα Κινητήρας: έναρξη λειτουργίας
=====
Κατάσταση: Μίζα σε λειτουργία
Κατάσταση: Μπαταρία σε λειτουργία
Κατάσταση: Αντλία Καυσίμου σε λειτουργία
Κατάσταση: Κινητήρας σε λειτουργία
=====
Εξάρτημα Αντλία Καυσίμου: παύση λειτουργίας
Εξάρτημα Κινητήρας: παύση λειτουργίας
Εξάρτημα Μπαταρία: παύση λειτουργίας
Εξάρτημα Μίζα: παύση λειτουργίας
=====
Κατάσταση: Μίζα εκτός λειτουργίας
Κατάσταση: Μπαταρία εκτός λειτουργίας
Κατάσταση: Αντλία Καυσίμου εκτός λειτουργίας
Κατάσταση: Κινητήρας εκτός λειτουργίας
=====
```

Γενικές Υποδείξεις:

I) Για τις απαντήσεις της εργασίας μπορείτε να ανατρέξετε στη συμπληρωματική βιβλιογραφία που δίνεται και στα βοηθητικά κείμενα που υπάρχουν στον δικτυακό τόπο / portal της θεματικής ενότητας. Συνιστάται να προσθέσετε στο τέλος της εργασίας σας κατάλογο βιβλιογραφίας.

II) Οδηγίες σχετικές με τον κώδικα

- Το όνομα κάθε .py αρχείου να περιλαμβάνει το επώνυμό σας με λατινικούς χαρακτήρες, το χαρακτήρα της υπογράμμισης και τον αριθμό του συγκεκριμένου υποερωτήματος (π.χ. αν το επώνυμό σας είναι Γεωργίου, τότε ο κώδικας για την υποεργασία 1β θα έχει το όνομα Georgiou_1b.py). Κάθε αρχείο κώδικα που θα παραδοθεί θα πρέπει τουλάχιστον να περνάει τη φάση της διερμηνείας χωρίς συντακτικά σφάλματα.
- Τα αρχεία .py θα πρέπει να τα ανοίξετε και να τα επεξεργαστείτε με το ολοκληρωμένο περιβάλλον ανάπτυξης κώδικα IDLE. Ο κώδικας να είναι επαρκώς σχολιασμένος, σωστά στοιχισμένος και ενσωματωμένος μέσα στο έγγραφο Word, με τις απαντήσεις σας σε γραμματοσειρά courier.
- Στο έγγραφο της απάντησής σας και στο αρχείο του κώδικα θα πρέπει να δίνεται ολόκληρο το πρόγραμμα, επισημαίνοντας με σχόλια πού απαντάτε κάθε ερώτημα ώστε να θεωρούνται πλήρεις οι απαντήσεις.
- Όλα τα .py αρχεία με τον πηγαίο κώδικα και το .doc αρχείο κειμένου να υποβληθούν μέσω της πλατφόρμας <https://study.eap.gr>.

III) Τρόπος παράδοσης εργασίας:

α) Οι απαντήσεις πρέπει να είναι γραμμένες με χρήση **επεξεργαστή κειμένου** (π.χ. **Word**) σε σελίδες **διαστάσεων Α4**. Το αρχείο να περιέχει ως **πρώτη σελίδα** το κείμενο του **Εντύπου Υποβολής – Αξιολόγησης** και ως δεύτερη σελίδα τον τίτλο «Σχόλια προς τον φοιτητή» (θα συμπληρωθεί από τον καθηγητή σας). Οι απαντήσεις στις υποεργασίες θα αρχίζουν από την τρίτη σελίδα, **χωρίς να επαναλαμβάνονται οι εκφωνήσεις**. Κάθε υποεργασία θα αρχίζει από νέα σελίδα. Για την απάντησή σας θα πρέπει να χρησιμοποιείτε υποχρεωτικά το **Πρότυπο Υποβολής Γραπτής Εργασίας**.

β) Το .doc αρχείο κειμένου να υποβληθεί στη διεύθυνση <https://study.eap.gr> με **όνομα αρχείου το επώνυμό σας με λατινικούς χαρακτήρες και τον Αριθμό Μητρώου σας**, π.χ. Ioannou_82345.

IV) Η καλή παρουσίαση της εργασίας λαμβάνεται υπόψη στην αξιολόγηση της εργασίας.
