**Διδακτική της Πληροφορικής – Εργασία Δ2**

Χρήστος Χριστίδης

ΑΕΜ 3350

[christpc@csd.auth.gr](mailto:christpc@csd.auth.gr)

18/2/2022

Τμήμα Πληροφορικής ΑΠΘ**Εισαγωγή στις συναρτήσεις και στις μεθόδους στο προγραμματιστικό περιβάλλον της Python.**

**Μια μελέτη περίπτωσης για τη Γ’ τάξη ΕΠΑΛ.**

1. ΤΙΤΛΟΣ & ΔΗΜΙΟΥΡΓΟΣ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Η εισαγωγή στην έννοια των μεθόδων και και συναρτήσεων που αφορά τη Γ’ τάξη Επαγγελματικού Λυκείου σε προγραμματιστικό περιβάλλον Python. Μία μελέτη περίπτωσης Γ’ Λυκείου ΕΠΑΛ. Το σενάριο αυτό όπως και ο κώδικας των δραστηριοτήτων του φύλλου εργασίας (εκτός της πρώτης δραστηριότητας) δημιουργήθηκε από τον **Χρήστο Χριστίδη.**

1. ΕΚΤΙΜΩΜΕΝΗ ΔΙΑΡΚΕΙΑ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Προβλέπεται να διαρκέσει συνολικά 1 διδακτική ώρα. Ο χρόνος αυτός σε γενικές γραμμές κρίνεται επαρκής. Αφορά κυρίως την κατανόηση για ανάγκη δημιουργίας μεθόδων ώστε να μην επαναλαμβάνονται όμοια τμήματα κώδικα. Ο εκπαιδευτικός δύναται να επιλέξει το είδος των μεθόδων που θα υλοποιήσει, τροποποιώντας (ανάλογα με τη δυναμική των μαθητών) τις προτεινόμενες δραστηριότητες του φύλλου εργασίας καθώς και να προσθέσει ή να αφαιρέσει μερικές, εφόσον το κρίνει κατάλληλο, ώστε να ανταποκρίνονται στις μαθησιακές ανάγκες και δυνατότητες των μαθητών.

1. ΕΝΤΑΞΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ / ΠΡΟΑΠΑΙΤΟΥΜΕΝΕΣ ΓΝΩΣΕΙΣ

Το διδακτικό σενάριο εντάσσεται στο Αναλυτικό Πρόγραμμα Σπουδών της Γ’ Τάξης ΕΠΑΛ και συγκεκριμένα στο μάθημα Προγραμματισμός Υπολογιστών.

Σύμφωνα με το αναλυτικό πρόγραμμα σπουδών, ο σκοπός των παραγράφων 4.2, 7.1, 7.2 αφορά την συγγραφή συναρτήσεων σε γλώσσα Python και την κατανόηση περί ανάγκης χρήσης τους. Πάνω σε αυτό θα στηριχτεί η διδασκαλία και η κατανόηση του τρόπου που ορίζονται οι συναρτήσεις, τις παραμέτρους που δέχονται και επεξεργάζονται καθώς των λειτουργιών που επιτελούν.

Οι γνώσεις που είναι αναγκαίο να κατέχουν ήδη οι μαθητές είναι η εξοικείωση με τη γλώσσα Python καθώς και γνώση και χειρισμός δομών επιλογής if-elif-else, επανάληψης for και while, η χρήση αθροιστών και συγκριτικών τελεστών, η εξοικείωση με δομές δεδομένων όπως λίστες, ουρές, πίνακες και tuples.

1. ΣΚΟΠΟΙ ΚΑΙ ΣΤΟΧΟΙ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

Το σενάριο έχει ως γενικούς στόχους το να είναι οι μαθητές σε θέση να κατανοούν γενικά τον τρόπο λειτουργίας των μεθόδων.

Με το πέρας της διδακτικής ωρας οι μαθητές θα πρέπει να μπορούν να αναγνωρίζουν υπάρχουσες συναρτήσεις και να αντιλαμβάνονται τον τρόπο με τον οποίο οι συναρτήσεις επεξεργάζονται και επιστρέφουν δεδομένα καθώς και να εργάζονται ομαδικά προς ανάπτυξη κώδικα.

Πιο συγκεκριμένα, οι αντικειμενικοί στόχοι του διδακτικού σεναρίου είναι:

**Τομέας Γνώσεων**

1. Να αναγνωρίζουν τη λειτουργία μίας μεθόδου και τις αναμενόμενες εξόδους της.
2. Να αναγνωρίζουν την εμβέλεια των μεταβλητών και των παραμέτρων των μεθόδων.

**Τομέας Δεξιοτήτων**

1. Να επιδεικνύουν σε ποια επαναλαμβανόμενα τμήματα κώδικα υπάρχει ανάγκη δημιουργίας των παραπάνω μεθόδων.
2. Να είναι σε θέση να τροποποιούν με αφαιρετικό τρόπο υπάρχουσες συναρτήσεις.
3. ΕΠΙΠΕΔΑ ΣΤΟΧΩΝ ΜΑΘΗΣΗΣ (ΤΑΞΙΝΟΜΙΕΣ Bloom & FULLER)

Τα επίπεδα στόχων μάθησης του σεναρίου είναι:

Α) Σύμφωνα με την ταξινομία Bloom: Το μάθημα είναι μέσης δυσκολίας και αφορά τα επίπεδα:

* «Κατανοώ» και «Αναλύω»: μάθηση της σύνταξης των συναρτήσεων σε γλώσσα Python και αναγνώριση των τμημάτων κώδικα που επαναλαμβάνονται και άρα απαιτούν τη δημιουργία των συναρτήσεων αυτών.
* «Εφαρμόζω» και «Δημιουργώ»: Εφαρμογή της χρήσης μεθόδων που εκτελούν απλές εργασίες και στη συνέχεια προχωρημένων, αφαιρετικών συναρτήσεων που εκτελούν πιο σύνθετες εργασίες, όπως για παράδειγμα της διαχείρισης κειμένων.

Β) Σύμφωνα με την ταξινομία Fuller:



Γ) Σύμφωνα με τους Άξονες στόχων μάθησης: Το μάθημα αφορά τον πρώτο άξονα («χειρίζομαι και δημιουργώ με τα εργαλεία των ΤΠΕ») καθώς και τον τρίτο άξονα («διερευνώ, ανακαλύπτω και λύνω προβλήματα με ΤΠΕ»).

1. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΔΙΔΑΚΤΙΚΟΥ ΣΕΝΑΡΙΟΥ

To σενάριο είναι σχεδιασμένο ώστε να υλοποιείται στο σχολικό εργαστήριο εντός μίας διδακτικής ώρας και αποτελείται από ένα φύλλο εργασίας. Οι μαθητές ενδείκνυται να είναι χωρισμένοι σε ομάδες των δύο ή τριών ατόμων και να χρησιμοποιήσουν το προγραμματιστικό περιβάλλον της τελευταίας έκδοσης της γλώσσας Python (Python 3.10) που θα βρίσκεται εγκατεστημένη στους υπολογιστές του εργαστηρίου.

Κατά τη διδακτική ώρα (45 λεπτά) ο καθηγητής θα χωρίσει τους μαθητές ανά ομάδες, θα εξηγήσει την έννοια του υποπρογράμματος και θα παραθέσει τον αντίστοιχο ορισμό της συνάρτησης μέσω διαφανειών που θα προετοιμάσει ο ίδιος και που θα είναι βασισμένες στο σχολικό βιβλίο ή μέσω του ίδιου του σχολικού βιβλίου και γράφοντας σε μαυροπίνακα. Στη συνέχεια θα παρουσιάσει τη σύνταξη των συναρτήσεων σε γλώσσα Python και θα εξηγήσει την εμβέλεια τόσο των παραμέτρων, όσο και των μεταβλητών της συνάρτησης. Θα διαμοιράσει το φύλλο εργασίας (κατόπιν διαχωρισμού των ομάδων) και οι μαθητές θα κληθούν να απαντήσουν σε ερωτήσεις που αφορούν την κατανόηση του τρόπου λειτουργίας των συναρτήσεων, της εμβέλειας των μεταβλητών και των παραμέτρων τους. Επίσης θα κληθούν να εντοπίσουν τμήματα κώδικα που απαιτούν τη δημιουργία μίας συνάρτησης. Αυτό θα γίνει παρατηρώντας έτοιμο κώδικα συναρτήσεων που θα παρέχει ο καθηγητής. Αφού οι μαθητές προτείνουν μία λύση συνεργατικά, στη συνέχεια ο καθηγητής θα παρουσιάσει υλοποίηση μίας συνάρτησης, εισάγοντας παράλληλα την έννοια της αφαίρεσης με πλάγιο τρόπο.

1. ΧΡΗΣΗ Η.Υ. ΚΑΙ ΓΕΝΙΚΑ ΨΗΦΙΑΚΩΝ ΜΕΣΩΝ ΓΙΑ ΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ («προστιθέμενη αξία» και αντίλογος, επιφυλάξεις, προβλήματα)

Το παρόν διδακτικό σενάριο απαιτεί τη χρήση ηλεκτρονικών υπολογιστών οι οποίοι έχουν προεγκατεστημένη την τελευταία έκδοση της γλώσσας Python (η παρούσα είναι η [Python 3.10](https://www.python.org/downloads/release/python-3100/)) καθώς και ένα περιβάλλον ανάπτυξης κώδικα, κατά προτίμηση το open source λογισμικό από το «Φωτόδενδρο» [Geany](http://photodentro.edu.gr/edusoft/r/8531/350?locale=el) (εναλλακτικά το [PyCharm Community Edition](https://www.jetbrains.com/pycharm/download/#section=windows) ή το [Spyder](https://www.spyder-ide.org/)).

Έχοντας ως δεδομένο ότι τα τμήματα δεν ξεπερνούν σε μέγεθος τους 24 μαθητές, κρίνεται ωφέλιμο να χωριστούν οι μαθητές σε 12 ομάδες των δύο ατόμων, επομένως υπάρχει ανάγκη για ύπαρξη 12 Η/Υ.

1. ΠΡΟΒΛΕΨΗ ΔΥΣΚΟΛΙΩΝ ΣΤΟ ΔΙΔΑΚΤΙΚΟ ΣΕΝΑΡΙΟ ΚΑΙ ΠΡΟ-ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΩΝ ΜΑΘΗΤΩΝ

Η έννοια του υποπρογράμματος εμφανίζεται αρχικά αφού οι μαθητές έχουν αποκτήσει σχετική εξοικείωση με δομές επιλογής, επανάληψης, των αθροιστών και της διαχείρισης αρχείων. Επομένως οι προαναπαραστάσεις που ενδέχεται να έχουν σχετικά με τα υποπρογράμματα αφορούν κυρίως τη γενικότερη αντίληψη τους γύρω από την επαναχρησιμοποίηση και την τμηματοποίηση κώδικα. Έτσι, η κατανόηση και η ορθή σύνταξη συναρτήσεων φαίνεται να παρουσιάζει τις εξής δυσκολίες:

* Το ζήτημα της κατασκευής αφαιρετικών συναρτήσεων ώστε να μην εξαρτώνται από το συγκεκριμένο παράδειγμα που θα τους δίνεται αποτελεί ίσως την κορυφαία δυσκολία κατά την κατασκευή υποπρογραμμάτων.
* Οι συναρτήσεις σε γλώσσα Python έχουν τη δυνατότητα να επιστρέφουν πολλά αποτελέσματα με τη μορφή tuples (Mark Lutz). Οι μαθητές ενδέχεται να το αγνοούν αυτό και να καλούν μία συνάρτηση πολλές φορές για την ίδια λειτουργία ενώ μπορούν απλά να τροποποιήσουν τις παραμέτρους και τις εντολές της ώστε να επεξεργάζεται ταυτόχρονα πολλά δεδομένα και να τα επιστρέφει όλα μαζί.
* Οι συναρτήσεις της Python αποτελούν και αυτές με τη σειρά τους αντικείμενα (Rosales Silva, n.d.). Ενδεχομένως οι μαθητές, έχοντας εισαχθεί στην έννοια της αντικειμενοστρέφιας, να αντιλαμβάνονται ως αντικείμενα μόνο τις μεταβλητές και τους τελεστές και όχι τις συναρτήσεις (εσφαλμένη προαναπαράσταση που μπορεί να οδηγήσει σε μαθησιακή δυσκολία).
* Μία συνάρτηση μπορεί να κληθεί εντός άλλων συναρτήσεων ανεξάρτητα από το σημείο του κώδικα που έχουν οριστεί χωρίς αυτό να επηρεάζει κάπως την εκτέλεση κώδικα. Οι μαθητές ενδέχεται να το αγνοούν αυτό και να μη το εφαρμόζουν, αναπαράγοντας έτσι λειτουργίες συναρτήσεων που υπάρχουν ήδη σε νέες συναρτήσεις που κατασκευάζουν αντί απλώς να τις καλούν. Ο εκπαιδευτικός καλείται να δώσει έμφαση στην καλή πρακτική ιεραρχικού ορισμού των συναρτήσεων συναρτήσεις που καλούνται πολλές φορές από πολλές διαφορετικές συναρτήσεις να βρίσκονται στο αρχικό τμήμα του κώδικα.

1. ΧΡΗΣΗ ΕΞΩΤΕΡΙΚΩΝ ΠΗΓΩΝ

Αράπογλου Αριστείδης, Κανίδης Ευάγγελος, Λέκκα Δήμητρα Μακρυγιάννης Παναγιώτης, Μπελεσιώτης Βασίλειος, Παπαδάκης Σπυρίδων, Τζήμας Δημήτριος «Προγραμματισμός Υπολογιστών – Σημειώσεις Μαθητή», Ινστιτούτο Εκπαιδευτικής Πολιτικής

Αράπογλου Αριστείδης, Κανίδης Ευάγγελος, Λέκκα Δήμητρα Μακρυγιάννης Παναγιώτης, Μπελεσιώτης Βασίλειος, Παπαδάκης Σπυρίδων, Τζήμας Δημήτριος «Προγραμματισμός Υπολογιστών – Τετράδιο Εργασιών», Ινστιτούτο Εκπαιδευτικής Πολιτικής

J. Rosales Silva, “Python Tutorial Release 3.8.1 Guido van Rossum and the Python development team”.

Mark Lutz, «Learning Python», Chapter 4, Fifth Edition, 2013.

1. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΦΥΛΛΩΝ ΕΡΓΑΣΙΑΣ

**Φύλλο Εργασίας 1ης Διδακτικής ώρας**

1. Αφού γίνει η παρουσίαση του θεωρητικού πλαισίου τα 10 πρώτα λεπτά της ώρας, ξεκινάει η πρώτη δραστηριότητα του φύλλου εργασίας. Κατά τη δραστηριότητα αυτή, διάρκειας 5 λεπτών, θα παρουσιαστούν στους μαθητές δύο συναρτήσεις έτοιμου κώδικα (από το τετράδιο εργασιών του μαθήματος που υπάρχει στο Φωτόδενδρο σελ. 80-82) και θα κληθούν να αναγνωρίσουν τη λειτουργία που εκτελούν, την εμβέλεια των μεταβλητών και τα πιθανά αποτελέσματά τους. Το όνομα της συνάρτησης θα είναι αόριστο έτσι ώστε να μην φαίνεται εκ πρώτης όψεως η εργασία που επιτελούν. Παράλληλα ο διδάσκων θα εξηγήσει την ανάγκη για ορθή ονομασία των συναρτήσεων ώστε να μην υπάρχουν παρανοήσεις αργότερα κατά την κλήση τους εντός ενός άλλου προγράμματος. Ουσιαστικά η πρώτη δραστηριότητα αποτελείται από ερωτήσεις εμπέδωσης πάνω στις έννοιες που μόλις παρουσίασε ο καθηγητής. H δραστηριότητα αυτή πραγματοποιείται με την τεχνική των διερευνητικών ερωτήσεων και των αντίστοιχων απαντήσεων (Ερώτηση Πρόβλεψης και Ερώτηση Πιθανής Παρανόησης) και της Διερεύνησης Κώδικα. Αφορά τα στάδια Στόχοι Μάθησης (οι μαθητές θα πρέπει να είναι σε θέση να αναγνωρίζουν τη λειτουργία συναρτήσεων και να τις ονοματίζουν ορθά), Ανάκληση Πρότερης Γνώσης (οι μαθητές θα κληθούν να χρησιμοποιήσουν τις γνώσεις τους πάνω στην ονοματοδοσία) και της Εξάσκησης.
2. Η δεύτερη δραστηριότητα θα διαρκέσει 10 λεπτά περιλαμβάνει δύο έτοιμες συναρτήσεις (Συνάρτηση 1: υπολογισμός της ταυτότητας διαφοράς τετραγώνου και Συνάρτηση 2: ευκλείδεια απόσταση) ενώ οι μαθητές θα κληθούν να συμπληρώσουν τις κενές εντολές μίας τρίτης συνάρτησης η οποία θα συγκρίνει την ευκλείδεια απόσταση από ένα σημείο εκκίνησης προς δύο διαφορετικά και θα αποφασίζει ποια απόσταση είναι μικρότερη. Η συνάρτηση 3 θα καλεί τη Συνάρτηση 2 η οποία θα καλεί με τη σειρά της τη Συνάρτηση 1. Μέσω της δραστηριότητας αυτής ο εκπαιδευτικός θα εξηγήσει στους μαθητές ότι οι συναρτήσεις μπορούν να κληθούν εντός άλλων συναρτήσεων. Η δραστηριότητα αυτή θα χρησιμοποιήσει τις τεχνικές της Πρακτικής Άσκησης, των Ομάδων Εργασίας και Διερευνητικών Ερωτήσεων ενώ αφορά τα στάδια Ανάκλησης Πρότερης Γνώσης, Υποστήριξης, Εξάσκησης και Ανάδρασης.
3. Στη δραστηριότητα αυτή, διάρκειας 10 λεπτών, θα παρουσιαστεί στους μαθητές ένα πρόγραμμα σε γλώσσα Python το οποίο θα περιέχει επαναλαμβανόμενες αντιμεταθέσεις στοιχείων μίας λίστας. Οι μαθητές θα κληθούν να τις εντοπίσουν και να προτείνουν ιδέες (όχι κώδικα!) και να σκεφθούν υπολογιστικά για το πώς αυτό μπορεί να αποφευχθεί, συζητώντας ανά ομάδες. Η δραστηριότητα αυτή χρησιμοποιεί τις τεχνικές Συζήτηση και Χιονοστιβάδα (κάθε μαθητής θα αναπτύσσει μία ιδέα και στη συνέχεια οι συμμαθητές του θα την επεκτείνουν) και περιέχει Ερώτηση Πρόβλεψης και Ερώτηση Πιθανής Παρανόησης. Η δραστηριότητα αυτή αφορά το στάδιο της Ανάδρασης.
4. Η τελευταία δραστηριότητα διαρκεί 10 λεπτά. Ο εκπαιδευτικός θα παρουσιάσει μία προτεινόμενη υλοποίηση της συνάρτησης που κλήθηκαν να κατασκευάσουν νοητικά οι μαθητές στην τρίτη δραστηριότητα η οποία περιέχει την επιστροφή πολλών στοιχείων. Μέσω αυτής της δραστηριότητας ο εκπαιδευτικός θα καταστήσει κατανοητό ότι ο κώδικας μπορεί να συμπυκνωθεί περαιτέρω και αντί για πολλές κλήσεις της συνάρτησης αντιμετάθεσης στοιχείων μίας λίστας μπορεί να επιστρέφει όλες τις τιμές που χρειάζεται να επεξεργαστεί η συνάρτηση. Ταυτόχρονα θα επιχειρήσει να εισάγει την έννοια της αφαίρεσης, ωθώντας τους μαθητές να τροποποιήσουν τη συνάρτηση κατάλληλα ώστε να λειτουργεί υπό οποιοδήποτε πλαίσιο. Η δραστηριότητα αυτή αφορά το στάδιο της Παρουσίασης Υλικού και θα εφαρμοστεί η τεχνική της Επίδειξης, του Διαλόγου και των διερευνητικών Ερωτήσεων Πιθανής Παρανόησης ενώ παράλληλα θα επιχειρηθεί να προκληθεί κοινωνιογνωστική σύγκρουση. Αυτή η δραστηριότητα αφήνεται για το τέλος ως πιο δύσκολη καθώς αφορά την προχωρημένη έννοια της αφαίρεσης με την οποία οι μαθητές δεν είναι εξοικειωμένοι.
5. ΑΞΙΟΛΟΓΗΣΗ

**Αξιολόγηση Φύλλου Εργασίας**

|  |  |  |  |
| --- | --- | --- | --- |
| **Κριτήριο** | **Μη Ικανοποιητικά** | **Μέτρια** | **Ικανοποιητικά** |
| Δυνατότητα να εξηγούν τη λειτουργία δοσμένων συναρτήσεων και να προβλέπουν την έξοδό τους (Τομέας Γνώσεων στόχος 1) |  |  |  |
| Δυνατότητα να αναγνωρίζουν την εμβέλεια των παραμέτρων και των μεταβλητών μίας συνάρτησης καθώς και το περιεχόμενό τους *(Τομέας Γνώσεων στόχος 2)* |  |  |  |
| Δυνατότητα να αναγνωρίζουν την ύπαρξη επαναλαμβανόμενου κώδικα *(Τομέας Δεξιοτήτων στόχος 1)* |  |  |  |
| Δυνατότητα να επεκτείνουν συναρτήσεις αφαιρετικά *(Τομέας Δεξιοτήτων στόχος 2)* |  |  |  |

1. ΦΥΛΛΟ ΕΡΓΑΣΙΑΣ

Συναρτήσεις-μέθοδοι σε προγραμματιστικό περιβάλλον Python

1. *Συναρτήσεις σε Python*

Μελέτησε τις παρακάτω συναρτήσεις:

def coo():   
 for i in range(100):  
 print ("python")

def foo(sequence, item):   
 counter = 0   
 for i in sequence:   
 if i == item:   
 counter += 1   
 return counter  
foo([1,3,1,1],1)

1. Τι τιμή περιέχει η μεταβλητή sequence και τι η item; Μπορώ να χρησιμοποιήσω τη μεταβλητή counter και εκτός της συνάρτησης;
2. Πόσες φορές θα εκτελεστεί η συνάρτηση coo;
3. Τι πιστεύεις ότι θα επιστρέψει/εμφανίσει η συνάρτηση foo;
4. Θα είχε νόημα να ονομάζαμε τη συνάρτηση foo “fileReader”;

Αιτιολόγησε όλες τις απαντήσεις σου. Συζήτησε με τον/την συνεργάτη/συνεργάτιδά και για τις τέσσερις ερωτήσεις και συγκρίνετε τις απαντήσεις σας.

1. ***Εσωτερικές Κλήσεις Συναρτήσεων***

Παρακάτω ακολουθούν δύο συναρτήσεις ορισμένες κατά σειρά: υπολογισμός διαφοράς τετραγώνου και υπολογισμός ευκλείδειας απόστασης 2 σημείων.

import math as m  
  
def squareDif(a, b):  
 return a \* a - 2 \* a \* b - b\*b

def eucledianDistance(pointA, pointB): #pointA[0] is xA and pointA[1] is yA  
 return m.sqrt(abs(squareDif(pointA[0], pointB[0]) + squareDif(pointA[1], pointB[1])))

1. Εάν η συνάρτηση eucledianDistance είχε οριστεί πριν από την squareDif, θα μπορούσε να την καλέσει παρόλο που αυτή ορίζεται μετά ή θα εμφάνιζε σφάλμα ο μεταγλωττιστής; Να αιτιολογήσεις την απάντηση της ομάδας σου.
2. Αφού συνεργαστείς με τον/ην συνεργάτη/δα σου, συμπληρώστε τα κενά στην παρακάτω συνάρτηση. Ποια περίπτωση δεν ελέγχει ο κώδικας και τι θα έπρεπε να προστεθεί στη συνάρτηση shortestPath προκειμένου να συμπεριληφθεί και αυτή η περίπτωση;

def shortestPath(startingPoint, targetPoint1, targetPoint2):  
 if \_(startingPoint, targetPoint1) < \_(startingPoint, targetPoint2):  
 msg = "Shortest distance is Starting Point -> Target point1"  
 else:  
 \_\_\_\_

return msg  
  
startingPoint = [0,0]  
targetPoint1 = [8, 13]  
targetPoint2 = [9, 12]  
print(shortestPath(startingPoint, targetPoint1, targetPoint2))

1. Καλεί η συνάρτηση shortestPath άμεσα τη συνάρτηση squareDif; Σε ποιο σημείο του κώδικα θα πρέπει να έχει οριστεί η shortestPath ώστε να μην υπάρχει σφάλμα κατά την εκτέλεση; Τι άποψη σχημάτισες αφού συνεργάστηκες για τη συμπλήρωση της shortestPath;
2. *Προβληματικός Κώδικας*

i) Ποια από τις παρακάτω λειτουργίες επιτελεί ο κώδικας που ακολουθεί; (μπορείτε να επιλέξετε περισσότερες από μία απαντήσεις). Α) Άθροισμα στοιχείων μίας λίστας Β) Αντιμετάθεση Στοιχείων μίας λίστας Γ) Διαγραφή Στοιχείων μίας λίστας Δ) Μετονομασία μίας λίστας.

myList = [3, 7, 5, 5, 8, 2]  
sum1 = myList[0] + myList[1]  
temp = myList[0]  
myList[0] = myList[1]  
myList[1] = temp  
sum2 = myList[2] + myList[3]  
temp = myList[2]  
myList[2] = myList[3]  
myList[3] = temp  
sum3 = myList[4] + myList[5]  
temp = myList[4]  
myList[4] = myList[5]  
myList[5] = temp  
print(myList, sum1, sum2, sum3)

ii) Ένας συμμαθητής σου ισχυρίζεται ότι ο παραπάνω κώδικας δεν έχει κανένα απολύτως πρόβλημα. Συμφωνείς μαζί του; Εάν όχι, πώς μπορεί να διορθωθεί το πρόβλημα αυτό; Συζήτησε με τον/ην συνεργάτη/δα σου και αφού καταλήξετε σε μία κοινώς αποδεκτή λύση, προτείνετέ την στην τάξη.

1. Αφού άκουσες και τις προτάσεις των υπόλοιπων ομάδων, πιστεύεις ότι η πρόταση

της ομάδας σου έχει κάποιες αδυναμίες; Συζήτησέ το με τον/ην συνεργάτη/δα σου.

1. *Γενικευμένη συνάρτηση*

Ας δούμε λοιπόν μία υλοποίηση της συνάρτησης που συζητήσαμε στην προηγούμενη δραστηριότητα.

myList = [3, 7, 5, 5, 8, 2]

def listEditor(aList):  
 i=0  
 k=0  
 sumF=[0,0,0]  
 while i<=4:  
 sumF[k]= aList[i] + aList[i + 1]  
 localTemp=aList[i]  
 aList[i]=aList[i + 1]  
 aList[i + 1]=localTemp  
 i+=2  
 k+=1  
 return aList, sumF  
  
print(listEditor(myList))

1. Πιστεύεις ότι η συνάρτηση θα επιστρέψει κάτι ώστε να το εμφανίσει η εντολή print ή ο μεταγλωττιστής θα επιστρέψει μήνυμα σφάλματος; Να αιτιολογήσεις την απάντησή σου και να τη συγκρίνεις με αυτήν του/ης συνεργάτη/δας σου. Συμφωνείτε ή διαφωνείτε και για ποιο λόγο;
2. Θα μπορούσαμε ίσως να τροποποιήσουμε τη λογική συνθήκη της εντολής while έτσι ώστε να μην εξαρτάται από το μέγεθος της συγκεκριμένης λίστας που δίνεται στο παράδειγμα αλλά να μπορεί να υλοποιηθεί για οποιοδήποτε μέγεθος λίστας; Με ποιον τρόπο θα γίνει αυτό; Ποια λύση προτείνει η ομάδα σου;