****

***“ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ”***

***ΣΥΜΠΛΗΡΩΜΑΤΙΚΟ ΥΛΙΚΟ***

***ΝΕΑ ΥΛΗ 2019-2020***

***ΒΟΗΘΗΜΑ ΜΑΘΗΤΗ***

**ΕΠΙΜΕΛΕΙΑ ΣΗΜΕΙΩΣΕΩΝ: ΛΑΜΠΡΑΚΗΣ ΜΑΝΩΛΗΣ**

**(ΜΗΧΑΝΙΚΟΣ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ, ΜΕΤΑΠΤΥΧΙΑΚΟ ΔΙΠΛΩΜΑ ΣΤΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΚΤΥΩΝ ΥΠΟΛΟΓΙΣΤΩΝ)**

**Κεφάλαιο 1: Ανάλυση προβλήματος**

1. Να δώσετε τον ορισμό του προβλήματος.

**(Ενότητα 1.1, ορισμός)** Με τον όρο πρόβλημα εννοείται μια κατάσταση η οποία χρήζει αντιμετώπισης, απαιτεί λύση, η δε λύση της δεν είναι ούτε γνωστή, ούτε προφανής.

1. Η έννοια του προβλήματος είναι χαρακτηριστικό μόνο της εποχής μας;

**(Ενότητα 1.1, κουκίδες)** Η έννοια του προβλήματος δεν είναι χαρακτηριστικό μόνο της εποχής μας, αντίθετα υφίσταται από την αρχαιότητα.

1. Περιγράψτε το πρόβλημα του έτους 2000 (millennium bug).

**(Ενότητα 1.1, παράρτημα)** Οι υπολογιστές για εξοικονόμηση αποθηκευτικού χώρου, αποθήκευαν τις ημερομηνίες με 2 ψηφία για κάθε ένα από τα τρία συστατικά της. Για παράδειγμα η ημερομηνία 15 Απριλίου 1999 αναπαριστάται με 150499. Με τον παραπάνω τρόπο αποθήκευσης, η ημερομηνία 1 Ιανουαρίου 2000 θα είχε την μορφή 010100, γεγονός που θα αποτελούσε πρόβλημα στους υπολογισμούς που θα πραγματοποιούσαν οι υπολογιστές.

1. Από ποιους παράγοντες επηρεάζεται η κατανόηση ενός προβλήματος;

**(Ενότητα 1.2)** Η κατανόηση ενός προβλήματος αποτελεί συνάρτηση δύο παραγόντων, της σωστής διατύπωσης εκ μέρους του δημιουργού του και της αντίστοιχα σωστής ερμηνείας από τη μεριά εκείνου που καλείται να το αντιμετωπίσει.

1. Ποια είναι η μορφή με την οποία πρέπει να παρουσιάζεται ένα πρόβλημα;

**(Ενότητα 1.2)** Η μορφή με την οποία παρουσιάζεται ένα πρόβλημα μπορεί να είναι οποιαδήποτε αρκεί να μπορεί να γίνει αντιληπτή από μία από τις πέντε ανθρώπινες αισθήσεις.

1. Τι γνωρίζετε για την σαφήνεια της διατύπωσης ενός προβλήματος;

**(Ενότητα 1.2 - Σαφήνεια)** Τα προβλήματα που μπορεί να κληθούμε να αντιμετωπίσουμε κατά τη διάρκεια της ζωής μας μπορούν να αναφέρονται σε οποιοδήποτε τομέα, μπορεί να αφορούν στα μαθηματικά, στη φυσική, στη λογική, στην καθημερινή ζωή ή οτιδήποτε άλλο θα μπορούσε κάποιος να σκεφτεί Η κατανόηση ενός προβλήματος εξαρτάται σε μεγάλο βαθμό από την διατύπωσή του. Οποιοδήποτε μέσο μπορεί να χρησιμοποιηθεί για να αποδοθεί η διατύπωση ενός προβλήματος. Συνηθέστερο από όλα είναι ο λόγος, είτε ο προφορικός, είτε ο γραπτός. Ο λόγος σαν μέσο επικοινωνίας και συνεννόησης πρέπει να χαρακτηρίζεται από σαφήνεια. Άστοχη χρήση ορολογίας, λανθασμένη σύνταξη, είναι δύο στοιχεία που μπορούν να προκαλέσουν παρερμηνείες και παραπλανήσεις.

1. Τι γνωρίζετε για τον χώρο στον οποίο αναφέρεται ένα πρόβλημα;

**(Ενότητα 1.2 – Σαφήνεια)** Σημαντικός παράγοντας για την κατανόηση ενός προβλήματος είναι η αποσαφήνιση του χώρου στον οποίο αναφέρεται. Η πληροφορία αυτή παρέχεται από την διατύπωση-δεδομένα του προβλήματος. Παράδειγμα στον χώρο: Έγραψα στο διαγώνισμα 7,5. Είναι καλός ή κακός βαθμός; Η απάντηση εξαρτάται από τον χώρο: Αν μιλάμε για πανεπιστήμιο είναι καλός!

1. Να δώσετε τους ορισμούς των εννοιών: δεδομένο, πληροφορία, επεξεργασία δεδομένων και να δώσετε μία διαγραμματική σχέση μεταξύ τους.

**(Ενότητα 1.2 – ορισμοί)**

**Δεδομένο:** Οποιοδήποτε στοιχείο μπορεί να γίνει αντιληπτό από έναν τουλάχιστον παρατηρητή με μία από τις πέντε αισθήσεις του.

**Επεξεργασία δεδομένων:** Μία διαδικασία κατά την οποία ένας μηχανισμός (π.χ. ο Η/Υ, η ο ανθρώπινος εγκέφαλος) δέχεται δεδομένα, κάνει πράξεις και υπολογισμούς σύμφωνα με ένα προκαθορισμένο τρόπο και αποδίδει πληροφορίες. Για παράδειγμα: η μέση θερμοκρασία κατά τον μήνα Μάρτιο (πληροφορία), προκύπτει από τον υπολογισμό του μέσου όρου (επεξεργασία) των ημερησίων θερμοκρασιών του μήνα (δεδομένα).

**Πληροφορία:** Οποιοδήποτε γνωσιακό στοιχείο προέρχεται από την επεξεργασία δεδομένων.

**Δεδομένα**

**Επεξεργασία Δεδομένων**

**Πληροφορία** *(ζητούμενο)*

1. Τι ονομάζεται δομή ενός προβλήματος;

**(Ενότητα 1.3 - ορισμός)**. Με τον όρο **δομή προβλήματος**, εννοούμε τα συστατικά μέρη από τα οποία συντίθεται το πρόβλημα, δηλαδή τα επιμέρους τμήματα που το αποτελούν καθώς και τον τρόπο με τον οποίο αυτά συνδέονται μεταξύ τους. Χωρίζοντας το πρόβλημα σε μικρότερα και απλούστερα υποπροβλήματα, αυτά λύνονται ευκολότερα, καταλήγοντας στη λύση του αρχικού προβλήματος.

1. Τι γνωρίζετε για την διαγραμματική αναπαράσταση ενός προβλήματος;

**(Ενότητα 1.3)** \_

* Το αρχικό πρόβλημα αναπαρίσταται από ένα ορθογώνιο παραλληλόγραμμο.
* Κάθε ένα από τα απλούστερα προβλήματα στα οποία αναλύεται ένα οποιοδήποτε πρόβλημα, αναπαρίσταται επίσης από ένα ορθογώνιο παραλληλόγραμμο.
* Τα παραλληλόγραμμα που αντιστοιχούν στα απλούστερα προβλήματα στα οποία αναλύεται ένα οποιοδήποτε πρόβλημα, σχηματίζοντας ένα επίπεδο χαμηλότερα. Έτσι σε κάθε κατώτερο επίπεδο, δημιουργείται η γραφική αναπαράσταση των προβλημάτων στα οποία αναλύονται τα προβλήματα του αμέσως ψηλότερου επιπέδου.

1. Τι είναι ο καθορισμός απαιτήσεων ενός προβλήματος;

**(Ενότητα 1.4)** Για τη σωστή επίλυση του προβλήματος βασική προϋπόθεση είναι ο **καθορισμός απαιτήσεων**. Είναι η διαδικασία κατά την οποία πρέπει να κάνουμε:

* τον επακριβή προσδιορισμό των δεδομένων που παρέχει το πρόβλημα καθώς και
* την λεπτομερειακή καταγραφή των ζητούμενων που αναμένονται σαν αποτελέσματα της επίλυσης,

Τα δεδομένα δεν είναι πάντα εύκολο να διακριθούν.(Σε πολλά προβλήματα τα δεδομένα θα πρέπει να «ανακαλυφθούν» μέσα στα λεγόμενα)

1. Ποια είναι τα στάδια αντιμετώπισης ενός προβλήματος;

**(Ενότητα 1.4)**

**Κατανόηση**

**Ανάλυση**

**Επίλυση**

* κατανόηση, όπου απαιτείται η σωστή και πλήρης αποσαφήνιση των δεδομένων και των ζητούμενων του προβλήματος
* ανάλυση, όπου το αρχικό πρόβλημα διασπάται σε άλλα επί μέρους απλούστερα προβλήματα
* επίλυση, όπου υλοποιείται η λύση του προβλήματος, μέσω της λύσης των επιμέρους προβλημάτων.

**ΑΣΚΗΣΕΙΣ ΓΙΑ ΛΥΣΗ**

**Άσκηση 1:** Ερωτήσεις Σωστό – Λάθος.

1. Πρόβλημα είναι μία οποιαδήποτε κατάσταση η οποία απαιτεί λύση.
2. Η έννοια του προβλήματος υφίσταται από την αρχαιότητα.
3. Η κατανόηση ενός προβλήματος συνδέεται αποκλειστικά με την σωστή διατύπωση από την πλευρά του δημιουργού του.
4. Η μορφή με την οποία παρουσιάζεται ένα πρόβλημα, μπορεί να είναι οποιαδήποτε αρκεί να μπορεί να γίνει αντιληπτή από μία από τις πέντε ανθρώπινες αισθήσεις.
5. Το πρόβλημα του έτους 2000 (millennium bug) σχετίζονταν με τον τρόπο αναπαράστασης μίας ημερομηνίας από τους υπολογιστές.
6. Η κατανόηση ενός προβλήματος εξαρτάται σε μεγάλο βαθμό από την διατύπωσή του.
7. Ένας σημαντικός παράγοντας για την κατανόηση ενός προβλήματος είναι η αποσαφήνιση του χώρου στον οποίο αναφέρεται.
8. Η πληροφορία αντλείται απευθείας από τα δεδομένα του προβλήματος, δίχως να απαιτείται κάποια επεξεργασία.
9. Με τον όρο «Δομή Προβλήματος», ένα πρόβλημα αντιμετωπίζεται σαν μία ενιαία οντότητα προς επεξεργασία.
10. Στην διαγραμματική αναπαράσταση ενός προβλήματος, το αρχικό πρόβλημα αναπαρίσταται από ένα ορθογώνιο παραλληλόγραμμο.
11. Ο καθορισμός των απαιτήσεων ενός προβλήματος, έχει να κάνει αυστηρά με την καταγραφή των δεδομένων του προβλήματος.
12. Ένα από τα στάδια αντιμετώπισης ενός προβλήματος είναι και η Ανάλυση.

**Άσκηση 2:** Να τοποθετήσετε τις παρακάτω λέξεις στην σωστή σειρά, ώστε να περιγράφονται σωστά οι αντίστοιχες έννοιες:

1. **Παραγωγή Δεδομένων:** Πληροφορία, Δεδομένα, Επεξεργασία Δεδομένων.
2. **Στάδια Αντιμετώπισης προβλήματος:** Κατανόηση, Επίλυση, Ανάλυση

**Άσκηση 3:** Να συμπληρώσετε τα κενά στις ακόλουθες προτάσεις:

1. Με τον όρο πρόβλημα εννοείται μια κατάσταση η οποία χρήζει \_\_\_\_ , απαιτεί \_\_\_\_, η δε λύση της δεν είναι ούτε \_\_\_\_ , ούτε \_\_\_\_.
2. Η \_\_\_\_ ενός προβλήματος αποτελεί συνάρτηση δύο παραγόντων, της σωστής \_\_\_\_ εκ μέρους του \_\_\_\_ του και της αντίστοιχα σωστής \_\_\_\_ από τη μεριά εκείνου που καλείται να το \_\_\_\_.
3. Η κατανόηση ενός προβλήματος εξαρτάται σε μεγάλο βαθμό από την \_\_\_\_ του. Οποιοδήποτε \_\_\_\_ μπορεί να χρησιμοποιηθεί για να αποδοθεί η διατύπωση ενός προβλήματος. Συνηθέστερο από όλα είναι ο \_\_\_\_, είτε ο \_\_\_\_, είτε ο \_\_\_\_. Ο λόγος σαν μέσο επικοινωνίας και συνεννόησης πρέπει να χαρακτηρίζεται από \_\_\_\_. Άστοχη χρήση ορολογίας, λανθασμένη \_\_\_\_, είναι δύο στοιχεία που μπορούν να προκαλέσουν παρερμηνείες και παραπλανήσεις.
4. Με τον όρο **δομή προβλήματος**, εννοούμε τα \_\_\_\_ μέρη από τα οποία συντίθεται το \_\_\_\_, δηλαδή τα \_\_\_\_ τμήματα που το αποτελούν καθώς και τον τρόπο με τον οποίο αυτά \_\_\_\_ μεταξύ τους.
5. **Επεξεργασία \_\_\_\_ :** Μία διαδικασία κατά την οποία ένας μηχανισμός (π.χ. ο \_\_\_\_, η ο \_\_\_\_ \_\_\_\_ ) δέχεται δεδομένα, κάνει \_\_\_\_ και υπολογισμούς σύμφωνα με ένα \_\_\_\_ τρόπο και αποδίδει \_\_\_\_.

**Κεφάλαιο 2 – Ενότητα 2.2: Σπουδαιότητα αλγορίθμων**

1. Ποιες είναι οι σκοπιές από τις οποίες μελετάει τους αλγορίθμους η Πληροφορική;

**(Ενότητα 2.2)**

* **Υλικού (Hardware):** Η ταχύτητα εκτέλεσης ενός αλγορίθμου επηρεάζεται από τις διάφορες τεχνολογίες υλικού, δηλαδή από τον τρόπο που είναι δομημένα σε μία ενιαία αρχιτεκτονική τα διάφορα συστατικά του υπολογιστή (δηλαδή ανάλογα με το αν ο υπολογιστής έχει κρυφή μνήμη και πόση, ανάλογα με την ταχύτητα της κύριας και δευτερεύουσας μνήμης κοκ.)
* **Γλωσσών προγραμματισμού (Programming Languages):** Το είδος της γλώσσας προγραμματισμού που χρησιμοποιείται (δηλαδή, χαμηλότερου ή υψηλότερου επιπέδου) αλλάζει τη δομή και τον αριθμό των εντολών ενός αλγορίθμου. Γενικά μία γλώσσα που είναι χαμηλότερου επιπέδου (όπως η assembly ή η γλώσσα C) είναι ταχύτερη από μία άλλη γλώσσα που είναι υψηλοτέρου επιπέδου (όπως η Basic ή Pascal). Ακόμη, σημειώνεται ότι διαφορές συναντώνται μεταξύ των γλωσσών σε σχέση με το πότε εμφανίσθηκαν. Για παράδειγμα, παλαιότερα μερικές γλώσσες προγραμματισμού δεν υποστήριζαν την αναδρομή.
* **Θεωρητική (Theoretical):** Το ερώτημα που συχνά τίθεται είναι, αν πράγματι υπάρχει ή όχι κάποιος αποδοτικός αλγόριθμος για την επίλυση ενός προβλήματος. Η προσέγγιση αυτή είναι ιδιαίτερα σημαντική, γιατί προσδιορίζει τα όρια της λύσης που θα βρεθεί σε σχέση με ένα συγκεκριμένο πρόβλημα.
* ***Αναλυτική*** (analytical). Μελετώνται οι υπολογιστικοί πόροι (computer resources) που απαιτούνται από έναν αλγόριθμο, όπως για παράδειγμα το μέγεθος της κύριας και της δευτερεύουσας μνήμης, ο χρόνος για λειτουργίες CPU και για λειτουργίες εισόδου/εξόδου κ.λπ.

**ΑΣΚΗΣΕΙΣ ΓΙΑ ΛΥΣΗ**

**Άσκηση 1:** Ερωτήσεις Σωστό – Λάθος.

1. Η ταχύτητα εκτέλεσης ενός αλγορίθμου δεν επηρεάζεται από τις διάφορες τεχνολογίες υλικού.
2. Γενικά, μία γλώσσα προγραμματισμού χαμηλότερου επιπέδου είναι ταχύτερη από μία γλώσσα προγραμματισμού υψηλού επιπέδου.
3. Δεν υπάρχουν διαφορές στις γλώσσες προγραμματισμού, σε σχέση με το πότε αυτές εμφανίστηκαν.
4. Η θεωρητική προσέγγιση απαντάει στο ερώτημα αν πράγματι υπάρχει ή όχι κάποιος αποδοτικός αλγόριθμος για την επίλυση ενός προβλήματος.
5. Η θεωρητική προσέγγιση είναι ιδιαίτερα σημαντική, καθώς προσδιορίζει τα όρια της λύσης βρεθεί σε σχέση με ένα συγκεκριμένο πρόβλημα.
6. Οι υπολογιστικοί πόροι που απαιτούνται σε έναν αλγόριθμο, δεν επηρεάζονται από τις λειτουργίες εισόδου/εξόδου.

**Άσκηση 2:** Να συμπληρώσετε τα κενά στις ακόλουθες προτάσεις:

1. Η \_\_\_\_ εκτέλεσης ενός αλγορίθμου επηρεάζεται από τις διάφορες τεχνολογίες \_\_\_\_, δηλαδή από τον τρόπο που είναι δομημένα σε μία ενιαία \_\_\_\_ τα διάφορα συστατικά του υπολογιστή (δηλαδή ανάλογα με το αν ο υπολογιστής έχει \_\_\_\_ μνήμη και πόση, ανάλογα με την \_\_\_\_ της κύριας και δευτερεύουσας μνήμης κοκ.).
2. Το είδος της γλώσσας \_\_\_\_ που χρησιμοποιείται (δηλαδή, χαμηλότερου ή υψηλότερου επιπέδου) αλλάζει τη \_\_\_\_ και τον αριθμό των \_\_\_\_ ενός αλγορίθμου. Γενικά μία γλώσσα που είναι \_\_\_\_ επιπέδου (όπως η assembly ή η γλώσσα C) είναι ταχύτερη από μία άλλη γλώσσα που είναι \_\_\_\_ επιπέδου (όπως η Basic ή Pascal). Ακόμη, σημειώνεται ότι διαφορές συναντώνται μεταξύ των γλωσσών σε σχέση με το πότε \_\_\_\_. Για παράδειγμα, παλαιότερα μερικές γλώσσες προγραμματισμού δεν υποστήριζαν την \_\_\_\_.
3. Το ερώτημα που συχνά τίθεται είναι, αν πράγματι υπάρχει ή όχι κάποιος \_\_\_\_ αλγόριθμος για την επίλυση ενός \_\_\_\_. Η προσέγγιση αυτή είναι ιδιαίτερα σημαντική, γιατί προσδιορίζει τα \_\_\_\_ της λύσης που θα βρεθεί σε σχέση με ένα συγκεκριμένο \_\_\_\_.
4. Μελετώνται οι \_\_\_\_ πόροι (computer resources) που απαιτούνται από έναν \_\_\_\_, όπως για παράδειγμα το \_\_\_\_ της κύριας και της δευτερεύουσας μνήμης, ο χρόνος για \_\_\_\_ CPU και για λειτουργίες \_\_\_\_ / \_\_\_\_ κ.λπ.

**Ενότητα 3 Συμπληρωματικού υλικού – Εντολή Επίλεξε**

Η εντολή «Επίλεξε» είναι μία μορφή δομή πολλαπλής επιλογής. Η γενική μορφή της δομής αυτής και η γενική λειτουργία της είναι η ακόλουθη:

|  |  |
| --- | --- |
| **Γενική Σύνταξη** | **Παρατηρήσεις** |
| **ΕΠΙΛΕΞΕ** <έκφραση>  **ΠΕΡΙΠΤΩΣΗ** <λίστα\_τιμών\_1>  <Εντολές\_1>  **ΠΕΡΙΠΤΩΣΗ** <λίστα\_τιμών\_2>  <Εντολές\_2>  **ΠΕΡΙΠΤΩΣΗ** <λίστα\_τιμών\_3>  <Εντολές\_3>  ………….  **ΠΕΡΙΠΤΩΣΗ** **ΑΛΛΙΩΣ**  <Εντολές\_Αλλιώς>  **ΤΕΛΟΣ**\_**ΕΠΙΛΟΓΩΝ** | Η <έκφραση> μπορεί να είναι:   * Μεταβλητή * Αριθμητική πράξη * Συγκριτική Πράξη   Η <λίστα τιμών> μπορεί να είναι:   * Διακριτές τιμές * Περιοχή τιμών από … έως * Να υπακούουν σε μία συνθήκη |
| **Τρόπος Εκτέλεσης:** Υπολογίζεται η τιμή της έκφρασης και στη συνέχεια εκτελούνται οι εντολές που ανήκουν στην αντίστοιχη περίπτωση τιμών. Στην περίπτωση που η τιμή της έκφρασης δεν αντιστοιχεί σε καμία περίπτωση, τότε εκτελούνται οι εντολές της ΠΕΡΙΠΤΩΣΗΣ\_ΑΛΛΙΩΣ.  Η ΠΕΡΙΠΤΩΣΗ\_ΑΛΛΙΩΣ είναι προαιρετική.  Η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί μετά το ΤΕΛΟΣ\_ΕΠΙΛΟΓΩΝ.  **Μειονεκτήματα:** σε όλες τις περιπτώσεις ελέγχεται ή ίδια μεταβλητή ή έκφραση, δεν μπορούμε δηλαδή να ελέγξουμε περίπλοκες λογικές εκφράσεις που εμπεριέχουν δύο ή περισσότερες μεταβλητές ή εκφράσεις.  **Πλεονεκτήματα:** Σε κάποιες περιπτώσεις προσφέρει πλεονεκτήματα λόγω της συμπαγούς μορφής της.  **Διάγραμμα ροής:** είναι το ίδιο με το αντίστοιχο διάγραμμα της δομής «ΑΝ» | |

**Παράδειγμα 1 – Μετατροπή δομής Αν σε δομή Επίλεξε:** Να γράψετε ξανά τα ακόλουθα τμήματα αλγορίθμου, χρησιμοποιώντας την δομή Επίλεξε.

|  |  |  |
| --- | --- | --- |
| **Δομή ΑΝ** | **Δομή ΕΠΙΛΕΞΕ** | **Παρατηρήσεις** |
| **Διάβασε** κ  **Αν** κ<0 **τότε**  **Εμφάνισε** ‘Αρνητικός’  **Αλλιώς\_αν** κ<=9 **τότε**  **Εμφάνισε** ‘Μονοψήφιος’  **Αλλιώς\_αν** κ<=99 **τότε**  **Εμφάνισε** ‘Διψήφιος’  **Αλλιώς**  **Εμφάνισε** ‘Λάθος’  **Τέλος\_αν** | **Διάβασε** κ  **Επίλεξε** κ  **Περίπτωση** <0  **Εμφάνισε** ‘Αρνητικός’  **Περίπτωση** <=9  **Εμφάνισε** ‘Μονοψήφιος’  **Περίπτωση** <=99  **Εμφάνισε** ‘Διψήφιος’  **Περίπτωση** **Αλλιώς**  **Εμφάνισε** ‘Λάθος’  **Τέλος**\_**επιλογών** | Στην περίπτωση που η Επίλεξε πρέπει να διερευνήσει τις τιμές μίας συνθήκης, η μεταβλητή (ή η έκφραση) τοποθετείται στο τμήμα «Επίλεξε» και οι συγκρίσεις στα τμήματα «Περίπτωση».  Δεν μπορούμε να μετατρέψουμε κάθε δομή «ΑΝ» σε δομή «ΕΠΙΛΕΞΕ», καθώς η τελευταία έχει δεν μπορεί να χρησιμοποιηθεί σε κάθε περίπτωση. |

|  |  |  |
| --- | --- | --- |
| **Δομή ΑΝ** | **Δομή ΕΠΙΛΕΞΕ** | **Παρατηρήσεις** |
| **Διάβασε** κ  **Αν** κ=0 **ή** κ=5 **ή** κ=10 **τότε**  λ🡨1  **Αλλιώς\_αν** κ=1 **ή** κ=2 **τότε**  λ🡨2  **Αλλιώς**  λ🡨3  **Τέλος\_αν**  **Εμφάνισε** λ | **Διάβασε** κ  **Επίλεξε** κ  **Περίπτωση** 0, 5, 10  λ🡨1  **Περίπτωση** 1, 2  λ🡨2  **Περίπτωση** **Αλλιώς**  λ🡨3  **Τέλος**\_**επιλογών**  **Εμφάνισε** λ | Στην περίπτωση που η Επίλεξε πρέπει να διακριτές τιμές, η μεταβλητή (ή η έκφραση) τοποθετείται στο τμήμα «Επίλεξε» και οι διακριτές στα τμήματα «Περίπτωση». Αν οι διακριτές τιμές είναι περισσότερες από μία, χωρίζονται με κόμμα (**,**).  Η δομή «ΕΠΙΛΕΞΕ» μπορεί πάντα να μετατραπεί σε αντίστοιχη δομή «ΑΝ». |

**Παράδειγμα 2 – Πίνακας τιμών της δομής Επίλεξε:** Να παρουσιάσετε τον πίνακα τιμών του ακόλουθου αλγορίθμου.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Αλγόριθμος** πίνακας   1. **Διάβασε** κ 2. **Επίλεξε** κ-2 3. **Περίπτωση** 1, 2, 3 4. λ🡨κ+4 5. **Περίπτωση** >15 6. λ🡨κ-2 7. **Περίπτωση** **Αλλιώς** 8. λ🡨κ\*3 9. **Τέλος**\_**επιλογών** 10. **Εμφάνισε** λ   **Τέλος** πίνακας | **Εντολές** | **κ** | **λ** | **Συνθήκη** | **Έξοδος** |
| **Περίπτωση Α:** 1 | 3 |  |  |  |
| 3 |  |  | Αληθής |  |
| 4 |  | 7 |  |  |
| 10 |  |  |  | 7 |
|  | | | | |
| **Περίπτωση Β:** 1 | 10 |  |  |  |
| 3 |  |  | Ψευδής |  |
| 5 |  |  | Ψευδής |  |
| 8 |  | 30 |  |  |
| 10 |  |  |  | 30 |

**Παράδειγμα 3 – Διάγραμμα ροής με δομή Επίλεξε:** Να παρουσιάσετε το διάγραμμα ροής του παρακάτω αλγορίθμου.

|  |  |
| --- | --- |
| **Αλγόριθμος** διάγραμμα\_ροής  **Διάβασε** x  **Επίλεξε** x  **Περίπτωση** 0  Γράψε ‘μηδέν’  **Περίπτωση** <0  **Γράψε** ‘αρνητικός’  **Περίπτωση** Αλλιώς  **Γράψε** ‘θετικός’  **Τέλος\_Επιλογών**  **Τέλος** διάγραμμα\_ροής  Σημείωση: το διάγραμμα ροής της δομής «Επίλεξε» είναι ίδιο με της αντίστοιχης δομής «Αν». |  |

**Παράδειγμα 4 – Άσκηση περιπτώσεων πίνακα με δομή Επίλεξε:** Ένα ξενοδοχείο χρεώνει τους πελάτες του ανά ημέρα ανάλογα με τον μήνα που θα το επισκεφτούν, σύμφωνα με τον ακόλουθο πίνακα. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο: **1)** θα διαβάζει για έναν πελάτη τον μήνα που επισκέφτηκε το ξενοδοχείο και τον αριθμό ημερών παραμονής **2)** θα υπολογίζει και θα εμφανίζει το ποσό που θα πληρώσει.

|  |  |
| --- | --- |
| **Μήνας** | **Χρέωση ανά ημέρα** |
| Από Ιανουάριο μέχρι και Απρίλιο | 50 ευρώ |
| Από Μάιο μέχρι και Αύγουστο | 80 ευρώ |
| Από Σεπτέμβρη μέχρι και Δεκέμβριο | 65 ευρώ |

|  |  |
| --- | --- |
| **Κώδικας** | **Παρατηρήσεις** |
| **ΠΡΟΓΡΑΜΜΑ** άσκηση\_επίλεξε  **ΜΕΤΑΒΛΗΤΕΣ**  **ΧΑΡΑΚΤΗΡΕΣ:** μήνας  **ΑΚΕΡΑΙΕΣ:** ημέρες, κόστος  **ΑΡΧΗ**  **ΔΙΑΒΑΣΕ** ημέρες, μήνας  **ΕΠΙΛΕΞΕ** μήνας **! μεταβλητή ελέγχου**  **ΠΕΡΙΠΤΩΣΗ** ‘Ιανουάριος’, ‘ Φεβρουάριος’, ‘Μάρτης’, **&**‘Απρίλιος’ **! ανάλυση περιπτώσεων διακριτών τιμών**  κόστος🡨ημέρες\*50  **ΠΕΡΙΠΤΩΣΗ** ‘Μάιος’, ‘ Ιούνιος’, ‘Ιούλιος’, ‘Αύγουστος’  κόστος🡨ημέρες\*80  **ΠΕΡΙΠΤΩΣΗ** **ΑΛΛΙΩΣ**  κόστος🡨κόστος\*65  **ΤΕΛΟΣ**\_**ΕΠΙΛΟΓΩΝ**  **ΓΡΑΨΕ** κόστος  **ΤΕΛΟΣ**\_**ΠΡΟΓΡΑΜΜΑΤΟΣ** | Η δομή Επίλεξε λόγω της συμπαγούς μορφής της βολεύει σε περιπτώσεις με πολλές διακριτές τιμές όπως στην συγκεκριμένη περίπτωση. |

**ΑΣΚΗΣΕΙΣ ΓΙΑ ΛΥΣΗ**

**Άσκηση 1:** Ερωτήσεις Σωστό – Λάθος.

1. Σε μία δομή «ΕΠΙΛΕΞΕ» εκτελούνται με τη σειρά όλες οι εντολές που βρίσκονται μέσα στη δομή
2. Σε μία δομή «ΕΠΙΛΕΞΕ» η «ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ» είναι προαιρετική.
3. Η χρήση της δομής «ΕΠΙΛΕΞΕ» αποτελεί πλεονέκτημα στον προγραμματισμό, λόγω της συμπαγούς μορφής της.
4. Η έκφραση που διερευνάται σε μία δομή «ΕΠΙΛΕΞΕ» δεν μπορεί να είναι αριθμητική πράξη
5. Οι εντολές που βρίσκονται μέσα σε μία δομή «ΕΠΙΛΕΞΕ», μπορεί να είναι διακριτές τιμές, περιοχή τιμών από … έως ή να υπακούουν σε μία συνθήκη.
6. Μία δομή «ΕΠΙΛΕΞΕ» μπορεί να περιέχει σύνθετες συνθήκες σε κάθε «ΠΕΡΙΠΤΩΣΗ».
7. Σε μία δομή «ΕΠΙΛΕΞΕ» δεν ελέγχονται πάντα με τη σειρά όλες οι περιπτώσεις μέχρι το «ΤΕΛΟΣ\_ΠΕΡΙΠΤΩΣΕΩΝ».
8. Μία δομή «ΕΠΙΛΕΞΕ» δεν μπορεί να διερευνήσει περιπτώσεις αλφαριθμητικών τιμών.
9. Κάθε δομή «ΕΠΙΛΕΞΕ» μπορεί να μετατραπεί σε αντίστοιχη δομή «ΑΝ».
10. Κάθε δομή «ΑΝ» μπορεί να μετατραπεί σε αντίστοιχη δομή «ΕΠΙΛΕΞΕ».

**Άσκηση 2:** Να βρείτε τα λάθη που υπάρχουν στον ακόλουθο κώδικα (να αναφέρετε την γραμμή που βρίσκεται το λάθος και ποιο είναι αυτό).

1. **Διάβασε** κ
2. **Επίλεξε** κ
3. **Περίπτωση** 5,10,5 **τότε**
4. **Εμφάνισε** ‘Α’
5. **Περίπτωση** >15 **και** <50
6. **Εμφάνισε** ‘Β’
7. **Περίπτωση**\_**Αλλιώς**

**8.** **Εμφάνισε** ‘Γ’

**9.** **Τέλος**

**Άσκηση 3:** Να παρουσιάσετε τον πίνακα τιμών και το διάγραμμα ροής των ακόλουθων τμημάτων αλγορίθμου.

|  |  |
| --- | --- |
| 1. κ🡨15 2. **Επίλεξε** κ 3. **Περίπτωση** 3, 5, 12 4. λ🡨κ-3 5. **Περίπτωση** <50 6. λ🡨κ+4 7. **Περίπτωση** >100 8. λ🡨κ 9. **Περίπτωση Αλλιώς** 10. λ🡨10   **11.** **Τέλος\_επιλογών**  **12. Εμφάνισε** λ | 1. κ🡨5 2. **Επίλεξε** κ**mod**3 3. **Περίπτωση** 1 4. μ🡨κ\*2 5. **Περίπτωση** 2 6. μ🡨κ\*3 7. **Περίπτωση** **Αλλιώς** 8. μ🡨κ+10 9. κ🡨κ+μ 10. **Τέλος\_επιλογών** 11. **Εμφάνισε** κ,μ |

**Άσκηση 4:** Να μετατρέψετε τις ακόλουθες δομές «ΑΝ» χρησιμοποιώντας την δομή «ΕΠΙΛΕΞΕ».

|  |  |
| --- | --- |
| **ΓΡΑΨΕ** "Δώσε αριθμό από 0 έως και 5" **ΔΙΑΒΑΣΕ** Χ **ΑΝ** Χ = 0 **ΤΟΤΕ** **ΓΡΑΨΕ**"μηδέν" **ΑΛΛΙΩΣ\_ΑΝ** (Χ = 1) **ή** (Χ = 3) **ή** (Χ = 5) **ΤΟΤΕ** **ΓΡΑΨΕ** "περιττός αριθμός" **ΑΛΛΙΩΣ\_ΑΝ** (Χ = 2) **ή** (Χ = 4) **ΤΟΤΕ** **ΓΡΑΨΕ** "άρτιος αριθμός" **ΑΛΛΙΩΣ** **ΓΡΑΨΕ** "έδωσες λάθος αριθμό" **ΤΕΛΟΣ\_ΑΝ**  **(Πανελλήνιες 2008)** | **Διάβασε** έτη  **Αν** έτη<=3 **τότε**  μπόνους🡨μισθός\*0.10  **Αλλιώς**\_**αν** έτη<=6 **τότε**  μπόνους🡨μισθός\*0.15  **Αλλιώς\_αν** έτη<=10 **τότε**  μπόνους🡨μισθός\*0.18  **Αλλιώς**  μπόνους🡨μισθός\*0.20  **Τέλος**\_**αν**  **Εμφάνισε** μπόνους |

**Άσκηση 5:** Να μετατρέψετε τις ακόλουθες δομές «ΕΠΙΛΕΞΕ» χρησιμοποιώντας την δομή «ΑΝ».

|  |  |
| --- | --- |
| **Διάβασε** κ  **Επίλεξε** κ  **Περίπτωση** 1, 2, 3  λ🡨κ+4  **Περίπτωση** >15  λ🡨κ-2  **Περίπτωση** **Αλλιώς**  λ🡨κ\*3  **Τέλος**\_**επιλογών**  **Εμφάνισε** λ | κ🡨5  **Επίλεξε** κ  **Περίπτωση** 10  μ🡨κ\*2  **Περίπτωση** 3  μ🡨κ\*3  **Περίπτωση** **Αλλιώς**  μ🡨κ+10  κ🡨κ+μ  **Τέλος\_επιλογών**  **Εμφάνισε** κ,μ |

**Άσκηση 6:** Να αναπτύξετε αλγόριθμο ο οποίος: **1)** θα διαβάζει έναν αριθμό από τον χρήση **2)** θα εμφανίζει μήνυμα σχετικά με το αν είναι πολλαπλάσιο του 20 ή όχι

**Άσκηση 7:** ο πληθυσμός των 4 νομών της Κρήτης παρουσιάζονται στον ακόλουθο πίνακα. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο: **1)** θα διαβάζει ένα νομό από τον χρήση **2)** στην περίπτωση που είναι ένας από τους 4 νομούς της Κρήτης θα εμφανίζει τον πληθυσμό του, διαφορετικά θα εμφανίζει «Λάθος Νομός».

|  |  |
| --- | --- |
| **Νομός** | **Πληθυσμός** |
| Ηράκλειο | 351145 |
| Ρέθυμνο | 120336 |
| Χανιά | 195934 |
| Άγιος Νικόλαος | 98038 |

**Άσκηση 8:** Μία εταιρεία κινητής τηλεφωνίας ακολουθεί ανά μήνα την πολιτική τιμών που φαίνονται στον παρακάτω πίνακα (κλιμακωτή χρέωση):

|  |  |
| --- | --- |
| **Πάγιο** 5 ευρώ / **Χρέωση μηνύματος** 0.07 ευρώ ανά μήνυμα | |
| **Χρόνος τηλεφωνημάτων (δευτερόλεπτα)** | **Χρονοχρέωση (ευρώ/ δευτερόλεπτο)** |
| 1 – 200 | 0.005 |
| 201 – 500 | 0.003 |
| 501 – 700 | 0.002 |
| 701 και πάνω | 0.001 |

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο: **1)** θα διαβάζει τη χρονική διάρκεια των τηλεφωνημάτων ενός συνδρομητή και τον αριθμό μηνυμάτων που έστειλε **2)** θα υπολογίζει και θα εμφανίζει τη μηνιαία χρέωση του συνδρομητή

**Άσκηση 9:** Μία καφετέρια πληρώνει τους υπαλλήλους της ανάλογα με την ημέρα που θα εργαστούν, σύμφωνα με τον ακόλουθο πίνακα. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο: **1)** θα διαβάζει για έναν υπάλληλο την ημέρα που δούλεψε την εβδομάδα αυτή (υποθέστε δούλεψε μόνο μία ημέρα) **2)** θα εμφανίζει το ποσό που θα λάβει.

|  |  |
| --- | --- |
| **Ημέρα** | **Ποσό** |
| Από Δευτέρα μέχρι και Τετάρτη | 35 |
| Πέμπτη και Παρασκευή | 45 |
| Σάββατο και Κυριακή | 60 |

**Ενότητα 1 Συμπληρωματικού υλικού: Στοίβα, Ουρά, Άλλες δομές δεδομένων**

**Ενότητα 1.1: Η δομή δεδομένων «Στοίβα».**

1. Ποια δομή δεδομένων ονομάζεται στοίβα;

**(Ενότητα 1.1, Συμπληρωματικό Υλικό) Στοίβα (stack)**, ονομάζεται μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που βρίσκονται στην κορυφή της στοίβας λαμβάνονται πρώτα, ενώ αυτά που βρίσκονται στο βάθος της στοίβας λαμβάνονται τελευταία.

1. Πως ονομάζεται η μέθοδος λειτουργίας μίας στοίβας; Δώστε ένα παράδειγμα από την καθημερινή σας ζωή.

**(Ενότητα 1.1, Συμπληρωματικό Υλικό)** Η μέθοδος λειτουργίας της στοίβας ονομάζεται «Τελευταίο μέσα πρώτο Έξω» (LIFO – Last In First Out). Χαρακτηριστικό παράδειγμα είναι η στοίβα με τα πιάτα. Όταν τα πλένουμε, τα τοποθετούμε το ένα πάνω στο άλλο και χρησιμοποιούμε πρώτο το πιάτο που βρίσκεται πάνω-πάνω (δηλαδή αυτό που μπήκε τελευταίο στη στοίβα).

1. Ποιες είναι οι κύριες λειτουργίες που εφαρμόζονται σε μία στοίβα;

**(Ενότητα 1.1, Συμπληρωματικό Υλικό)** Οι δύο βασικές λειτουργίες που εφαρμόζονται σε μία στοίβα είναι:

1. Η **ώθηση** (push) στοιχείου στην κορυφή της στοίβας. Στη διαδικασία της ώθησης ελέγχουμε αν η στοίβα είναι γεμάτη. Στην περίπτωση που προσπαθήσουμε να «προσθέσουμε» ένα στοιχείο σε μια ήδη γεμάτη στοίβα, έχουμε **υπερχείλιση** (overflow) της στοίβας.
2. Η **απώθηση** (pop) στοιχείου από τη στοίβα. Στη διαδικασία της απώθησης ελέγχουμε αν υπάρχει ένα τουλάχιστον στοιχείο στη στοίβα. Στην περίπτωση που προσπαθήσουμε να «αφαιρέσουμε» ένα στοιχείο από μία κενή στοίβα, έχουμε **υποχείλιση** (underflow) της στοίβας.

**Παράδειγμα 1 – Υλοποίηση στοίβας με μονοδιάστατο πίνακα:** Να δώσετε ένα παράδειγμα υλοποίησης στοίβας με μονοδιάστατο πίνακα, εκτελώντας τις διαδικασίες της ώθησης και της απώθησης.

|  |  |  |
| --- | --- | --- |
| **Αρχική μορφή στοίβας** | **Ώθηση του στοιχείου 3 στην αρχική στοίβα** | **Απώθηση στην αρχική στοίβα.** |
| |  |  |  | | --- | --- | --- | | 6 |  |  | | 5 |  |  | | 4 | 12 | top=4 | | 3 | -5 |  | | 2 | 0 |  | | 1 | 10 |  |   Χρησιμοποιούμε έναν δείκτη (top) που δείχνει το τελευταίο στοιχείο που μπήκε στην στοίβα. | |  |  |  | | --- | --- | --- | | 6 |  |  | | 5 | 3 | top=5 | | 4 | 12 |  | | 3 | -5 |  | | 2 | 0 |  | | 1 | 10 |  |   Ώθηση του στοιχείου 3 στην αρχική στοίβα.  Αυξάνουμε τον δείκτη top κατά 1 (top🡨top+1) και στη συνέχεια γίνεται η ώθηση του στοιχείου. | |  |  |  | | --- | --- | --- | | 6 |  |  | | 5 |  |  | | 4 |  |  | | 3 | -5 | top=3 | | 2 | 0 |  | | 1 | 10 |  |   Εξάγεται το στοιχείο που δείχνει η μεταβλητή top και στην συνέχεια η μεταβλητή top μειώνεται κατά 1 (top🡨top-1). |

**Παράδειγμα 2 – Θεωρητικό παράδειγμα στοίβας:** Σε μία στοίβα 6 θέσεων έχουν τοποθετηθεί με την σειρά οι χαρακτήρες Σ,Η,Κ,Λ **1)** Να σχεδιάσετε την μορφή της στοίβας και να προσδιορίσετε την τιμή του δείκτη top **2)** Να εκτελέσετε τις ακόλουθες ενέργειες και να παρουσιάσετε την μορφή της στοίβας μετά τις ενέργειες: απώθηση, απώθηση, ώθηση Ρ, ώθηση Α.

|  |  |  |
| --- | --- | --- |
| **Αρχική μορφή στοίβας** | **Απώθηση** | **Απώθηση** |
| |  |  |  | | --- | --- | --- | | 6 |  |  | | 5 |  |  | | 4 | Λ | top=4 | | 3 | Κ |  | | 2 | Η |  | | 1 | Σ |  | | |  |  |  | | --- | --- | --- | | 6 |  |  | | 5 |  |  | | 4 |  |  | | 3 | Κ | top=3 | | 2 | Η |  | | 1 | Σ |  | | |  |  |  | | --- | --- | --- | | 6 |  |  | | 5 |  |  | | 4 |  |  | | 3 |  |  | | 2 | Η | top=5 | | 1 | Σ |  | |

|  |  |  |
| --- | --- | --- |
| **Αρχική μορφή στοίβας** | **Απώθηση** | **Απώθηση** |
| |  |  |  | | --- | --- | --- | | 6 |  |  | | 5 |  |  | | 4 | Λ |  | | 3 | Κ |  | | 2 | Η |  | | 1 | Σ |  | | |  |  |  | | --- | --- | --- | | 6 |  |  | | 5 |  |  | | 4 |  |  | | 3 | Κ | top=3 | | 2 | Η |  | | 1 | Σ |  | | |  |  |  | | --- | --- | --- | | 6 |  |  | | 5 |  |  | | 4 |  |  | | 3 |  |  | | 2 | Η | top=5 | | 1 | Σ |  | |
| **Ώθηση Ρ** | **Ώθηση Α** |  |
| |  |  |  | | --- | --- | --- | | 6 |  |  | | 5 |  |  | | 4 |  |  | | 3 | Ρ | top=3 | | 2 | Η |  | | 1 | Σ |  | | |  |  |  | | --- | --- | --- | | 6 |  |  | | 5 |  |  | | 4 | Α | top=4 | | 3 | Ρ |  | | 2 | Η |  | | 1 | Σ |  | |  |

**Παράδειγμα 3 – Υλοποίηση ώθησης και απώθησης σε κώδικα:** Να γράψετε τμήμα αλγορίθμου το οποίο θα υλοποιεί τις λειτουργίες της ώθησης και της απώθησης σε μία στοίβα A 5 θέσεων.

|  |  |
| --- | --- |
| **Κώδικας για Ώθηση στοιχείου** | **Παρατηρήσεις** |
| **Διάβασε** στοιχείο  **Αν** top<5 **τότε ! έλεγχος για γεμάτη στοίβα**  top🡨top+1 **! αύξηση δείκτη top κατά 1**  A[top]🡨στοιχείο **! τοποθέτηση στοιχείου**  **Αλλιώς**  **Γράψε** ‘Η στοίβα είναι γεμάτη, υπερχείλιση’  **Τέλος\_αν** | Κατά την διαδικασία της ώθησης, θα πρέπει να ελέγξουμε πως η στοίβα δεν είναι γεμάτη, για να μην δημιουργήσουμε υπερχείλιση.  Εφόσον υπάρχει χώρος, πρώτα αυξάνουμε τον δείκτη top κατά 1 και στη νέα θέση τοποθετούμε το στοιχείο. |

|  |  |
| --- | --- |
| **Κώδικας για Απώθηση στοιχείου** | **Παρατηρήσεις** |
| **Αν** top>=1 **τότε ! έλεγχος για άδεια στοίβα**  **Γράψε** A[top] **! εμφάνιση στοιχείου**  top🡨top-1 **! μείωση δείκτη top κατά 1**  **Αλλιώς**  **Γράψε** ‘Η στοίβα είναι άδεια, υποχείλιση’  **Τέλος\_αν** | Κατά την διαδικασία της απώθησης, θα πρέπει να ελέγξουμε πως η στοίβα δεν είναι άδεια, για να μην δημιουργήσουμε υποχείλιση.  Εφόσον υπάρχει στοιχείο, πρώτα το εμφανίζουμε και στη συνέχεια μειώνουμε τον δείκτη top κατά 1. |

**Παράδειγμα 4 – Άσκηση με στοίβα:** Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο: **1)** θα χρησιμοποιεί στοίβα 20 θέσεων για αποθήκευση ακέραιων αριθμών **2)** θα διαβάζει επαναληπτικά από τον χρήστη, την λειτουργία που επιθυμεί να εκτελέσει, με έλεγχο δεδομένων για μία εκ των «Ώθηση», «Απώθηση» ή «Τερματισμός» μέχρι να δοθεί ως λειτουργία η λέξη «Τερματισμός» που θα τερματίζει την επανάληψη **3)** στην περίπτωση της ώθησης στοιχείου, αν υπάρχει χώρος στην στοίβα, θα διαβάζει έναν αριθμό και θα τον τοποθετεί στην στοίβα, διαφορετικά θα εμφανίζει «Γεμάτη στοίβα» **4)** στην περίπτωση της απώθησης, θα την εκτελεί στην περίπτωση που η στοίβα δεν είναι άδεια, διαφορετικά θα εμφανίζει μήνυμα «Άδεια στοίβα» 5) μετά το τέλος της επαναληπτικής διαδικασίας θα εμφανίζει: **α)** πόσες φορές γέμισε η στοίβα μετά από κάποια ώθηση **β)** το άθροισμα όλων των στοιχείων που έγιναν απώθηση **γ)** πόσες φορές δεν πραγματοποιήθηκε κάποια απώθηση επειδή η στοίβα ήταν άδεια.

|  |  |
| --- | --- |
| **Κώδικας** | **Παρατηρήσεις** |
| **ΠΡΟΓΡΑΜΜΑ** άσκηση\_στοίβα  **ΜΕΤΑΒΛΗΤΕΣ**  **ΑΚΕΡΑΙΕΣ:** top, sum, γέμισε, άδεια, στοίβα[20], αρ, top  **ΧΑΡΑΚΤΗΡΕΣ**: λειτουργία  **ΑΡΧΗ**  sum🡨0, top🡨0  γέμισε🡨0, άδεια🡨0  **ΑΡΧΗ**\_**ΕΠΑΝΑΛΗΨΗΣ**  **ΔΙΑΒΑΣΕ** επιλογή  **ΜΕΧΡΙΣ**\_**ΟΤΟΥ** επιλογή=’Ώθηση’ Η επιλογή=’Απώθηση’  & Η επιλογή=’Τερματισμός’  **ΟΣΟ** επιλογή<>’Τερματισμός’ **ΕΠΑΝΑΛΑΒΕ**  **ΑΝ** επιλογή=’Ώθηση’ **ΤΟΤΕ**  **ΑΝ** top<20 **ΤΟΤΕ ! υπάρχει χώρος**  **ΔΙΑΒΑΣΕ** αρ  top🡨top+1 **! λειτουργία ώθησης**  στοίβα[top]🡨αρ  **ΑΝ** top=20 **TOTE ! η στοίβα γέμισε**  γέμισε🡨γέμισε+1  **ΤΕΛΟΣ**\_**ΑΝ**  **ΑΛΛΙΩΣ**  **ΓΡΑΨΕ** ‘Γεμάτη στοίβα’  **ΤΕΛΟΣ**\_**ΑΝ**  **ΑΛΛΙΩΣ**\_**ΑΝ** επιλογή = ‘Απώθηση’ **ΤΟΤΕ**  **ΑΝ** top>=1 **TOTE ! υπάρχει στοιχείο**  sum🡨sum+στοίβα[top]  top🡨top-1 **! λειτουργία απώθησης**  **ΑΛΛΙΩΣ**  **ΓΡΑΨΕ** ‘Άδεια στοίβα’  άδεια🡨άδεια+1 **! η στοίβα ήταν άδεια**  **ΤΕΛΟΣ**\_**ΑΝ**  **ΤΕΛΟΣ\_ΑΝ**  **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**  **ΔΙΑΒΑΣΕ** επιλογή  **ΜΕΧΡΙΣ**\_**ΟΤΟΥ** επιλογή=’Ώθηση’ Η επιλογή=’Απώθηση’  & Η επιλογή=’Τερματισμός’  **ΤΕΛΟΣ**\_**ΕΠΑΝΑΛΗΨΗΣ**  **ΓΡΑΨΕ** sum, άδεια, γέμισε  **ΤΕΛΟΣ**\_**ΠΡΟΓΡΑΜΜΑΤΟΣ** | Χρήσιμες συνθήκες για την δομή της στοίβας Ν θέσεων:  **Γεμάτη Στοίβα**🡪 top=N  **Άδεια Στοίβα**🡪 top=0  **Έλεγχος αν υπάρχει χώρος για ώθηση**🡪top<N  **Έλεγχος αν υπάρχει κάποιο στοιχείο για απώθηση**🡪top>=1  Στοιχεία που υπάρχουν ανά πάσα στιγμή  στην στοίβα 🡪 top |

**ΑΣΚΗΣΕΙΣ ΓΙΑ ΛΥΣΗ**

**Άσκηση 1:** Ερωτήσεις Σωστό – Λάθος.

1. Σε μία στοίβα, το στοιχείο που τοποθετείται τελευταίο σε αυτή, βγαίνει πρώτο.
2. Μία στοίβα χρησιμοποιεί την μέθοδο «Πρώτο Μέσο – Πρώτο Έξω».
3. Μία στοίβα δεν μπορεί να υλοποιηθεί με μονοδιάστατο πίνακα.
4. Οι βασικές λειτουργίες μίας στοίβας είναι η ώθηση και η απώθηση.
5. Στην διαδικασία την ώθησης, πρέπει να προσέξουμε την περίπτωση της υποχείλισης.
6. Στην διαδικασία της απώθησης, πρέπει να ελέγξουμε αν υπάρχει ένα τουλάχιστον στοιχείο στην στοίβα.
7. Για την υλοποίηση μίας στοίβας, χρειαζόμαστε έναν δείκτη ο οποίος «δείχνει» το στοιχείο που τοποθετήθηκε τελευταίο στη στοίβα.
8. Σε μία στοίβα Ν θέσεων, αν ισχύει «top=N», τότε η στοίβα είναι γεμάτη.
9. Η λειτουργία της ώθησης μπορεί να πραγματοποιηθεί ακόμη και στην περίπτωση που η στοίβα είναι άδεια.
10. Σε μία στοίβα χρειάζονται δύο δείκτες για την υλοποίηση της με χρήση ενός μονοδιάστατου πίνακα.

**Άσκηση 2:**  Σε μία στοίβα 8 θέσεων έχουν τοποθετηθεί με τη σειρά οι αριθμοί: 5, 10, -3, 8. **1)** να σχεδιάσετε την στοίβα και να προσδιορίσετε την τιμή του δείκτη top **2)** να παρουσιάσετε την μορφή της στοίβας μετά από καθεμία από τις παρακάτω ενέργειες: ώθηση 0, ώθηση 10, απώθηση, ώθηση 9, ώθηση 7.

**Άσκηση 3:**  Σε μία στοίβα 5 θέσεων έχουν τοποθετηθεί με την σειρά οι αριθμοί 1,2,3. **1)** να σχεδιάσετε την στοίβα και να προσδιορίσετε την τιμή του δείκτη top **2)** αν θέλετε να τοποθετήσετε στην στοίβα τον αριθμό 4, ποια λειτουργία θα χρησιμοποιήσετε; Πως θα μεταβληθεί ο δείκτης top; Να σχεδιάσετε τη νέα μορφή της στοίβας **3)** αν (στην αρχική στοίβα) θέλετε να αφαιρέσετε ένα στοιχείο, ποια λειτουργία θα χρησιμοποιήσετε; Πως θα μεταβληθεί ο δείκτης top; Να σχεδιάσετε τη νέα μορφή της στοίβας.

**Άσκηση 4:** Σε μία στοίβα 6 θέσεων έχουν τοποθετηθεί με τη σειρά οι αριθμοί 100 και 200. **1)** να σχεδιάσετε την στοίβα και να προσδιορίσετε την τιμή του δείκτη top **2)** στην συγκεκριμένη στοίβα, θα μπορούσατε να εκτελέσετε τις ενέργειες: απώθηση, ώθηση του 300, απώθηση, απώθηση, απώθηση; Να αιτιολογήσετε την απάντηση σας **3)** πόσες συνεχόμενες φορές μπορείτε να εκτελέσετε την λειτουργία της ώθησης στην αρχική στοίβα; Να αιτιολογήσετε την απάντηση σας.

**Άσκηση 5:** Δίνεται στοίβα Ν θέσεων. **1)** να γράψετε τις εντολές που υλοποιούν την λειτουργία της ώθησης, με έλεγχο για υπερχείλιση της στοίβας, όπου θα εμφανίζεται το μήνυμα «Γεμάτη στοίβα» **2)** να γράψετε τις εντολές που υλοποιούν την λειτουργία της απώθησης, με έλεγχο για υποχείλιση της στοίβας, όπου θα εμφανίζεται το μήνυμα «Άδεια στοίβα».

**Άσκηση 6:** Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο: **1)** θα χρησιμοποιεί στοίβα Α[30] για αποθήκευση ονομάτων **2)** θα εκτελεί 100 φορές την ακόλουθη επαναληπτική διαδικασία: **α)** θα ρωτάει τον χρήστη την λειτουργία που επιθυμεί να εκτελέσει, με έλεγχο δεδομένων για τιμές «Ώθηση» ή «Απώθηση» **β)** στην περίπτωση της ώθησης, θα ελέγχει αν υπάρχει χώρος στην στοίβα, θα διαβάζει ένα όνομα και θα το τοποθετεί στην στοίβα, διαφορετικά θα εμφανίζει μήνυμα «Γεμάτη στοίβα» **γ)** στην περίπτωση της απώθησης, θα ελέγχει αν υπάρχει κάποιο στοιχείο στην στοίβα, θα το εμφανίζει και θα το βγάζει από την στοίβα, διαφορετικά θα εμφανίζει μήνυμα «Άδεια στοίβα» **3)** στο τέλος θα εμφανίζει πόσες φορές πραγματοποιήθηκε ώθηση στοιχείου και πόσες φορές μετά από κάποια απώθηση η στοίβα ήταν άδεια.

**Άσκηση 7:** Να αναπτύξετε αλγόριθμο ο οποίος: **1)** θα χρησιμοποιεί στοίβα Σ[100] για αποθήκευση ακέραιων αριθμών **2)** θα διαβάζει επαναληπτικά από τον χρήστη την λειτουργία που επιθυμεί να εκτελέσει με έλεγχο δεδομένων για τιμές «Α» για απώθηση, «Ω» για ώθηση και «Τ» για τερματισμό λειτουργίας **3)** στην περίπτωση της ώθησης, θα ελέγχει αν υπάρχει χώρος στην στοίβα, θα διαβάζει έναν αριθμό και θα τον τοποθετεί στην στοίβα, διαφορετικά θα εμφανίζει μήνυμα «Γεμάτη στοίβα» **4)** στην περίπτωση της απώθησης, θα ελέγχει αν υπάρχει κάποιο στοιχείο στην στοίβα, θα το εμφανίζει και θα το βγάζει από την στοίβα, διαφορετικά θα εμφανίζει μήνυμα «Άδεια στοίβα» **5)** η παραπάνω επαναληπτική διαδικασία θα ολοκληρωθεί όταν δοθεί ως λειτουργία το «Τ» **6)** στο τέλος θα εμφανίζει: **α)** το άθροισμα των αριθμών που έγιναν ώθηση στην στοίβα **β)** πόσες φορές πραγματοποιήθηκε η λειτουργία της απώθησης **γ)** μήνυμα «Άδεια στοίβα» αν η στοίβα είναι άδεια, διαφορετικά θα εμφανίζει πόσα στοιχεία έχει μέσα η στοίβα.

**Άσκηση 8:**  Ένας μαθητής χρησιμοποιεί στοίβα εργασίες[10] για την διαχείριση των εργασιών του. Όταν έχει μία νέα εργασία, την τοποθετεί στην κορυφή της στοίβας, την οποία και επιλύει πρώτη. Τι εργασίες που έρχονται ενώ η στοίβα του είναι γεμάτη, ο μαθητής έχει συμφωνήσει να τις χρωστάει στον καθηγητή. Η στοίβα του μαθητή έχει ήδη 4 εργασίες μέσα. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο: **1)** θα διαβάζει από τον χρήστη τα ονόματα των 4 εργασιών που υπάρχουν ήδη μέσα στην στοίβα **2)** θα διαβάζει επαναληπτικά από τον μαθητή την λειτουργία που επιθυμεί να εκτελέσει με έλεγχο δεδομένων για τιμές «Π» για προσθήκη εργασίας, «Ε» για επίλυση εργασίας **3)** στην περίπτωση της προσθήκης εργασίας, θα ελέγχει αν υπάρχει χώρος στην στοίβα, θα διαβάζει το όνομα της και θα την τοποθετεί στην στοίβα, διαφορετικά θα εμφανίζει μήνυμα «Θα την χρωστάω» **4)** στην περίπτωση της επίλυσης, θα ελέγχει αν υπάρχει κάποια εργασία στην στοίβα, θα την εμφανίζει και θα την βγάζει από την στοίβα, διαφορετικά θα εμφανίζει μήνυμα «Τέλος οι εργασίες» **5)** η παραπάνω επαναληπτική διαδικασία θα ολοκληρωθεί όταν ο μαθητής επιλύσει όλες τις εργασίες που έχει στην στοίβα του **6)** στο τέλος θα εμφανίζει: **α)** πόσες εργασίες χρωστάει ο μαθητής **β)** πόσες εργασίες επίλυσε **γ)** πόσες φορές ο μαθητής είχε 5 εργασίες στην στοίβα του μετά από την προσθήκη ή την επίλυση κάποιας εργασίας.

**Άσκηση 9:** Ένα εστιατόριο διαθέτει ένα σύγχρονο πλυντήριο πιάτων το οποίο χωράει μέχρι και 200 πιάτα. Τα πιάτα τοποθετούνται το ένα πάνω στο άλλο και πλένονται όλα όσα υπάρχουν στην στοίβα, όταν ο υπάλληλος επιλέξει την αντίστοιχη ενέργεια. Κάθε πιάτο χρειάζεται 20 δευτερόλεπτα για να πλυθεί. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο: **1)** θα χρησιμοποιεί στοίβα πιάτα[200] για την διαχείριση του πλυντηρίου **2)** θα διαβάζει από τον υπάλληλο του εστιατορίου πόσα πιάτα υπάρχουν ήδη μέσα στο πλυντήριο, εξασφαλίζοντας πως θα είναι από 30 μέχρι 100 και θα εισάγει τον χαρακτήρα ‘Π’ στην στοίβα για τα πιάτα αυτά **3)** θα εκτελεί επαναληπτικά τα ακόλουθα: **α)** θα εμφανίζει το ακόλουθο μενού επιλογών: 1.Εισαγωγή πιάτου 2. Πλύσιμο όλων των πιάτων 3. Έξοδος **β)** θα διαβάζει την απάντηση του υπαλλήλου εξασφαλίζοντας πως θα λάβει μία από τις τιμές 1,2 ή 3 **γ)** στην περίπτωση της εισαγωγής πιάτου, θα ελέγχει αν υπάρχει χώρος στο πλυντήριο, οπότε το εισάγει και θα τοποθετεί το γράμμα ‘Π’ στην στοίβα, διαφορετικά θα εμφανίζει μήνυμα «Γεμάτο πλυντήριο». Στην περίπτωση που το πλυντήριο είναι γεμάτο, θα ρωτάει τον χρήστη με κατάλληλο μήνυμα αν επιθυμεί να πλύνει τα πιάτα, θα διαβάζει την απάντηση του με έλεγχο για τιμές «ναι» / «όχι» και αν απαντήσει θετικά θα πλένει όλα τα πιάτα **δ)** στην περίπτωση που ο υπάλληλος επιθυμεί να πλύνει όλα τα πιάτα, το πλυντήριο θα αδειάζει **ε)** στην περίπτωση που επιλέξει έξοδο από το πρόγραμμα, θα εμφανίζει μήνυμα «Πραγματοποιείται Έξοδος» **στ)** η παραπάνω επαναληπτική διαδικασία θα ολοκληρωθεί όταν ο υπάλληλος επιλέξει την έξοδο από το πρόγραμμα **4)** μετά το τέλος της επαναληπτικής διαδικασίας θα εμφανίζει: **α)** τον συνολικό χρόνο που χρειάστηκε το πλυντήριο για το πλύσιμο των πιάτων σε μορφή «Ώρες:Λεπτά:Δευτερόλεπτα» **β)** τον μεγαλύτερο αριθμό πιάτων που έπλυνε το πλυντήριο.

**Άσκηση 10 (παραλλαγή στοίβας):** Το παρκινγκ στο αεροδρόμιο Ηρακλείου μπορεί να φιλοξενήσει μέχρι 300 χρησιμοποιεί δύο στοίβες για την διαχείριση των αυτοκινήτων: α) στοίβα ΑΚ[300] για τον αριθμό κυκλοφορίας του αυτοκινήτου και στοίβα ΗΜ[300] για τις ημέρες παραμονής στο παρκινγκ. Κάθε ημέρα κοστίζει 3 ευρώ, ενώ αν οι ημέρες είναι περισσότερες από 15, υπάρχει έκπτωση 10%. Να αναπτύξετε αλγόριθμο ο οποίος: **1)** θα διαβάζει από τον υπάλληλο του παρκινγκ πόσα αυτοκίνητα υπάρχουν ήδη μέσα στο παρκινγκ, εξασφαλίζοντας πως δώσει έγκυρη τιμή. Για κάθε ένα από τα αυτοκίνητα αυτά θα διαβάζει τον αριθμό κυκλοφορίας του και τις ημέρες παραμονής στο παρκινγκ και θα ενημερώνει κατάλληλα τις στοίβες **2)** θα εκτελεί επαναληπτικά τα ακόλουθα: **α)** θα εμφανίζει το ακόλουθο μενού επιλογών: 1. Είσοδος αυτοκινήτου 2. Έξοδος αυτοκινήτου 3. Έξοδος **β)** θα διαβάζει την απάντηση του υπαλλήλου εξασφαλίζοντας πως θα λάβει μία από τις τιμές 1,2 ή 3 **γ)** στην περίπτωση της εισόδου αυτοκινήτου, θα ελέγχει αν υπάρχει χώρος στο παρκινγκ, οπότε θα διαβάζει τον αριθμό κυκλοφορίας του και τις ημέρες παραμονής, ενημερώνοντας κατάλληλα τις στοίβες, διαφορετικά θα εμφανίζει μήνυμα «Γεμάτο παρκινγκ» **δ)** στην περίπτωση εξόδου αυτοκινήτου, θα ελέγχει αν υπάρχει κάποιο αυτοκίνητο στο παρκινγκ, διαφορετικά θα εμφανίζει «Άδειο παρκινγκ». Αν η έξοδος πραγματοποιηθεί, θα διαβάζει τον αριθμό κυκλοφορίας του αυτοκινήτου, θα τον αναζητεί στην στοίβα και θα εμφανίζει το ποσό που θα πληρώσει ο οδηγός. Τέλος, θα μεταφέρει κατάλληλα όσα αυτοκίνητα χρειάζονται, ώστε να είναι ξανά σε μορφή στοίβας **ε)** στην περίπτωση που επιλέξει έξοδο από το πρόγραμμα, θα εμφανίζει μήνυμα «Πραγματοποιείται Έξοδος» **στ)** η παραπάνω επαναληπτική διαδικασία θα ολοκληρωθεί όταν ο υπάλληλος επιλέξει την έξοδο από το πρόγραμμα **3)** μετά το τέλος της επαναληπτικής διαδικασίας θα εμφανίζει: **α)** τα συνολικά έσοδα του παρκινγκ **β)** πόσες φορές το παρκινγκ γέμισε μετά από κάποια είσοδο αυτοκινήτου και πόσες φορές άδειασε μετά την έξοδο κάποιου αυτοκινήτου.

**Άσκηση 11:** Να αναπτύξετε αλγόριθμο ο οποίος: **1)** θα χρησιμοποιεί στοίβα Σ[20] για την αποθήκευση ακέραιων αριθμών **2)** θα διαβάζει από τον χρήστη πόσα στοιχεία υπάρχουν ήδη μέσα στην στοίβα, εξασφαλίζοντας πως η στοίβα μπορεί να είναι άδεια, να έχει κάποια στοιχεία ή να είναι γεμάτη **3)** θα διαβάζει από τον χρήστη ποια επιλογή επιθυμεί να εκτελέσει, εξασφαλίζοντας πως θα είναι μία από τις «Άδειασμα Στοίβας» ή «Γέμισμα Στοίβας» **4)** στην περίπτωση που ο χρήστης επιλέξει «Άδειασμα Στοίβας», θα εμφανίζει όλα τα στοιχεία που είναι στην στοίβα, θα τα απωθεί όλα και θα εμφανίζει το άθροισμά τους **5)** στην περίπτωση που ο χρήστης επιλέξει «Γέμισμα Στοίβας», θα διαβάζει επαναληπτικά αριθμούς μέχρι να γεμίσει η στοίβα, εξασφαλίζοντας πως είναι θετικοί και θα εμφανίζει τον μεγαλύτερο αριθμό που δόθηκε.

**Ενότητα 1.2: Η δομή δεδομένων «Ουρά».**

1. Ποια δομή δεδομένων ονομάζεται «ουρά»;

**(Συμπληρωματικό υλικό, ενότητα 1.2) Ουρά (Queue),** ονομάζεται μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που τοποθετήθηκαν πρώτα στην ουρά να λαμβάνονται επίσης πρώτα.

1. Πως ονομάζεται η μέθοδος λειτουργίας μίας ουράς; Να παρουσιάσετε ένα παράδειγμα ουράς από τη καθημερινής σας ζωή.

**(Συμπληρωματικό υλικό, ενότητα 1.2)** Η μέθοδος λειτουργίας της ουράς ονομάζεται Πρώτο Μέσα – Πρώτο Έξω ( FIFO – First In First Out). Ένα παράδειγμα είναι η ουρά στην τράπεζα, στα ταμεία ενός σούπερ μάρκετ κτλ.

1. Ποιες είναι οι βασικές λειτουργίες που εφαρμόζονται σε μία ουρά;

**(Συμπληρωματικό υλικό, ενότητα 1.2)** Οι δύο βασικές λειτουργίες που εφαρμόζονται σε μία ουρά είναι:

* Η εισαγωγή (enqueue) ενός στοιχείου στο πίσω άκρο της ουράς. Θα πρέπει να προσέξουμε να μην είναι γεμάτη η ουρά.
* Η εξαγωγή (dequeue) ενός στοιχείου από το εμπρός άκρο της ουράς. Θα πρέπει να προσέξουμε να υπάρχει κάποιο στοιχείο στην ουρά.

**Παράδειγμα 1 – Υλοποίηση ουράς με μονοδιάστατο πίνακα:** Να δώσετε ένα παράδειγμα υλοποίησης ουράς με μονοδιάστατο πίνακα, εκτελώντας τις διαδικασίες της εισαγωγής και της εξαγωγής.

|  |  |  |
| --- | --- | --- |
| **Αρχική μορφή ουράς** | **Εισαγωγή του στοιχείου 7 στην αρχική ουρά** | **Εξαγωγή στην αρχική ουρά** |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 1 | 2 | 3 | 4 | 5 | 6 | |  | 5 | 2 | 3 |  |  |     εμπρός πίσω  Χρησιμοποιούμε δύο δείκτες: τον εμπρός (ή front) που δείχνει την θέση του πρώτου στοιχείου της ουράς και τον πίσω (ή rear) που δείχνει τη θέση του τελευταίου στοιχείου. Ως αρχικές τιμές τους θεωρούμε το 0. | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 1 | 2 | 3 | 4 | 5 | 6 | |  | 5 | 2 | 3 | 7 |  |     εμπρός πίσω  Η εισαγωγή ενός στοιχείου γίνεται στο πίσω άκρο της ουράς. Η τιμή του δείκτη πίσω αυξάνεται κατά 1 (πίσω🡨πίσω+1) και στη νέα θέση γίνεται η εισαγωγή του στοιχείου. | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 1 | 2 | 3 | 4 | 5 | 6 | |  |  | 2 | 3 |  |  |   εμπρός πίσω  Η εξαγωγή ενός στοιχείου γίνεται από το μπροστά άκρο της ουράς. Στην ουσία αυξάνεται η τιμή του δείκτη εμπρός κατά 1 (εμπρός🡨εμπρός+1), ο οποίος δείχνει στην επόμενη θέση, δίχως να διαγράφεται στην πραγματικότητα κάποιο στοιχείο. |

**Παράδειγμα 2 – Θεωρητικό παράδειγμα ουράς:** Σε μία ουρά 6 θέσεων έχουν τοποθετηθεί με την σειρά αριθμοί 8,2,4 **1)** Να σχεδιάσετε την μορφή της ουράς και να προσδιορίσετε την τιμή των δεικτών εμπρός και πίσω **2)** Να εκτελέσετε τις ακόλουθες ενέργειες και να παρουσιάσετε την μορφή της ουράς μετά τις ενέργειες: εξαγωγή, εισαγωγή 3, εισαγωγή 9, εξαγωγή.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 8 | 2 | 4 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
|  | 2 | 4 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
|  | 2 | 4 | 3 |  |  |

εμπρος=1 εξαγωγή εισαγωγή 3

πίσω=3 εμπρός=2 εμπρός=2

πίσω=3 πίσω=4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
|  | 2 | 4 | 3 | 9 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
|  |  | 4 | 3 | 9 |  |

εισαγωγή 9 εξαγωγή

εμπρός=2 εμπρός=3

πίσω=5 πίσω=5

**Παράδειγμα 3 – Υλοποίηση εισαγωγής και εξαγωγής σε κώδικα:** Να γράψετε τμήμα αλγορίθμου το οποίο θα υλοποιεί τις λειτουργίες της εισαγωγής και της εξαγωγής σε μία ουράς A 5 θέσεων.

|  |  |
| --- | --- |
| **Κώδικας για εισαγωγή στοιχείου** | **Παρατηρήσεις** |
| **Διάβασε** στοιχείο  **Αν** πίσω= 5 **τότε ! έλεγχος για γεμάτη ουρά**  **Γράψε** ‘γεμάτη ουρά’  **Αλλιώς\_αν** πίσω=0 **και** εμπρός=0 **τότε ! άδεια ουρά**  εμπρός🡨1 **! οι δείκτες εμπρός και πίσω**  πίσω🡨1 **! γίνονται 1**  Α[πίσω]🡨στοιχείο **! και εισάγεται το στοιχείο**  **Αλλιώς ! υπάρχει χώρος για το στοιχείο**  πίσω🡨πίσω+1 **! ο δείκτης πίσω αυξάνεται + 1**  Α[πίσω]🡨στοιχείο **! και εισάγεται το στοιχείο**  **Τέλος\_αν**  **Παρατήρηση:** επειδή στην ουρά η εισαγωγή γίνεται μόνο από το πίσω άκρος της, η συνθήκη πίσω=μέγεθος\_ουράς σημαίνει πως είναι γεμάτη, ακόμη και αν υπάρχει χώρος στο μπροστά άκρο. | Κατά την διαδικασία της εισαγωγής θα πρέπει αρχικά να ελέγξουμε να μην είναι γεμάτη η ουρά. Αν δεν είναι, διακρίνουμε δύο περιπτώσεις:   1. Να είναι άδεια, οπότε αρχικοποιούμε τους δύο δείκτες με 1 και εισάγουμε το στοιχείο. 2. Να έχει χώρο για το στοιχείο, οπότε αυξάνουμε τον δείκτη πίσω κατά 1 και στη νέα θέση εισάγουμε το στοιχείο. |

|  |  |
| --- | --- |
| **Κώδικας για εξαγωγή στοιχείου** | **Παρατηρήσεις** |
| **Αν** εμπρός=0 και πίσω=0 **τότε ! άδεια ουρά**  **Γράψε** ‘Η ουρά είναι άδεια’  **Αλλιώς\_αν** εμπρός=πίσω **τότε ! ένα μόνο στοιχείο**  **Γράψε** ‘Εξάγεται το στοιχείο’, Α[εμπρός]  εμπρός🡨0 **! οι δείκτες εμπρός και πίσω**  πίσω🡨0 **! αρχικοποιούνται με 0 (άδεια ουρά)**  **Αλλιώς ! υπάρχουν και άλλα στοιχεία**  **Γράψε ‘** Εξάγεται το στοιχείο’, Α[εμπρός]  εμπρός🡨εμπρός+1  **Τέλος\_αν** | Κατά την διαδικασία της εξαγωγής, θα πρέπει να ελέγξουμε πως η ουρά δεν είναι άδεια. Αν δεν είναι, διακρίνουμε δύο περιπτώσεις:   1. Να υπάρχει μόνο ένα στοιχείο, οπότε οι δείκτες εμπρός και πίσω αρχικοποιούνται με 0, καθώς η ουρά αδειάζει. 2. Να υπάρχουν και άλλα στοιχεία, οπότε ο δείκτης εμπρός αυξάνεται κατά. |

**Παράδειγμα 4 – Άσκηση με ουρά:** Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο: **1)** θα χρησιμοποιεί ουρά 30 θέσεων για αποθήκευση ακέραιων αριθμών **2)** θα διαβάζει επαναληπτικά από τον χρήστη, την λειτουργία που επιθυμεί να εκτελέσει, με έλεγχο δεδομένων μία εκ των 1 (εισαγωγή) ή 2 (εξαγωγή) **3)** στην περίπτωση της εισαγωγής στοιχείου, αν υπάρχει χώρος στην ουρά, θα διαβάζει έναν αριθμό και θα τον τοποθετεί στην ουρά, διαφορετικά θα εμφανίζει «Γεμάτη ουρά» **4)** στην περίπτωση της εξαγωγής, θα την εκτελεί στην περίπτωση που η ουρά δεν είναι άδεια, διαφορετικά θα εμφανίζει μήνυμα «Άδεια ουρά» 5) στο τέλος θα ρωτάει τον χρήστη αν επιθυμεί να συνεχίσει με νέα λειτουργία, θα διαβάζει την απάντηση του και η επαναληπτική διαδικασία θα τερματιστεί όταν δοθεί ως απάντηση το «όχι» 6) μετά το τέλος της επαναληπτικής διαδικασίας θα εμφανίζει: **α)** το μήνυμα «Άδεια ουρά», στην περίπτωση που η ουρά είναι άδεια, διαφορετικά θα εμφανίζει πόσα στοιχεία βρίσκονται σε αυτή **β)** πόσες φορές άδειασε η ουρά μετά από κάποια εξαγωγή **γ)** το άθροισμα των αριθμών που εξήχθησαν από την ουρά.

|  |  |
| --- | --- |
| **Κώδικας** | **Παρατηρήσεις** |
| **ΠΡΟΓΡΑΜΜΑ** ασκηση\_ουρά  **ΜΕΤΑΒΛΗΤΕΣ**  **ΑΚΕΡΑΙΕΣ**: εμπρός, πίσω, Β[30], x ,άδειασε, sum, **&**λειτουργία, στοιχεία  **ΧΑΡΑΚΤΗΡΕΣ**: απάντηση  **ΑΡΧΗ**  εμπρός 🡨0  πίσω 🡨0  άδειασε🡨0  sum🡨0  **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**  **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**  **ΔΙΑΒΑΣΕ** λειτουργία  **ΜΕΧΡΙΣ\_ΟΤΟΥ** λειτουργία=1 ή λειτουργία=2  **ΑΝ** λειτουργία=1 **ΤΟΤΕ**  **ΔΙΑΒΑΣΕ** x  **ΑΝ** εμπρός=0 **και** πίσω=0 **ΤΟΤΕ ! άδεια ουρά**  εμπρός🡨1  πίσω🡨1  Β[πίσω]🡨x  **ΑΛΛΙΩΣ\_ΑΝ** πίσω<30 **ΤΟΤΕ ! υπάρχει χώρος**  πίσω 🡨πίσω+1  Β[πίσω] 🡨 x  **ΑΛΛΙΩΣ** **! η ουρά είναι γεμάτη**  **ΓΡΑΨΕ** ‘ γεμάτη ουρά’  **ΤΕΛΟΣ\_ΑΝ**  **ΑΛΛΙΩΣ**  **ΑΝ** πίσω=0 **και** εμπρός=0 **ΤΟΤΕ ! είναι άδεια**  **ΓΡΑΨΕ ΄**άδεια ουρά **΄**  **ΑΛΛΙΩΣ\_ΑΝ** εμπρός=πίσω **ΤΟΤΕ ! ένα στοιχείο**  **ΓΡΑΨΕ** ‘εξαγεται το:’, Β[εμπρός]  sum🡨sum+ Β[εμπρός] **! άθροισμα στοιχείων**  πίσω🡨0  εμπρός🡨0  άδειασε🡨άδειασε+1 **! άδειασε η ουρά**  **ΑΛΛΙΩΣ**  **ΓΡΑΨΕ** ‘εξάγεται το:’, Β[εμπρός]  sum🡨sum+ Β[εμπρός] **! ξανά το άθροισμα**  εμπρός🡨εμπρός+1  **ΤΕΛΟΣ\_ΑΝ**  **ΤΕΛΟΣ\_ΑΝ**  **ΓΡΑΨΕ** ‘ΘΕΛΕΙΣ ΝΑ ΣΥΝΕΧΙΣΕΙΣ;’  **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**  **ΔΙΑΒΑΣΕ** απάντηση  **ΜΕΧΡΙΣ\_ΟΤΟΥ** απάντηση=’ΝΑΙ’ **Η** απάντηση=’ΟΧΙ’  **ΜΕΧΡΙΣ\_ΟΤΟΥ** απάντηση=’όχι’  **ΑΝ** εμπρός=0 και πίσω=0 **ΤΟΤΕ**  **ΓΡΑΨΕ** ‘ΟΥΡΑ ΑΔΕΙΑ’  **ΑΛΛΙΩΣ**  στοιχεία🡨πίσω-εμπρός+1  **ΓΡΑΨΕ** ‘Υπάρχουν’, στοιχεία, ‘στοιχεία στην ουρά’  **ΤΕΛΟΣ\_ΑΝ**  **ΓΡΑΨΕ** άδειασε, sum  **ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ** | Χρήσιμες συνθήκες για την δομή της ουράς Ν θέσεων:  **Γεμάτη Ουρά**🡪 πίσω=Ν  **Άδεια Ουρά** 🡪 πίσω =0 και εμπρός=0  **Στοιχεία που βρίσκονται στην ουρά:** πίσω-εμπρός+1 |

**ΑΣΚΗΣΕΙΣ ΓΙΑ ΛΥΣΗ**

**Άσκηση 1:** Ερωτήσεις Σωστό – Λάθος.

1. Σε μία ουρά, το στοιχείο που τοποθετείται πρώτο σε αυτή, βγαίνει πρώτο.
2. Μία ουρά χρησιμοποιεί την μέθοδο «Πρώτο Μέσο – Τελευταίο Έξω».
3. Σε μία ουρά τα στοιχεία μπορούν να εισέλθουν και από τα δύο άκρα της.
4. Οι βασικές λειτουργίες μίας ουράς είναι η ώθηση και η απώθηση.
5. Η λειτουργία της εισαγωγής, μπορεί να εκτελεστεί και σε άδεια ουρά.
6. Στην διαδικασία της εξαγωγής, πρέπει να ελέγξουμε να μην είναι γεμάτη η ουρά.
7. Για την υλοποίηση μίας ουράς, χρειαζόμαστε έναν δείκτη ο οποίος «δείχνει» το στοιχείο που τοποθετήθηκε τελευταίο στην ουρά.
8. Σε μία ουρά Ν θέσεων, αν ισχύει «πίσω=N», τότε η ουρά είναι γεμάτη.
9. Σε μία ουρά, αν ισχύει «εμπρός=πίσω» και «πίσω<>0», τότε στην ουρά υπάρχει ένα στοιχείο.
10. Σε μία ουρά, αν ισχύει «εμπρός=πίσω», τότε στην ουρά υπάρχει ένα στοιχείο.

**Άσκηση 2:**  Η ουρά είναι μία δομή δεδομένων. **1)** να δώσετε ένα παράδειγμα ουράς από την καθημερινή ζωή **2)** να αναφέρετε τις λειτουργίες της ουράς και τους δείκτες που απαιτούνται **3)** σε μία ουρά 10 θέσεων έχουν τοποθετηθεί διαδοχικά τα στοιχεία: Μ, Κ, Δ, Α, Σ στην πρώτη, δεύτερη, τρίτη, τέταρτη και πέμπτη θέση αντίστοιχα **α)** να προσδιορίσετε τις τιμές των δεικτών της παραπάνω ουράς **β)** στη συνέχεια να αφαιρέσετε ένα στοιχείο από την ουρά. Ποιος δείκτης μεταβάλλεται και ποια η νέα του τιμή; **γ)** τέλος να τοποθετήσετε το στοιχείο Λ στην ουρά. Ποιος δείκτης μεταβάλλεται και ποια η νέα του τιμή; **(Εσπερινά 2004)**

**Άσκηση 3:**  Δίνεται η παρακάτω ακολουθία αριθμών: 25, 8, 12, 14, 71, 41, 1. Τοποθετούμε τους αριθμούς σε στοίβα και σε ουρά. 1) ποια λειτουργία θα χρησιμοποιηθεί για την τοποθέτηση των αριθμών στη στοίβα και ποια για την τοποθέτησή τους στην ουρά; 2) να σχεδιάσετε τις δύο δομές (στοίβα και ουρά) μετά την τοποθέτηση των αριθμών 3) Ποια λειτουργία θα χρησιμοποιηθεί για την έξοδο αριθμών από τη στοίβα και ποια για την έξοδό τους από την ουρά; 4. Πόσες φορές θα πρέπει να γίνει η παραπάνω λειτουργία στη στοίβα και πόσες στην ουρά για να εξέλθει ο αριθμός 71; **(Επαναληπτικές 2006)**

**Άσκηση 4:** Δίνεται ουρά Ν θέσεων. **1)** να γράψετε τις εντολές που υλοποιούν την λειτουργία της εισαγωγής. Στην περίπτωση που η ουρά είναι γεμάτη, να εμφανίζει «Γεμάτη ουρά» **2)** να γράψετε τις εντολές που υλοποιούν την λειτουργία της εξαγωγής. Σε περίπτωση που δεν υπάρχει κάποιο στοιχείο στην ουρά, να εμφανίζει «Άδεια ουρά».

**Άσκηση 5:** Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο: **1)** θα χρησιμοποιεί ουρά Α[30] για αποθήκευση χαρακτήρων **2)** θα εκτελεί 200 φορές την ακόλουθη επαναληπτική διαδικασία: **α)** θα ρωτάει τον χρήστη την λειτουργία που επιθυμεί να εκτελέσει, με έλεγχο δεδομένων για τιμές «Εισαγωγή» ή «Εξαγωγή» **β)** στην περίπτωση της εισαγωγής, θα ελέγχει αν υπάρχει χώρος στην ουρά, θα διαβάζει έναν χαρακτήρα και θα τον τοποθετεί στην ουρά, διαφορετικά θα εμφανίζει μήνυμα «Γεμάτη ουρά» **γ)** στην περίπτωση της εξαγωγής, θα ελέγχει αν υπάρχει κάποιο στοιχείο στην ουρά, θα το εμφανίζει και θα το βγάζει από την ουρά, διαφορετικά θα εμφανίζει μήνυμα «Άδεια ουρά» **3)** στο τέλος θα εμφανίζει πόσες φορές πραγματοποιήθηκε εξαγωγή στοιχείου και πόσα στοιχεία έγιναν εισαγωγή συνολικά.

**Άσκηση 6:** Να αναπτύξετε αλγόριθμο ο οποίος: **1)** θα χρησιμοποιεί ουρά Ο[100] για αποθήκευση ακέραιων αριθμών **2)** θα διαβάζει επαναληπτικά από τον χρήστη την λειτουργία που επιθυμεί να εκτελέσει με έλεγχο δεδομένων για τιμές «ΕΙ» για εισαγωγή, «ΕΞ» για εξαγωγή και «Τ» για τερματισμό λειτουργίας **3)** στην περίπτωση της εισαγωγής, θα ελέγχει αν υπάρχει χώρος στην ουρά, θα διαβάζει έναν αριθμό και θα τον τοποθετεί στην ουρά, διαφορετικά θα εμφανίζει μήνυμα «Γεμάτη ουρά» **4)** στην περίπτωση της εξαγωγής, θα ελέγχει αν υπάρχει κάποιο στοιχείο στην ουρά, θα το εμφανίζει και θα το βγάζει από την ουρά, διαφορετικά θα εμφανίζει μήνυμα «Άδεια ουρά» **5)** η παραπάνω επαναληπτική διαδικασία θα ολοκληρωθεί όταν δοθεί ως λειτουργία το «Τ» ή όταν η ουρά γεμίσει 3 φορές μετά από κάποια εισαγωγή **6)** στο τέλος θα εμφανίζει: **α)** το άθροισμα των θετικών αριθμών που έγιναν εισαγωγή στην ουρά **β)** πόσες φορές πραγματοποιήθηκε η λειτουργία της εισαγωγής **γ)** μήνυμα «Άδεια ουρά» αν η στοίβα είναι άδεια, διαφορετικά θα εμφανίζει πόσα στοιχεία έχει μέσα η ουρά.

**Άσκηση 7:**  Μια αεροπορική εταιρεία εκτελεί το δρομολόγιο Αθήνα – Θεσσαλονίκη κατά την περίοδο του Σεπτέμβρη. Λόγω της Δ.Ε.Θ. υπάρχει αυξημένη ζήτηση και η εταιρεία διατηρεί λίστα αναμονής για τους επιβάτες που δεν πρόλαβαν να κλείσουν εισιτήριο, ώστε αν προκύψει κάποια ακύρωση, να ενημερώσει τον πρώτο στη σειρά πελάτη που εισήχθη στη λίστα αναμονής προκειμένου να κλείσει εισιτήριο. Η λίστα αναμονής δεν μπορεί να περιλαμβάνει περισσότερα από 10 ονόματα. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:1) να υπάρχει μενού επιλογής: **1.** ΕΓΓΡΑΦΗ **2.** ΑΚΥΡΩΣΗ **3.** ΤΕΛΟΣ 2) αν ο χρήστης επιλέξει την τιμή «**1.**ΕΓΓΡΑΦΗ», τότε θα ζητείται το όνομα του χρήστη και θα καταχωρίζεται στη λίστα αναμονής, εφόσον η λίστα αναμονής δεν έχει γεμίσει. Διαφορετικά, θα εμφανίζεται το μήνυμα: «Η λίστα αναμονής είναι πλήρης» 3) Αν ο χρήστης επιλέξει την τιμή «**2.**ΑΚΥΡΩΣΗ», τότε κάποιος από τους επιβάτες της πτήσης έχει ακυρώσει την κράτησή του, συνεπώς, το πρόγραμμα θα πρέπει να εμφανίσει το όνομα του ατόμου που είναι το πρώτο διαθέσιμο στη λίστα αναμονής. Αν δεν υπάρχουν άτομα στη λίστα αναμονής, εμφανίζεται το μήνυμα «Η λίστα αναμονής είναι άδεια» 4) η παραπάνω διαδικασία επαναλαμβάνεται μέχρι ο χρήστης να επιλέξει την τιμή«**3.**ΤΕΛΟΣ» 5) το πρόγραμμα εμφανίζει το πλήθος των ατόμων που κατάφεραν να κάνουν κράτηση στην λίστα αναμονής καθώς και το μέγιστο πλήθος ατόμων που περίμεναν στην ουρά αναμονής. Παρατήρηση: να γίνεται έλεγχος των τιμών εισόδου. **(Παράρτημα «Πληροφορική»)**

**Άσκηση 8:** Σε ένα ταχυδρομικό κατάστημα, οι πελάτες εξυπηρετούνται με βάση τη σειρά άφιξής τουςσε αυτό. Το ταχυδρομικό κατάστημα έχει ένα ταμείο και ο μέσος χρόνος εξυπηρέτησηςκάθε πελάτη είναι 3 λεπτά. Η ουρά αναμονής στο κατάστημα δεν μπορεί να ξεπερνά τα 30άτομα.Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:1) να δέχεται σαν είσοδο από τον χρήστη μία εκ των δύο τιμών εισαγωγής: «**1.**ΕΙΣΑΓΩΓΗ» ή «**2.**ΕΠΟΜΕΝΟΣ» (με έλεγχο εγκυρότητας) 2) αν δοθεί η τιμή «**1.**ΕΙΣΑΓΩΓΗ», τότε το πρόγραμμα να διαβάζει το ονοματεπώνυμοτου πελάτη και αμέσως μετά να εμφανίζει το πλήθος των ατόμων που περιμένουν πριναπό αυτόν, εκτός αν η ουρά αναμονής είναι γεμάτη, οπότε να εμφανίζει το μήνυμα «Τοκατάστημα γέμισε. Παρακαλούμε ελάτε άλλη φορά» 3) αν δοθεί η τιμή «**2.**ΕΠΟΜΕΝΟΣ», τότε το πρόγραμμα να εμφανίζει το ονοματεπώνυμοτου πελάτη προς εξυπηρέτηση 4) η παραπάνω διαδικασία να επαναλαμβάνεται μέχρι να εξυπηρετηθούν όλοι οι πελάτες 5) στο τέλος το πρόγραμμα να εμφανίζει το πλήθος των ατόμων που εξυπηρετήθηκαν, καθώς και τον μέσο χρόνο αναμονής των πελατών. **(Παράρτημα «Πληροφορική»)**

**Άσκηση 9:** Ένας εκτυπωτής χρησιμοποιεί μια ουρά εκτύπωσης για να τοποθετεί σε αυτήν τα αρχεία που έχουν σταλεί προς εκτύπωση με τη σειρά που αυτά στάλθηκαν. Κάθε φορά εκτυπώνει το αρχείο που βρίσκεται στην αρχή της ουράς εκτύπωσης, το οποίο και εξάγει. Λόγω της περιορισμένης μνήμης του εκτυπωτή, θεωρούμε ότι στην ουρά μπορούν να εισαχθούν το πολύ 15 αρχεία. Στην ουρά εκτύπωσης υπάρχουν ήδη τα αρχεία Α.doc, B.doc, Γ.doc. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο: 1) θα καταχωρεί στην ουρά τα αρχεία που υπάρχουν ήδη μέσα 2) θα διαβάζει επαναληπτικά, με έλεγχο εγκυρότητας, το γράμμα “N” που καθορίζει την έλευση νέου αρχείου ή το γράμμα “Ε” που δηλώνει την προσπάθεια εκτύπωσης ενός αρχείου 3) κατά την έλευση ενός αρχείου, διαβάζει το όνομά του και εξετάζει αν υπάρχει ο διαθέσιμος χώρος στην ουρά και το αρχείο καταχωρίζεται σε αυτήν με τη διαδικασία της εισαγωγής. Στην περίπτωση που δεν υπάρχει χώρος, εμφανίζεται το μήνυμα «Η ουρά γέμισε. Δε μπορεί να εκτυπωθεί το αρχείο» 4) όταν ο χρήστης δώσει το γράμμα “Ε”, εξετάζει αν υπάρχουν αρχεία προς εκτύπωση και στην περίπτωση αυτή εξάγεται το κατάλληλο αρχείο εμφανίζοντας τη λέξη «Εκτύπωση» ακολουθούμενη από το όνομα του αρχείου που τυπώνεται 5) στο τέλος θα ρωτάει τον χρήστη αν επιθυμεί να σταματήσει, και θα διαβάζει την απάντηση του, εξασφαλίζοντας πως θα λάβει μία από τις τιμές «ναι» / «όχι». Η επαναληπτική διαδικασία ολοκληρώνεται, όταν εκτυπωθούν όλα τα αρχεία που έχουν τοποθετηθεί στην ουρά και ταυτόχρονα δεν υπάρχει αρχείο προς εκτύπωση στην ουρά 6) Μετά το τέλος της διαδικασίας, το πρόγραμμα εμφανίζει: α) τον συνολικό αριθμό αρχείων που εκτυπώθηκαν β) πόσες φορές στην ουρά εκτύπωσης υπήρχαν 3 αρχεία, μετά από κάποια έλευση ή εκτύπωση αρχείου γ) πόσες φορές δεν μπορούσε να εκτυπωθεί κάποιο αρχείο επειδή η ουρά ήταν γεμάτη

**Άσκηση 10:** Να αναπτύξετε αλγόριθμο ο οποίος: **1)** θα χρησιμοποιεί ουρά Α[30] για την αποθήκευση ακέραιων αριθμών **2)** θα διαβάζει από τον χρήστη πόσα στοιχεία υπάρχουν ήδη μέσα στην ουρά, εξασφαλίζοντας πως η ουρά έχει κάποια στοιχεία μέσα (αρχίζοντας από την θέση 1) αλλά δεν είναι γεμάτη **3)** θα διαβάζει από τον χρήστη ποια επιλογή επιθυμεί να εκτελέσει, εξασφαλίζοντας πως θα είναι μία από τις «Άδειασμα Ουράς» ή «Γέμισμα Ουράς» **4)** στην περίπτωση που ο χρήστης επιλέξει «Άδειασμα Ουράς», θα εμφανίζει όλα τα στοιχεία που είναι στην ουρά, θα τα εξάγει όλα και θα εμφανίζει ποιος ήταν ο μεγαλύτερος αριθμός από αυτούς **5)** στην περίπτωση που ο χρήστης επιλέξει «Γέμισμα Ουράς», θα διαβάζει επαναληπτικά αριθμούς μέχρι να γεμίσει η ουρά και θα εμφανίζει τον μέσο όρο τους.

**Ενότητα 1.3.1: Λίστες**

1. Ποια είναι τα γενικά χαρακτηριστικά μίας λίστας;

**(Συμπληρωματικό υλικό, ενότητα 1.3.1)** Οι λίστες είναι δυναμικές δομές δεδομένων, οπότε μπορούμε να προσθέσουμε και να αφαιρέσουμε κόμβους από αυτές κατά την διάρκεια της εκτέλεσης του προγράμματος.

1. Περιγράψτε την μορφή της ελεύθερης μνήμης ενός υπολογιστή και πως αυτή επηρεάζει την αποθήκευση δεδομένων.

**(Συμπληρωματικό υλικό, ενότητα 1.3.1)** Συνήθως η μνήμη του υπολογιστή δεν περιέχει τις ελεύθερες θέσεις της με την σειρά σε συνεχόμενες θέσεις. Άρα στην περίπτωση αυτή δεν θα μπορούσαμε να αποθηκεύσουμε τα δεδομένα μας σε πίνακα. Μπορούμε όμως να χρησιμοποιήσουμε τις διάσπαρτες αυτές θέσεις για να αποθηκεύσουμε τα δεδομένα μας σε μορφή λίστας.

1. Από ποια επιμέρους τμήματα αποτελείται μία συνδεδεμένη λίστα;

**(Συμπληρωματικό υλικό, ενότητα 1.3.1)** Μία συνδεδεμένη λίστα αποτελείται από μία σειρά από κόμβους, που συνήθως βρίσκονται σε απομακρυσμένες θέσεις μνήμης. Κάθε κόμβος αποτελείται από δύο μέρη:

* Το πρώτο τμήμα περιέχει τα δεδομένα, τα οποία μπορεί να περιέχει μία ή περισσότερες αλφαριθμητικές ή λογικές τιμές.
* Το δεύτερο τμήμα είναι ονομάζεται δείκτης (pointer). Είναι ένας ιδιαίτερος τύπος δεδομένων, που προσφέρεται από τις περισσότερες γλώσσες προγραμματισμού. Δεν λαμβάνει αριθμητικές τιμές, αλλά οι τιμές του είναι διευθύνσεις στην κύρια μνήμη του υπολογιστή που χρησιμοποιούνται για τη σύνδεση στοιχείων μιας δομής που βρίσκονται σε μη συνεχόμενες θέσεις μνήμης. Στην ουσία «δείχνει» σε ποια διεύθυνση βρίσκεται ο επόμενος κόμβος μίας συνδεδεμένης λίστας

1. Να δώσετε τον ορισμό της συνδεδεμένης λίστας.

**(Συμπληρωματικό υλικό, ενότητα 1.3.1)**  Μία (απλά) **συνδεδεμένη λίστα (linked list)** είναι ένα σύνολο κόμβων διατεταγμένων γραμμικά (ο ένας μετά τον άλλο). Κάθε κόμβος περιέχει εκτός από τα **δεδομένα** του και έναν **δείκτη** που δείχνει προς τον επόμενο κόμβο.

* Ο δείκτης του τελευταίου κόμβου δε δείχνει σε κάποιον κόμβο (δείκτης στο κενό). Για να το δηλώσουμε αυτό λέμε ότι το πεδίο δείκτη του τελευταίου κόμβου έχει την τιμή **NULL** και αναπαρίσταται με το σύμβολο «».
* Οι δείκτες των υπόλοιπων κόμβων περιέχουν την διεύθυνση του επόμενου κόμβου και τους συμβολίζουμε με ένα βέλος που υποδηλώνει την σύνδεση μεταξύ του προηγούμενου και του επόμενου κόμβου.
* Για να προσπελάσουμε τους κόμβους της λίστας χρειάζεται να γνωρίζουμε τη διεύθυνση (θέση στη μνήμη) του πρώτου κόμβου της λίστας. Η διεύθυνση αυτή αποθηκεύεται σε μία ειδική μεταβλητή που την ονομάζουμε συνήθως **Κεφαλή (Head)**.

Δεδοµένα

Κεφαλή

45

8



13

67

Δείκτης (pointer)

Μορφή Κόμβου Συνδεδεμένη λίστα με 3 κόμβους

1. Πως γίνεται η πρόσβαση στους κόμβους μίας συνδεδεμένης λίστας;

**(Συμπληρωματικό υλικό, ενότητα 1.3.1)** Οι κόμβοι μιας (απλά) συνδεδεμένης λίστας δεν έχουν ονόματα και είναι διατεταγμένοι σε μια συγκεκριμένη σειρά, χωρίς αυτό να σημαίνει ότι αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη. Αντίθετα, είναι διασκορπισμένοι σε όλη τη μνήμη και η σύνδεση μεταξύ τους γίνεται μέσω των δεικτών. Έχουμε άμεση πρόσβαση μόνο στον πρώτο κόμβο της λίστας, μέσω της διεύθυνσης που περιέχεται στον δείκτη Κεφαλή. Επομένως, για να εντοπίσουμε κάποιον από τους ενδιάμεσους κόμβους, πρέπει να ξεκινήσουμε από τον πρώτο κόμβο της λίστας και να ακολουθήσουμε τους δείκτες με τη σειρά, μέχρι να φτάσουμε στον επιθυμητό κόμβο. Για παράδειγμα αν επιθυμούμε να προσπελάσουμε τον τρίτο κόμβο μίας λίστας, θα ξεκινήσουμε από την κεφαλή, η οποία θα μας δείξει την διεύθυνση του πρώτου κόμβου, ο δείκτης του πρώτου θα μας δείξει την διεύθυνση του δεύτερου και ο δείκτης του δεύτερου την διεύθυνση του τρίτου.

1. Να δώσετε ένα παράδειγμα προσθήκης και αφαίρεσης κόμβου σε μία συνδεδεμένη λίστα.

**(Συμπληρωματικό υλικό, ενότητα 1.3.1)** Το παρακάτω σχήμα δείχνει μία συνδεδεμένη λίστα που αναπαριστά αν ταξίδι, με αφετηρία την Αθήνα και τερματισμό τις Σέρρες.

Κεφαλή

Λαµία

Λάρισα



Σέρρες

Θεσσαλονίκη

Αθήνα



Κατερίνη

**Μορφή αρχικής λίστας**

Αν αποφασίζουμε να επισκεφτούμε και τον Βόλο μετά τη Λαμία και πριν τη Λάρισα, τότε θα πρέπει να εισάγουμε άλλο ένα κόμβο. Οι ενέργειες που απαιτούνται είναι:

* Ο δείκτης του δεύτερου κόμβου θα «δείχνει» το νέο κόμβο.
* Ο δείκτης του νέου κόμβου θα «δείχνει» τον τέταρτο κόμβο.

Έτσι οι κόμβοι της λίστας διατηρούν τη λογική τους σειρά, αλλά οι φυσικές θέσεις στη μνήμη μπορεί να είναι τελείως διαφορετικές.

Κεφαλή



Bόλος

Λάρισα



Σέρρες

Θεσσαλονίκη

Αθήνα



Κατερίνη

Λαµία

**Εισαγωγή του κόμβου «Βόλος» μετά την Λαμία και πριν τη Λάρισα**

Αν επιπλέον αποφασίζουμε να μην επισκεφτούμε τη Λαμία, θα πρέπει να διαγράψουμε τον αντίστοιχο κόμβο. Πρακτικά θα πρέπει να αλλάξει τιμή ο δείκτης του προηγούμενου κόμβου και να δείχνει πλέον στον επόμενο αυτού που διαγράφεται. Ο κόμβος που διαγράφεται αποτελεί «άχρηστο δεδομένο» και ο χώρος μνήμης που καταλάμβανε παραχωρείται για άλλη χρήση.

Κεφαλή



Σέρρες

Θεσσαλονίκη



Αθήνα

Λαµία

Bόλος

Λάρισα

Κατερίνη

**Αφαίρεση του κόμβου «Λαμία»**

1. Σε ποιες περιπτώσεις αξιοποιούνται οι συνδεδεμένες λίστες;

**(Συμπληρωματικό υλικό, ενότητα 1.3.1)** Οι συνδεδεμένες λίστες αξιοποιούνται για την υλοποίηση της στοίβας και της ουράς λόγω της δυνατότητας αυξομείωσης του μεγέθους τους.

1. Τι ονομάζουμε «απλά συνδεδεμένη λίστα»;

**(Συμπληρωματικό υλικό, ενότητα 1.3.1)** Στην απλή συνδεδεμένη λίστα μπορούμε να κινηθούμε προς μία μόνο κατεύθυνση, ξεκινώντας από τον αρχικό κόμβο και μετακινούμενοι προς τον τελευταίο.

1. Τι ονομάζουμε «διπλά συνδεδεμένη»; Να αναλύσετε τα χαρακτηριστικά της.

**(Συμπληρωματικό υλικό, ενότητα 1.3.1)** Μία λίστα ονομάζεται «διπλά συνδεδεμένη» όταν μπορούμε να τη διατρέξουμε και προς τις δύο κατευθύνσεις. Για την υλοποίηση της χρειαζόμαστε δύο δείκτες:

* Τον δείκτη «κεφαλή» που δείχνει τον πρώτο κόμβο της λίστας.
* Τον δείκτη «ουρά» που δείχνει τον τελευταίο κόμβο της λίστας.

Με τον τρόπο αυτό μπορούμε, να την διαβάσουμε και προς τις δύο κατευθύνσεις. Ένα παράδειγμα είναι οι σταθμοί του μετρό. Άλλα χαρακτηριστικά είναι :

* Κάθε κόμβος της διπλά συνδεδεμένης λίστας συνδέεται με τον αμέσως προηγούμενο και τον αμέσως επόμενο κόμβο της λίστας, εκτός από τον πρώτο και τον τελευταίο.
* Κάθε κόμβος περιέχει δύο δείκτες.
* Ο πρώτος και ο τελευταίος κόμβος έχουν δείκτη με τιμή NULL.
* Σε μία διπλά συνδεδεμένη λίστα διευκολύνεται η ταξινόμηση και η αναζήτηση, ωστόσο, αυξάνεται η πολυπλοκότητα στη διαχείριση των κόμβων, καθώς απαιτείται επιπλέον χώρος για τον δεύτερο δείκτη (επιπρόσθετη μνήμη για κάθε κόμβο)

Κεφαλή



Oυρά

Πανεπιστήµιο

Σύνταγµα

|  |  |  |  |
| --- | --- | --- | --- |
|  | Aκρόπολη |  |  |

**Διπλά συνδεδεμένη λίστα**

1. Ποιες είναι οι διαφορές ενός πίνακα από μία λίστα;

**(Συμπληρωματικό υλικό, ενότητα 1.3.1)**

* + Ο πίνακας θεωρείται μια δομή τυχαίας προσπέλασης, σε αντίθεση με μια λίστα που είναι στην ουσία μια δομή ακολουθιακής ή σειριακής προσπέλασης. Για να φθάσουμε, δηλαδή, σ’ έναν κόμβο μιας λίστας πρέπει να περάσουμε από όλους τους προηγούμενους ξεκινώντας από τον πρώτο.
  + O πίνακας έχει σταθερό μέγεθος, το οποίο δηλώνεται εξαρχής κατά την υλοποίηση. Αυτό γίνεται, διότι ο πίνακας είναι στατική δομή δεδομένων σε αντίθεση με τη λίστα που είναι δυναμική δομή και το μέγεθός της μπορεί να μεταβάλλεται καθώς εισέρχονται νέοι κόμβοι στη λίστα ή διαγράφονται κάποιοι άλλοι.
  + Oι κόμβοι της λίστας αποθηκεύονται σε μη συνεχόμενες θέσεις μνήμης σε αντιδιαστολή με τους πίνακες, όπου τα στοιχεία αποθηκεύονται σε συνεχόμενες θέσεις μνήμης.

1. Να αναφέρετε τα πλεονεκτήματα και τα μειονεκτήματα των λιστών έναντι των πινάκων.

**(Συμπληρωματικό υλικό, ενότητα 1.3.1)**

**Πλεονεκτήματα:**

* + - * Το δυναμικό τους μέγεθος.
      * Η ευκολία εισαγωγής και διαγραφής από οποιοδήποτε μέρος της λίστας.
      * Η μη αναγκαιότητα δήλωσης του μεγέθους τους.

**Μειονεκτήματα:**

* + - * Η τυχαία πρόσβαση στη λίστα δεν επιτρέπεται. Είναι αδύνατο να φτάσετε στον n-οστό κόμβο μιας απλά συνδεδεμένης λίστας χωρίς πρώτα να περάσετε από όλους τους κόμβους διαδοχικά μέχρι τον συγκεκριμένο κόμβο ξεκινώντας από τον πρώτο κόμβο. Εναλλακτικά, στην περίπτωση της διπλά συνδεμένης λίστας μπορείτε να ξεκινήσετε και από τον τελευταίο κόμβο. Επομένως, δεν μπορούμε να πραγματοποιήσουμε με αποτελεσματικό τρόπο δυαδική αναζήτηση σε συνδεδεμένες λίστες.
      * Οι συνδεδεμένες λίστες έχουν πολύ μεγαλύτερη επιβάρυνση από τους πίνακες, αφού οι συνδεδεμένοι κόμβοι της λίστας είναι δυναμικά κατανεμημένοι (οι οποίοι είναι λιγότερο αποτελεσματικοί στη χρήση της μνήμης) και κάθε κόμβος στη λίστα πρέπει, επιπλέον, να αποθηκεύσει έναν πρόσθετο δείκτη που θα δείχνει στον επόμενο κόμβο. Στην περίπτωση των διπλά συνδεδεμένων λιστών χρειαζόμαστε επιπλέον έναν δεύτερο δείκτη που θα δείχνει στον προηγούμενο κόμβο.

1. Ποιες είναι οι βασικές πράξεις των συνδεδεμένων λιστών;

**(Συμπληρωματικό υλικό, ενότητα 1.3.1)** Οι βασικές πράξεις των συνδεδεμένων λιστών είναι οι ακόλουθες:

* + - * Εισαγωγή κόμβου στη λίστα (εισαγωγή κόμβου στην αρχή, στο τέλος της λίστας ή ενδιάμεσα).
      * Διαγραφή κόμβου από τη λίστα (διαγραφή από την αρχή, το τέλος της λίστας ή ενδιάμεσα).
      * Έλεγχος για το αν η λίστα είναι κενή.
      * Αναζήτηση κόμβου για την εύρεση συγκεκριμένου στοιχείου.
      * Διάσχιση της λίστας και προσπέλαση των στοιχείων της (π.χ. εκτύπωση των δεδομένων που περιέχονται σε όλους τους κόμβους της λίστας).

**Παράδειγμα 1 - Δημιουργία λίστας από τις πληροφορίες των δεικτών:** Για την δημιουργία μίας λίστας δίνονται οι ακόλουθες πληροφορίες: στον πίνακα δεδομένα[5] αποθηκεύονται τα δεδομένα των κόμβων και στον πίνακα δείκτης[5] η διεύθυνση του επόμενου κόμβου. Στην πρώτη θέση έχουν τοποθετηθεί τα στοιχεία του πρώτου κόμβου. Να δημιουργήσετε την λίστα.

Δεδομένα Δείκτης

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 | -3 | 0 | 1 | 9 |
| 1 | 2 | 3 | 4 | 5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 0 | 4 | 5 | 2 |
| 1 | 2 | 3 | 4 | 5 |

**Επεξήγηση λύσης:**

Σε κάθε κόμβο έχουμε δύο πληροφορίες: Δεδομένα και Δείκτης, ο οποίος δείχνει τον επόμενο κόμβο που θα εισαχθεί στη λίστα. Στην θέση 1 έχουμε τον πρώτο κόμβο από τα δεδομένα της άσκησης. Στη συνέχεια σκεπτόμαστε ως εξής:

* Έχουμε δείκτης[1]=3, άρα ο επόμενος κόμβος βρίσκεται στο δεδομένα[3]=0 το οποίο τοποθετούμε δεύτερο στη λίστα.
* Έχουμε δείκτης[3]=4, άρα ο επόμενος κόμβος βρίσκεται στο δεδομένα[4]=1, το οποίο τοποθετούμε τρίτο στη λίστα.
* Έχουμε δείκτης[4]=5, άρα ο επόμενος κόμβος βρίσκεται στο δεδομένα[5]=9, το οποίο τοποθετούμε τέταρτο στη λίστα.
* Έχουμε δείκτης[5]=2, άρα ο επόμενος κόμβος βρίσκεται στο δεδομένα[2]=-3, το οποίο τοποθετούμε τελευταίο στη λίστα. Επίσης έχουμε δείκτης[2]=0, κάτι που επαληθεύει πως είναι το τελευταίο στοιχείο (τιμή Null).



Κεφαλή



-3

9

1

0

15

**Παράδειγμα 2 - Δημιουργία λίστας με βάση τον επόμενο κόμβο:** Για την δημιουργία μίας λίστας, κάποιος μαθητής ακολούθησε την παρακάτω διαδικασία: για κάθε κόμβο σημείωσε δύο πληροφορίες: αρχικά τα δεδομένα και στη συνέχεια έναν αριθμό που έδειχνε ποιος είναι ο επόμενος στη σειρά κόμβος. Να δημιουργήσετε τη λίστα.

Γιώργος 3, Πέτρος 2, Νάνσυ 5, Νάντια 0, Σαμπρίνα 4.

**Επεξήγηση λύσης:**

* Ο πρώτος κόμβος θα είναι αυτός που έχει ως επόμενο κόμβο το 2, οπότε θα είναι ο Πέτρος.
* Τελευταίος θα είναι το κόμβος με επόμενο κόμβο το 0, δηλαδή η Νάντια.
* Οι υπόλοιποι κόμβοι τοποθετούνται με αύξουσα σειρά αριθμού.

Κεφαλή



Νάντια

Νάνσυ

Πέτρος

Σαμπρίνα

Γιώργος

**ΑΣΚΗΣΕΙΣ ΓΙΑ ΛΥΣΗ**

**Άσκηση 1:** Ερωτήσεις Σωστό – Λάθος.

1. Συχνά, η μνήμη του υπολογιστή περιέχει διάσπαρτες ελεύθερες θέσεις μνήμης και όχι υποχρεωτικά συνεχόμενες.
2. Μία συνδεδεμένη λίστα αποτελείται από κόμβους που βρίσκονται σε απομακρυσμένες θέσεις μνήμης.
3. Ένας κόμβος μίας συνδεδεμένης λίστας, αποτελείται από δύο τμήματα που περιέχουν δεδομένα .
4. Ο δείκτης ενός κόμβου μίας συνδεδεμένης λίστας περιέχει αλφαριθμητικές τιμές.
5. Για να προσπελάσουμε τους κόμβους μίας συνδεδεμένης λίστας, πρέπει να γνωρίζουμε την διεύθυνση του πρώτου κόμβου της λίστας .
6. Σε μία συνδεδεμένη λίστα, μπορούμε να προσθέσουμε κόμβους μόνο από το τέλος της.
7. Σε μία συνδεδεμένη λίστα, δεν μπορούμε να αφαιρέσουμε κόμβους.
8. Μία απλή συνδεδεμένη λίστα, μπορούμε να την διατρέξουμε και προς τις δύο κατευθύνσεις.
9. Για να υλοποιήσουμε μία διπλά συνδεδεμένη λίστα, χρειαζόμαστε δύο δείκτες, ώστε να μπορέσουμε να την διατρέξουμε και από τις δύο κατευθύνσεις.
10. Όλοι οι κόμβοι μίας διπλά συνδεδεμένης λίστας, συνδέονται και με τον προηγούμενο και με τον επόμενο κόμβο της λίστας.
11. Η λίστα είναι μία δομή ακολουθιακής προσπέλασης.
12. Οι κόμβοι μίας λίστας αποθηκεύονται σε συνεχόμενες θέσεις μνήμης.
13. Σε μία λίστα, δεν επιτρέπεται η τυχαία προσπέλαση στοιχείων.
14. Σε μία διπλά συνδεδεμένη λίστα διευκολύνεται η αναζήτηση και η ταξινόμηση.
15. Μία λίστα έχουν μεγαλύτερη επιβάρυνση στη μνήμη σε σχέση ένα πίνακα.

**Άσκηση 2:** Σε μία λίστα έχουμε τοποθετήσει διαδοχικά τους αριθμούς 5, 20, -3, 8. **1)** να σχεδιάσετε την μορφή της λίστας **2)** να περιγράψετε την διαδικασία που απαιτείται για την προσθήκη του αριθμού 7 μετά τον αριθμό -3 και να σχεδιάσετε τη νέα λίστα **3)** να περιγράψετε την διαδικασία που απαιτείται ώστε να διαγραφεί ο δεύτερος κόμβος στην λίστα που προέκυψε από το ερώτημα 2 και να σχεδιάσετε τη νέα λίστα.

**Άσκηση 3:** Σε μία λίστα έχουμε τοποθετήσει διαδοχικά τους χαρακτήρες Α, Κ, Λ, Φ, Β **1)** να σχεδιάσετε την μορφή της λίστας **2)** να περιγράψετε την διαδικασία που απαιτείται για την προσθήκη του χαρακτήρα Γ μετά τον χαρακτήρα Α και να σχεδιάσετε τη νέα λίστα **3)** να περιγράψετε την διαδικασία που απαιτείται ώστε να διαγραφεί ο τέταρτος κόμβος στην λίστα που προέκυψε από το ερώτημα 2 και να σχεδιάσετε τη νέα λίστα.

**Άσκηση 4:** Να δημιουργήσετε μία διπλά συνδεδεμένη λίστα που αν αναπαριστά τις διαδρομές μεταξύ των 4 νομών της Κρήτης.

**Άσκηση 5:** Για την δημιουργία μίας λίστας δίνονται οι ακόλουθες πληροφορίες: στον πίνακα δεδομένα[8] αποθηκεύονται τα δεδομένα των κόμβων και στον πίνακα δείκτης[8] η διεύθυνση του επόμενου κόμβου. Στην πρώτη θέση έχουν τοποθετηθεί τα στοιχεία του πρώτου κόμβου. Να δημιουργήσετε την λίστα.

Δεδομένα Δείκτης

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 20 | 10 | 35 | 22 | 12 | 18 | 23 | 19 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 6 | 5 | 2 | 0 | 7 | 3 | 4 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

**Άσκηση 6:** Ένας καθηγητής του ΑΕΠΠ, για να οργανώσει την ύλη του καλοκαιριού, αποθήκευσε σε πίνακα κεφάλαιο[8] το όνομα του κεφαλαίου που θα διδάξει και σε πίνακα σειρά[5] την σειρά με την οποία θα το κάνει. Να σχεδιάσετε την αντίστοιχη λίστα, αν γνωρίζεται πως η κεφαλή της λίστας είναι στη θέση 3

κεφάλαιο σειρά

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Δομή επιλογής | ΓΛΩΣΣΑ | Δομή ακολουθίας | Γενικές ασκήσεις | Ασκήσεις με ΑΝ |
| 1 | 2 | 3 | 4 | 5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | 4 | 1 | 0 | 2 |
| 1 | 2 | 3 | 4 | 5 |

**Άσκηση 7:** Για την δημιουργία μίας λίστας σχετικά με τα υψόμετρα βουνών της Ελλάδας, κάποιος μαθητής ακολούθησε την παρακάτω διαδικασία: για κάθε βουνό σημείωσε δύο πληροφορίες: αρχικά το όνομα του και στη συνέχεια έναν αριθμό που έδειχνε ποιο είναι το επόμενο στη σειρά βουνό. Να δημιουργήσετε τη λίστα.

Σμόλικας 3, Γκιώνας 0, Όλυμπος 2, Γράμμος 5, Βόρας 4.

**Άσκηση 8:** Για να οργανώσει ένας μαθητής το διάβασμα του, σημείωσε για κάθε μάθημα δύο πληροφορίες: αρχικά το όνομα του μαθήματος και στη συνέχεια έναν αριθμό που έδειχνε ποιο είναι το επόμενο μάθημα που θα μελετήσει. Να σχεδιάσετε τη λίστα,

Μαθηματικά 3, ΑΟΘ 2, Διάλλειμα 4, Έκθεση 0, ΑΕΠΠ 5

**Ενότητα 1.3.1: Λίστες**

1. Να περιγράψετε τις ακόλουθες έννοιες: δένδρο, ακμή, γονέας, παιδί, ρίζα, αδέρφια, φύλλα.

**(Συμπληρωματικό υλικό, Ενότητα 1.3.2)**

* **Δένδρο:** είναι ένα σύνολο από κόμβους οι οποίοι συνδέονται μεταξύ τους με ακμές.
* **Ακμή:** είναι μία «γραμμή» που συνδέει δύο κόμβους.
* **Γονέας + Παιδί:**  όταν δύο κόμβοι ενώνονται μεταξύ τους με μία ακμή, γονέας είναι ο κόμβος από τον οποίο ξεκινάει η ακμή και παιδί είναι ο κόμβος στον οποίο καταλήγει η ακμή.
  + Ένας κόμβος μπορεί να έχει κανένα, ένα η περισσότερα παιδιά.
* **Ρίζα (root):**  είναι ο κόμβος που δεν έχει γονέα και βρίσκεται στην κορυφή του δένδρου.
* **Αδέρφια:** είναι κόμβοι που έχουν τον ίδιο γονέα.

**Ρίζα**

**p**

**Γονέας του c**

**c**

**Αδέλφια µε το c**

**Παιδί του p**

**Φύλλα**

**Φύλλα**

* **Φύλλα:** είναι κόμβοι που δεν έχουν παιδιά.

1. Να δώσετε τον ορισμό του δένδρου.

**(Συμπληρωματικό υλικό, ενότητα 1.3.2)** Ένα **δένδρο (tree)** είναι μία δομή που αποτελείται από ένα σύνολο κόμβων και ένα σύνολο ακμών μεταξύ των κόμβων με βάση τους εξής κανόνες:

* Υπάρχει ένας ξεχωριστός κόμβος που ονομάζεται ρίζα. Αυτός είναι ένας κόμβος χωρίς γονέα.
* Για κάθε κόμβο c, εκτός από τη ρίζα, υπάρχει μόνο μια ακμή που καταλήγει στον κόμβο αυτόν ξεκινώντας από κάποιον άλλον κόμβο p. Ο κόμβος p ονομάζεται γονέας του c και ο κόμβος c παιδί του p.
* Για κάθε κόμβο υπάρχει μία μοναδική διαδρομή, δηλαδή, μια ακολουθία διαδοχικών ακμών, που ξεκινάει από τη ρίζα και τερματίζει σε αυτόν τον κόμβο.
  + Μπορούμε να έχουμε και ένα απλό δέντρο το οποίο απαρτίζεται από έναν μόνο κόμβο. Αυτός ο κόμβος είναι και η ρίζα του δέντρου, διότι δεν έχει γονέα και φύλο, και διότι δεν έχει παιδιά.
  + Δένδρο θεωρούμε και το κενό δένδρο, δηλαδή το δένδρο που δεν έχει ούτε κόμβους, ούτε ακμές. Το κενό δένδρο είναι το μόνο δένδρο χωρίς ρίζα.

1. Να περιγράψετε τι είναι ένα «υποδένδρο».

**(Συμπληρωματικό υλικό, ενότητα 1.3.2)**  Μέσα σε ένα δένδρο μπορούμε να εντοπίσουμε και άλλα μικρότερα δένδρα, που ονομάζονται υποδένδρα. Πιο συγκεκριμένα, κάθε κόμβος ενός δένδρου μπορεί να θεωρηθεί ως ρίζα ενός υποδένδρου, δηλαδή ενός άλλου μικρότερου δένδρου, που ξεκινάει από τον κόμβο αυτόν. Όπως φαίνεται και από το ακόλουθο σχήμα, ο κόμβος 48 είναι ρίζα και έχει δύο υποδένδρα που ξεκινούν από τους κόμβους 45 και 70 αντίστοιχα. Ο κόμβος 45 έχει ένα υποδένδρο που αποτελείται από τον κόμβο 2. Ο κόμβος 70 έχει δύο υποδένδρα που αποτελούνται από τους κόμβους 66 και 83 αντίστοιχα. Τα υποδένδρα των κόμβων 2, 66 και 83 είναι κενά.

**48**

**45**

**70**

**2**

**66**

**83**

1. Δώστε ένα παράδειγμα περίπτωσης ενός «διατεταγμένου δένδρου».

**(Συμπληρωματικό υλικό, ενότητα 1.3.2)** Αν για παράδειγμα θέλουμε να μοντελοποιήσουμε την ιεραρχική σχέση των μελών μας οικογένειας και μας ενδιαφέρει να οργανώσουμε τα αδέλφια σύμφωνα με την ηλικία τους, τότε τα αδέλφια που θα έχουν γεννηθεί νωρίτερα θα τοποθετηθούν στην δενδρική δομή πιο αριστερά σε σχέση με αυτά που θα έχουν γεννηθεί αργότερα. Σε αυτή την περίπτωση, που για κάθε κόμβο υπάρχει μία γραμμική σχέση μεταξύ των παιδιών του κόμβου αυτού, αναφερόμαστε σε ένα διατεταγμένο δένδρο.

1. Να δώσετε παραδείγματα της χρησιμότητας των δένδρων.

**(Συμπληρωματικό υλικό, ενότητα 1.3.2)** Τα δένδρα είναι μία μη – γραμμική ευέλικτη δομή και έχουν αρκετές δυνατότητες και εφαρμογές, κάποιες από τις οποίες είναι οι ακόλουθες:

* Έχουν μεγάλη «δυναμικότητα», είναι δηλαδή εύκολο να προσθέσουμε, να αφαιρέσουμε ή αν αναζητήσουμε στοιχεία σε ένα δένδρο.
* Η δομή των δένδρων μεταφέρει πληροφορίες. Για παράδειγμα, σε ένα σύστημα αρχείων, αν ο κατάλογος “users” είναι παιδί της ρίζας “/” και ότι ο κατάλογος “student” είναι παιδί του “users” μπορούμε να συμπεράνουμε ότι υπάρχει η διαδρομή “/users/student”.
* Χρησιμοποιούνται για αναπαραστάσεις δεδομένων του πραγματικού χρόνου, αλλά και της πληροφορικής, που διέπονται από ένα είδος ιεραρχίας, όπως πχ οικογενειακό δένδρο, πίνακας περιεχομένων ενός βιβλίου, τα αρχεία και οι φάκελοι ενός υπολογιστή κτλ.
* Μπορούν να χρησιμοποιηθούν για την επίλυση προβλημάτων, όπως για παράδειγμα η συμπίεση εικόνων, η ταξινόμηση, η αυτόματη συμπλήρωση λέξεων σε συσκευές κινητών, η μεταγλώττιση ενός προγράμματος, η λήψη αποφάσεων. Στα δένδρα απόφασης, κάθε κόμβος αντιπροσωπεύει ένα χαρακτηριστικό (ιδιότητα), κάθε ακμή αντιπροσωπεύει μία απόφαση (κανόνα) και κάθε φύλλο αντιπροσωπεύει ένα αποτέλεσμα.
* Τα «δένδρα παιχνιδιού» μπορούν να χρησιμοποιηθούν για την μοντελοποίηση πιθανών κινήσεων σε παιχνίδια όπως πχ το σκάκι ή η τρίλιζα.
* Διαδεδομένα είναι και τα δένδρα για την αναπαράσταση και κατ επέκταση τον υπολογισμό αριθμητικών εκφράσεων.

1. Τι είναι ένα δυαδικό δένδρο;

**(Συμπληρωματικό υλικό, ενότητα 1.3.2)** Ένα δυαδικό δένδρο αναζήτησης είναι ένα διατεταγμένο δένδρο, στο οποίο κάθε κόμβος έχει το πολύ δύο παιδιά, το αριστερό και το δεξί παιδί.

**1**

**2**

**3**

**4**

**5**

**6**

**Δεξιό παιδί του 1**

**7**

**8**

**Αριστερό παιδί του 3**

**Δεξιό υπόδεντρο του 2**

1. Τι είναι το δυαδικό δένδρο αναζήτησης;

**(Συμπληρωματικό υλικό, ενότητα 1.3.2)** Ένα δυαδικό δένδρο αναζήτησης, είναι ένα δυαδικό δένδρο, όπου για κάθε κόμβο u, όλοι οι κόμβοι του αριστερού υποδένδρου έχουν τιμές μικρότερες της τιμής του κόμβου u και όλοι οι κόμβοι του δεξιού υποδένδρου έχουν τιμές μεγαλύτερες (ή ίσες) της τιμής του κόμβου u.

**19**

Μικρότερο

Μεγαλύτερο

Μικρότερο

**11**

Μεγαλύτερο

**35**

Μικρότερο

**7**

**16**

**23**

Μικρότερο

Μεγαλύτερο

**13**

**17**

1. Ποια είναι τα πλεονεκτήματα των δυαδικών δένδρων αναζήτησης;

**(Συμπληρωματικό υλικό, ενότητα 1.3.2)** Τα δυαδικά δένδρα αναζήτησης συνδυάζουν τα εξής πλεονεκτήματα:

* Των λιστών, όσον αφορά τις πράξεις της εισαγωγής και της εξαγωγής, δηλαδή μπορούν να προσθέτουν και να αφαιρούν εύκολα κόμβους.
* Των ταξινομημένων πινάκων, όσον αφορά την πράξη της αναζήτησης, την οποίο εκτελεί ταχύτερα χάρη στον τρόπο αποθήκευσης τιμών.
  + Γενικά όταν θέλουμε να έχουμε γρήγορους αλγόριθμους αναζήτησης, πρέπει να αποθηκεύουμε τις τιμές στα δένδρα με ένα συγκεκριμένο τρόπο.

**Παράδειγμα 1 - Δημιουργία δένδρου:** Να δημιουργήσετε το δένδρο που προκύπτει από τις ακόλουθες πληροφορίες: **1)** ο κόμβος Κ έχει παιδιά τους κόμβους Λ και Δ **2)** ο κόμβος Μ έχει πατέρα τον κόμβο Λ **3)** ο κόμβος Ν είναι αδέλφια με τον κόμβο Μ έχει παιδιά τους κόμβους Α και Β **3)** ο κόμβος Ε έχει πατέρα τον κόμβο Δ.

|  |  |
| --- | --- |
| **Λύση** | **Παρατηρήσεις** |
| **Κ**  **Λ**  **Δ**  **Μ**  **Ν**  **Ε**  **Α**  **Β** | Για να συνδυάσουμε τις παραπάνω πληροφορίες θα πρέπει να προσέξουμε:   * Ό κόμβος-πατέρας βρίσκεται πάνω από τον κόμβο-παιδί και ενώνονται με ακμή. * Όλοι οι κόμβοι-παιδιά ενός κόμβου-πατέρα βρίσκονται ένα επίπεδο κάτω από τον κόμβο πατέρα και στο ίδιο επίπεδο μεταξύ τους. * Όταν δύο κόμβοι είναι αδέλφια έχουν τον ίδιο κόμβο-πατέρα. * Όταν ένας κόμβος – παιδί έχει δικά του παιδιά, τότε αυτά βρίσκονται ένα επίπεδο κάτω. * Η ρίζα του δένδρου είναι ο κόμβος που δεν έχει πατέρα. |

**Παράδειγμα 2 – Δυαδικό δένδρο αναζήτησης:** Να συμπληρώσετε τα κενά Κ1, Κ2,Κ3, Κ4 στο παρακάτω δένδρο με κατάλληλες τιμές, ώστε να προκύψει ένα δυαδικό δένδρο αναζήτησης.

|  |  |
| --- | --- |
| **Λύση** - **Παρατηρήσεις** | |
| **30**  **Κ1**  **40**  **44**  **15**  **Κ4**  **Κ2**  **Κ3**  Για να προκύψει δυαδικό δένδρο αναζήτησης θα πρέπει κάθε κόμβος, να έχει τιμή μεγαλύτερη από όλους τους κόμβους του αριστερού υποδένδρου του και τιμή μικρότερη από όλους τους κόμβους του δεξιού υποδένδρου του. Οπότε: | * **Κ1:** θα πρέπει να έχει τιμή μικρότερη από το 30 (το Κ1 βρίσκεται στο αριστερό υποδένδρο του 30), μεγαλύτερη από το 4 (το 4 βρίσκεται στο αριστερό υποδένδρο του Κ1) και μικρότερη από το 15 (το 15 βρίσκεται στο δεξί υποδένδρο του Κ1), **για παράδειγμα το 10**. * **Κ2:** θα πρέπει να έχει τιμή μεγαλύτερη από το 10 που επιλέξαμε στο Κ1 (το Κ2 βρίσκεται στο δεξί υποδένδρο του Κ1) και τιμή μικρότερη από το 15 (το Κ2 βρίσκεται στο αριστερό υποδένδρο του 15), **για παράδειγμα το 14**. * **Κ3:** θα πρέπει να έχει τιμή μεγαλύτερη από το 10 που επιλέξαμε στο Κ1 (το Κ3 βρίσκεται στο δεξί υποδένδρο του Κ1) και τιμή μεγαλύτερη από το 15 (το Κ3 βρίσκεται στο δεξί υποδένδρο του 15), **για παράδειγμα το 16**. * **Κ4:** θα πρέπει να έχει τιμή μεγαλύτερη από το 30 (το Κ4 βρίσκεται στο δεξί υποδένδρο του 30) και τιμή μικρότερη από το 40 (το Κ4 βρίσκεται στο αριστερό υποδένδρο του 40), **για παράδειγμα το 35** |

**Παράδειγμα 3 – Δένδρο απόφασης:** Να δημιουργήσετε ένα δένδρο απόφασης, που θα κατηγοριοποιεί τους προορισμούς: Ηράκλειο, Αθήνα, Παρίσι, Νέα Υόρκη σύμφωνα με τα χαρακτηριστικά: **1)** αν είναι προορισμός εσωτερικού ή εξωτερικού **2)** στην περίπτωση του εσωτερικού αν είναι νησί ή όχι **3)** στην περίπτωση του εξωτερικού αν είναι Ευρώπη ή όχι.

**Επεξήγηση λύσης:** Σε ένα δένδρο απόφασης, κάθε κόμβος αποτελεί ένα χαρακτηριστικό, κάθε ακμή μία απόφαση (Ναι – Όχι, Αληθής - Ψευδής) και κάθε φύλο ένα χαρακτηριστικό. Στο παράδειγμά μας, αρχικά θα χρησιμοποιήσουμε ένα κόμβο για το ερώτημα να είναι προορισμός εσωτερικού η όχι, με δύο ακμές (μία για Αληθής και μία για Ψευδής). Στην περίπτωση Αληθής θα χρειαστούμε άλλο ένα κόμβο, για το ερώτημα αν είναι νησί ή όχι (ξανά με ακμές Αληθής - Ψευδής) και θα καταλήξουμε σε δύο φύλλα, ένα για Κρήτη και ένα για Ηράκλειο. Στην περίπτωση Ψευδής, θα χρειαστούμε άλλο ένα κόμβο, για το ερώτημα αν είναι Ευρώπη ή όχι (ξανά με ακμές Αληθής – Ψευδής) και θα καταλήξουμε σε δύο φύλα, ένα για Παρίσι και ένα για Νέα Υόρκη.

Εσωτερικό;

**Α**

**Ψ**

Νησί ;

Ευρώπη;

**Α Ψ Α Ψ**

Κρήτη

Αθήνα

Παρίσι

Νέα Υόρκη

**Παράδειγμα 4 – Δένδρο υπολογισμού αριθμητικών εκφράσεων:** Να δημιουργήσετε ένα δένδρο το οποίο θα αναπαριστά την λύση της πράξης (α+β)^(γ-2).

|  |  |
| --- | --- |
| **Λύση** **Παρατηρήσεις** | |
| ^  6  +  -    α  β  γ  2 | * Η πράξη (τελεστής) που θα εκτελεστεί τελευταία στην ιεραρχία θα τοποθετηθεί ως ρίζα του δένδρου. * Οι πράξεις (τελεστές) που θα εκτελεστούν πρώτες, θα τοποθετηθούν ως παιδιά της ρίζας. * Στα φύλα θα τοποθετήσουμε της μεταβλητές ή αριθμούς στους οποίους εφαρμόζονται οι πράξεις (τελεστές) |

**ΑΣΚΗΣΕΙΣ ΓΙΑ ΛΥΣΗ**

**Άσκηση 1:** Ερωτήσεις Σωστό – Λάθος.

1. Σε ένα δένδρο, όλοι οι κόμβοι έχουν ακριβώς ένα γονέα.
2. Οι κόμβοι χωρίς παιδιά σε ένα δένδρο ονομάζονται φύλλα.
3. Οι κόμβοι με τον ίδιο πατέρα ονομάζονται αδέρφια.
4. Ένας κόμβος-πατέρας, μπορεί να έχει μόνο ένα κόμβο-παιδί.
5. Στα δένδρα μπορούμε εύκολα να προσθέσουμε και να αφαιρέσουμε κόμβους .
6. Ένα δένδρο μπορεί να έχει περισσότερες από μία ρίζες.
7. Ένα από τα προβλήματα στην επίλυση των οποίων βοηθούν τα δένδρα είναι η αυτόματη συμπλήρωση λέξεων σε συσκευές κινητών.
8. Σε ένα δυαδικό δένδρο αναζήτησης, κάθε κόμβος μπορεί να έχει 2 ή περισσότερα παιδιά.
9. Σε ένα δυαδικό δένδρο αναζήτησης, κάθε κόμβος έχει τιμή μεγαλύτερη από όλους τους κόμβους του αριστερού υποδένδρου του και τιμή μικρότερη ή ίση από όλους τους κόμβους του δεξιού υποδένδρου του .
10. Τα δυαδικά δένδρα αναζήτησης προσφέρουν πλεονεκτήματα στην λειτουργία της αναζήτησης.
11. Σε ένα δυαδικό δένδρο αναζήτησης, κάποιος κόμβος μπορεί να έχει ένα μόνο παιδί.
12. Υπάρχουν δένδρα που βοηθούν στην μοντελοποίηση των κινήσεων σε παιχνίδια όπως η τρίλιζα.
13. Σε ένα δένδρο, υπάρχει μία μοναδική διαδρομή για κάθε κόμβο, δηλαδή μία ακολουθία ακμών, που ξεκινάει από τη ρίζα και τερματίζει σε αυτόν τον κόμβο.
14. Ένας κόμβος που είναι παιδί για έναν κόμβο, μπορεί να είναι πατέρας για κάποιον άλλο κόμβο.
15. Σε ένα δυαδικό δένδρο, κάποιος κόμβος μπορεί να έχει τρία υποδένδρα.

**Άσκηση 2:** Να απαντήσετε στις ερωτήσεις σχετικά με το ακόλουθο δένδρο: **1)** ποια είναι η ρίζα του δένδρου; **2)** ποια είναι τα παιδιά της Άννας; **3)** ποια είναι τα αδέρφια του Μάρκου; **4)** ποιος είναι ο πατέρας του Βασίλη; **5)** ποια παιδιά είναι αδέρφια μεταξύ τους; **6)** Ποιοι κόμβοι αποτελούν φύλλα; **7)** να προσθέσετε στον Μάρκο δύο παιδιά, τον Κωνσταντίνο και την Δήμητρα και στον Βασίλη ένα παιδί, την Ελπίδα και να παρουσιάσετε τη νέα μορφή του δένδρου.

Πέτρος

Βασίλης Άννα Μάρκος

Μάγδα Ρωµανός Αχιλλέας

**Άσκηση 3:** Να σχεδιάσετε το δένδρο που προκύπτει από τις ακόλουθες πληροφορίες: **1)** ο κόμβος Α έχει παιδιά τους κόμβους Β και Γ **2)** ο κόμβος Β έχει παιδιά τους κόμβους Δ και Ε **3)** ο κόμβος Γ έχει παιδιά τους κόμβους Ζ και Η.

**Άσκηση 4:** Να σχεδιάσετε το δένδρο που προκύπτει από τις ακόλουθες πληροφορίες: **1)** ο κόμβος Α έχει παιδιά τους κόμβους Β, Γ και Δ **2)** οι κόμβοι Ε και Ζ έχουν πατέρα τον κόμβο Δ **3)** ο κόμβος Κ έχει αδέλφια τους κόμβους Λ και Μ και πατέρα τον κόμβο Γ **4)** κόμβος Π έχει παιδί τον κόμβο Ρ και πατέρα τον κόμβο Β.

**Άσκηση 5:** Να συμπληρώσετε τα κενά Κ1, Κ2,Κ3, Κ4 στα παρακάτω δένδρα με κατάλληλες τιμές, ώστε να προκύψει ένα δυαδικό δένδρο αναζήτησης.

|  |  |
| --- | --- |
| **60**  **Κ1**  **Κ4**  **84**  **30**  **70**  **Κ2**  **Κ3** | **Κ1**  **20**  **50**  **44**  **46**  **Κ4**  **Κ2**  **Κ3** |

**Άσκηση 6:**  Να εξηγήσετε αν τα ακόλουθα δένδρα αποτελούν δυαδικά δένδρα αναζήτησης ή όχι.



γ.

β.

α.

**7**

**4**

**6**

**1**

**6**

**6**

**3**

**1**

**5**

**2**

**3**

**1**

**8**

**5**

**8**

**2**

**7**

**3**

**2**

**7**

**5**

**4**

**4**

δ.

γ.

α. β.

**45**

**12**

**38**

**30**

**12**

**56**

**32**

**65**

**32**

**12**

**32**

**12**

**32**

**42**

**42**

**42**

**42**

**Τα δένδρα είναι από το Συμπληρωματικό Υλικό σελίδα 63.**

**Άσκηση 7:** Δίνονται οι αριθμοί 11,7, 16, 19, 35, 23. Να δημιουργήσετε ένα δυαδικό δένδρο αναζήτησης.

**Άσκηση 8:** Δίνονται οι αριθμοί 50, 30, 60, 31, 58, 62, 61, 29, 70. Να δημιουργήσετε ένα δυαδικό δένδρο αναζήτησης.

**Άσκηση 9:** Να δημιουργήσετε ένα δένδρο απόφασης, που θα κατηγοριοποιεί τις παρακάτω πόλεις: Ρέθυμνο, Ρόδος, Αθήνα, Θεσσαλονίκη σύμφωνα με τα χαρακτηριστικά: **1)** αν είναι νησί ή όχι **2)** στην περίπτωση που είναι νησί, αν είναι στην Κρήτη ή όχι **3)** στην περίπτωση που δεν είναι νησί, αν βρίσκεται στην Μακεδονία ή όχι.

**Άσκηση 10:** Να δημιουργήσετε ένα δένδρο απόφασης, που θα κατηγοριοποιεί τα παρακάτω ζώα: λιοντάρι, κότα, χελιδόνι, πρόβατο, σκύλος, γάτα, άλογο, σύμφωνα με τα χαρακτηριστικά: **1)** αν έχουν δύο πόδια ή όχι **2)** στην περίπτωση που δεν έχουν δύο πόδια, αν είναι χορτοφάγα ή σαρκοφάγα.

**Άσκηση 11:** Να δημιουργήσετε ένα δένδρο το οποίο θα αναπαριστά την λύση της πράξης (α+β+γ)/3.

**Άσκηση 12:** Να δημιουργήσετε ένα δένδρο το οποίο θα αναπαριστά την λύση της πράξης (α+β)\*(γ+δ)\*(κ+λ).

**Ενότητα 1.3.3: Γράφοι**

1. Να περιγράψετε τις γραμμικές και τις μη γραμμικές δομές δεδομένων.

**(Συμπληρωματικό υλικό, ενότητα 1.3.2 και 1.3.3)**

* **Γραμμικές δομές:** σε μία γραμμική δομή, μετά από κάθε στοιχείο ακολουθεί ένα άλλο στοιχείο, εκτός αν είναι τελευταίο. Παράδειγμα γραμμικής δομής είναι οι πίνακες και οι συνδεδεμένες λίστες.
* **Μη-Γραμμικές δομές:** Τα δεδομένα τους δεν ακολουθούν κάποια σειρά, όπως για παράδειγμα στ δένδρα, όπου ξεκινούν από έναν κόμβο-ρίζα και μπορεί να συνδεθεί με άλλους κόμβους, περισσότερους από έναν. Επίσης κάθε δένδρο μπορεί να περιέχει άλλα υποδένδρα.

1. Να δώσετε τον ορισμό του γράφου.

**(Συμπληρωματικό υλικό, ενότητα 1.3.3)** Ένας **γράφος (graph)** είναι μία δομή που αποτελείται από ένα σύνολο κόμβων (ή σημείων ή κορυφών) και ένα σύνολο γραμμών (ή ακμών ή τόξων) που ενώνουν μερικούς ή όλους τους κόμβους. Ο γράφος αποτελεί την πιο γενική δομή δεδομένων, με την έννοια ότι όλες οι προηγούμενες δομές που παρουσιάστηκαν μπορούν να θεωρηθούν περιπτώσεις γράφων



1. Να αναφέρετε κάποιες διαφορές των γράφων σε σχέση με το δένδρα.

**(Συμπληρωματικό υλικό, ενότητα 1.3.3)**

* Ένα δένδρο μπορεί να ρέει μόνο προς μία κατεύθυνση, από τον κόμβο ρίζα σε κόμβους φύλλων ή κόμβους παιδιών. Αντίθετα, οι γράφοι δεν έχουν την έννοια ενός κόμβου ρίζας.
* Τα δένδρα μπορούν να έχουν μόνο μονόδρομες συνδέσεις, δηλαδή ένας κόμβος παιδιού μπορεί να έχει μόνο έναν γονέα. Επίσης ένα δένδρο δεν μπορεί να έχει βρόγχους ή κυκλικούς δεσμούς. Αντίθετα, οι γράφοι μπορούν να συνδεθούν με οποιονδήποτε τρόπο, για παράδειγμα ένας κόμβος μπορεί να συνδεθεί με άλλους πέντε. Επίσης, οι γράφοι δεν έχουν «μονοκατευθυντική» ροή, αντ αυτού μπορεί να έχουν κατεύθυνση ή να μην έχουν καμία κατεύθυνση.
* Το δένδρα είναι περιορισμένοι τύποι γράφων. Ένα δένδρο θα είναι πάντα γράφος, ενώ ένας γράφος δεν είναι πάντα δένδρο.
* Τα δένδρα, γενικά διέπονται από κανόνες, όπως για παράδειγμα στα δυαδικά δένδρα που κάθε κόμβος μπορεί να έχει συνδέσεις μέχρι δύο κόμβους, ενώ οι γράφοι δεν έχουν αυτούς τους περιορισμούς και είναι η πιο γενική δομή δεδομένων.

1. Τι ονομάζουμε κατευθυνόμενες ακμές και τι μη κατευθυνόμενη ακμή;

**(Συμπληρωματικό υλικό, ενότητα 1.3.3)**

* Κατευθυνόμενη ακμή: είναι η ακμή που έχει κατεύθυνση. Στην περίπτωση αυτή οι κόμβοι συνδέονται σε πολύ συγκεκριμένο τρόπο.
* Μη κατευθυνόμενη ακμή: είναι η ακμή που δεν έχει κατεύθυνση. Στην περίπτωση αυτή η σύνδεση μεταξύ των κόμβων είναι πιο γενική.

|  |  |
| --- | --- |
| **Α** **Β**   * Στο συγκεκριμένο παράδειγμα, η ακμή είναι κατευθυνόμενη. * Ο κόμβος Α ονομάζεται προέλευση και ο κόμβος Β προορισμός. * Μπορούμε να «ταξιδέψουμε» μόνο από την πηγή στον προορισμό, δηλαδή από τον κόμβο Α στον κόμβο Β. | **Α Β**   * Αντίθετα, εδώ η ακμή είναι μη – κατευθυνόμενη. * Οι κόμβοι προέλευσης και προορισμού δεν είναι σταθεροί. * Μπορούμε να «ταξιδέψουμε» και προς τις δύο κατευθύνσεις, δηλαδή η διαδρομή είναι αμφίδρομη. |

1. Τι ονομάζουμε κατευθυνόμενο και τι μη- κατευθυνόμενο γράφο;

**(Συμπληρωματικό υλικό, ενότητα 1.3.3)**

* Εάν όλες οι ακμές σε έναν γράφο έχουν κατεύθυνση, ο γράφος ονομάζεται **κατευθυνόμενος γράφος (directed graph)**.
* Εάν όλες οι ακμές σε έναν γράφο δεν έχουν κατεύθυνση, ο γράφος ονομάζεται **μη κατευθυνόμενος γράφος (undirected graph).**

1. Να δώσετε διάφορα παραδείγματα γράφων.

**(Συμπληρωματικό υλικό, ενότητα 1.3.3)**

* **Παγκόσμιος Ιστός (WWW):** Είναι ένας τεράστιος τύπος γράφου, όπου κάποιοι κόμβοι έχουν μη κατευθυνόμενες ακμές (δηλαδή μπορούμε να πάμε από την ιστοσελίδα Α στην ιστοσελίδα Β και το αντίστροφο) και κάποιοι κατευθυνόμενες ακμές (μπορούμε να πάμε μόνο από την ιστοσελίδα Α στην ιστοσελίδα Β και όχι το αντίστροφο).
* **Facebook:**  Είναι τύπος γράφου, με αμφίδρομη σχέση κόμβων, δηλαδή μη κατευθυνόμενες ακμές, καθώς αν κάποιος χρήστης είναι φίλος με κάποιον άλλο, ισχύει και το αντίστροφο.
* **Twitter:** Είναι τύπος γράφου, που γενικά υπάρχουν κατευθυνόμενες ακμές, καθώς αν ο χρήστης Α ακολουθεί τον χρήστη Β, δεν είναι απαραίτητο να ισχύει και το αντίθετο.

**Παράδειγμα 1 – Δημιουργία γράφου:** Να σχεδιάσετε μία μορφή δεδομένων «Γράφος» θα αναπαριστά οδικές συνδέσεις μεταξύ των πόλεων Α, Β, Γ, Δ, Ε, με βάση τις ακόλουθες πληροφορίες:

* Η πόλη Α ενώνεται με τις πόλεις Β και Γ.
* Η πόλη Γ ενώνεται με τις πόλεις Ε και Δ.
* Η πόλη Β συνδέεται με τις πόλεις Ε και Δ

**Επεξήγηση λύσης:** Θα χρησιμοποιήσουμε μη κατευθυνόμενες ακμές, καθώς οι οδικές συνδέσεις είναι αμφίδρομες. Κάθε πόλη που έχει σύνδεση με μία άλλη, ενώνονται με μία ακμή.

**Αιωρ**

**Β**

**Δ**

**Ε**

**Γ**

**ΑΣΚΗΣΕΙΣ ΓΙΑ ΛΥΣΗ**

**Άσκηση 1:** Ερωτήσεις Σωστό – Λάθος.

1. Ένας γράφος έχει υποχρεωτικά μία ρίζα.
2. Ο γράφος είναι η πιο γενική δομή δεδομένων.
3. Σε ένα γράφο, ένας κόμβος μπορεί να συνδέεται μέχρι και με δύο άλλους κόμβους.
4. Σε μία γραμμική δομή δεδομένων, τα δεδομένα ακολουθούν μία συγκεκριμένη σειρά.
5. Ένα δένδρο είναι πάντα γράφος.
6. Ένας γράφος είναι πάντα δένδρο.
7. Υπάρχουν δύο είδη ακμών, οι κατευθυνόμενες και οι μη κατευθυνόμενες.
8. Εάν όλες οι ακμές σε έναν γράφο έχουν κατεύθυνση, ο γράφος ονομάζεται κατευθυνόμενος..
9. Σε μία μη κατευθυνόμενη ακμή, η σύνδεση των κόμβων είναι αμφίδρομη.
10. Στο Facebook, υπάρχει μη αμφίδρομη σχέση μεταξύ των χρηστών.

**Άσκηση 2:** Να σχεδιάσετε έναν γράφο, για την αναπαράσταση οδικών διαδρομών, με βάση τις παρακάτω πληροφορίες. Να εξηγήσετε επίσης τι είδους ακμές θα χρησιμοποιήσετε και γιατί.

* Η πόλη Α συνδέεται με τις πόλεις Β και Ε.
* Η πόλη Β συνδέεται με τις πόλεις Γ και Ζ.
* Η πόλη Γ συνδέεται με τις πόλεις Ζ και Δ.
* Η πόλη Ε συνδέεται με τις πόλεις Δ και Ζ.

**Άσκηση 3:** Να σχεδιάσετε έναν γράφο, για την αναπαράσταση σύνδεσης ιστοσελιδών, με βάση τις παρακάτω πληροφορίες. Να εξηγήσετε επίσης τι είδους ακμές θα χρησιμοποιήσετε και γιατί.

* Η ιστοσελίδα Α συνδέεται με τις ιστοσελίδες Β και Γ με αμφίδρομη σχέση.
* Η ιστοσελίδα Β συνδέεται με τις ιστοσελίδες Δ και Ε με αμφίδρομη σχέση.
* Η ιστοσελίδα Ε μπορεί να συνδεθεί με την ιστοσελίδα Ζ αλλά όχι το αντίθετο.
* Η ιστοσελίδα Γ μπορεί να συνδεθεί με την ιστοσελίδα Π αλλά όχι το αντίθετο.

**Παράγραφος 3.1 (Δεδομένα) σχολικού, παράγραφος 4.1 σχολικού (Ανάλυση προβλημάτων) + Ενότητα 2.1 συμπληρωματικού υλικού, Παράγραφος 6.1 σχολικού (Η έννοια του προγράμματος)**

1. Από ποιες σκοπιές μελετά τα δεδομένα η πληροφορική;

**(Παράγραφος 3.1 σχολικού βιβλίου)**

* **Υλικού**: Το υλικό (hardware), δηλαδή η μηχανή, επιτρέπει στα δεδομένα ενός προγράμματος να αποθηκεύονται στην κύρια μνήμη και στις περιφερειακές συσκευές του υπολογιστή με διάφορες αναπαραστάσεις (representations). Τέτοιες μορφές είναι η δυαδική, ο κώδικας ΑSCII, ο κώδικας EBCDIC, το συμπλήρωμα του 1 ή του 2 κ.λπ.
* **Γλωσσών προγραμματισμού**. Οι γλώσσες προγραμματισμού υψηλού επιπέδου (high level programming languages) επιτρέπουν τη χρήση διάφορων τύπων (types) μεταβλητών (variables) για να περιγράψουν ένα δεδομένο. Ο μεταφραστής κάθε γλώσσας φροντίζει για την αποδοτικότερη μορφή αποθήκευσης, από πλευράς υλικού, κάθε μεταβλητής στον υπολογιστή.
* **Δομών Δεδομένων**. Δομή δεδομένων (data structure) είναι ένα σύνολο δεδομένων μαζί με ένα σύνολο επιτρεπτών λειτουργιών επί αυτών. Για παράδειγμα, μία τέτοια δομή είναι η εγγραφή (record), που μπορεί να περιγράφει ένα είδος, ένα πρόσωπο κλπ.
* **Ανάλυσης Δεδομένων**. Τρόποι καταγραφής και αλληλοσυσχέτισης των δεδομένων μελετώνται έτσι ώστε να αναπαρασταθεί η γνώση για πραγματικά γεγονότα. Οι τεχνολογίες των Βάσεων Δεδομένων (Databases), της Μοντελοποίησης Δεδομένων (Data Modelling) και της Αναπαράστασης Γνώσης (Knowledge Representation) ανήκουν σε αυτή τη σκοπιά μελέτης των δεδομένων.

1. Γιατί πρέπει να γίνεται ανάλυση του προβλήματος πριν την επίλυση του;

**(Παράγραφος 4.1 σχολικού βιβλίου)**  Είναι πιθανόν ένα πρόβλημα να μην επιλύεται με μία μόνο λύση αλλά με περισσότερες. Γενικά, η λύση σε ένα πρόβλημα μπορεί να προέλθει από ποικίλες και διαφορετικές προσεγγίσεις, τεχνικές και μεθόδους. Έτσι, είναι απαραίτητο να γίνεται μία καλή ανάλυση του κάθε προβλήματος και να προτείνεται συγκεκριμένη μεθοδολογία και ακολουθία βημάτων, η οποία θα οδηγήσει σε έξυπνες και αποδοτικές λύσεις.

1. Ποια βήματα περιλαμβάνει η ανάλυση ενός προβλήματος σε ένα σύγχρονο προγραμματιστικό περιβάλλον;

**(Παράγραφος 4.1 σχολικού βιβλίου)** **1)** την καταγραφή της υπάρχουσας πληροφορίας για το πρόβλημα **2)** την αναγνώριση των ιδιαιτεροτήτων του προβλήματος **3)** την αποτύπωση των συνθηκών και προϋποθέσεων υλοποίησής του και στη συνέχεια: **4)** την πρόταση επίλυσης με χρήση κάποιας μεθόδου **5)** την τελική επίλυση με χρήση υπολογιστικών συστημάτων.

1. Σε ποιες ερωτήσεις θα πρέπει να απαντήσουμε κατά την ανάλυση ενός προβλήματος;

**(Παράγραφος 4.1 σχολικού βιβλίου) 1)** Ποια είναι τα δεδομένα και το μέγεθος του προβλήματος **2)** Ποιες είναι οι συνθήκες που πρέπει να πληρούνται για την επίλυσητου προβλήματος **3)** Ποια είναι η πλέον αποδοτική μέθοδος επίλυσής τους (σχεδίαση αλγορίθμου) **4)** Πώς θα καταγραφεί η λύση σε ένα πρόβλημα (π.χ. σε ψευδογλώσσα) **5)** Ποιος είναι ο τρόπος υλοποίησης στο συγκεκριμένο υπολογιστικό σύστημα (π.χ. επιλογή γλώσσας προγραμματισμού).

1. Να δώσετε ένα παράδειγμα ανάλυσης προβλήματος.

**(Παράγραφος 4.1 σχολικού βιβλίου)** Ας θεωρήσουμε έναν ταχυδρόμο, ο οποίος πρέπει να ξεκινήσει από ένα χωριό, να επισκεφθεί έναν αριθμό γειτονικών χωριών για να μοιράζει γράμματα και να επιστρέψει στο αρχικό χωριό, περνώντας μόνο μία φορά από το κάθε φορά. Θα μπορούσαμε να αναλύσουμε το πρόβλημα με τους ακόλουθους τρόπους:

* Να επιλέγει ο ταχυδρόμος την πιο κοντινή απόσταση σε σχέση με το χωριό που βρίσκεται.
* Να υπολογιστούν όλες οι πιθανές διαδρομές και να επιλεχθεί αυτή με την ελαχιστοποίηση της συνολικής απόστασης και όχι την ελαχιστοποίηση της απόστασης κάθε φορά. Η προσέγγιση αυτή απαιτεί περισσότερους υπολογιστικού πόρους.

1. Ποια είναι η σημασία της ανάλυσης των προβλημάτων;

**(Παράγραφος 4.1 σχολικού βιβλίου)** Η ανάλυση κάθε προβλήματος είναι απαραίτητη, έτσι ώστε να αναζητηθεί η πλέον κατάλληλη μέθοδος που να παρέχει τη ζητούμενη λύση, όσο γίνεται ταχύτερα και με το λιγότερο δυνατό κόστος σε υπολογιστικούς πόρους.

1. Υπάρχει κάποιος γενικός κανόνας ανάλυσης προβλημάτων;

**(Παράγραφος 4.1 σχολικού βιβλίου)** Δεν υπάρχει ένας ενιαίος κανόνας, μία γενική φόρμουλα που να αναφέρεται στην επίλυση του συνόλου των προβλημάτων. Υπάρχουν όμως “συγγενή” προβλήματα, δηλαδή προβλήματα που μπορούν να αναλυθούν με παρόμοιο τρόπο και να αντιμετωπισθούν με αντίστοιχες μεθόδους και τεχνικές.

1. Ποια είναι τα χαρακτηριστικά της ανάλυσης των προβλημάτων;

**(Παράγραφος 4.1 σχολικού βιβλίου) 1)** παρέχουν ένα γενικό πρότυπο κατάλληλο για την επίλυση προβλημάτων ευρείας κλίμακας **2)** μπορούν να αναπαρασταθούν με κοινές δομές δεδομένων και ελέγχου(που υποστηρίζονται από τις περισσότερες σύγχρονες γλώσσες προγραμματισμού) **3)** παρέχουν τη δυνατότητα καταγραφής των χρονικών και “χωρικών” απαιτήσεων της μεθόδου επίλυσης, έτσι ώστε να μπορεί να γίνει επακριβής εκτίμηση των αποτελεσμάτων.

1. Περιγράψτε την μέθοδο «Διαίρει και Βασίλευε».

**(Συμπληρωματικό υλικό, ενότητα 2.1)** Η **«Διαίρει και Βασίλευε»** (divide and conquer) αποτελεί μια μέθοδο σχεδίασης αλγορίθμων στην οποία εντάσσονται οι τεχνικές που υποδιαιρούν ένα πρόβλημα σε μικρότερα υποπροβλήματα, που έχουν την ίδια τυποποίηση με το αρχικό πρόβλημα, αλλά είναι μικρότερα σε μέγεθος. Με όμοιο τρόπο, τα υποπροβλήματα αυτά μπορούν να διαιρεθούν σε ακόμη μικρότερα υποπροβλήματα κ.ο.κ. Έτσι η επίλυση ενός προβλήματος έγκειται στη σταδιακή επίλυση των όσο το δυνατόν μικρότερων υποπροβλημάτων, ώστε τελικά να προκύψει η συνολική λύση του αρχικού ευρύτερου προβλήματος.Η προσέγγιση αυτή ονομάζεται «από πάνω προς τα κάτω» (top-down).

1. Με ποια βήματα μπορεί να αποδοθεί η μέθοδος «Διαίρει και Βασίλευε»;

**(Συμπληρωματικό υλικό, ενότητα 2.1) 1)**  δίνεται για επίλυση ένα στιγμιότυπο ενός προβλήματος **2)** το στιγμιότυπο του προβλήματος υποδιαιρείται σε υποστιγμιότυπα του ίδιου προβλήματος **3)** Δίνεται ανεξάρτητη λύση σε κάθε ένα υποστιγμιότυπο **4)** συνδυάζονται όλες οι μερικές λύσεις που βρέθηκαν για τα υποστιγμιότυπα, έτσι ώστε να δοθεί η συνολική λύση του προβλήματος.

1. Πως μπορούμε να υπολογίσουμε τον μέγιστο αριθμό συγκρίσεων που απαιτούνται για την εύρεση ενός στοιχείου σε ένα σύνολο ταξινομημένων στοιχείων;

**(Συμπληρωματικό υλικό, ενότητα 2.1)** Στο πλαίσιο του μαθήματος η υλοποίηση της μεθόδου «Διαίρει και Βασίλευε» γίνεται με την επαναληπτική προσέγγιση (με διαδοχικές επαναλήψεις).Ο μέγιστος αριθμός των συγκρίσεων (επαναλήψεων) που απαιτούνται για την εύρεση ενός στοιχείου σε ένα σύνολο «n» ταξινομημένων στοιχείων, συμπεριλαμβανομένης και της περίπτωσης μη ύπαρξης του στοιχείου, δίνεται από το ακέραιο μέρος του [log2(n)+1] (με στρογγυλοποίηση προς τα κάτω), η απόδειξη του οποίου υπερβαίνει τα όρια της διδακτέας ύλης του μαθήματος. Επομένως, για την εύρεση του μέγιστου πλήθους των επαναλήψεων θεωρείται γνωστό το log2(n). Για παράδειγμα, σε ένα σύνολο 100 ταξινομημένων στοιχείων (n=100), ο μέγιστος αριθμός συγκρίσεων (επαναλήψεων) είναι: [log2(100)+1]=[6,643856+1]=[7,643856]=7. Χαρακτηριστικό παράδειγμα που ακολουθεί την λογική της μεθόδου «Διαίρει και Βασίλευε» είναι η Δυαδική αναζήτηση.

1. Ποια είναι τα στάδια επίλυσης ενός προβλήματος;

**(Παράγραφος 6.1 σχολικού βιβλίου) 1)** τον ακριβή προσδιορισμό του προβλήματος **2)** την ανάπτυξη του αντίστοιχου αλγορίθμου **3)** η διατύπωση του αλγορίθμου σε κατανοητή μορφή από τον υπολογιστή.

1. Με ποιο από τα παραπάνω στάδια ασχολείται ο προγραμματισμός;

**(Παράγραφος 6.1 σχολικού βιβλίου)** Ο προγραμματισμός ασχολείται με το τρίτο αυτό στάδιο, τη δημιουργίατου προγράμματος δηλαδή του συνόλου των εντολών που πρέπει να δοθούν στον υπολογιστή, ώστε να υλοποιηθεί ο αλγόριθμος για την επίλυσητου προβλήματος. Το πρόγραμμα, το οποίο γράφεται σε κάποια γλώσσαπρογραμματισμού, δεν είναι απλά η υλοποίηση του αλγορίθμου, αλλά βασικό στοιχείο του είναι τα δεδομένα και οι δομές δεδομένων επί των οποίων ενεργεί.

1. Ποιες είναι οι καταστάσεις που καταλαβαίνει ένας υπολογιστής και ποιες είναι οι στοιχειώσεις ενέργειες που μπορεί να εκτελέσει ένας υπολογιστής;

**(Παράγραφος 6.1 σχολικού βιβλίου)**

* Ο υπολογιστής, ως γνωστόν, είναι μία μηχανή που καταλαβαίνει μόνο δύο καταστάσεις, οι οποίες αντιπροσωπεύονται με δύο αριθμούς το μηδέν και το ένα, τα ψηφία του δυαδικού συστήματος. Το μόνο πράγμα που κάνει ο υπολογιστής είναι στοιχειώδεις ενέργειες σε ακολουθίες αυτών των δύο ψηφίων, αλλά αυτές τις ενέργειες τις εκτελεί μα ασύλληπτη ταχύτητα.
* Ο υπολογιστής μπορεί απλά να αποθηκεύει στη μνήμη τις ακολουθίες των δυαδικών ψηφίων, να τις ανακτά, να κάνει στοιχειώδεις αριθμητικές πράξεις με αυτές και να τις συγκρίνει.

**ΑΣΚΗΣΕΙΣ ΓΙΑ ΛΥΣΗ**

**Άσκηση 1:** Ερωτήσεις Σωστό – Λάθος.

1. Το υλικό του υπολογιστή δεν σχετίζεται με την αποθήκευση των δεδομένων στην μνήμη του υπολογιστή.
2. Οι γλώσσες προγραμματισμού υψηλού επιπέδου επιτρέπουν τη χρήση διάφορων τύπων μεταβλητών για να περιγράψουν ένα δεδομένο.
3. Ένα πρόβλημα επιλύεται με ένα μοναδικό τρόπο.
4. Η ανάλυση ενός προβλήματος αρχίζει με την καταγραφή της υπάρχουσας πληροφορίας για το πρόβλημα.
5. Κατά την ανάλυση ενός προβλήματος, σκοπός είναι να επιλέξουμε την πλέον αποδοτική επίλυση του.
6. Ένα από τα ερωτήματα στα οποία πρέπει να απαντήσουμε κατά την ανάλυση ενός προβλήματος, είναι ο τρόπος υλοποίησης του σε ένα συγκεκριμένο υπολογιστικό σύστημα.
7. Γενικά υπάρχει ένας ενιαίος κανόνας για την ανάλυση και επίλυση των προβλημάτων.
8. Ένα από τα χαρακτηριστικά της ανάλυσης των προβλημάτων, είναι ότι μπορούν να αναπαρασταθούν με κοινές δομές δεδομένων και ελέγχου.
9. Στην μέθοδο «Διαίρει και Βασίλευε», ένα πρόβλημα υποδιαιρείται σε μικρότερα υποπροβλήματα.
10. Η μέθοδος «Διαίρει και Βασίλευε» χρησιμοποιεί την «από κάτω προς τα πάνω» προσέγγιση.
11. Ο μέγιστος αριθμός επαναλήψεων που απαιτούνται για την εύρεση ενός στοιχείου σε ένα σύνολο ταξινομημένων στοιχείων από την μέθοδο «Διαίρει και Βασίλευε» υπολογίζεται από τον τύπο [log2(n)+1].
12. Βασικό στοιχείο των προγραμμάτων είναι οι αλγόριθμοι, τα δεδομένα και οι δομές δεδομένων.
13. Ο υπολογιστής καταλαβαίνει μόνο δύο καταστάσεις, το μηδέν και το ένα.
14. Ο υπολογιστής μπορεί να εκτελέσει στοιχειώδεις ενέργειες σε ακολουθίες από μηδέν και ένα.
15. Ο υπολογιστής δεν έχει την δυνατότητα αποθήκευσης στην μνήμη ακολουθίες δυαδικών ψηφίων.

**Άσκηση 2:** Να συμπληρώσετε τα κενά στις ακόλουθες προτάσεις:

1. Το υλικό (hardware), δηλαδή η \_\_\_\_, επιτρέπει στα \_\_\_\_ ενός προγράμματος να αποθηκεύονται στην κύρια \_\_\_\_ και στις περιφερειακές συσκευές του υπολογιστή με διάφορες \_\_\_\_ . Τέτοιες μορφές είναι η \_\_\_\_, ο κώδικας ΑSCII, ο κώδικας EBCDIC, το συμπλήρωμα του 1 ή του 2 κ.λ.π.
2. Δεν υπάρχει ένας ενιαίος \_\_\_\_ , μία γενική φόρμουλα που να αναφέρεται στην \_\_\_\_ του συνόλου των προβλημάτων. Υπάρχουν όμως “\_\_\_\_” προβλήματα, δηλαδή προβλήματα που μπορούν να \_\_\_\_ με παρόμοιο τρόπο και να αντιμετωπισθούν με αντίστοιχες μεθόδους και \_\_\_\_.
3. Η **«Διαίρει και Βασίλευε»** (divide and conquer) αποτελεί μια μέθοδο \_\_\_\_ αλγορίθμων στην οποία εντάσσονται οι τεχνικές που υποδιαιρούν ένα πρόβλημα σε μικρότερα \_\_\_\_, που έχουν την ίδια \_\_\_\_ με το αρχικό πρόβλημα, αλλά είναι μικρότερα σε \_\_\_\_.
4. Ο υπολογιστής μπορεί απλά να \_\_\_\_ στη μνήμη τις ακολουθίες των \_\_\_\_ ψηφίων, να τις \_\_\_\_, να κάνει στοιχειώδεις \_\_\_\_ πράξεις με αυτές και να τις \_\_\_\_.

**Παράγραφοι 13.1 και 13.2 σχολικού, Ενότητα 5 συμπληρωματικού υλικού: Κατηγορίες λαθών και εκσφαλμάτωση**

1. Να αναφέρετε τις κατηγορίες λαθών που μπορεί να παρουσιαστούν σε κάποιο πρόγραμμα.

**(Κεφάλαιο 13.1 σχολικού βιβλίου)** Τα λάθη που μπορεί να παρουσιαστούν σε ένα πρόγραμμα είναι: **1)** λάθη κατά την υλοποίηση (συντακτικά λάθη) **2)** λάθη κατά την εκτέλεση (λάθη χρόνου εκτέλεσης) **3)** λογικά λάθη.

1. Τι γνωρίζετε για τα λάθη κατά την υλοποίηση;

**(Κεφάλαιο 13.1 σχολικού βιβλίου)**

* Τα λάθη κατά το χρόνο υλοποίησης, προκαλούνται κυρίως από λανθασμένη σύνταξη εντολών προγράμματος. Τέτοια λάθη μπορεί να είναι η λανθασμένη συγγραφή μιας δεσμευμένης λέξης της γλώσσας προγραμματισμού ή η χρήση μιας δομής ελέγχου χωρίς την εντολή τερματισμού της.
* Ένα λάθος που προκαλείται κατά τη συγγραφή του προγράμματος, ανιχνεύεται από το μεταγλωττιστή, ο οποίος εμφανίζει προς το προγραμματιστή κάποιο προειδοποιητικό μήνυμα. Αν το πρόγραμμα περιέχει ένα λάθος αυτής της μορφής, δεν επιτρέπεται η εκτέλεσή του, μέχρι να το διορθώσει ο προγραμματιστής.
* Τα σύγχρονα προγραμματιστικά περιβάλλοντα μας προφυλάσσουν αυτόματα από τα λάθη κατά την υλοποίηση , αφού παρέχουν εργαλεία αυτόματου ελέγχου σύνταξης των εντολών και παρακολουθούν τον προγραμματιστή κατά τη συγγραφή του προγράμματος. Μόλις διαπιστώσουν κάποιο συντακτικό λάθος, σταματούν και απαιτούν τη διόρθωσή του. Συνήθως αντιλαμβάνονται ακριβώς το λάθος που δημιουργήθηκε και προτείνουν αναλυτικά τον τρόπο διόρθωσής του, εμφανίζοντας σε ενημερωτικό πλαίσιο την ορθή σύνταξη της εντολής που προκλήθηκε το λάθος.

1. Τι γνωρίζετε για τα λάθη κατά την εκτέλεση;

**(Κεφάλαιο 13.1 σχολικού βιβλίου)**

* Τα λάθη που προκαλούνται κατά το χρόνο εκτέλεσης του προγράμματος, είναι πιο επώδυνα γιατί συνήθως εμφανίζονται σε πραγματικό περιβάλλον εκτέλεσης και τις περισσότερες φορές προκαλούν τον αντικανονικό τερματισμό της εφαρμογής και το *κρέμασμα* (crash) του συστήματος.
* Όταν ένα λάθος προκληθεί κατά την εκτέλεση της εφαρμογής, είναι δυνατό να αντιμετωπισθεί μόνο με τη χρήση εντολών προγράμματος που το παγιδεύουν και εκτελούν τις κατάλληλες διαδικασίες χειρισμού του.
* Η πρόληψη τέτοιων λαθών είναι αρκετά δύσκολη, αφού συνήθως οφείλονται σε καταστάσεις που δεν είναι εύκολο να ελεγχθούν από τον προγραμματιστή, ενώ πολλές φορές εμφανίζονται μετά από μεγάλο χρονικό διάστημα. Τέτοια λάθη είναι δυνατό να προκληθούν από την κλήση μιας διαδικασίας με δεδομένα που δεν μπορεί να χειριστεί, όπως η αναζήτηση διαγραμμένων αρχείων, η προσπάθεια διαίρεσης ενός αριθμού με το μηδέν, η υπερχείλιση μιας αριθμητικής μεταβλητής ή από δυσλειτουργία του υλικού μέρους του υπολογιστή, όπως η καταστροφή του σκληρού δίσκου του συστήματος, ο τερματισμός μιας σύνδεσης δικτύου και η αποσύνδεση του εκτυπωτή.

1. Τι γνωρίζετε για τα λογικά λάθη;

**(Κεφάλαιο 13.1 σχολικού βιβλίου)**

* Τα λογικά λάθη είναι συνήθως λάθη σχεδιασμού και δεν προκαλούν τη διακοπή της εκτέλεσης του προγράμματος. Ενώ ο μεταγλωττιστής της γλώσσας προγραμματισμού δεν ανιχνεύει κανένα συντακτικό λάθος και κατά την εκτέλεση του προγράμματος δεν παρουσιάζονται ανεπιθύμητες καταστάσεις σφαλμάτων, τελικά δεν παράγονται τα επιθυμητά αποτελέσματα.
* Η ανίχνευση τέτοιων λαθών δεν είναι δυνατό να πραγματοποιηθεί από κάποιο εργαλείο του υπολογιστή και διαπιστώνονται μόνο με τη *διαδικασία ελέγχου* (testing) και την ανάλυση των αποτελεσμάτων των προγραμμάτων.

1. Τι γνωρίζεται για την διαδικασία της εκσφαλμάτωσης;

**(Παράγραφος 13.2 σχολικού βιβλίου)**

* Η διαδικασία ελέγχου, εντοπισμού και διόρθωσης των σφαλμάτων ενός προγράμματος καλείται *εκσφαλμάτωση* (debugging). Στόχος της διαδικασίας εκσφαλμάτωσης είναι ο εντοπισμός των σημείων του προγράμματος που προκαλούν προβλήματα στη λειτουργία του.
* Η εργασία της εκσφαλμάτωσης δεν είναι εύκολη, απαιτεί βαθιά γνώση της γλώσσας προγραμματισμού και φυσικά αντίστοιχες ικανότητες από τον προγραμματιστή.
* Για τον εντοπισμό ενός λάθους δεν υπάρχουν ιδιαίτερα μυστικά και τρυκ. Η εκσφαλμάτωση είναι ένα πρόβλημα λογικής και όσο πιο καλά αντιλαμβάνεται ο προγραμματιστής τον τρόπο που εργάζεται το πρόγραμμα, τόσο πιο εύκολα και σύντομα θα εντοπίσει λάθη που προκαλούν δυσλειτουργίες.
* Σε ένα σύγχρονο προγραμματιστικό περιβάλλον δεν χρειάζεται ιδιαίτερα μνεία για τα λάθη που παρουσιάζονται κατά το χρόνο σχεδιασμού, αφού αυτά, όπως αναφέρθηκε, είναι συντακτικά λάθη και τις περισσότερες φορές το περιβάλλον προγραμματισμού τα ανιχνεύει αυτόματα και προτείνει τη διόρθωσή τους ή τα εντοπίζει ο μεταγλωττιστής.
* Τα λάθη που κυρίως μας απασχολούν στη φάση της εκσφαλμάτωσης είναι τα λογικά λάθη και τα λάθη που παρουσιάζονται κατά το χρόνο εκτέλεσης του προγράμματος.
* Η εκσφαλμάτωση τέτοιων λαθών μπορεί να γίνει μέσα από εργαλεία εκσφαλμάτωσης ή από ειδικές εντολές ή συναρτήσεις που προσφέρει το περιβάλλον προγραμματισμού.

**Παράδειγμα 1 – εντοπισμός συντακτικών λαθών:** Να εντοπίσετε τα συντακτικά λάθη που υπάρχουν στο ακόλουθο πρόγραμμα και να προτείνετε διορθώσεις.

|  |  |
| --- | --- |
| **Συντακτικά λάθη** | **Διορθώσεις** |
| **ΠΡΟΓΡΑΜΜΑ** λάθη\_1  **ΜΕΤΑΒΛΗΤΕΣ**  **ΑΛΦΑΡΙΜΗΤΙΚΕΣ:** όνομα  **ΠΡΑΓΜΑΤΙΚΕΣ:** βαθμός  **ΑΚΕΡΑΙΕΣ:** i  **ΑΡΧΗ**  **ΓΙΑ** i **ΑΠΟ** 1 **ΕΩΣ** 10  **ΔΙΑΒΑΣΕ** όνομα, βαθμός  **ΕΜΦΑΝΙΣΕ** όνομα, βαθμός  **ΤΕΛΟΣ**  **ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ** | **ΠΡΟΓΡΑΜΜΑ** σωστό\_1  **ΜΕΤΑΒΛΗΤΕΣ**  **ΧΑΡΑΚΤΗΡΕΣ:** όνομα **! λάθος🡪ΑΛΦΑΡΙΘΜΗΤΙΚΕΣ**  **ΠΡΑΓΜΑΤΙΚΕΣ:** βαθμός  **ΑΚΕΡΑΙΕΣ:** i  **ΑΡΧΗ**  **ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 10 **! λάθος🡪ΕΩΣ**  **ΔΙΑΒΑΣΕ** όνομα, βαθμός  **ΓΡΑΨΕ** όνομα, βαθμός **! λάθος🡪ΕΜΦΑΝΙΣΕ**  **ΤΕΛΟΣ\_ ΕΠΑΝΑΛΗΨΗΣ ! λάθος🡪ΤΕΛΟΣ**  **ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ** |

**Παράδειγμα 2 – εντοπισμός λαθών που οδηγούν σε αντικανονικό τερματισμό του προγράμματος:** Να εντοπίσετε τα λάθη που υπάρχουν στον ακόλουθο αλγόριθμο και να προτείνετε διορθώσεις.

|  |  |  |
| --- | --- | --- |
| **Κώδικας με λάθη** | **Ανάλυση** | **Διορθώσεις** |
| **Αλγόριθμος** λάθη\_2  **Διάβασε** κ, λ  μ🡨κ/λ  **Εμφάνισε** μ  **Τέλος** λάθη\_2 | * Αν ο χρήστης δώσει τιμή 0 στο λ, θα επιχειρηθεί διαίρεση με το 0 και το πρόγραμμα θα τερματιστεί αντικανονικά. * Θα προσθέσουμε μία δομή επιλογής με την οποία θα ελέγχουμε την περίπτωση αυτή. | **Αλγόριθμος** σωστός\_2  **Διάβασε** κ, λ  **Αν** λ<>0 **τότε ! έλεγχος**  μ🡨κ/λ  **Εμφάνισε** μ  **Αλλιώς**  **Εμφάνισε** ‘δεν ορίζεται’  **Τέλος**\_**αν**  **Τέλος** σωστός\_2 |

**Παράδειγμα 3 – εντοπισμός λογικών λαθών:** Να εντοπίσετε τα λογικά λάθη που υπάρχουν στον ακόλουθο αλγόριθμο και να προτείνετε διορθώσεις. Η εκφώνηση είναι «Να αναπτυχθεί αλγόριθμος που θα διαβάζει 3 αριθμούς και θα εμφανίζει τον μέσο όρο τους.»

|  |  |  |
| --- | --- | --- |
| **Κώδικας με λάθη** | **Ανάλυση** | **Διορθώσεις** |
| **Αλγόριθμος** λάθη\_3  **Διάβασε** α,β,γ  μο🡨α+β+γ/3  **Εμφάνισε** μο  **Τέλος** λάθη\_3 | * Ο χρήστης έχει παραλείψει την παρένθεση στον υπολογισμό του μέσου όρου. * Θα εμφανιστεί αποτέλεσμα αλλά θα είναι λάθος. * Πολλές φορές τα λάθη δεν βρίσκονται στην εντολή που εμφανίζει το λάθος αποτέλεσμα, αλλά σε κάποια προηγούμενη, οπότε θα πρέπει να ελέγχουμε το σύνολο του αλγορίθμου. | **Αλγόριθμος** σωστός\_3  **Διάβασε** α,β,γ  μο🡨(α+β+γ)/3  **Εμφάνισε** μο  **Τέλος** σωστός\_3 |

**Παράδειγμα 4 – εκσφαλμάτωση λογικών λαθών σε δομή επιλογής:** Ένας μαθητής είχε ως εργασία να επιλύσει το πρόβλημα «Να αναπτυχθεί αλγόριθμος που θα διαβάζει δύο αριθμούς από τον χρήστη, αν το άθροισμά τους είναι θετικό να εμφανίζει την απόλυτη τιμή της διαφοράς τους, αν είναι αρνητικό θα εμφανίζει το γινόμενό τους, διαφορετικά θα εμφανίζει μήνυμα «μηδέν».Να εντοπίσετε τα λογικά λάθη που υπάρχουν στον ακόλουθο αλγόριθμο και να προτείνετε διορθώσεις.

|  |  |  |
| --- | --- | --- |
| **Κώδικας με λάθη** | **Ανάλυση** | **Διορθώσεις** |
| **Αλγόριθμος** λάθη\_4  **Διάβασε** κ, λ  **Αν** κ+λ>3 **τότε**  **Εμφάνισε** Α\_Τ(α-β)  **Αλλιώς\_αν** κ+λ<0 **τότε**  **Εμφάνισε** α\*β  **Αλλιώς**  **Εμφάνισε** ‘δεν ορίζεται’  **Τέλος**\_**αν**  **Τέλος** λάθη\_4 | * Στις δομές επιλογής ελέγχουμε για λάθη **α)** στην συνθήκη **β)** στις εντολές μέσα στις περιπτώσεις. * Στο παράδειγμά μας, υπάρχει λάθος και στην πρώτη συνθήκη και στις εντολές * Στην ανίχνευση των λαθών δεν αρκεί η μεμονωμένη μελέτη των συνθηκών αλλά χρειάζεται να μελετηθεί το αποτέλεσμα που παράγει ο συνδυασμός των συνθηκών και των ομάδων εντολών. | **Αλγόριθμος** σωστό\_4  **Διάβασε** κ, λ  **Αν** κ+λ>0 **τότε**  **Εμφάνισε** Α\_Τ(α-β)  **Αλλιώς\_αν** κ+λ<0 **τότε**  **Εμφάνισε** α\*β  **Αλλιώς**  **Εμφάνισε** ‘μηδέν  **Τέλος**\_**αν**  **Τέλος** σωστό\_4 |
| Για να εντοπίσουμε τα λογικά λάθη, μπορούμε να δώσουμε κάποιες ενδεικτικές τιμές για να εντοπίσουμε αν λειτουργεί σωστά σε όλες τις περιπτώσεις. Για παράδειγμα:  α=5 και β= 8: Αναμενόμενο αποτέλεσμα 🡪3 | Αποτέλεσμα 🡪 3 (Σωστό)  α= -3 και β= -2: Αναμενόμενο αποτέλεσμα 🡪 6 | Αποτέλεσμα 🡪 6 (Σωστό)  α=2 και β= 1: Αναμενόμενο αποτέλεσμα 🡪1 | Αποτέλεσμα 🡪 «Δεν ορίζεται» (Λάθος)  α=4 και β= -4: Αναμενόμενο αποτέλεσμα🡪 «Μηδέν» | Αποτέλεσμα 🡪 «Δεν ορίζεται» (Λάθος) | | |

**Παράδειγμα 5 – εκσφαλμάτωση λογικών λαθών σε δομή επανάληψης:** Ένας μαθητής είχε ως εργασία να επιλύσει το πρόβλημα «Να αναπτυχθεί αλγόριθμος που θα διαβάζει επαναληπτικά αριθμούς μέχρι να δοθεί ο αριθμός 5, που θα υπολογίζεται στους αριθμούς και θα τερματίζει την επανάληψη. Στο τέλος να εμφανίζει το άθροισμα των αριθμών».Να εντοπίσετε τα λογικά λάθη που υπάρχουν στον ακόλουθο αλγόριθμο και να προτείνετε διορθώσεις.

|  |  |  |
| --- | --- | --- |
| **Κώδικας με λάθη** | **Ανάλυση** | **Διορθώσεις** |
| **Αλγόριθμος** λάθη\_5  sum🡨0  **Διάβασε** κ  **Όσο** κ<>5 **επανάλαβε**  sum🡨sum+κ  **Διάβασε** κ  **Τέλος\_επανάληψης**  **Εμφάνισε** sum  **Τέλος** λάθη\_5 | Στις δομές επανάληψης ελέγχουμε για λάθη: **1)** στην συνθήκη επανάληψης ή τερματισμού **2)** στην αρχικοποίηση της συνθήκης **3)** στην ενημέρωση της συνθήκης εντός του βρόγχου επανάληψης **4)** στις εντολές μέσα στον βρόγχο **5)** στους λογικούς και συγκριτικούς τελεστές **6)** στο κριτήριο της περατότητας **7)** στην πρώτη και την τελευταία επανάληψη για το αν θα πρέπει να συμπεριληφθεί στους υπολογισμούς ή όχι. | **Αλγόριθμος** σωστό\_5  sum🡨0  **Αρχή\_επανάληψης**  **Διάβασε** κ  sum🡨sum+κ  **Μέχρις\_ότου** κ=5  **Εμφάνισε** sum  **Τέλος** σωστό\_5 |
| Στο παράδειγμα μας, η τιμή 5 πρέπει να συμπεριληφθεί στους υπολογισμούς, κάτι που δεν γίνεται στην δομή ΌΣΟ της λύσης. Για παράδειγμα:  Τιμές: 2,3,5: Αναμενόμενο αποτέλεσμα 🡪30 | Αποτέλεσμα 🡪 6 (Λάθος)  Μία λύση είναι η δομή Μέχρις\_ότου, η οποία συμπεριλαμβάνει και τον τελευταίο αριθμό. | | |

**Παράδειγμα 6 – εκσφαλμάτωση λογικών λαθών στους πίνακες:** Δίνεται το ακόλουθο τμήμα αλγορίθμου το οποίο εμφανίζει τα στοιχεία του πίνακα Α[5] που είναι μεγαλύτερα ή ίσα από το επόμενο στοιχείο τους. Τα στοιχεία του πίνακα Α είναι [10, 10, 5, 8, 6]

|  |  |  |
| --- | --- | --- |
| **Κώδικας με λάθη** | **Ανάλυση** | **Διορθώσεις** |
| **Για** i **από** 1 **μέχρι** 5  **Αν Α**[i]>A[i+1] **τότε**  **Εμφάνισε** Α[i]  **Αλλιώς**  **Εμφάνισε** ‘όχι’  **Τέλος\_αν**  **Τέλος\_επανάληψης** | Στους πίνακες ελέγχουμε για λάθη:   * Στο μέγεθος των πινάκων κατά την δήλωση τους. * Στους δείκτες των πινάκων κατά την προσπέλαση τους. * Στη μη υπέρβαση των ορίων του πίνακα. | **Για** i **από** 1 **μέχρι** 4  **Αν Α**[i]>=A[i+1] **τότε**  **Εμφάνισε** Α[i]  **Αλλιώς**  **Εμφάνισε** ‘όχι’  **Τέλος\_αν**  **Τέλος\_επανάληψης** |
| Στο παράδειγμα μας, **1)** η επανάληψη πρέπει να σταματάει στην τιμή 4 **2)** πρέπει να προσθέσουμε = στην συνθήκη.  i=1: Αναμενόμενο αποτέλεσμα 🡪10 | Αποτέλεσμα 🡪 «όχι» (Λάθος)  i=2: Αναμενόμενο αποτέλεσμα 🡪 10 | Αποτέλεσμα 🡪 10 (Σωστό)  i=3: Αναμενόμενο αποτέλεσμα 🡪«όχι» | Αποτέλεσμα 🡪 «όχι» (Σωστό)  i=4: Αναμενόμενο αποτέλεσμα🡪 8| Αποτέλεσμα 🡪 8 (Σωστό)  i=5: Αντικανονικός τερματισμός | | |

**Παράδειγμα 7 – εκσφαλμάτωση λογικών λαθών στα υποπρογράμματα:** «Να αναπτύξετε πρόγραμμα το οποίο θα διαβάζει από τον χρήστη 4 ακέραιους αριθμούς και θα εμφανίζει το άθροισμα τους, με την βοήθεια συνάρτησης η οποία θα δέχεται ως παραμέτρους το άθροισμα και τον αριθμό και θα επιστρέφει τη νέα τιμή του αθροίσματος». Να εντοπίσετε τα λάθη.

|  |  |
| --- | --- |
| **Κώδικας με λάθη** | **Διορθώσεις** |
| **ΠΡΟΓΡΑΜΜΑ** λάθη  **ΜΕΤΑΒΛΗΤΕΣ**  **ΑΚΕΡΑΙΕΣ:** i, sum, x  **ΑΡΧΗ**  sum🡨1  **ΓΙΑ** i **ΑΠΟ** 1 **ΕΩΣ** 4  **ΔΙΑΒΑΣΕ** x  sum🡨F(sum,i)  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  **ΓΡΑΨΕ** sum  **ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**  **ΣΥΝΑΡΤΗΣΗ** F(sum,x):**ΑΚΕΡΑΙΑ**  **ΜΕΤΑΒΛΗΤΕΣ**  **ΑΚΕΡΑΙΕΣ:** sum, x  **ΑΡΧΗ**  F🡨sum-x  **ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ** | **ΠΡΟΓΡΑΜΜΑ** σωστό  **ΜΕΤΑΒΛΗΤΕΣ**  **ΑΚΕΡΑΙΕΣ:** i, sum, x  **ΑΡΧΗ**  sum🡨0 **! αντί για 1, λάθος αρχικοποίηση**  **ΓΙΑ** i **ΑΠΟ** 1 **ΕΩΣ** 4  **ΔΙΑΒΑΣΕ** x  sum🡨F(sum,x) **! αντί για f(sum,i), λάθος παράμετρος**  **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**  **ΓΡΑΨΕ** sum  **ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**  **ΣΥΝΑΡΤΗΣΗ** F(sum,x):**ΑΚΕΡΑΙΑ**  **ΜΕΤΑΒΛΗΤΕΣ**  **ΑΚΕΡΑΙΕΣ:** sum, x  **ΑΡΧΗ**  F🡨sum+x **! αντί για –x, λάθος πράξη**  **ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ** |
| Στα υποπρογράμματα ελέγχουμε για λάθη: **1)** στην κλήση του υποπρογράμματος και στο πέρασμα των παραμέτρων **2)** στα γενικά λάθη που υπάρχουν και σε ένα πρόγραμμα.  **Δοκιμαστική εκτέλεση:** αν δώσουμε τους αριθμούς 2, 4, 1, 6 το αναμενόμενο αποτέλεσμα είναι το 13, ενώ το αρχικό πρόγραμμα θα δώσει -12. | | |

**Παράδειγμα 8 – μέθοδος ελέγχου «Μαύρο κουτί»: «**Η βαθμολογία σε ένα διαγώνισμα στο μάθημα της πληροφορικής, κυμαίνεται στο διάστημα [1,100] σε ακέραιες τιμές. Για να πετύχει ο μαθητής στο διαγώνισμα, θα πρέπει να συγκεντρώσει τουλάχιστον 60 μονάδες. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο: **1)** θα διαβάζει την βαθμολογία που συγκέντρωσε ένας μαθητής **2)** θα εμφανίζει «επιτυχία» αν πέτυχε στο διαγώνισμα, ή «αποτυχία» σε αντίθετη περίπτωση **3)** αν δοθεί βαθμολογία εκτός [1,100] να εμφανίζει «λάθος δεδομένα»». Με βάση την παραπάνω εκφώνηση, να δημιουργήσετε κατάλληλα σενάρια για να πραγματοποιήσετε έλεγχο ακραίων τιμών.

* Ένα **σενάριο ελέγχου (test case)** περιγράφει τα δεδομένα εισόδου ολόκληρου του προγράμματος ή τμήματος του προγράμματος (διαδικασία, συνάρτηση) και τα αναμενόμενα αποτελέσματα. Τα σενάρια ελέγχου εκτελούνται, είτε σε πραγματικό περιβάλλον προγραμματισμού είτε εικονικά με δημιουργία πίνακα τιμών των μεταβλητών. Σε περίπτωση αποκλίσεων μεταξύ των αναμενόμενων και των πραγματικών αποτελεσμάτων, υπάρχει λάθος το οποίο πρέπει να εντοπιστεί και να διορθωθεί.
* Μια δημοφιλής τεχνική ελέγχου είναι ο **έλεγχος μαύρου κουτιού (black-box testing)**. Ονομάζεται έτσι επειδή τα δεδομένα εισόδου στα σενάρια ελέγχου προκύπτουν από τις προδιαγραφές του προγράμματος, αγνοώντας εντελώς τον κώδικα. Δηλαδή το πρόγραμμα μοιάζει σαν να βρίσκεται μέσα σε ένα μαύρο κουτί που κρύβει το περιεχόμενό του.
* Ιδανικά θα θέλαμε να ελέγξουμε όλες τις τιμές εισόδου και όλα τα πιθανά αποτελέσματα. Αυτό όμως είναι αδύνατο. Γι’ αυτό προσπαθούμε να βρούμε αντιπροσωπευτικές τιμές για τα δεδομένα εισόδου που θα παράγουν αντιπροσωπευτικά αποτελέσματα.
* Το πρώτο βήμα είναι η **δημιουργία ισοδύναμων διαστημάτων τιμών (equivalence partitioning)** για τα δεδομένα εισόδου. Τα διαστήματα θεωρούνται ισοδύναμα, καθώς αν δεν υπάρχουν λάθη, τότε όλες οι τιμές ενός διαστήματος εισόδου θα παράγουν τιμές που θα ανήκουν στο ίδιο διάστημα αποτελεσμάτων.
* Μετά τον καθορισμό των διαστημάτων πρέπει να επιλεγούν τιμές για τα σενάρια ελέγχου που να καλύπτουν όλα τα διαστήματα. Αφού τα διαστήματα είναι ισοδύναμα, μπορεί να επιλεγεί οποιαδήποτε τιμή από κάθε διάστημα.
* Μια καλύτερη στρατηγική είναι να γίνει **έλεγχος των ακραίων τιμών κάθε διαστήματος (boundary value analysis)**, καθώς η εμπειρία έχει δείξει ότι τα περισσότερα λάθη γίνονται σε αυτά τα σημεία. Αυτό είναι λογικό, αν σκεφτούμε ότι τα διαστήματα τιμών θα υλοποιηθούν με κάποια μορφή δομής επιλογής, οπότε μπορεί να υπάρχουν λάθη στις λογικές συνθήκες, π.χ. συμπερίληψη ακραίας τιμής (<= αντί για <, >= αντί για >), παράλειψη ακραίας τιμής (< αντί για <=, > αντί για >=).
* **Βήμα 1:**  Δημιουργία ισοδύναμων διαστημάτων.

Υπάρχουν δύο διαστήματα για τις τιμές εισόδου: 1<=βαθμός<60 και 60<=βαθμός<=100.

Υπάρχουν δύο διαστήματα μη έγκυρων τιμών: βαθμός<0 και βαθμός >100.

-------------------- > | 1 < --------------------- > | 60 < -------------------- > 100 | < ----------------------

Λάθος τιμές Αποτυχία Επιτυχία Λάθος τιμές

* **Βήμα 2:** Καθορισμός ακραίων τιμών των ισοδύναμων διαστημάτων.

-------------------- > 0 | 1 < --------------------- >59 | 60 < -------------------- > 100 | 101 < ----------------

Λάθος τιμές Αποτυχία Επιτυχία Λάθος τιμές

* **Βήμα 3:** Δημιουργία σεναρίων ελέγχου για κάθε ακραία τιμή.

**Είσοδος**🡪 0 (άνω άκρο για βαθμός<1) **Αναμενόμενο αποτέλεσμα**🡪Λάθος δεδομένα

**Είσοδος**🡪 1 (κάτω άκρο για 1<=βαθμός<60) **Αναμενόμενο αποτέλεσμα**🡪Αποτυχία

**Είσοδος**🡪 59 (άνω άκρο για 1<βαθμός<60) **Αναμενόμενο αποτέλεσμα**🡪Αποτυχία

**Είσοδος**🡪 60 (κάτω άκρο για 60<=βαθμός<=100) **Αναμενόμενο αποτέλεσμα**🡪Επιτυχία

**Είσοδος**🡪 100 (άνω άκρο για 60<=βαθμός<=100) **Αναμενόμενο αποτέλεσμα**🡪Επιτυχία

**Είσοδος**🡪 101 (κάτω άκρο για βαθμός>100) **Αναμενόμενο αποτέλεσμα**🡪Λάθος δεδομένα

**ΑΣΚΗΣΕΙΣ ΓΙΑ ΛΥΣΗ**

**Άσκηση 1:** Ερωτήσεις Σωστό – Λάθος.

1. Τα λάθη κατά τον χρόνο υλοποίησης προκαλούνται κυρίως από λανθασμένη σύνταξη εντολών του προγράμματος.
2. Τα λάθη κατά τον χρόνο υλοποίησης δεν εντοπίζονται από τον μεταγλωττιστή.
3. Τα σύγχρονα προγραμματιστικά περιβάλλοντα παρέχουν εργαλεία αυτόματου έλεγχου σύνταξης των εντολών κατά την συγγραφή του προγράμματος.
4. Τα λάθη κατά τον χρόνο εκτέλεσης τις περισσότερες φορές προκαλούν την αντικανονική διακοπή του προγράμματος.
5. Ένα παράδειγμα λάθους κατά τον χρόνο εκτέλεσης είναι η προσπάθεια διαίρεσης ενός αριθμού με το 0.
6. Τα λογικά λάθη προκαλούν την διακοπή εκτέλεσης του προγράμματος.
7. Τα λογικά λάθη μπορούν να εντοπιστούν από τον μεταγλωττιστή.
8. Στόχος της εκσφαλμάτωσης είναι ο εντοπισμός των σημείων του προγράμματος που προκαλούν προβλήματα στη λειτουργία του .
9. Η τεχνική ελέγχου «Μαύρο Κουτί» στηρίζεται άμεσα στην κώδικα της λύσης.
10. Στις τεχνικές ελέγχου είναι δύσκολο να ελέγξουμε όλες τις πιθανές τιμές.

**Άσκηση 2:** Να εντοπίσετε τα λάθη που υπάρχουν στις παρακάτω κωδικοποιήσεις, να τα χαρακτηρίσετε ως λογικά, συντακτικά ή χρόνου εκτέλεσης και να προτείνετε κάποια λύση για κάθε μία περίπτωση.

|  |  |  |
| --- | --- | --- |
| «Το παρακάτω πρόγραμμα πρέπει να διαβάζει 3 αριθμούς από τον χρήστη και να εμφανίζει το άθροισμά τους».  **ΠΡΟΓΡΑΜΜΑ** λάθη\_1  **ΜΕΤΑΒΛΗΤΕΣ**  **ΑΡΙΘΜΗΤΙΚΕΣ:** κ, α, β, γ  **ΑΡΧΗ**  **ΔΙΑΒΑΣΕ** α, β, γ  κ:=α+β+γ  **ΤΥΠΩΣΕ** όνομα, βαθμός  **ΤΕΛΟΣ** | «Ο παρακάτω αλγόριθμος πρέπει να διαβάζει 20 αριθμούς από τον χρήστη και να εμφανίζει το πλήθος των αρνητικών από αυτούς».  **Αλγόριθμος** λάθη\_2  πλήθος🡨-1  **Για** κ **από** 1 **μέχρι** 30  **Διάβασε** λ  **Αν** λ>0 **τότε**  πλήθος🡨πλήθος-1  **Τέλος**\_**αν**  **Τέλος**\_**επανάληψης**  **Εμφάνισε** πλήθος  **Τέλος** λάθη\_2 | «Ο παρακάτω αλγόριθμος πρέπει να διαβάζει έναν αριθμό από τον χρήστη και να εμφανίζει την τετραγωνική ρίζα του».  **Αλγόριθμος** λάθη\_3  **Διάβασε** κ  λ🡨Τ\_Ρ(κ)  **Εμφάνισε** λ  **Τέλος** λάθη\_3 |

**Άσκηση 3:** Το ακόλουθο τμήμα αλγορίθμου έχει ως σκοπό να εμφανίσει για κάθε στοιχείο Α[5] με τιμές 3, -1 ,0, 10 , -4 μήνυμα σχετικά με το αν είναι θετικό, αρνητικό ή μηδέν. **1)** να συμπληρώσετε τον ακόλουθο πίνακα ώστε να πραγματοποιήσετε δοκιμαστική εκτέλεση των εντολών

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Τιμή κ | Τιμή Α[κ] | Έξοδος εντολών | Αναμενόμενη έξοδος | Ορθότητα τιμών (Σωστό η Λάθος) |

**2)** να βρείτε τα λάθη που περιέχουν οι εντολές και να τα χαρακτηρίσετε ως προς τι είδος τους **3)** να προτείνετε λύσεις που θα διορθώνουν τα παραπάνω λάθη.

**Για** κ **από** 1 **μέχρι** 6

**Αν** Α[κ]>=0 **τότε**

**Εμφάνισε** ‘Θετικός’

**Αλλιώς**\_**αν** Α[κ] < -1 **τότε**

**Εμφάνισε** ‘Αρνητικός’

**Αλλιώς**

**Εμφάνισε** ‘Μηδέν’

**Τέλος**\_**αν**

**Τέλος**\_**επανάληψης**

**Άσκηση 4:** «Να αναπτύξετε πρόγραμμα το οποίο **1)** θα διαβάζει 5 αριθμούς από τον χρήστη με έλεγχο για τιμές από 10 μέχρι 200 **2)** θα καλεί διαδικασία η οποία θα εμφανίζει αν ο αριθμός είναι άρτιος ή περιττός **3)** θα εμφανίζει το γινόμενο όλων των αριθμών που δόθηκαν». Για την λύση του παραπάνω προβλήματος αναπτύχθηκε το ακόλουθο πρόγραμμα το οποίο περιέχει ΛΟΓΙΚΆ λάθη. Να τα εντοπίσετε και να τα διορθώσετε.

**ΠΡΟΓΡΑΜΜΑ** λάθη

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** i, γιν, x

**ΑΡΧΗ**

γιν🡨0

**ΓΙΑ** i **ΑΠΟ** 1 **ΜΕΧΡΙ** 4

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΔΙΑΒΑΣΕ** x

**ΜΕΧΡΙΣ**\_**ΟΤΟΥ** x<10 **H** x>200

γιν🡨γιν+x

**ΚΑΛΕΣΕ** Δ(γιν)

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** γιν

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

**ΔΙΑΔΙΚΑΣΙΑ** Δ(x)

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** x

**ΑΡΧΗ**

**ΑΝ** x**mod**0=3 **TOTE**

**ΓΡΑΨΕ** ‘ΑΡΤΙΟΣ’

**ΑΛΛΙΩΣ**

**ΓΡΑΨΕ** ‘ΠΕΡΙΤΤΟΣ’

**ΤΕΛΟΣ**\_**ΑΝ**

**ΤΕΛΟΣ\_ΔΙΑΔΙΚΑΣΙΑΣ**

**Άσκηση 5:** Να παρουσιάσετε κατάλληλα σενάρια για να πραγματοποιήσετε έλεγχο ακραίων τιμών στις ακόλουθες περιπτώσεις:

|  |  |
| --- | --- |
| Η βαθμολογία σε ένα τεστ στα μαθηματικά ορίζεται στα όρια [1, 50]. Για να πετύχει ο μαθητής στο τεστ χρειάζεται βαθμολογία τουλάχιστον 30. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο θα διαβάζει την βαθμολογία ενός μαθητή (υποθέστε ακέραια τιμή) και θα εμφανίζει μήνυμα «Επιτυχία» ή «Αποτυχία». Στην περίπτωση που δοθεί βαθμός εκτός ορίων, να εμφανίζει «Εκτός ορίων». | Αν ένας μαθητής είναι μέχρι και 50 κιλά θεωρείται αδύνατος, αν είναι πάω από 50 κιλά μέχρι και 85 κιλά θεωρείται κανονικός, αν είναι πάνω από 85 κιλά θεωρείται βαρύς. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο θα διαβάζει τα κιλά ενός μαθητή (υποθέστε πραγματικός αριθμός με ένα δεκαδικό ψηφίο) και θα εμφανίζει κατάλληλο μήνυμα «αδύνατος», «κανονικός», «βαρύς», σύμφωνα με την περιγραφή. Αν δοθεί τιμή αρνητική ή μεγαλύτερη του 120 να εμφανίζει «Λάθος». |

**Ενότητα 4 συμπληρωματικού υλικού: Αντικειμενοστραφής προγραμματισμός**

1. Ποια είναι η βασική φιλοσοφία του αντικειμενοστραφούς προγραμματισμού;

**(Συμπληρωματικό υλικό, ενότητα 4.1)**

* Σύμφωνα με την αντικειμενοστραφή θεωρία ανάπτυξης εφαρμογών, η προσέγγιση κάθε προβλήματος πρέπει να γίνεται με φυσική ερμηνεία και να μη στηρίζεται σε πολύπλοκα τεχνικά ζητήματα.
* Αυτή η αντίληψη, ότι δηλαδή η επίλυση ενός προβλήματος επιτυγχάνεται με τη σύνθεση ικανοτήτων (ο τρόπος υλοποίησης των οποίων μας είναι άγνωστος) που διαθέτουν διαφορετικές ανεξάρτητες οντότητες, οι οποίες αλληλεπιδρούν για τον σκοπό αυτό, βρίσκεται στην καρδιά της αντικειμενοστραφούς προσέγγισης.
* Για παράδειγμα, για το χτίσιμο ενός σπιτιού, συνεργάζονται αρχιτέκτονες, μηχανικοί, υδραυλικοί, οικοδόμοι κτλ. Η κάθε ειδικότητα δεν γνωρίζει τις λεπτομέρειες για το πώς υλοποιείται η εργασία των άλλων ιδιοτήτων, όμως συνεργάζονται μεταξύ τους, άμεσα η έμμεσα, ώστε να λύσουν το αρχικό πρόβλημα.
* Στηρίζεται στο γεγονός ότι για να μπορέσει κάποιος να κατανοήσει άγνωστες σε αυτόν έννοιες, θα πρέπει να καθοδηγηθεί μέσω της προσομοίωσης των άγνωστων αυτών εννοιών αντιστοιχίζοντας αυτές σε πρακτικές γνώσεις και εικόνες από το περιβάλλον του, τις οποίες γνωρίζει και μπορεί πολύ εύκολα να χειριστεί.

1. Να δώσετε τους ορισμούς για τις ακόλουθες έννοιες: Αντικειμενοστραφής προγραμματισμός, αντικείμενα, ιδιότητες, μέθοδοι.

**(Συμπληρωματικό υλικό, ενότητα 4.1)**

* **Αντικειμενοστραφής προγραμματισμός** (object-oriented programming) ή αντικειμενοστραφής σχεδίαση είναι μια μεθοδολογία ανάπτυξης εφαρμογών η οποία στηρίζεται σε αυτόνομες προγραμματιστικές οντότητες με δική τους ταυτότητα και συμπεριφορά.
* Οι οντότητες αυτές καλούνται **αντικείμενα** (objects), αντιστοιχούν σε φυσικές οντότητες ή έννοιες του φυσικού μας κόσμου, και δομούνται με βάση δεδομένα (ιδιότητες) που προσδιορίζουν την υπόστασή τους και ενέργειες (κανόνες συμπεριφοράς) που εφαρμόζονται πάνω στα δεδομένα. Σε μια εφαρμογή, ένα αντικείμενο είναι ο ομαδοποιημένος συνδυασμός δεδομένων και κώδικα, τα οποία έχουμε τη δυνατότητα να χειριστούμε ενιαία.
* Tα δεδομένα αποτελούν τα χαρακτηριστικά ενός αντικειμένου και αναφέρονται ως **ιδιότητες** (properties) ενώ οι ενέργειες καθορίζουν τη συμπεριφορά του. Οι ενέργειες στον αντικειμενοστραφή προγραμματισμό αναφέρονται και ως **μέθοδοι** (methods).

1. Ποια είναι τα βασικά συστατικά που πρέπει να καταγράψουμε για την επίλυση ενός προβλήματος με βάση τον αντικειμενοστραφή προγραμματισμό;

**(Συμπληρωματικό υλικό, ενότητα 4.2.1)**

* τα **αντικείμενα** που συμμετέχουν με βάση τον ρόλο τους στο συγκεκριμένο σενάριο
* οι **ιδιότητες** κάθε αντικειμένου, δηλ. τα σχετικά με το συγκεκριμένο πρόβλημα χαρακτηριστικά του
* οι **υπηρεσίες** που προσφέρει ή οι **ενέργειες** που υλοποιεί κάθε αντικείμενο (μέθοδοι) προς αξιοποίηση από άλλες, ώστε να αναπτυχθούν οι απαραίτητες **συνεργασίες** μεταξύ των αντικειμένων για την επίλυση του προβλήματος.

1. Μεθοδολογία Εύρεσης αντικειμένων, ιδιοτήτων, ενεργειών / υπηρεσιών και ειδών συνεργασίας.

**«**Η μητέρα σας, η οποία ήταν αδύνατο να στείλει αυτοπροσώπως λουλούδια στη φίλη της την Άννα, η οποία ζει στη Ρώμη, έπρεπε να επισκεφθεί το ανθοπωλείο της γειτονιάς, να δώσει τα προσωπικά της στοιχεία (όνομα, επώνυμο, διεύθυνση, τηλέφωνο, email), να περιγράψει στον κ. Γιώργο τον ανθοπώλη το είδος της ανθοδέσμης που ήθελε να στείλει, να δώσει τη διεύθυνση και τα στοιχεία επικοινωνίας της Άννας στη Ρώμη. Ο κ. Γιώργος, ο οποίος έπρεπε να στείλει ένα μήνυμα στο δίκτυο ανθοπωλών που συμμετείχε για να εντοπίσει συνεργαζόμενο ανθοπωλείο στη Ρώμη και να αποστείλει τα στοιχεία της παραγγελίας της μαμάς, ο τοπικός ανθοπώλης κ. Τζιοβάνι να αποδεχτεί τη συνεργασία, να αναθέσει στον ανθοδέτη του κ. Αντόνιο να φτιάξει την ανθοδέσμη και στη συνέχεια να καλέσει τον ταχυμεταφορέα κ. Πέπε να παραδώσει την ανθοδέσμη στην κα Άννα.» Με βάση το παραπάνω σενάριο, να εντοπίσετε τα αντικείμενα, τις ιδιότητες, τις ενέργειες / υπηρεσίες και τα είδης συνεργασίας.

**(Συμπληρωματικό υλικό, ενότητα 4.2.1)**

* **Αντικείμενα:** Οι οντότητες που συμμετέχουν στο σενάριο μας είναι: Μαμά (Πελάτης), Γιώργος (Ανθοπώλης), Τζιοβάνι (Ανθοπώλης), Αντώνιο (Ανθοδέτης), Πέπε (Ταχυμεταφορέας), Άννα (Πελάτης).
  + Στα αντικείμενα προσδιορίσαμε και τον ρόλο τους, κάτι που θα μας βοηθήσει στην δημιουργία κλάσεων στη συνέχεια.
* **Ιδιότητες:** Τα χαρακτηριστικά κάθε αντικειμένου, δηλαδή οι πληροφορίες που πρέπει να γνωρίζουμε για αυτό είναι (στο παράδειγμα μας δεν αναφέρονται αναλυτικά, θεωρούμε τα παρακάτω):
  + **Μαμά (Πελάτης):** όνομα, επώνυμο, διεύθυνση, τηλέφωνο, email
  + **Άννα (Πελάτης):** όνομα, επώνυμο, διεύθυνση, τηλέφωνο, email
  + **Γιώργος (Ανθοπώλης):** επωνυμία εταιρίας, όνομα ιδιοκτήτη, επώνυμο ιδιοκτήτη, διεύθυνση, ΑΦΜ, τηλέφωνο, email, τραπεζικός λογαριασμός, κωδικός δικτύου συνεργασίας.
  + **Τζιοβάνι (Ανθοπώλης):** επωνυμία εταιρίας, όνομα ιδιοκτήτη, επώνυμο ιδιοκτήτη, διεύθυνση, ΑΦΜ, τηλέφωνο, email, τραπεζικός λογαριασμός, κωδικός δικτύου συνεργασίας.
  + **Αντόνιο (Ανθοδέτης):** επωνυμία εταιρίας, όνομα ιδιοκτήτη, επώνυμο ιδιοκτήτη, διεύθυνση, ΑΦΜ, τηλέφωνο, email, ειδικότητα, ωριαία αμοιβή.
  + **Πέπε (Ανθοδέτης):** επωνυμία εταιρίας, όνομα ιδιοκτήτη, επώνυμο ιδιοκτήτη, διεύθυνση, ΑΦΜ, τηλέφωνο, email, τύπος.
* **Ενέργειες / υπηρεσίες:** Θα καταγράψουμε τις ενέργειες ή τις υπηρεσίες (μέθοδοι) που πραγματοποιεί κάθε αντικείμενο. Τα ονόματα των μεθόδων παρουσιάζονται με βάση την ονοματοδοσία των υποπρογραμμάτων ( δηλαδή όνομα\_μεθόδου() ).
  + **Μαμά (Πελάτης):** ΚάνειΠαραγγελία()
  + **Άννα (Πελάτης):** ΠαραλαμβάνειΑνθοδέσμη()
  + **Γιώργος (Ανθοπώλης):** ΔέχεταιΠαραγγελία(), ΖητάΣυνεργασία()
  + **Τζιοβάνι (Ανθοπώλης):** ΑποδέχεταιΣυνεργασία(), ΑναθέτειΑνθοσεσία(), ΑναθέτειΠαράδοση()
  + **Αντόνιο (Ανθοδέτης):** ΕτοιμάζειΑνθοδέσμη()
  + **Πέπε (Ανθοδέτης):** ΠαραδίδειΑνθοδέσμη()
* **Είδος Συνεργασίας:** θα καταγράψουμε τον τρόπο με τον οποίο συνεργάζονται τα αντικείμενα μεταξύ τους
  + **Παραγγελία:** Μαμά (Πελάτης) – Γιώργος (Ανθοπώλης)
  + **Συνεργασία:** Γιώργος (Ανθοπώλης) – Τζιοβάνι (Ανθοπώλης)
  + **Ανάθεση ανθοδεσίας:** Τζιοβάνι (Ανθοπώλης) – Αντόνιο (Ανθοδέτης)
  + **Ανάθεση παράδοσης:** Τζιοβάνι (Ανθοπώλης) – Πέπε (Ταχυμεταφορέας)
  + **Παράδοση:** Πέπε (Ταχυμεταφορέας) – Άννα (Πελάτης)

1. Διαγραμματική Αναπαράσταση των συστατικών ενός προβλήματος.

**(Συμπληρωματικό υλικό, ενότητα 4.2.2)** Αφού εντοπίσαμε τα συστατικά επίλυσης του προβλήματος, μπορούμε να τα αναπαραστήσουμε διαγραμματικά, ώστε να έχουμε μία εικόνα των συνεργαζόμενων οντοτήτων. Αποτελεί ουσιαστικά το σχέδιο επίλυσης του με βάση την αντικειμενοστραφής προσέγγιση. Χρησιμοποιούμε:

* **Παραλληλόγραμμα**, για την αποτύπωση των αντικειμένων, των ιδιοτήτων και των μεθόδων τους.
* **Γραμμές σύνδεσης,** για την περιγραφή του είδους συνεργασίας τους

|  |
| --- |
| Όνοµα Αντικειµένου 2 (Ρόλος) |
| Ιδιότητα1:  Ιδιότητα2:  ... |
|
| Μέθοδος1() Μέθοδος2()  ... |

|  |
| --- |
| Όνοµα Αντικειµένου 1 (Ρόλος) |
| Ιδιότητα1:  Ιδιότητα2:  ... |
|
| Μέθοδος1() Μέθοδος2()  ... |

\_\_\_\_ συνεργασία \_\_\_\_

Ολοκληρωμένο το παραπάνω παράδειγμα αποστολής λουλουδιών:

Μαμά (Πελάτης)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Όνομα: Μαρία

Επώνυμο: Πέτρου

Διεύθυνση: Λαυρίου 2, Αθήνα

Τηλέφωνο: 6972233444

Email: [maria@hotmail.com](mailto:maria@hotmail.com)

………………………….

ΚάνειΠαραγγελεία()

Άννα (Πελάτης)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Όνομα: Άννα

Επώνυμο: Γεωργίου

Διεύθυνση: Μοντάνα 4, Ρώμη

Τηλέφωνο: 4932323234

Email: [anna@hotmail.com](mailto:anna@hotmail.com)

………………………….

ΠαραλαμβάνειΑνθοδέσμη()

Γιώργος (Ανθοπώλης)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Επωνυμία εταιρίας: Φλόρα ΕΠΕ

Όνομα ιδιοκτήτη: Γιώργος

Επώνυμο ιδιοκτήτη: Παπαδόπουλος

Διεύθυνση: Πηνειού 120, Αθήνα

ΑΦΜ:999888777

Τηλέφωνο:2101234567

Email: info@fiora.gr

Tραπεζικός λογαριασμός: GR111222

Kωδικός Δικτύου Συνεργασίας: ww12

………………………….

ΑπόδέχεταιΠαραγγελεία()

ΖητάΣυνεργασία()

Τζιβάνι (Ανθοπώλης)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Επωνυμία εταιρίας: Φιόρι ΟΕ

Όνομα ιδιοκτήτη: Τζιοβάνι

Επώνυμο ιδιοκτήτη: Λορέτο

Διεύθυνση: Ρικολέτο 43, Ρώμη

ΑΦΜ:333444555

Τηλέφωνο:062345678

Email: info@fiori.it

Tραπεζικός λογαριασμός: IT122233

Kωδικός Δικτύου Συνεργασίας: ww34

………………………….

ΑπόδέχεταιΣυνεργασία()

ΑναθέτειΑνθοδεσία()

ΑναθέτειΠαράδοση()

Αντόνιο (Ανθοδέτης)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Επωνυμία εταιρίας: Κρετσιόνε ΕΠΕ

Όνομα ιδιοκτήτη: Αντόνιο

Επώνυμο ιδιοκτήτη: Περότι

Διεύθυνση: Κολοσέο 10, Ρώμη

ΑΦΜ:121212121212

Τηλέφωνο:069876543

Email: info@creazzione.it

Ειδικότητα: Ανθοδέσμες

Ωριαία αμοιβή: 20

………………………….

ΕτοιμάζειΑνθοδέσμη()

Πέπε (Ταχυμεταφορέας)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Επωνυμία εταιρίας: Ράπιντο ΟΕ

Όνομα ιδιοκτήτη: Πέπε

Επώνυμο ιδιοκτήτη:

Διεύθυνση: Καρακόλε 67, Ρώμη

ΑΦΜ:454545454545

Τηλέφωνο:061276343

Email: info@rapido.it

Τύπος: Ταχυμεταφορές

………………………….

ΠαραδίδειΑνθοδέσμη()

Παραγγελία

Συνεργασία

Ανάθεση Ανθοδεσίας

Παράδοση

Ανάθεση Παράδοσης

1. Πως δομείται ένα αντικειμενοστραφές πρόγραμμα;

**(Συμπληρωματικό υλικό, ενότητα 4.2.2)** Ένα **αντικειμενοστραφές πρόγραμμα** δομείται ως **ένα δίκτυο συνεργαζόμενων οντοτήτων** που είναι τα αντικείμενα. Κάθε αντικείμενο έχει ένα συγκεκριμένο ρόλο στην εφαρμογή και παρέχει μια υπηρεσία ή εκτελεί μια ενέργεια (μέθοδο) που χρησιμοποιείται από άλλα μέλη του δικτύου, δηλαδή από άλλα αντικείμενα, για την υλοποίηση της συνεργασίας που θα επιλύσει το πρόβλημα

1. Τι ονομάζουμε ενθυλάκωση;

**(Συμπληρωματικό υλικό, ενότητα 4.3)** Σε μια αντικειμενοστραφή εφαρμογή κάθε αντικείμενο αποτελεί ξεχωριστή οντότητα και περιέχει ενσωματωμένες τις ιδιότητες (δεδομένα) και τους κανόνες συμπεριφοράς του (μεθόδους). Η δυνατότητα ενός αντικειμένου να συνδυάζει εσωτερικά τα δεδομένα και τις μεθόδους χειρισμού του καλείται **ενθυλάκωση** (encapsulation). Την ενθυλάκωση μπορούμε να την παρομοιάσουμε σαν ένα κέλυφος που υπάρχει γύρω από κάθε αντικείμενο και διαχωρίζει τον εσωτερικό από τον εξωτερικό του κόσμο.

1. Τι ονομάζουμε κλάση;

**(Συμπληρωματικό υλικό, ενότητα 4.3)** Ο γενικός τύπος ενός αντικειμένου καλείται **κλάση** (class) και καθορίζει τις αρχικές ιδιότητες και τη συμπεριφορά κάθε αντικειμένου που προέρχεται από αυτή. Μια κλάση αποτελεί ένα **αφαιρετικό** (abstract) στοιχείο (τύπο) και μπορεί να παράγει ένα απεριόριστο πλήθος δομικά ίδιων αντικειμένων.

1. Δώστε παραδείγματα κλάσεων.

|  |  |
| --- | --- |
| Σκύλος | |
| Όνοµα: Ράτσα: Μέγεθος: Ηλικία: Χρώµα: | |
| Τρώω() Κοιµάµαι() Κάθοµαι() Τρέχω() | |
|  |  |

**(Συμπληρωματικό υλικό, ενότητα 4.3)** Ένα παράδειγμα από τον πραγματικό κόσμο είναι τα αυτοκίνητα. Όλα τα αυτοκίνητα ενός συγκεκριμένου μοντέλου παράγονται με βάση το ίδιο σχέδιο που καθορίζει τις προδιαγραφές του οχήματος, π.χ. διαστάσεις αμαξώματος, διαστάσεις τροχών, κυβισμός, είδος κιβωτίου ταχυτήτων, είδος καυσίμου, χρώμα αμαξώματος, επένδυση καθισμάτων, κ.λπ. Με βάση αυτό το κοινό σχέδιο παράγονται από το εργοστάσιο πολλά διαφορετικά οχήματα του ίδιου μοντέλου. Κάθε όχημα διαφοροποιείται από τα υπόλοιπα στις τιμές κάποιων ιδιοτήτων. Ακόμα όμως και αν παραχθούν δύο οχήματα με τις ίδιες ακριβώς τιμές για τις ιδιότητές τους (κάτι που είναι συνηθισμένο), τα οχήματα συνεχίζουν να αποτελούν διαφορετικές οντότητες. Μπορούμε λοιπόν να θεωρήσουμε το σχέδιο του συγκεκριμένου μοντέλου αυτοκινήτου ως κλάση και τα οχήματα που κατασκευάζονται με βάση το σχέδιο ως αντικείμενα της κλάσης. Ένα δεύτερο παράδειγμα, το οποίο απεικονίζεται διαγραμματικά παρακάτω, είναι η κλάση «Σκύλος» και τα αντικείμενα της.



Όνοµα: Κάντι

Ράτσα: Τσιουάουα

Μέγεθος: Πολύ µικρό

Ηλικία: 1 έτους

Χρώµα: Καφέ

Όνοµα: Σούγκαρ Όνοµα: Μπρούνο Ράτσα: Γουέστ Χάιλαντ Τεριέ Ράτσα: Σέτλαντ

Μέγεθος: Μικρό Μέγεθος: Μεσαίο

Ηλικία: 6 µηνών Ηλικία: 4 ετών

Χρώµα: Άσπρο Χρώµα: Άσπρο-Καφέ

Όνοµα: Γκρέις

Ράτσα: Γκόλντεν Ριτρίβερ

Μέγεθος: Μεγάλο Ηλικία: 3 ετών

Χρώµα: Χρυσό

Όνοµα: Ρούντυ

Ράτσα: Γκρέιτ Ντέιν

Μέγεθος: Πολύ µεγάλο Ηλικία: 3 ετών

Χρώµα: Μαύρο-Άσπρο

1. Να δημιουργήσετε την κλάση «Ανθοπώλης» για το πρόβλημα αποστολή λουλουδιών του ερωτήματος 4.

**(Συμπληρωματικό υλικό, ενότητα 4.3)** Ο «Γιώργος», ο «Τζιοβάνι» και το νέο μέλος από το Παρίσι, ο Ζαν Κλωντ, είναι όλα αντικείμενα της κλάσης «Ανθοπώλης», άρα έχουν τις ίδιες ιδιότητες και μεθόδους. Αυτή είναι μία πλήρης απεικόνιση των αντικειμένων, καθώς στην διαγραμματική απεικόνιση που είχαμε κάνει στο ερώτημα 4, ο «Γιώργος» και ο «Τζιοβάνι» είχαν διαφορετικές μεθόδους. Όμως θα μπορούσε και κάποιος από τη Ρώμη να στείλει λουλούδια στην Αθήνα, οπότε οι ρόλοι θα αντιστρεφόταν.

|  |  |
| --- | --- |
| Ανθοπώλης | |
| Επωνυµία: Όνοµα Ιδιοκτήτη:  Επώνυµο Ιδιοκτήτη: Διεύθυνση:  ΑΦΜ:  Τηλέφωνο: Email:  Τραπεζικός Λογαριασµός: Κωδ Δικ Συν: | |
| ΔέχεταιΠαραγγελία() ΖητάΣυνεργασία() ΑποδέχεταιΣυνεργασία() ΑναθέτειΑνθοδεσία() ΑναθέτειΠαράδοση() | |
|  |  |



ΔέχεταιΠαραγγελία() ΖητάΣυνεργασία() ΑποδέχεταιΣυνεργασία() ΑναθέτειΑνθοδεσία() ΑναθέτειΠαράδοση()

Επωνυµία: Πτι Φλερ ΕΠΕ Όνοµα Ιδιοκτήτη: Ζαν Κλωντ Επώνυµο Ιδιοκτήτη: Ρισιέ Διεύθυνση: Ζερµέν 12, Παρίσι

ΑΦΜ: 111222333

Τηλέφωνο: 145007155 Email: [info@petit-fleur.fr](mailto:info@petit-fleur.fr)

Τραπεζικός Λογαριασµός: FR445566 Κωδ Δικ Συν: WorldWideFlower2

Ζαν Κλωντ (Ανθοπώλης)

ΔέχεταιΠαραγγελία() ΖητάΣυνεργασία() ΑποδέχεταιΣυνεργασία() ΑναθέτειΑνθοδεσία() ΑναθέτειΠαράδοση()

Επωνυµία: Φιόρι ΟΕ Όνοµα Ιδιοκτήτη: Τζιοβάνι

Επώνυµο Ιδιοκτήτη: Λορέτο Διεύθυνση: Ρικολέτο 43, Ρώµη ΑΦΜ: 333444555

Τηλέφωνο: 062345678 Email: [info@fiori.it](mailto:info@fiori.it)

Τραπεζικός Λογαριασµός: IT556677 Κωδ Δικ Συν: WorldWideFlower34

Τζιοβάνι (Ανθοπώλης)

ΔέχεταιΠαραγγελία() ΖητάΣυνεργασία() ΑποδέχεταιΣυνεργασία() ΑναθέτειΑνθοδεσία() ΑναθέτειΠαράδοση()

Επωνυµία: Φλόρα ΕΠΕ Όνοµα Ιδιοκτήτη: Γιώργος

Επώνυµο Ιδιοκτήτη: Παπαδόπουλος Διεύθυνση: Πηνειού 120, Αθήνα ΑΦΜ: 999888777

Τηλέφωνο: 2101234567 Email: [info@flora.gr](mailto:info@flora.gr)

Τραπεζικός Λογαριασµός: GR111222 Κωδ Δικ Συν: WorldWideFlower157

Γιώργος (Ανθοπώλης)

1. Να παρουσιάσετε μία διαγραμματική απεικόνιση των κλάσεων του προβλήματος αποστολής λουλουδιών του ερωτήματος 4.

**(Συμπληρωματικό υλικό, ενότητα, 4.3)** Αναλύοντας το πρόβλημα, προκύπτουν οι ακόλουθες κλάσεις:

* **Ανθοπώλης:** με τις ιδιότητες και μεθόδους που αναλύσαμε στο προηγούμενο ερώτημα. Θα αναπαραστήσουμε την συνεργασία μεταξύ δύο αντικειμένων της ίδιας κλάσης, με ένα βέλος (μέθοδος) που θα αρχίζει και θα επιστρέφει στην ίδια την κλάση.
* **Πελάτης:** με ιδιότητες όνομα, επώνυμο, διεύθυνση, τηλέφωνο, email και μεθόδους ΚάνειΠαραγελία() και ΠαραλαμβάνειΑνθοδέση().
* **Ταχυμεταφορέας:** Με ιδιότητες επωνυμία, όνομα ιδιοκτήτη, επώνυμο ιδιοκτήτη, διεύθυνση, ΑΦΜ, τηλέφωνο, email, τύπος και μεθόδους ΠαραδίδειΑνθοδέσμη().
* **Ανθοδέτης:** Με ιδιότητες επωνυμία, όνομα ιδιοκτήτη, επώνυμο ιδιοκτήτη, διεύθυνση, ΑΦΜ, τηλέφωνο, email, ειδικότητα, ωριαία αμοιβή και μεθόδους ΠαραδίδειΑνθοδέσμη().

Η διαγραμματική απεικόνιση κλάσεων είναι η ακόλουθη:

Μαμά (Πελάτης)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Όνομα: Μαρία

Επώνυμο: Πέτρου

Διεύθυνση: Λαυρίου 2, Αθήνα

Τηλέφωνο: 6972233444

Email: [maria@hotmail.com](mailto:maria@hotmail.com)

………………………….

ΚάνειΠαραγγελεία()

Άννα (Πελάτης)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Όνομα: Άννα

Επώνυμο: Γεωργίου

Διεύθυνση: Μοντάνα 4, Ρώμη

Τηλέφωνο: 4932323234

Email: [anna@hotmail.com](mailto:anna@hotmail.com)

………………………….

ΠαραλαμβάνειΑνθοδέσμη()

Γιώργος (Ανθοπώλης)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Επωνυμία εταιρίας: Φλόρα ΕΠΕ

Όνομα ιδιοκτήτη: Γιώργος

Επώνυμο ιδιοκτήτη: Παπαδόπουλος

Διεύθυνση: Πηνειού 120, Αθήνα

ΑΦΜ:999888777

Τηλέφωνο:2101234567

Email: info@fiora.gr

Tραπεζικός λογαριασμός: GR111222

Kωδικός Δικτύου Συνεργασίας: ww12

………………………….

ΑπόδέχεταιΠαραγγελεία()

ΖητάΣυνεργασία()

Τζιβάνι (Ανθοπώλης)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Επωνυμία εταιρίας: Φιόρι ΟΕ

Όνομα ιδιοκτήτη: Τζιοβάνι

Επώνυμο ιδιοκτήτη: Λορέτο

Διεύθυνση: Ρικολέτο 43, Ρώμη

ΑΦΜ:333444555

Τηλέφωνο:062345678

Email: info@fiori.it

Tραπεζικός λογαριασμός: IT122233

Kωδικός Δικτύου Συνεργασίας: ww34

………………………….

ΑπόδέχεταιΣυνεργασία()

ΑναθέτειΑνθοδεσία()

ΑναθέτειΠαράδοση()

Αντόνιο (Ανθοδέτης)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Επωνυμία εταιρίας: Κρετσιόνε ΕΠΕ

Όνομα ιδιοκτήτη: Αντόνιο

Επώνυμο ιδιοκτήτη: Περότι

Διεύθυνση: Κολοσέο 10, Ρώμη

ΑΦΜ:121212121212

Τηλέφωνο:069876543

Email: info@creazzione.it

Ειδικότητα: Ανθοδέσμες

Ωριαία αμοιβή: 20

………………………….

ΕτοιμάζειΑνθοδέσμη()

Πέπε (Ταχυμεταφορέας)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Επωνυμία εταιρίας: Ράπιντο ΟΕ

Όνομα ιδιοκτήτη: Πέπε

Επώνυμο ιδιοκτήτη:

Διεύθυνση: Καρακόλε 67, Ρώμη

ΑΦΜ:454545454545

Τηλέφωνο:061276343

Email: info@rapido.it

Τύπος: Ταχυμεταφορές

………………………….

ΠαραδίδειΑνθοδέσμη()

Παραγγελία

Συνεργασία

Ανάθεση Ανθοδεσίας

Παράδοση

Ανάθεση Παράδοσης

Πελάτης

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Όνομα:

Επώνυμο:

Διεύθυνση:

Τηλέφωνο:

Email:

………………………….

ΚάνειΠαραγγελεία()

ΠαραλαμβάνειΑνθοδέση()

Ανθοπώλης

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Επωνυμία εταιρίας:

Όνομα ιδιοκτήτη:

Επώνυμο ιδιοκτήτη:

Διεύθυνση:

ΑΦΜ:

Τηλέφωνο:

Email:

Tραπεζικός λογαριασμός:

Kωδικός Δικτύου Συνεργασίας:

………………………….

ΑπόδέχεταιΠαραγγελεία()

ΖητάΣυνεργασία()

ΑπόδέχεταιΣυνεργασία()

ΑναθέτειΑνθοδεσία()

ΑναθέτειΠαράδοση()

Ανθοδέτης

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Επωνυμία εταιρίας:

Όνομα ιδιοκτήτη:

Επώνυμο ιδιοκτήτη:

Διεύθυνση:

ΑΦΜ:

Τηλέφωνο:

Email:

Ειδικότητα:

Ωριαία αμοιβή:

………………………….

ΕτοιμάζειΑνθοδέσμη()

Ταχυμεταφορέας

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Επωνυμία εταιρίας:

Όνομα ιδιοκτήτη:

Επώνυμο ιδιοκτήτη:

Διεύθυνση:

ΑΦΜ:

Τηλέφωνο:

Email:

Τύπος:

………………………….

ΠαραδίδειΑνθοδέσμη()

Παραγγελία

Ανάθεση Ανθοδεσίας

Συνεργασία

Παράδοση

Ανάθεση Παράδοσης

1. Τι είναι η κληρονομικότητα;

**(Συμπληρωματικό υλικό, ενότητα 4.4)** Η δυνατότητα δημιουργίας ιεραρχιών αντικειμένων καλείται **κληρονομικότητα** (inheritance). Με βάση την κληρονομικότητα, μια κλάση μπορεί να περιγραφεί γενικά και στη συνέχεια μέσω αυτής της κλάσης να οριστούν υποκλάσεις αντικειμένων. Η κλάση απόγονος (υποκλάση) κληρονομεί και μπορεί να χρησιμοποιήσει όλα τα δεδομένα (ιδιότητες) και τις μεθόδους που περιέχει η κλάση πρόγονος (υπερκλάση).

1. Να δημιουργήσετε σχέσεις κληρονομικότητας στο παράδειγμα 4 με το πρόβλημα αποστολής λουλουδιών.

**(Συμπληρωματικό υλικό, ενότητα 4.4)**

* Μπορούμε να δημιουργήσουμε μία κλάση «Επαγγελματίας», η οποία διαθέτει τις κοινές ιδιότητες των κλάσεων «Ανθοπώλης», «Ανθοδέτης» και «Ταχυμεταφορέας».
* Σε μια σχέση κληρονομικότητας, η κλάση-πρόγονος περιλαμβάνει τις κοινές ιδιότητες και μεθόδους όλων των κλάσεων-απογόνων της, ενώ οι κλάσεις-απόγονοι εμφανίζουν μόνο τις διαφορετικές τους ιδιότητες και μεθόδους αφού τις κοινές τις κληρονομούν από τη «μητέρα» τους.
* Η διαγραμματική αναπαράσταση της σχέσης κληρονομικότητας που μόλις περιγράψαμε γίνεται με τη βοήθεια του ειδικού συμβόλου γενίκευσης .
* Οι ιδιότητες της κλάσης «Επαγγελματίας» δεν επαναλαμβάνονται στην κλάση «Ανθοπώλης», όμως το αντικείμενο «Γιώργος» θα έχει τις ιδιότητες και των δύο.

ΕτοιµάζειΑνθοδέσµη()

ΔέχεταιΠαραγγελία() ΖητάΣυνεργασία() ΑποδέχεταιΣυνεργασία() ΑναθέτειΑνθοδεσία() ΑναθέτειΠαράδοση()

ΠαραδίδειΑνθοδέσµη()

Τύπος:

Ειδικότητα: Ωριαία Αµοιβή:

Τραπεζικός Λογαριασµός: Κωδ Δικ Συν:

Ταχυµεταφορέας

Ανθοδέτης

Ανθοπώλης

Επωνυµία: Όνοµα Ιδιοκτήτη:

Επώνυµο Ιδιοκτήτη: Διεύθυνση:

ΑΦΜ:

Τηλέφωνο: Email:

Επαγγελµατίας

1. Να εξηγήσετε πότε μία κλάση Α μπορεί να είναι έγκυρη υποκλάση της Β.

**(Συμπληρωματικό υλικό, ενότητα 4.4)** Μία κλάση Α μπορεί αν είναι έγκυρη υποκλάση της Β όταν ισχύει ο κανόνας «το Α είναι ένα (is\_a) B». Για παράδειγμα:

* «Το αυτοκίνητο είναι μέσο μεταφοράς»: ισχύει ο κανόνας, «Αυτοκίνητο is\_a Μέσο μεταφοράς» οπότε η κλάση «Αυτοκίνητο» θα μπορούσε να είναι υποκλάση της κλάσης «Μέσο μεταφοράς».
* «Ο άνθρωπος είναι μέσο μεταφοράς»: δεν ισχύει ο κανόνας, «Άνθρωπος is\_a Μέσο μεταφοράς» οπότε η κλάση «Άνθρωπος» θα μπορούσε να είναι υποκλάση της κλάσης «Μέσο μεταφοράς».

1. Να οργανώσετε σε ιεραρχία κλάσεων με σχέση κληρονομικότητας τις κλάσεις: Σημείο, Πολύγωνο, Πίνακας, Κύκλος, Δισδιάστατο σχήμα, Τετράγωνο, Κύλινδρος, Μαθηματικό αντικείμενο, Σφαίρα, Τρίγωνο.

**(Συμπληρωματικό υλικό, ενότητα 4.4)**

* Θα εντοπίσουμε τα επίπεδα ιεραρχίας από το πιο γενικό στο πιο ειδικό και θα κατατάξουμε κάθε κλάση στο αντίστοιχο επίπεδο ορίζοντας τις κατάλληλες σχέσεις κληρονομικότητας.
* Η πιο γενική έννοια είναι το «Μαθηματικό αντικείμενο», άρα θα είναι η υπερκλάση ανωτέρου επιπέδου.
* Κατεβαίνοντας ένα επίπεδο, μαθηματικά αντικείμενα είναι ο «Πίνακας», το «Σημείο» και το «Δισδιάστατο σχήμα».
* Δισδιάστατα σχήματα είναι το «Πολύγωνο», ο «Κύκλος», το «Τρίγωνο» και το «Τετράγωνο».
  + Το «Τετράγωνο» και το «Τρίγωνο» είναι πολύγωνα.
* Παρατηρούμε πως ο «Κύλινδρος» και η «Σφαίρα» είναι γεωμετρικά σχήματα αλλά όχι δισδιάστατα. Μία λύση είναι να εισάγουμε δύο κλάσεις:
  + «Τρισδιάστατο σχήμα», που θα είναι υπερκλάση των κλάσεων «Κύλινδρος» και «Σφαίρα».
  + «Γεωμετρικό σχήμα», που θα είναι υπερκλάση των κλάσεων «Δισδιάστατο σχήμα » και «Τρισδιάστατο σχήμα».

Με βάση την παραπάνω ανάλυση καταλήγουμε στο ακόλουθο διάγραμμα:

Μαθηματικό Αντικείμενο

…………………………………….....

Γεωμετρικό Σχήμα

…………………………………….....

Σημείο

…………………………………….....

Πίνακας

…………………………………….....

Δισδιάστατο Σχήμα

…………………………………….....

Τρισδιάστατο Σχήμα

…………………………………….....

Έλλειψη

…………………………………….....

Πολύγωνο

…………………………………….....

Τρίγωνο

…………………………………….....

Τετράγωνο

…………………………………….....

1. Τι είναι ο πολυμορφισμος;

**(Συμπληρωματικό υλικό, ενότητα 4.5) Πολυμορφισμός** (polymorphism) είναι μια ιδιότητα του αντικειμενοστραφούς προγραμματισμού με την οποία μια λειτουργία μπορεί να υλοποιείται με πολλούς διαφορετικούς τρόπους.

* Ο πολυμορφισμός μας επιτρέπει να επαναπροσδιορίσουμε τον τρόπο με τον οποίο λειτουργούν κάποια πράγματα, είτε αλλάζοντας τον τρόπο λειτουργίας τους είτε αλλάζοντας τα εργαλεία τα οποία χρησιμοποιούνται για την επίτευξη του στόχου.
* Μια μορφή πολυμορφισμού έχουμε όταν, αντί για το αυτόματο σύστημα κεντρικού κλειδώματος, χρησιμοποιήσουμε τα κλειδιά για να κλειδώσουμε το αυτοκίνητό μας. Και στις δύο περιπτώσεις το αποτέλεσμα είναι το ίδιο. Με βάση όμως το εργαλείο που ενεργοποιούμε κάθε φορά αλλάζει και ο τρόπος υλοποίησης της λειτουργίας.

1. Παράδειγμα πολυμορφισμού: αριθμητικός τελεστής «+».

**(Συμπληρωματικό υλικό, ενότητα 4.5)** Σε κάποιες γλώσσες προγραμματισμού, ο τελεστής «+» δεν χρησιμοποιείται για αριθμητικά δεδομένα, αλλά διαφοροποιείται η λειτουργία του ες εξής:

* Αν δοθούν δύο αριθμοί, για παράδειγμα 60 και 20, τότε θα τους προσθέσει και θα βγάλει ως αποτέλεσμα το 60.
* Αν δοθούν δύο συμβολοσειρές, για παράδειγμα «Γεια» και «σας», ο τελεστής «+» θα προσαρμοστεί στα νέα δεδομένα και θα εκτελέσει μία πράξη που έχει νόημα στο χώρο των συμβολοσειρών, αυτή της «συνένωσης αλφαριθμητικών» και θα δώσει ως αποτέλεσμα την συμβολοσειρά «Γεια σας».
* Αν δοθούν η συμβολοσειρά «Καλώς ήρθες» και ο αριθμός 2020, ο τελεστής «+» θα αλλάξει ξανά συμβολοσειρά, θα μετατρέψει τον αριθμό 2020 στην συμβολοσειρά «2020», θα συνενώσει τις δύο συμβολοσειρές και θα δώσει ως αποτέλεσμα την συμβολοσειρά «Καλώς ήρθες 2020».
* Αν υποστηριζόταν από την γλώσσα προγραμματισμού, θα μπορούσαμε να δημιουργήσουμε μία συνάρτηση με το όνομα «Πρόσθεση» και συμπεριφορά ανάλογη με τα δεδομένα εισόδου:

|  |  |
| --- | --- |
| **Συνάρτηση Πρόσθεση (a, b) ΑΡΧΗ … ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ** | |
| **Είσοδος** | **Έξοδος** |
| Πρόσθεση(20,40) | 60 |
| Πρόσθεση(«Γεια », «σας!») | «Γεια σας!» |
| Πρόσθεση(«Καλώς ήρθες », 2020) | «Καλώς ήρθες 2020» |

* Τρεις είναι οι γενικές περιπτώσεις κατά τις οποίες αναγκάζεται η συνάρτηση «Πρόσθεση» να προσαρμοστεί και να αλλάξει συμπεριφορά σύμφωνα με τον τύπο των παραμέτρων της: **1)** αριθμοί **2)** συμβολοσειρές **3)** αριθμοί και συμβολοσειρές.

1. Παράδειγμα υλοποίησης μεθόδου.

**(Συμπληρωματικό υλικό, ενότητα 4.5)** Έστω πως έχουμε τρία γεωμετρικά σχήματα σε κλάσεις, «Τρίγωνο», «Παραλληλόγραμμο» και «Κύκλος». Και οι τρεις κλάσεις έχουν την ίδια μέθοδο «ΥπολογισμόςΕμβαδού()», ο υπολογισμός όμως γίνεται από διαφορετικό μαθηματικό τύπο, επομένως η λειτουργία είναι πολυμορφική.

|  |  |  |
| --- | --- | --- |
| **Τρίγωνο** | **Παραλληλόγραμμο** | **Κύκλος** |
| Εμ <- Βάση\*Ύψος/2 | Εμ <- Μήκος\*Πλάτος | Εμ <- 3.14\*Ακτίνα\*Ακτίνα |

Οι μέθοδοι είναι στην ουσία υποπρογράμματα, όπως αυτά που έχετε ήδη μάθετε. Μόνο που ενώ οι γνωστές σας διαδικασίες και συναρτήσεις εντάσσονταν απευθείας στο κύριο πρόγραμμα, η κάθε αντικειμενοστραφής μέθοδος εντάσσεται σε μια κλάση και «περιορίζεται» στα δεδομένα που αυτή περιέχει. Αν θεωρήσουμε λοιπόν ότι η γλώσσα προγραμματισμού υποστηρίζει το παραπάνω μοντέλο, ο κώδικας υλοποίησης της πολυμορφικής μεθόδου «ΥπολογισμόςΕμβαδού()» για κάθε ένα από τα τρία γεωμετρικά σχήματα του παραδείγματός μας παρουσιάζεται παρακάτω:

|  |  |  |
| --- | --- | --- |
| **Τρίγωνο** | **Παραλληλόγραμμο** | **Κύκλος** |
| **ΣΥΝΑΡΤΗΣΗ** | **ΣΥΝΑΡΤΗΣΗ** | **ΣΥΝΑΡΤΗΣΗ** |
| ΥπολογισμόςΕμβαδού(): | ΥπολογισμόςΕμβαδού(): | ΥπολογισμόςΕμβαδού(): |
| **ΠΡΑΓΜΑΤΙΚΗ** | **ΠΡΑΓΜΑΤΙΚΗ** | **ΠΡΑΓΜΑΤΙΚΗ** |
| **ΜΕΤΑΒΛΗΤΕΣ** | **ΜΕΤΑΒΛΗΤΕΣ** | **ΣΤΑΘΕΡΕΣ** |
| **ΠΡΑΓΜΑΤΙΚΕΣ:**Eμ | **ΠΡΑΓΜΑΤΙΚΕΣ:**Eμ | Π=3.14 |
| **ΑΡΧΗ** | **ΑΡΧΗ** | **ΜΕΤΑΒΛΗΤΕΣ** |
| Eμ <- Βάση\*Ύψος/2 | Eμ <- Μήκος\*Πλάτος | **ΠΡΑΓΜΑΤΙΚΕΣ**:Eμ |
| ΥπολογισμόςΕμβαδού <- Εμ | ΥπολογισμόςΕμβαδού<-Εμ | **ΑΡΧΗ** |
| **ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ** | **ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ** | Eμ <- Π\*Ακτίνα\*Ακτίνα |
|  |  | ΥπολογισμόςΕμβαδού <- Εμ |
|  |  | **ΤΕΛΟΣ\_ΣΥΝΑΡΤΗΣΗΣ** |

**ΑΣΚΗΣΕΙΣ ΓΙΑ ΛΥΣΗ**

**Άσκηση 1:** Ερωτήσεις Σωστό – Λάθος.

1. Σύμφωνα με την αντικειμενοστραφή θεωρία, η προσέγγιση κάθε προβλήματος πρέπει να γίνεται με φυσική ερμηνεία και να μη στηρίζεται σε πολύπλοκα τεχνικά ζητήματα.
2. Τα βασικά δομικά στοιχεία του αντικειμενοστραφούς προγραμματισμού είναι τα αντικείμενα, οι ιδιότητες, οι υπηρεσίες/ ενέργειες και οι συνεργασίες.
3. Η δυνατότητα ενός αντικειμένου να αποκρύπτει τα δεδομένα του από τις μεθόδους του, ονομάζεται ενθυλάκωση.
4. Μία κλάση μπορεί να δημιουργήσει μόνο ένα αντικείμενο.
5. Στην κληρονομικότητα, η κλάση – πρόγονος κληρονομεί όλα τα χαρακτηριστικά της κλάσης - απογόνου.
6. Σε μία σχέση κληρονομικότητας, η κλάση- πρόγονος περιλαμβάνει όλα τα κοινά χαρακτηριστικά των κλάσεων-απογόνων της.
7. Πολυμορφισμός είναι μία ιδιότητα κατά την οποία μία λειτουργία μπορεί να υλοποιηθεί με πολλούς τρόπους.
8. Οι μέθοδοι των κλάσεων υλοποιούνται ουσιαστικά με υποπρογράμματα, τα οποία εντάσσονται απευθείας στο κύριο πρόγραμμα.
9. Μία κλάση Α είναι έγκυρη υποκλάση της κλάσης Β, αν ισχύει ο κανόνας «ένα Α είναι ένα Β».
10. Ένα αντικείμενο αναπαρίσταται με παραλληλόγραμμο σε μία διαγραμματική αναπαράσταση.

**Άσκηση 2:**  Έχουμε το ακόλουθο σενάριο: Ο πελάτης Άρης Χατζάκης, με τηλέφωνο 6971212123 και διεύθυνση Βασιλειές 54, επιθυμεί να παραγγείλει μία πίτσα, οπότε τηλεφωνεί στην τοπική πιτσαρία «PIZZA\_ONE» και δίνει στον Λευτέρη Σταγάκη, τον ιδιοκτήτη, την παραγγελία. Τα στοιχεία της πιτσαρίας είναι: διεύθυνση Δημοκρατίας 30, τηλέφωνο 6948989893, ΑΦΜ 102013394. Μόλις η πίτσα είναι έτοιμη, ειδοποιεί τον ταχυμεταφορέα, Νίκο Μπίστη, με τηλέφωνο 6984343436, ΑΦΜ 987987654 και ωριαία αμοιβή 15 ευρώ, ώστε να παραδώσει την πίτσα στον πελάτη». **1)** να εντοπίσετε τα αντικείμενα, τις ιδιότητες τους, τις ενέργειες υπηρεσίες και τα είδη συνεργασίας **2)** να παρουσιάσατε την διαγραμματική αναπαράσταση των αντικειμένων.

**Άσκηση 3:** Δίνονται οι ακόλουθες πληροφορίες για 3 αντικείμενα - καθηγητές:

**Καθηγητής 1:** *Όνομα:* Κάλλια *Επώνυμο:* Καραταράκη *Ειδικότητα:* Φιλόλογος *Προυπηρεσία:* 15 έτη, *Μέθοδοι:* Διδάσκει(), Διορθώνει().

**Καθηγητής 2:** *Όνομα:* Γιάννης *Επώνυμο:* Σταθόπουλος *Ειδικότητα:* Μαθηματικός *Προυπηρεσία:* 20 έτη, *Μέθοδοι:* Διδάσκει(), ΕτοιμάζειΔιαγωνίσματα().

**Καθηγητής 2:** *Όνομα:* Πέτρος *Επώνυμο:* Μπαλαδήμας *Ειδικότητα:* Μαθηματικός *Προυπηρεσία:* 18 έτη, *Μέθοδοι:* Διδάσκει(), ΠροετοιμάζειΠαρουσίασεις().

Με βάση τις παραπάνω πληροφορίες να σχεδιάσετε την κλάση «Καθηγητής».

**Άσκηση 4:** «Ο Γιώργος Πέτρου με κωδικό 676 που διαμένει στην οδό Σμύρνης 8 στο Χαλάνδρι με τηλέφωνο 2191234567 είναι πελάτης του καταστήματος «ΒιντεοΔράση» στην οδό Κορίνθου 4 στο Χαλάνδρι και, αφού ζητήσει είσοδο στην εφαρμογή και ταυτοποιηθεί ως πελάτης, ενοικιάζει την ταινία αρ. 1543,«Ο Πόλεμος των Άστρων» που είναι αποθηκευμένη στο συγκεκριμένο κατάστημα ενοικίασης ταινιών». Με βάση το ακόλουθο σενάριο: **1)** να εντοπίσετε τα αντικείμενα, τις ιδιότητες τους, τις ενέργειες υπηρεσίες και τα είδη συνεργασίας **2)** να παρουσιάσατε την διαγραμματική αναπαράσταση των κλάσεων.

**Άσκηση 5:** Δίνονται οι ακόλουθες πληροφορίες για τις κλάσεις «Τρίγωνο», «Παραλληλόγραμμο» και «Κύκλος»:

**Τρίγωνο: *Ιδιότητες:***Ύψος, Βάση, Σημείο Χ, Σημείο Υ, Χρώμα. ***Μέθοδοι:*** ΥπολογισμόςΕμβαδού(), ΑλλαγήΧρώματος().

**Παραλληλόγραμμο: *Ιδιότητες:***Ύψος, Πλάτος, Σημείο Χ, Σημείο Υ, Χρώμα. ***Μέθοδοι:*** ΥπολογισμόςΕμβαδού(), ΑλλαγήΧρώματος().

**Τρίγωνο: *Ιδιότητες:***Ακτίνα, Σημείο Χ, Σημείο Υ, Χρώμα. ***Μέθοδοι:*** ΥπολογισμόςΕμβαδού(), ΑλλαγήΧρώματος().

Λαμβάνοντας υπόψη πως ο υπολογισμός του εμβαδού κάθε σχήματος είναι διαφορετικός, να δημιουργήσετε σχέσεις κληρονομικότητας και να δημιουργήσετε την υπερκλάση «Γεωμετρικό Σχήμα».

**Άσκηση 6:** Να αναφέρετε ποια από τα παρακάτω ζευγάρια μπορούν να δημιουργήσουν σχέση «κλάσης - υποκλάσης». Ποιο κανόνα χρησιμοποιήσατε για να καταλήξετε σε συμπέρασμα;

**1)**Νόμισμα – ευρώ **2)** Σχήμα – Τρίγωνο **3)** Κύκλος – Τετράγωνο **4)** Άνδρας – Γυναίκα **5)** Άνθρωπος – Γυναίκα **6)** Σκύλος – Γάτα **7)** Ζώο – Ελέφαντας **8)** ΆΡΗΣ-ΠΑΟΚ **9)** Ομάδα – ΟΦΗ **10)** Πόλη - Ηράκλειο

**Άσκηση 7:** Να δημιουργήσετε ιεραρχίες κληρονομικότητας κλάσεων, που θα αναπαριστά τα κεφάλαια του μαθήματος της πληροφορικής: **1)** Δομή Απλής επιλογής **2)** Δομή ΟΣΟ **3)** Συνάρτηση **4)** Δυναμικές Δομές δεδομένων **5)** Λίστες **6)** Πίνακες **7)** Δομή Σύνθετης επιλογής **8)** Δομή ΜΕΧΡΙΣ\_ΟΤΟΥ **9)** Δένδρα **10)** Μονοδιάστατοι πίνακες **11)** Υποπρογράμματα **12)** Κεφάλαια πληροφορικής **13)** Διαδικασία **14)** Δομή Πολλαπλής επιλογής **15)** Δυαδικά Δένδρα Αναζήτησης **16)** Γράφοι **17)** Απλά συνδεδεμένες λίστες **18)** Δομή ΓΙΑ **19)** Διπλά Συνδεδεμένες λίστες **20)** Δισδιάστατοι πίνακες **21)** Δομές Επανάληψης.

Αν χρειαστείτε μπορείτε να δημιουργήσετε και επιπλέον δικές σας κλάσεις.