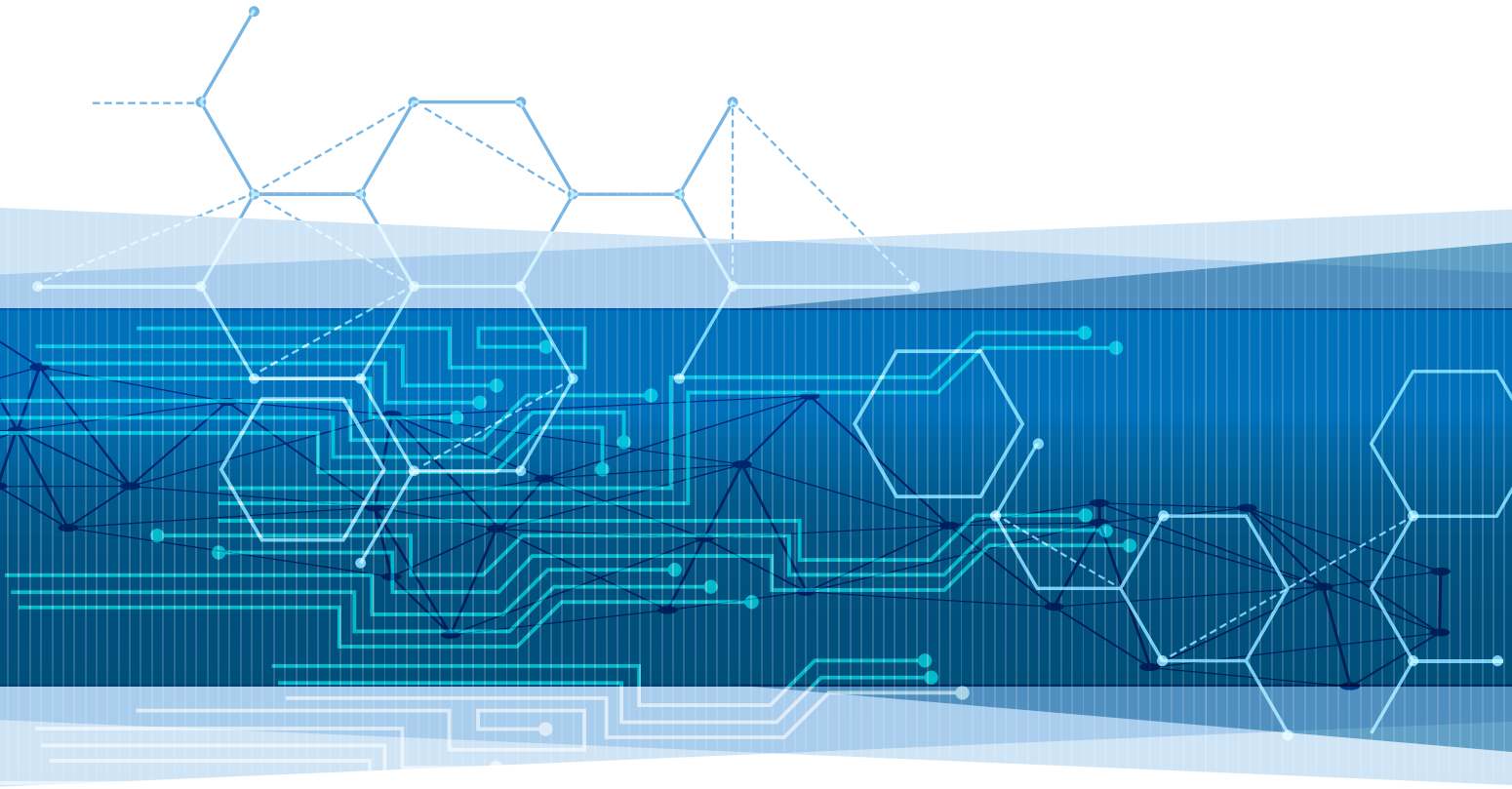
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΕΡΕΥΝΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

ΠΛΗΡΟΦΟΡΙΚΗ



ΒΙΒΛΙΟ ΜΑΘΗΤΗ
ΣΥΜΠΛΗΡΩΜΑΤΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΥΛΙΚΟ

Γ΄ ΓΕΝΙΚΟΥ ΛΥΚΕΙΟΥ

Ομάδας Προσανατολισμού Σπουδών Οικονομίας & Πληροφορικής

ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΕΚΔΟΣΕΩΝ
«ΔΙΟΦΑΝΤΟΣ»

ΠΛΗΡΟΦΟΡΙΚΗ

ΒΙΒΛΙΟ ΜΑΘΗΤΗ

ΣΥΜΠΛΗΡΩΜΑΤΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΥΛΙΚΟ

ΕΙΔΙΚΟΙ ΕΠΙΣΤΗΜΟΝΕΣ

ΣΥΝΤΟΝΙΣΜΟΣ
Γραμμένος Νικόλαος, Σύμβουλος Β' ΙΕΠ

ΕΠΙΣΤΗΜΟΝΙΚΗ ΕΠΟΠΤΕΙΑ
Βεντούρας Ερρίκος-Χαΐμ, Καθηγητής Πανεπιστημίου Δυτικής Αττικής
Σγουροπούλου Κλειώ, Καθηγήτρια Πανεπιστημίου Δυτικής Αττικής

ΕΚΠΟΝΗΣΗ-ΑΝΑΜΟΡΦΩΣΗ-ΕΠΙΜΕΛΕΙΑ ΠΕΡΙΕΧΟΜΕΝΟΥ
Μέλη της Μονάδας Φυσικών Επιστημών, Τεχνολογίας & Μαθηματικών του Ι.Ε.Π.
Γραμμένος Νικόλαος, Σύμβουλος Β΄ ΙΕΠ
Γούσιου Ανθή, Σύμβουλος Β΄ ΙΕΠ

ΕΚΠΟΝΗΣΗ
Εξωτερικοί Εμπειρογνώμονες
Αλεξούδα Γεωργία, Συντονίστρια Εκπαιδευτικού Έργου ΠΕ86
Κοτίνη Ισαβέλλα, Εκπαιδευτικός Δημοσίου Τομέα ΠΕ86
Κωτσάκης Σταύρος, Συντονιστής Εκπαιδευτικού Έργου ΠΕ86
Μωράκης Διονύσιος, Εκπαιδευτικός Δημοσίου Τομέα ΠΕ86
Νείρος Αντώνιος, Εκπαιδευτικός Δημοσίου Τομέα ΠΕ86
Ταταράκη Αλεξάνδρα, Εκπαιδευτικός Δημοσίου Τομέα ΠΕ86
Τζελέπη Σοφία, Συντονίστρια Εκπαιδευτικού Έργου ΠΕ86

ΠΡΟΕΚΤΥΠΩΤΙΚΕΣ ΕΡΓΑΣΙΕΣ
ΙΤΥΕ "ΔΙΟΦΑΝΤΟΣ"

ΔΡΑΣΗ ΓΙΑ ΤΗΝ ΑΝΑΜΟΡΦΩΣΗ Η/ΚΑΙ ΕΚΠΟΝΗΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ ΣΠΟΥΔΩΝ ΚΑΙ ΤΥΧΟΝ ΣΥΜΠΛΗΡΩΜΑΤΙΚΟΥ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΥΛΙΚΟΥ ΣΤΑ ΘΕΜΑΤΙΚΑ ΠΕΔΙΑ ΤΩΝ ΑΝΘΡΩΠΙΣΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ, ΚΟΙΝΩΝΙΚΩΝ ΕΠΙΣΤΗΜΩΝ, ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΜΑΘΗΜΑΤΙΚΩΝ ΠΡΩΤΟΒΑΘΜΙΑΣ ΚΑΙ ΔΕΥΤΕΡΟΒΑΘΜΙΑΣ ΕΚΠΑΙΔΕΥΣΗΣ (ΠΡΑΞΗ 43/19-10-2017 ΤΟΥ ΔΣ ΤΟΥ ΙΕΠ ΚΑΙ ΣΕ ΣΥΝΕΧΕΙΑ ΤΗΝ ΜΕ ΑΡ. ΠΡΩΤ. 7143/23-10-2017 ΚΑΙ ΑΔΑ: 7ΣΝΓΟΞΛΔ-ΞΟ5 ΠΡΟΣΚΛΗΣΗ ΕΚΔΗΛΩΣΗΣ ΕΝΔΙΑΦΕΡΟΝΤΟΣ)

ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ
Γεράσιμος Κουζέλης
Πρόεδρος του Ινστιτούτου Εκπαιδευτικής Πολιτικής

Υπεύθυνη Δράσης
Γεωργία Φέρμελη
Σύμβουλος Α΄ του Ινστιτούτου Εκπαιδευτικής Πολιτικής

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΕΡΕΥΝΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

Το παρόν εκπονήθηκε αμισθί, με ευθύνη της Μονάδας Φυσικών Επιστημών, Τεχνολογίας και Μαθηματικών του Ινστιτούτου Εκπαιδευτικής Πολιτικής, στο πλαίσιο της ανωτέρω Δράσης.

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΕΡΕΥΝΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

ΠΛΗΡΟΦΟΡΙΚΗ

ΒΙΒΛΙΟ ΜΑΘΗΤΗ

ΣΥΜΠΛΗΡΩΜΑΤΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΥΛΙΚΟ

Γ΄ ΤΑΞΗ ΓΕΝΙΚΟΥ ΛΥΚΕΙΟΥ

ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΕΚΔΟΣΕΩΝ
«ΔΙΟΦΑΝΤΟΣ»

ΠΕΡΙΕΧΟΜΕΝΑ

Πρόλογος – Εισαγωγικό Σημείωμα

Η δυναμικά αναπτυσσόμενη επιστήμη των Υπολογιστών και της Πληροφορικής υποδεικνύει την περιοδική αναπροσαρμογή και τον εμπλουτισμό του εκπαιδευτικού υλικού, ώστε οι μαθητές και οι μαθήτριες, οι ψηφιακά ενεργοί πολίτες του 21ου αιώνα, να συμβαδίζουν γνωστικά με τις σύγχρονες κάθε φορά απαιτήσεις της ραγδαία εξελισσόμενης τεχνολογίας.

Το παρόν εγχειρίδιο στοχεύει στην ανάπτυξη της αναλυτικής, της συνδυαστικής, της συνθετικής, της αλγοριθμικής και εν γένει της υπολογιστικής σκέψης των μαθητών και των μαθητριών. Αποτελεί συμπληρωματικό/εμπλουτισμένο εκπαιδευτικό υλικό που συνοδεύει αναπόσπαστα και επεκτείνει το βιβλίο μαθητή «Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον» (Α.Ε.Π.Π.) της Γ' Τάξης ΓΕ.Λ. και είναι διαρθρωμένο σε πέντε (5) διακριτές ενότητες.

Συγκεκριμένα:

Στην ενότητα **«Δομές Δεδομένων και Αλγόριθμοι»** περιγράφεται η δομή και η λειτουργία της **στοίβας** και της **ουράς**. Αναλύονται οι βασικές λειτουργίες αυτών των δύο δομών με κατά τον δυνατό διττό τρόπο: (α) σε αφαιρετικό επίπεδο, και (β) σε επίπεδο υλοποίησης σε προγραμματιστικό περιβάλλον με χρήση μονοδιάστατου πίνακα. Στην ίδια ενότητα περιγράφεται η δομή και η λειτουργία μιας σειράς δυναμικών δομών δεδομένων, όπως είναι **οι λίστες, τα δένδρα και οι γράφοι**, η χρησιμότητα και οι εφαρμογές των οποίων αναδεικνύονται μέσω απλών παραδειγμάτων που υλοποιούνται με διαγραμματικό τρόπο. Η προσέγγιση των δυναμικών αυτών δομών πραγματοποιείται σε αφαιρετικό επίπεδο, δίχως να θίγονται ζητήματα υλοποίησης. Σε αυτό το πλαίσιο εξετάζονται οι διαφορές μεταξύ των διαφόρων δομών δεδομένων και τα πλεονεκτήματά τους στην οργάνωση, αποθήκευση και διαχείριση δεδομένων.

Στην ενότητα **«Τεχνικές σχεδίασης αλγορίθμων»** παρουσιάζεται η μέθοδος **«Διαίρει και βασίλευε»** με τη χρήση αυθεντικών για τους μαθητές και τις μαθήτριες παραδειγμάτων. Ο αλγόριθμος της δυαδικής αναζήτησης υλοποιείται με τη χρήση επαναληπτικής διαδικασίας και παρουσιάζεται ως ένας κλασικός αλγόριθμος που ακολουθεί τη συγκεκριμένη μέθοδο. Δίνεται έμφαση σε παραδείγματα τα οποία βασίζονται σε παραλλαγές της δυαδικής αναζήτησης και ταυτόχρονα συνοδεύονται από την ανάλυση της μεθοδολογίας που ακολουθείται.

Ακολουθεί η ενότητα **«Επιλογή και επανάληψη»** στην οποία παρουσιάζεται η σύνταξη και η χρήση της **εντολής ΕΠΙΛΕΞΕ**. Δίνεται έμφαση στη συμπαγή δομή της και γίνεται σύνδεσή της με την εντολή ΑΝ ... ΑΛΛΙΩΣ_ΑΝ...

Η ενότητα **«Σύγχρονα Προγραμματιστικά Περιβάλλοντα»** παρουσιάζει τον **αντικειμενοστραφή προγραμματισμό** ως ένα σύγχρονο προγραμματιστικό παράδειγμα αποτελεσματικής επίλυσης σύνθετων προβλημάτων βασιζόμενο στη σύνθεση των ικανοτήτων που διαθέτουν διαφορετικές ανεξάρτητες οντότητες και στη μεταξύ τους συνεργασία. Η ενότητα αυτή αποτελεί μια εισαγωγή στις βασικές αρχές και τα χαρακτηριστικά της αντικειμενοστραφούς προσέγγισης (αφαίρεση, ενθυλάκωση, κληρονομικότητα, πολυμορφισμός) με στόχο οι μαθητές και οι μαθήτριες να αποκτήσουν βασικές δεξιότητες ανάλυσης αντικειμενοστραφών εφαρμογών αξιοποιώντας απλές διαγραμματικές τεχνικές αποτύπωσης των συστατικών τους στοιχείων.

Τέλος, στην ενότητα **«Εκσφαλμάτωση προγράμματος»** παρουσιάζονται οι κατηγορίες λαθών και αναδεικνύονται καλές πρακτικές εκσφαλμάτωσης. Περιλαμβάνονται δραστηριότητες για τις κατηγορίες λαθών και δίνεται ιδιαίτερη έμφαση στα λογικά λάθη που οδηγούν σε λανθασμένα αποτελέσματα.

Μέσω κατάλληλων παραδειγμάτων παρουσιάζονται συνηθισμένα και χαρακτηριστικά λάθη. Ιδιαίτερα για τα λογικά λάθη υπάρχουν ξεχωριστές υποενότητες για τις δομές επιλογής, τις δομές επανάληψης, τους πίνακες και τα υποπρογράμματα. Επίσης, παρουσιάζεται η μέθοδος ελέγχου προγραμμάτων του «Μαύρου κουτιού» για τον συστηματικό και μεθοδικό εντοπισμό των λαθών ενός προγράμματος.

Εκτός από διαγράμματα, πίνακες, εικόνες και πλαίσια, στο συγκεκριμένο εγχειρίδιο έχουν χρησιμοποιηθεί εικονίδια, τα οποία οπτικά χαρακτηρίζουν τα μέρη των κειμένων που συνοδεύουν, ως ακολούθως:

Εικονίδιο	Περιγραφή	Εικονίδιο	Περιγραφή
	Δραστηριότητα		Χρήσιμη Πληροφορία
	Ορισμός Θεωρίας		Κώδικας σε ΓΛΩΣΣΑ

Παραμένουμε στη διάθεση της εκπαιδευτικής κοινότητας για οποιεσδήποτε παρατηρήσεις, εποικοδομητικά σχόλια, αλλαγές και διορθώσεις με στόχο τη βελτίωση του παρόντος εγχειριδίου, ώστε να γίνει ένας εύχρηστος και αποτελεσματικός εμπλουτισμένος οδηγός για τη διδασκαλία του μαθήματος.

Σεπτέμβριος 2019

Η συγγραφική ομάδα

Πίνακας Αντιστοίχισης Ενοτήτων Εκπαιδευτικού Υλικού Μαθήματος

Βιβλίο Μαθητή «Πληροφορική» Συμπληρωματικό Εκπαιδευτικό Υλικό			Βιβλίο Μαθητή «Α.Ε.Π.Π.»
Δομές Δεδομένων και Αλγόριθμοι	1.1	Στοίβα	3.4
	1.2	Ουρά	3.5
	1.3	Άλλες Δομές Δεδομένων	3.9
	1.3.1	Λίστες	3.9.1
	1.3.2	Δένδρα	3.9.2
	1.3.3	Γράφοι	3.9.3
Τεχνικές Σχεδίασης Αλγορίθμων	2.1	Μέθοδος «Διαίρει και Βασίλευε»	4.3
Επιλογή και Επανάληψη	3.1	Εντολή ΕΠΙΛΕΞΕ	8.1.2
Σύγχρονα Προγραμματιστικά Περιβάλλοντα	4.1	Αντικειμενοστραφής Προγραμματισμός: ένας φυσικός τρόπος επίλυσης προβλημάτων	11.1, 11.1.1, 11.1.2, 11.1.3, 11.1.4
	4.2	Χτίζοντας Αντικειμενοστραφή Προγράμματα	
	4.3	Ομαδοποίηση Αντικειμένων σε Κλάσεις: Αφαιρετικότητα και Ενθυλάκωση	
	4.4	Η Αντικειμενοστραφής «Οικογένεια»: Κλάσεις - Πρόγονοι, Κλάσεις - Απόγονοι	
	4.5	Ορίζοντας την Κατάλληλη Συμπεριφορά: Πολυμορφισμός	
Εκσφαλμάτωση Προγράμματος	5.1	Κατηγορίες Λαθών	13.1
	5.2	Εκσφαλμάτωση	13.2

Ενότητα 1

Δομες Δεδομενων και Αλγοριθμοι

Στοίβα

Ουρά

Άλλες Δομές Δεδομένων
Λίστες
Γράφοι
Δένδρα

Ενότητα 1. Δομές Δεδομένων και Αλγόριθμοι

1.1 Στοίβα

Δύο τριτοετείς φοιτητές αποφάσισαν τον Δεκαπενταύγουστο να πάρουν το αυτοκίνητό τους και να περάσουν ένα τριήμερο στην Αίγινα. Θεωρούσαν ότι το οχηματαγωγό πλοίο χωρητικότητας 30 αυτοκινήτων, που εκτελεί το δρομολόγιο ΠΕΙΡΑΙΑΣ – ΑΙΓΙΝΑ ήταν αρκετά μεγάλο και δεν ήταν απαραίτητη η εκ των προτέρων κράτηση θέσης αυτοκινήτου. Όταν όμως έφθασαν στο λιμάνι, πληροφορήθηκαν, ότι το πλοίο ήταν πλήρες από οχήματα και μόνο σε περίπτωση ακύρωσης θα ελευθερώνονταν θέση για το αυτοκίνητό τους. Αυτό όμως θα το μάθαιναν λίγο πριν τον απόπλου. Τελικά, για καλή τους τύχη, έγινε μία ακύρωση και επιβιβάστηκαν τελευταίοι στο πλοίο μαζί με το αυτοκίνητό τους. Τότε, συνειδητοποίησαν ότι, το πλοίο είχε μία πόρτα, την ίδια για είσοδο και έξοδο των οχημάτων και θα αποβιβάζονταν πρώτοι αποφεύγοντας την ταλαιπωρία της αναμονής.
Παρατήρησαν, μάλιστα, ότι τα οχήματα που μπήκαν πρώτα ήταν σταθμευμένα στο βάθος του πλοίου **σε μία σειρά** το ένα μετά το άλλο και τα τελευταία συμπλήρωναν **αυτή τη σειρά** μέχρι και την πόρτα και μέχρι το πλήθος των 30 αυτοκινήτων.
Δηλαδή, η κατασκευή του «γκαράζ» του πλοίου και ο τρόπος στάθμευσης των οχημάτων προσομοίαζε στη δομή της «Στοίβας».

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.

Η παραπάνω μέθοδος ονομάζεται **Τελευταίο Μέσα, Πρώτο Έξω ή LIFO** (=Last In First Out).
Μπορούμε να φανταστούμε την τοποθέτηση των στοιχείων μιας στοίβας σε κατακόρυφη σειρά. Χαρακτηριστικό παράδειγμα είναι μια στοίβα από πιάτα. Παίρνουμε προς χρήση το πιάτο που τοποθετήσαμε τελευταίο.

Οι **κύριες λειτουργίες** σε μια στοίβα είναι δύο:

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.

Υλοποίηση στοίβας με χρήση μονοδιάστατου πίνακα

- Χρησιμοποιούμε μια βοηθητική μεταβλητή (top), που δείχνει το στοιχείο που τοποθετήθηκε τελευταίο στην κορυφή της στοίβας.

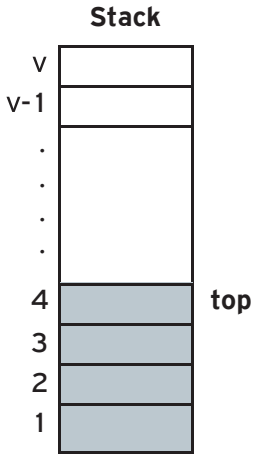
- Η **ώθηση** ενός νέου στοιχείου στη στοίβα (εισαγωγή στοιχείου στον πίνακα) γίνεται πάντα στην κορυφή της. Συγκεκριμένα, η μεταβλητή top αυξάνεται κατά ένα:

  top←top+1

  και στη συνέχεια γίνεται η ώθηση του στοιχείου.

- Η **απώθηση** ενός στοιχείου από τη στοίβα (εξαγωγή από τον πίνακα) γίνεται πάντα από την κορυφή της στοίβας. Συγκεκριμένα, εξάγεται το στοιχείο που δείχνει η μεταβλητή top και στη συνέχεια η μεταβλητή top μειώνεται κατά ένα:

  top←top-1



Εικόνα 1. 1. Υλοποίηση Στοίβας

Σε μια κενή στοίβα/πίνακα θεωρούμε ότι η αρχική τιμή της μεταβλητής top είναι μηδέν (top←0). Η μεταβλητή top είναι η μεταβλητή που δείχνει τη θέση που τοποθετήθηκε το τελευταίο στοιχείο στη στοίβα/πίνακα (δηλ. δείχνει την κορυφή της στοίβας).
Κατά την **ώθηση** ενός στοιχείου στη στοίβα (εισαγωγή ενός στοιχείου στον πίνακα), πρώτα αυξάνεται η τιμή της μεταβλητής top κατά ένα, δηλ. top←top+1, και στη συνέχεια γίνεται η ώθηση του στοιχείου στην κορυφή της στοίβας.
Κατά την **απώθηση** ενός στοιχείου από τη στοίβα (εξαγωγή στοιχείου από τον πίνακα) μειώνεται η τιμή της μεταβλητής top κατά ένα, δηλ. top← top-1. Στην απώθηση δε διαγράφεται το στοιχείο, στην πραγματικότητα δε γίνεται καμία παρέμβαση στα περιεχόμενα του πίνακα. Απλώς ο δείκτης κορυφή δείχνει στην προηγούμενη θέση.

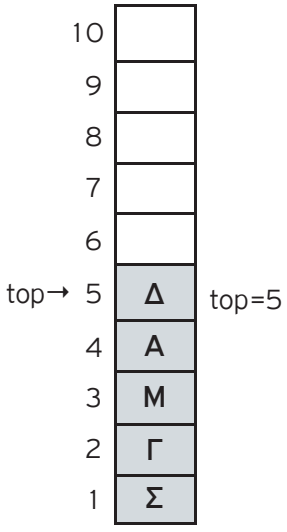
1.1.1 Παραδείγματα υλοποίησης στοίβας με χρήση μονοδιάστατου πίνακα

Παράδειγμα 1 – Ώθηση & Απώθηση στοιχείου σε στοίβα

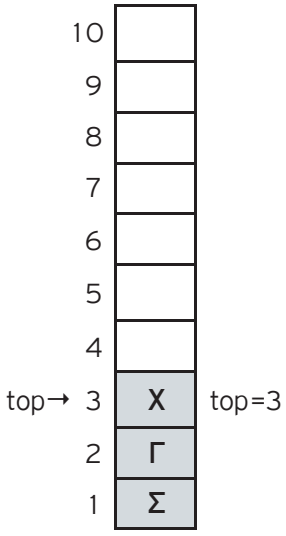
1) Σε μια στοίβα 10 θέσεων έχουν τοποθετηθεί διαδοχικά τα στοιχεία: Σ, Γ, Μ, Α, Δ στην 1η, 2η, 3η, 4η και 5η θέση αντίστοιχα.
   i. Να προσδιορίσετε την τιμή του δείκτη top και να σχεδιάσετε την παραπάνω στοίβα.
   ii. Αν εφαρμόσετε τις παρακάτω λειτουργίες: **Απώθηση, Απώθηση, Απώθηση, Ώθηση Χ, Ώθηση Δ** και **Απώθηση**, ποια είναι η νέα τιμή της top και ποια η τελική μορφή της στοίβας;
2) Σε μια άδεια στοίβα 10 θέσεων ωθούνται τα στοιχεία Ο, Σ, Λ, Τ, Ε. Με ποιον τρόπο πρέπει να γίνει η ώθηση και η απώθηση των στοιχείων, ώστε να έχουμε ως έξοδο τα στοιχεία Τ, Ε, Λ, Ο, Σ, με το στοιχείο Σ να βρίσκεται στην κορυφή της στοίβας;

Απάντηση

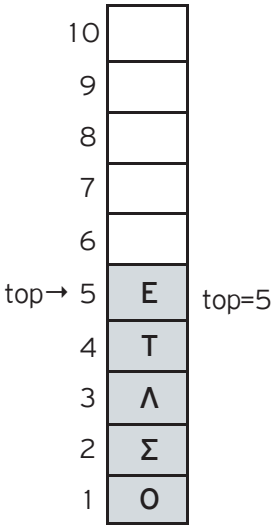
1) **i)**



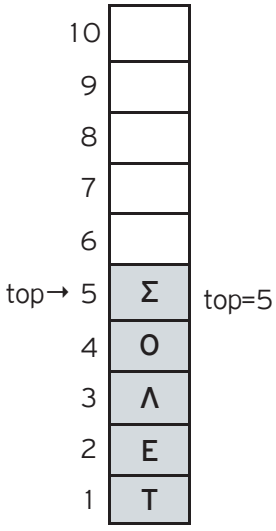
ii) Η νέα τιμή της top είναι 3 και η στοίβα γίνεται:



2) Η αρχική μορφή της στοίβας είναι:



Εκτελώντας τις λειτουργίες : **Απώθηση**, **Απώθηση, Απώθηση, Απώθηση, Απώθηση, Ώθηση** Τ, **Ώθηση** Ε**, Ώθηση** Λ**, Ώθηση** Ο**, Ώθηση** Σ, τότε η τελική μορφή της στοίβας γίνεται:



Παράδειγμα 2 – Ώθηση στοιχείου στην κορυφή της στοίβας με χρήση μονοδιάστατου πίνακα

Να αναπτύξετε τμήμα προγράμματος σε ΓΛΩΣΣΑ, που πραγματοποιεί ώθηση στοιχείου στην κορυφή της στοίβας με χρήση μονοδιάστατου πίνακα Α, 10 θέσεων.

Απάντηση
Θεωρούμε ότι η στοίβα δεν είναι κενή και η μεταβλητή top έχει μια έγκυρη τιμή.

Κώδικας σε ΓΛΩΣΣΑ [1.1]

```
1    ΓΡΑΨΕ 'Δώσε στοιχείο για να εισαχθεί στη στοίβα Α:'
2    ΔΙΑΒΑΣΕ στοιχείο
3    ΑΝ top<10 ΤΟΤΕ
4          top<- top+1
5          Α[top]<- στοιχείο
6    ΑΛΛΙΩΣ
7          ΓΡΑΨΕ 'Υπερχείλιση στοίβας'
8    ΤΕΛΟΣ_ΑΝ
```

Παράδειγμα 3 – Απώθηση στοιχείου από στοίβα με χρήση μονοδιάστατου πίνακα

Να αναπτύξετε τμήμα προγράμματος σε ΓΛΩΣΣΑ που πραγματοποιεί την απώθηση στοιχείου από στοίβα με χρήση ενός μονοδιάστατου πίνακα Α, 10 θέσεων

Απάντηση
Θεωρούμε ότι η στοίβα δεν είναι κενή και η μεταβλητή top έχει μια έγκυρη τιμή.

Κώδικας σε ΓΛΩΣΣΑ [1.2]

```
1    ΑΝ top>=1  ΤΟΤΕ
2          ΓΡΑΨΕ Α[top]
3          top<- top-1
4    ΑΛΛΙΩΣ
5          ΓΡΑΨΕ 'Υποχείλιση στοίβας'
6    ΤΕΛΟΣ_ΑΝ
```

Παράδειγμα 4 – Επιβίβαση & Αποβίβαση αυτοκινήτων σε πλοίο

Ένα οχηματαγωγό πλοίο, χωρητικότητας 250 αυτοκινήτων, τα οποία δύνανται να τοποθετηθούν αποκλειστικά σε μία σειρά, εκτελεί το δρομολόγιο ΠΕΙΡΑΙΑΣ – ΑΙΓΙΝΑ. Στο λιμάνι του Πειραιά προσέρχονται τα οχήματα για αναχώρηση. Τα οχήματα που επιβιβάζονται πρώτα είναι αυτά που θα αποβιβαστούν τελευταία.
Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

1. Να υλοποιεί μενού με τις επιλογές:

   1. Επιβίβαση **2.** Αποβίβαση **3.** Έξοδος
2. Στην περίπτωση που επιλεχθεί η **Επιβίβαση,** να ζητείται εισαγωγή του αριθμού κυκλοφορίας καθενός από τα οχήματα που προσέρχονται και ο αριθμός κυκλοφορίας του να καταχωρείται στη στοίβα ΟΧΗΜΑΤΑ. Κάθε φορά που επιβιβάζεται ένα όχημα να τυπώνεται το ερώτημα «**Υπάρχει όχημα για επιβίβαση; (Ν/Ο)**». Αν ο χρήστης απαντήσει Ν (=ΝΑΙ), τότε να επαναλαμβάνεται η διαδικασία επιβίβασης, ενώ αν απαντήσει Ο (=ΟΧΙ), τότε να σταματά η διαδικασία επιβίβασης και το πρόγραμμα να επιστρέφει στο μενού Επιλογής.
3. Στην περίπτωση που επιλεχθεί η **Αποβίβαση,** να τυπώνει τον αριθμό κυκλοφορίας όλων των οχημάτων με τη σειρά που αποβιβάζονται από το πλοίο στην ΑΙΓΙΝΑ.
4. Στο τέλος να τυπώνει το πλήθος των οχημάτων που επιβιβάστηκαν στο λιμάνι του ΠΕΙΡΑΙΑ.

Απάντηση

Παρακάτω υλοποιείται το πρόγραμμα του παραδείγματος επιβίβασης/αποβίβασης σε οχηματαγωγό πλοίο, όπου η «πόρτα» για την είσοδο και την έξοδο των οχημάτων στο πλοίο, είναι ίδια.

Ανάλυση της λύσης

Σύμφωνα με την εκφώνηση της άσκησης πρέπει:

- Να υλοποιηθεί μια δομή δεδομένων LIFO (στοίβα). Η υλοποίηση της στοίβας θα πραγματοποιηθεί με μονοδιάστατο πίνακα, όπου οι θέσεις του πίνακα θα είναι τόσες όσες και το μέγιστο πλήθος επιβιβαζόμενων οχημάτων (250). Επομένως για την υλοποίηση της στοίβας θα χρησιμοποιηθεί ένας μονοδιάστατος πίνακας **π** 250 θέσεων, π[250].
- Θα πρέπει να δηλωθεί η βοηθητική μεταβλητή (δείκτης) «**τοπ**», που δείχνει το στοιχείο που έχει τοποθετηθεί τελευταίο στη στοίβα. Η αρχική τιμή της μεταβλητής «**τοπ**» πρέπει να είναι Ο για να δείχνει στη «θέση» Ο του πίνακα, επισημαίνοντας κατά αυτόν τον τρόπο ότι το πλοίο/στοίβα είναι κενό. Η μεταβλητή «τοπ» αυξάνεται κάθε φορά κατά 1 (τοπ**<-**τοπ+1), και δείχνει την κενή θέση, την οποία καταλαμβάνει κάθε νέο επιβιβαζόμενο όχημα στο πλοίο.
- Να δηλωθούν οι μεταβλητές: **επ1** (επιλογή μενού), **πλ1** (μετρητής επιβιβαζόμενων οχημάτων, με αρχική τιμή Ο), **πλ2** (μετρητής αποβιβαζόμενων οχημάτων, με αρχική τιμή Ο), **επ2** (μεταβλητή δήλωσης οχήματος για επιβίβαση), **αρ** (αριθμός κυκλοφορίας οχήματος, μέσω του οποίου καταχωρίζεται-δηλώνεται η επιβίβασή του), π[250].

- Να εμφανίζεται ένα μενού με τις επιλογές: **1.** «Επιβίβαση», **2.** «Αποβίβαση», **3.** «Έξοδος» (χρήση δομής επανάληψης «**ΜΕΧΡΙΣ_ΟΤΟΥ…»,** μεταβλητή επιλογών «**επ1**»)
- Στην περίπτωση της **επιβίβασης** («**ΑΝ** επ1 **=** 1 **ΤΟΤΕ**»), θα πρέπει το πρόγραμμα να ελέγχει αν υπάρχει όχημα για επιβίβαση (μεταβλητή «**επ2**»). Αν υπάρχει όχημα για επιβίβαση, θα πρέπει παράλληλα να γίνεται έλεγχος αν υπάρχει κενή θέση στο πλοίο («τοπ **<** 250»). Αν υπάρχει κενή θέση, αυξάνεται κατά ένα η τιμή της μεταβλητής «τοπ» («τοπ**<-**τοπ**+**1») και στη συνέχεια γίνεται καταχώριση του αριθμού κυκλοφορίας του οχήματος σε αυτή την κενή θέση της στοίβας/πίνακα («π**[**τοπ**]<-**αρ»). Αμέσως μετά την τυχόν καταχώριση του αυτοκινήτου, η τιμή του μετρητή καταχώρισης επιβιβαζόμενων οχημάτων αυξάνεται κατά 1 (πλ1**<-**πλ1 **+** 1). Σε αντίθετη περίπτωση (γεμάτο πλοίο/υπερχείλιση στοίβας), να μη γίνεται καμιά καταχώριση και να εμφανίζεται το μήνυμα («**ΓΡΑΨΕ** 'Το πλοίο γέμισε και δε χωρά άλλα αυτοκίνητα'»).
- Στην περίπτωση της **αποβίβασης** («**ΑΛΛΙΩΣ_ΑΝ** επ1**=**2 **ΤΟΤΕ»)** θα πρέπει να ελέγχεται αν υπάρχει αυτοκίνητο για αποβίβαση, μέσω της μεταβλητής «**τοπ**» («**ΟΣΟ** τοπ**>=**1 **ΕΠΑΝΑΛΑΒΕ**»). Αν υπάρχει όχημα προς αποβίβαση, τυπώνεται ο αριθμός κυκλοφορίας του αποβιβαζόμενου οχήματος με κατάλληλο μήνυμα («**ΓΡΑΨΕ** 'Αποβιβάζεται το όχημα: ', π**[**τοπ**]**»). Κατόπιν μειώνεται η τιμή του δείκτη «**τοπ**» κατά 1 («τοπ**<-**τοπ**−**1»), για να δείχνει στην προηγούμενη θέση της στοίβας/πίνακα, θέση που περιέχει τον αριθμό κυκλοφορίας του επόμενου οχήματος προς αποβίβαση. Παράλληλα, αυξάνουμε την τιμή του μετρητή καταχώρισης αποβιβαζόμενων οχημάτων (πλ2 **<-** πλ2 **+** 1). Σε αντίθετη περίπτωση, αν ο δείκτης «τοπ» από την αρχή δείχνει 0 (υποχείλιση, δεν υπάρχουν αυτοκίνητα για αποβίβαση ή έχουν αποβιβασθεί όλα τα αυτοκίνητα), θα εμφανίζεται κατάλληλο μήνυμα με το σύνολο των αποβιβασθέντων οχημάτων («**ΓΡΑΨΕ** 'Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ: ', πλ2»).
  Να σημειωθεί ότι αν δεν αποβιβασθούν οχήματα ή για κάθε νέα επιλογή επιβίβασης ή αποβίβασης (νέα εκτέλεση του μενού), η μεταβλητή πλ2 έχει την αρχική της τιμή, που είναι 0.
- Τέλος, με την εκτύπωση του πλήθους επιβιβασθέντων οχημάτων στον Πειραιά («**ΓΡΑΨΕ** 'Οχήματα που επιβιβάστηκαν στον ΠΕΙΡΑΙΑ: ', πλ1»), γίνεται και τυπικός έλεγχος μεταξύ των οχημάτων που έχουν επιβιβαστεί στο πλοίο κι έχουν αποβιβαστεί από αυτό.

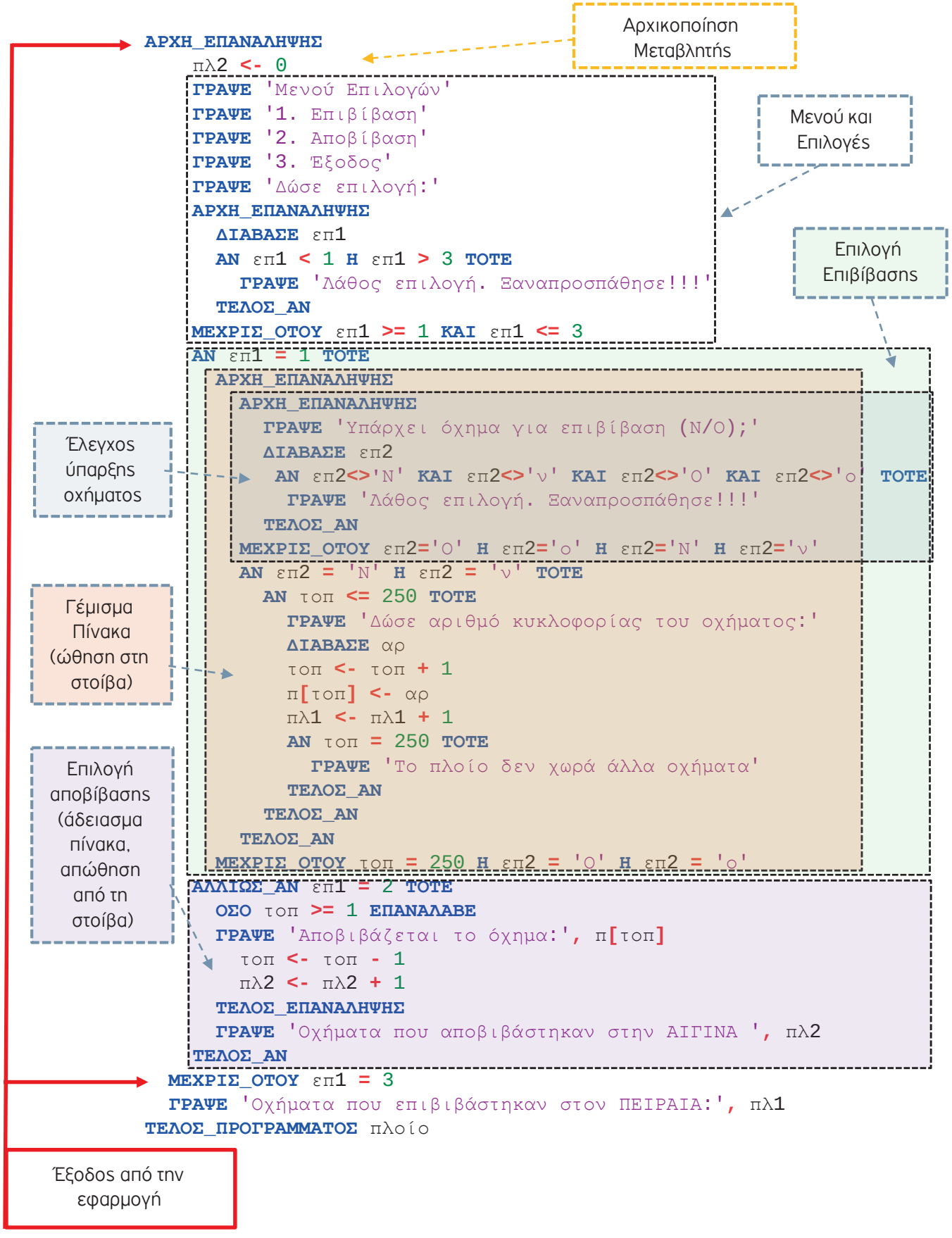
Ακολουθεί η υλοποίηση του προγράμματος σε ΓΛΩΣΣΑ.

Κώδικας σε ΓΛΩΣΣΑ [1.3]

```
ΠΡΟΓΡΑΜΜΑ πλοίο
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: τοπ, επ1, πλ1, πλ2
  ΧΑΡΑΚΤΗΡΕΣ: επ2, αρ, π[250]
ΑΡΧΗ
  τοπ <- 0
  πλ1 <- 0
```

Δήλωση Μεταβλητών

Αρχικοποίηση Μεταβλητών


Αρχικοποίηση Μεταβλητής
Μενού και Επιλογές
Επιλογή Επιβίβασης
Έλεγχος ύπαρξης οχήματος
Γέμισμα Πίνακα (ώθηση στη στοίβα)
Επιλογή αποβίβασης (άδειασμα πίνακα, απώθηση από τη στοίβα)
Έξοδος από την εφαρμογή
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
πλ2 <- 0
ΓΡΑΨΕ 'Μενού Επιλογών'
ΓΡΑΨΕ '1. Επιβίβαση'
ΓΡΑΨΕ '2. Αποβίβαση'
ΓΡΑΨΕ '3. Έξοδος'
ΓΡΑΨΕ 'Δώσε επιλογή:'
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ επ1
ΑΝ επ1 < 1 Η επ1 > 3 ΤΟΤΕ
ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ επ1 >= 1 ΚΑΙ επ1 <= 3
ΑΝ επ1 = 1 ΤΟΤΕ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Υπάρχει όχημα για επιβίβαση (Ν/Ο);'
ΔΙΑΒΑΣΕ επ2
ΑΝ επ2<>'Ν' ΚΑΙ επ2<>'ν' ΚΑΙ επ2<>'Ο' ΚΑΙ επ2<>'ο' ΤΟΤΕ
ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ επ2='Ο' Η επ2='ο' Η επ2='Ν' Η επ2='ν'
ΑΝ επ2 = 'Ν' Η επ2 = 'ν' ΤΟΤΕ
ΑΝ τοπ <= 250 ΤΟΤΕ
ΓΡΑΨΕ 'Δώσε αριθμό κυκλοφορίας του οχήματος:'
ΔΙΑΒΑΣΕ αρ
τοπ <- τοπ + 1
π[τοπ] <- αρ
πλ1 <- πλ1 + 1
ΑΝ τοπ = 250 ΤΟΤΕ
ΓΡΑΨΕ 'Το πλοίο δεν χωρά άλλα οχήματα'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ τοπ = 250 Η επ2 = 'Ο' Η επ2 = 'ο'
ΑΛΛΙΩΣ_ΑΝ επ1 = 2 ΤΟΤΕ
ΟΣΟ τοπ >= 1 ΕΠΑΝΑΛΑΒΕ
ΓΡΑΨΕ 'Αποβιβάζεται το όχημα:', π[τοπ]
τοπ <- τοπ - 1
πλ2 <- πλ2 + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ ', πλ2
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ επ1 = 3
ΓΡΑΨΕ 'Οχήματα που επιβιβάστηκαν στον ΠΕΙΡΑΙΑ:', πλ1
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ πλοίο

```
1   ΠΡΟΓΡΑΜΜΑ πλοίο
2   ΜΕΤΑΒΛΗΤΕΣ
3     ΑΚΕΡΑΙΕΣ: τοπ, επ1, πλ1, πλ2
4     ΧΑΡΑΚΤΗΡΕΣ: επ2, αρ, π[250]
5   ΑΡΧΗ
6     τοπ <- 0
7     πλ1 <- 0
8     ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
9       πλ2 <- 0
10      ΓΡΑΨΕ 'Μενού Επιλογών'
11      ΓΡΑΨΕ '1. Επιβίβαση'
12      ΓΡΑΨΕ '2. Αποβίβαση'
13      ΓΡΑΨΕ '3. Έξοδος'
14      ΓΡΑΨΕ 'Δώσε επιλογή:'
15      ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
16        ΔΙΑΒΑΣΕ επ1
17        ΑΝ επ1 < 1 Η επ1 > 3 ΤΟΤΕ
18          ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'
19        ΤΕΛΟΣ_ΑΝ
20      ΜΕΧΡΙΣ_ΟΤΟΥ επ1 >= 1 ΚΑΙ επ1 <= 3
21      ΑΝ επ1 = 1 ΤΟΤΕ
22        ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
23          ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
24            ΓΡΑΨΕ 'Υπάρχει όχημα για επιβίβαση (Ν/Ο);'
25            ΔΙΑΒΑΣΕ επ2
26             ΑΝ επ2<>'Ν' ΚΑΙ επ2<>'ν' ΚΑΙ επ2<>'Ο' ΚΑΙ επ2<>'ο' ΤΟΤΕ
27              ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'
28            ΤΕΛΟΣ_ΑΝ
29          ΜΕΧΡΙΣ_ΟΤΟΥ επ2='Ο' Η επ2='ο' Η επ2='Ν' Η επ2='ν'
30          ΑΝ επ2 = 'Ν' Η επ2 = 'ν' ΤΟΤΕ
31            ΑΝ τοπ <= 250 ΤΟΤΕ
32              ΓΡΑΨΕ 'Δώσε αριθμό κυκλοφορίας του οχήματος:'
33              ΔΙΑΒΑΣΕ αρ
34              τοπ <- τοπ + 1
35              π[τοπ] <- αρ
36              πλ1 <- πλ1 + 1
37              ΑΝ τοπ = 250 ΤΟΤΕ
38                ΓΡΑΨΕ 'Το πλοίο δεν χωρά άλλα οχήματα'
39              ΤΕΛΟΣ_ΑΝ
40            ΤΕΛΟΣ_ΑΝ
```

```
41            ΤΕΛΟΣ_ΑΝ
42          ΜΕΧΡΙΣ_ΟΤΟΥ τοπ = 250 Η επ2 = 'Ο' Η επ2 = 'ο'
43        ΑΛΛΙΩΣ_ΑΝ επ1 = 2 ΤΟΤΕ
44          ΟΣΟ τοπ >= 1 ΕΠΑΝΑΛΑΒΕ
45          ΓΡΑΨΕ 'Αποβιβάζεται το όχημα:', π[τοπ]
46            τοπ <- τοπ - 1
47            πλ2 <- πλ2 + 1
48          ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
49          ΓΡΑΨΕ 'Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ ', πλ2
50        ΤΕΛΟΣ_ΑΝ
51      ΜΕΧΡΙΣ_ΟΤΟΥ επ1 = 3
52      ΓΡΑΨΕ 'Οχήματα που επιβιβάστηκαν στον ΠΕΙΡΑΙΑ:', πλ1
53    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ πλοίο
```

1.1.2 Ερωτήσεις - Ασκήσεις

Ε. 1: Δίνεται η επόμενη ακολουθία αριθμών: 4, 8, 2, 5, 9, 13.
α) Ποια λειτουργία θα χρησιμοποιηθεί για την τοποθέτηση των αριθμών σε στοίβα;
β) Σχεδιάστε τη στοίβα μετά την τοποθέτηση των αριθμών.
γ) Ποια λειτουργία θα χρησιμοποιηθεί για την έξοδο των αριθμών από τη στοίβα;
δ) Πόσες φορές θα πρέπει να εκτελεστεί η προηγούμενη λειτουργία στη στοίβα για να εξαχθεί ο αριθμός 5;

Ε. 2: Σε μια στοίβα έχουν τοποθετηθεί κατά σειρά οι αριθμοί : 24, 7, 11, 13, 65, 39, 5.
α) Να σχεδιάσετε την παραπάνω δομή.
β) Ποια θα είναι η τιμή του δείκτη της παραπάνω στοίβας;
γ) Αν θέλετε να τοποθετήσετε τον αριθμό 25 στην στοίβα, ποια λειτουργία θα χρησιμοποιήσετε;
δ) Ποια θα είναι η τιμή του δείκτη μετά την λειτουργία αυτή;
ε) Αν θέλετε να εξάγετε τον αριθμό 65 από τη στοίβα, ποια λειτουργία θα χρησιμοποιήσετε;
στ) Ποια θα είναι η τιμή του δείκτη μετά τη λειτουργία αυτή;

Ε. 3: Χαρακτηρίστε τις παρακάτω προτάσεις ως Σωστές ή Λάθος. Στην περίπτωση που πιστεύετε ότι είναι λανθασμένες δικαιολογήστε την επιλογή σας και σκεφτείτε ποια θα μπορούσε να είναι η αντίστοιχη σωστή πρόταση.

α/α	Προτάσεις	Σ	Λ
1	Για την υλοποίηση μιας στοίβας μπορεί να χρησιμοποιηθεί ένας πίνακας.	❒	❒
2	Στη στοίβα το στοιχείο που μπαίνει πρώτο βγαίνει πρώτο.	❒	❒
3	Στην υλοποίηση της στοίβας χρειάζονται δύο μεταβλητές-δείκτες για την υλοποίηση των δύο βασικών λειτουργιών που εκτελούνται σε αυτή.	❒	❒
4	Η λειτουργία της ώθησης μπορεί να εκτελεστεί και σε μια άδεια στοίβα.	❒	❒
5	Η λειτουργία της ώθησης μπορεί να εκτελεστεί και σε μια γεμάτη στοίβα.	❒	❒
6	Η ώθηση στοιχείου γίνεται στην κορυφή της στοίβας.	❒	❒
7	Στη δομή της στοίβας απαιτούνται δύο δείκτες, ο εμπρός και ο πίσω.	❒	❒
8	Υπερχείλιση έχουμε όταν εισάγουμε ένα στοιχείο σε μια ήδη γεμάτη στοίβα.	❒	❒
9	Η μέθοδος LIFO περιγράφει τη διαδικασία εκείνη κατά την οποία το στοιχείο που τοποθετείται τελευταίο εξάγεται πρώτο	❒	❒
10	Κάθε στοιχείο που εισάγεται πρώτο σε μια στοίβα είναι αυτό που εξάγεται πρώτο.	❒	❒

1.2 Ουρά

Κατά την ημέρα αναχώρησης τους από την Αίγινα, οι δύο συμφοιτητές αποφάσισαν σκόπιμα να καθυστερήσουν, προκειμένου να μπουν τελευταίοι (ώστε να βγουν πρώτοι και χωρίς ταλαιπωρία στον Πειραιά) και ταυτόχρονα, να απολαύσουν για περισσότερο χρόνο τις τελευταίες στιγμές των διακοπών τους στο νησί.
Δυστυχώς, όμως, ατύχησαν! Το πλοίο της επιστροφής τους ήταν διαφορετικό. Είχε μία είσοδο στο πίσω μέρος προς επιβίβαση των οχημάτων και μία έξοδο στο μπροστινό μέρος προς αποβίβαση. Έτσι, το αυτοκίνητο που έμπαινε πρώτο, προχωρούσε ευθεία προς την άλλη μεριά του πλοίου και έπαιρνε θέση προς αποβίβαση προς την πόρτα εξόδου. Κάθε αυτοκίνητο που έμπαινε στο πλοίο πάρκαρε υποχρεωτικά πίσω από τα άλλα, που είχαν μπει νωρίτερα, στην ίδια σειρά. Δηλαδή, το αυτοκίνητο που επιβιβαζόταν πρώτο στην Αίγινα θα αποβιβαζόταν πρώτο και στον Πειραιά. Με τον τρόπο αυτό, η επιβίβαση και η αποβίβαση των οχημάτων γινόταν πιο γρήγορα. Η κατασκευή του «γκαράζ» του πλοίου και ο τρόπος στάθμευσης των οχημάτων (τρόπος οργάνωσης των δεδομένων) προσομοίαζε στη δομή της «Ουράς». Δηλαδή, τα δεδομένα που μπαίνουν πρώτα σε μια ουρά είναι αυτά που βγαίνουν και πρώτα.

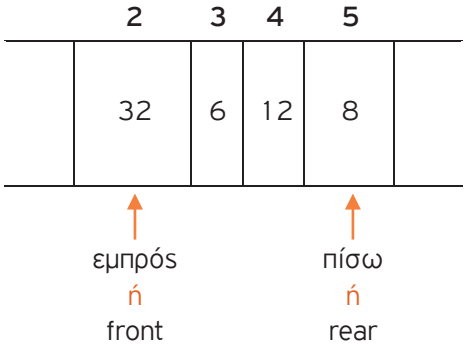
Ουρά (Queue), ονομάζεται μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που τοποθετήθηκαν πρώτα στην ουρά να λαμβάνονται επίσης πρώτα.

Η παραπάνω μέθοδος ονομάζεται **Πρώτο Μέσα, Πρώτο Έξω ή FIFO** (=First In First Out).
Μπορούμε να φανταστούμε την τοποθέτηση των στοιχείων μιας ουράς σε οριζόντια σειρά. Ο πελάτης μιας τράπεζας που μπαίνει πρώτος σε μια ουρά για να εξυπηρετηθεί, είναι αυτός που εξυπηρετείται και πρώτος, με την έξοδό του από την ουρά αναμονής.
Άλλα παραδείγματα είναι η ουρά στα λεωφορεία ή η ουρά στα ταμεία, όπου ο πρώτος που στέκεται στην ουρά εξυπηρετείται και πρώτος.

Οι **κύριες λειτουργίες** που εκτελούνται σε μια ουρά είναι δύο:
1. Η **εισαγωγή** (enqueue) στοιχείου στο πίσω άκρο της ουράς.
2. Η **εξαγωγή** (dequeue) στοιχείου από το εμπρός άκρο της ουράς.

Υλοποίηση ουράς με χρήση μονοδιάστατου πίνακα

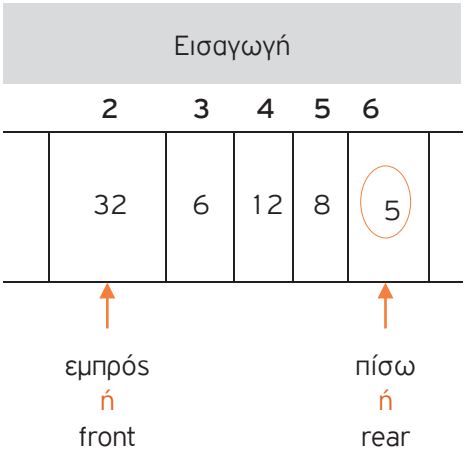
- Χρησιμοποιούμε δύο μεταβλητές, την front (ή εμπρός) που δείχνει τη θέση του 1ου στοιχείου της ουράς και την rear (ή πίσω) που δείχνει τη θέση του τελευταίου στοιχείου. Ως αρχικές τιμές των μεταβλητών rear και front θεωρούμε το μηδέν.



- Η **εισαγωγή** ενός νέου στοιχείου γίνεται από το πίσω άκρο της ουράς και η τιμή της μεταβλητής rear αλλάζει ως εξής:

  rear← rear+1

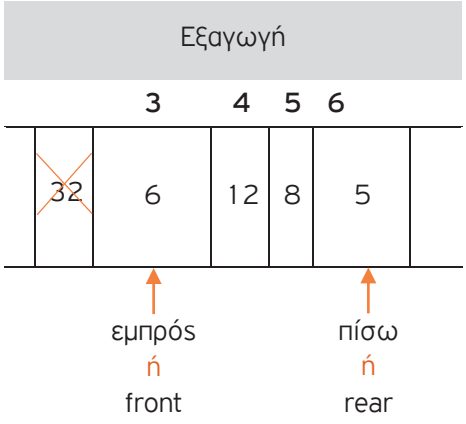
  Κατά την εισαγωγή, πρώτα αυξάνουμε τον δείκτη rear κατά ένα και μετά εισάγουμε το στοιχείο στον πίνακα.



- Η **εξαγωγή** ενός στοιχείου γίνεται από το εμπρός άκρο της ουράς και η τιμή της μεταβλητής front αλλάζει ως εξής:

  front ← front +1

  Κατά την εξαγωγή ενός στοιχείου, αυξάνεται ο δείκτης front κατά ένα (δείχνει στην επόμενη θέση του πίνακα) χωρίς στην πραγματικότητα να γίνεται καμία παρέμβαση στα περιεχόμενα του πίνακα (χωρίς να διαγράφεται κάποιο στοιχείο).



1.2.1 Παραδείγματα υλοποίησης ουράς με χρήση μονοδιάστατου πίνακα

Παράδειγμα 1 – Εισαγωγή και Εξαγωγή στοιχείων σε ουρά

1. Σε μια ουρά 10 θέσεων έχουν τοποθετηθεί διαδοχικά τα στοιχεία: Σ, Γ, Μ, Α, Δ στην 1η, 2η, 3η, 4η και 5η θέση αντίστοιχα.
   i. Να προσδιορίσετε τις τιμές των δεικτών rear και front και να σχεδιάσετε την παραπάνω ουρά.
   ii. Αν εφαρμόσετε τις παρακάτω λειτουργίες: **Εξαγωγή**, **Εξαγωγή**, **Εξαγωγή**, **Εισαγωγή Χ , Εισαγωγή Δ** και **Εξαγωγή,** ποιες είναι οι τιμές των δεικτών rear και front της ουράς και η τελική μορφή της;
2. Σε μια άδεια ουρά 10 θέσεων εισάγονται τα στοιχεία Ο, Σ, Λ, Τ, Ε . Με ποιον τρόπο πρέπει να εισαχθούν και να εξαχθούν τα στοιχεία, ώστε η έξοδος να εμφανίζει τα στοιχεία Τ, Ε, Λ, Ο, Σ;

Απάντηση

1)

i)

1η	2η	3η	4η	5η	6η	7η	8η	9η	10η
Σ	Γ	Μ	Α	Δ					
front				**rear**					

front=1 και rear=5

ii) Η τελική μορφή της ουράς είναι:

1η	2η	3η	4η	5η	6η	7η	8η	9η	10η
				Δ	Χ	Δ			
				front		**rear**			

και οι τιμές της front και της rear γίνονται: front=5 και rear=7

2)

i) Η αρχική μορφή της ουράς είναι:

1η	2η	3η	4η	5η	6η	7η	8η	9η	10η
Ο	Σ	Λ	Τ	Ε					
front				**rear**					

front=1 και rear=5

ii) Με την εκτέλεση των λειτουργιών: **Εξαγωγή**, **Εξαγωγή, Εξαγωγή, Εισαγωγή Λ**, **Εισαγωγή Ο, Εισαγωγή Σ**, η τελική μορφή της ουράς γίνεται:

1η	2η	3η	4η	5η	6η	7η	8η	9η	10η
			Τ	Ε	Λ	Ο	Σ		
			front				**rear**		

και οι τιμές της front και της rear γίνονται: front=4 και rear=8

Παράδειγμα 2 – Εισαγωγή στοιχείου σε ουρά με χρήση μονοδιάστατου πίνακα

Να αναπτύξετε τμήμα προγράμματος σε ΓΛΩΣΣΑ που υλοποιεί την εισαγωγή στοιχείου σε ουρά, με χρήση μονοδιάστατου πίνακα Α, 10 θέσεων.

Απάντηση

Κώδικας σε ΓΛΩΣΣΑ [1.4]

```
1    ΓΡΑΨΕ 'Δώσε στοιχείο για εισαγωγή στην ουρά Α:'
2    ΔΙΑΒΑΣΕ στοιχείο
3    ΑΝ rear = 10 ΤΟΤΕ
4       ΓΡΑΨΕ 'Γεμάτη ουρά'
5    ΑΛΛΙΩΣ_ΑΝ (front = 0 ΚΑΙ rear = 0) ΤΟΤΕ
6       front<- 1
7       rear<-1
8       A[rear]<-στοιχείο
9    ΑΛΛΙΩΣ
10         rear <- rear + 1
11         A[rear]<-στοιχείο
12   ΤΕΛΟΣ_ΑΝ
```

Παράδειγμα 3 – Εξαγωγή στοιχείου από ουρά με χρήση μονοδιάστατου πίνακα

Να αναπτύξετε τμήμα προγράμματος σε ΓΛΩΣΣΑ που υλοποιεί την εξαγωγή στοιχείου από ουρά, με χρήση μονοδιάστατου πίνακα Α, 10 θέσεων.

Απάντηση

Κώδικας σε ΓΛΩΣΣΑ [1.5]

```
1    ΑΝ (front = 0 ΚΑΙ rear = 0) ΤΟΤΕ
2        ΓΡΑΨΕ 'Άδεια ουρά'
3    ΑΛΛΙΩΣ ΑΝ (front = rear ) ΤΟΤΕ
4        ΓΡΑΨΕ 'Εξάγεται το στοιχείο:',A[front]
5        front <-  0
6        rear  <-  0
7    ΑΛΛΙΩΣ
8        ΓΡΑΨΕ 'Εξάγεται το στοιχείο:',A[front]
9        front <- front + 1
10   ΤΕΛΟΣ_ΑΝ
```

Παράδειγμα 4 – Επιβίβαση και Αποβίβαση αυτοκινήτων σε πλοίο

Ένα οχηματαγωγό πλοίο με δύο διαφορετικές πόρτες, μία για την είσοδο και μία για την έξοδο των οχημάτων, χωρητικότητας 250 αυτοκινήτων, τα οποία δύνανται να τοποθετηθούν αποκλειστικά σε μία σειρά, εκτελεί το δρομολόγιο ΠΕΙΡΑΙΑΣ – ΑΙΓΙΝΑ. Τα οχήματα που **επιβιβάζονται πρώτα είναι και αυτά που θα αποβιβαστούν πρώτα**. Στο λιμάνι του Πειραιά προσέρχονται τα αυτοκίνητα για αναχώρηση.

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

1. Να υλοποιεί μενού με τις επιλογές:

   1. Επιβίβαση **2.** Αποβίβαση **3.** Έξοδος

2. Στην περίπτωση που επιλεχθεί η **Επιβίβαση** το πρόγραμμα θα διαβάζει τον αριθμό κυκλοφορίας καθενός από τα οχήματα που επιβιβάζονται στο πλοίο και θα τον καταχωρίζει στην ουρά ΟΧΗΜΑΤΑ. Κάθε φορά που επιβιβάζεται ένα όχημα να τυπώνεται το ερώτημα «**Υπάρχει όχημα για επιβίβαση; (Ν/Ο)**». Αν ο χρήστης απαντήσει Ν (=ΝΑΙ), τότε να επαναλαμβάνεται η διαδικασία επιβίβασης, ενώ αν απαντήσει Ο (=ΟΧΙ), τότε να σταματά η διαδικασία επιβίβασης και να επιστρέφει το πρόγραμμα στο μενού Επιλογής.
3. Στην περίπτωση που επιλεχθεί η **Αποβίβαση** το πρόγραμμα θα εξάγει από την ουρά και θα εμφανίζει όλα τα αυτοκίνητα που αποβιβάστηκαν στην ΑΙΓΙΝΑ.

Απάντηση

Παρακάτω υλοποιείται το πρόγραμμα του παραδείγματος επιβίβασης/αποβίβασης σε οχηματαγωγό πλοίο, όπου η «πόρτα» για την είσοδο των οχημάτων στο πλοίο είναι διαφορετική από την «πόρτα» εξόδου των οχημάτων από αυτό.

Ανάλυση της λύσης

Σύμφωνα με την εκφώνηση της άσκησης πρέπει:

- Να υλοποιηθεί μια δομή δεδομένων FIFO (ουρά). Η υλοποίηση της ουράς θα πραγματοποιηθεί με μονοδιάστατο πίνακα 250 θέσεων (π[250]) όσο και το μέγιστο πλήθος επιβιβαζόμενων οχημάτων (250).
- Να δηλωθούν δύο βοηθητικές μεταβλητές (δείκτες) που δείχνουν την αρχή («**αρχ**») και το τέλος («**τέλος**») της ουράς. Οι αρχικές τιμές των δεικτών «**αρχ**» και «**τελος**» θα είναι μηδέν.
- Στην περίπτωση της **επιβίβασης**, πρώτα θα αυξάνεται ο δείκτης «τέλος» κατά 1, για να δείχνει στην επόμενη κενή θέση της ουράς (τέλος<-τέλος+1). Σε αυτή την κενή θέση θα καταχωρίζεται ο αριθμός κυκλοφορίας («**αρ**») του επιβιβαζόμενου οχήματος (π[τελος]<-αρ).
- Στην περίπτωση της **αποβίβασης**, πρώτα θα αποβιβάζεται το όχημα και μετά ο δείκτης «**αρχ**» θα αυξάνεται κατά 1 (αρχ<-αρχ+1), έτσι ώστε να δείχνει στη θέση που αντιστοιχίζεται στο επόμενο όχημα προς αποβίβαση.
- Να δηλωθούν και οι μεταβλητές: **επ1** (επιλογή μενού), **πλ** (μετρητής αποβιβαζόμενων οχημάτων), **επ2** (μεταβλητή δήλωσης οχήματος για επιβίβαση), **αρ** (αριθμός κυκλοφορίας οχήματος, μέσω του οποίου καταχωρίζεται-δηλώνεται η επιβίβασή του), **π**[250] (πίνακας 250 θέσεων, που φαίνεται το μέγιστο πλήθος των οχημάτων που μπορούν να χωρέσουν στο πλοίο).

- Να εμφανίζεται ένα μενού επιλογών με τις επιλογές: **1.** «Επιβίβαση», **2.** «Αποβίβαση», **3.** «Έξοδος» (δομή επανάληψης «**ΜΕΧΡΙΣ_ΟΤΟΥ…»**, με τη βοήθεια της μεταβλητής «**επ1**» η οποία θα παίρνει τις τιμές 1 ή 2 ή 3).
- Στην περίπτωση της **επιβίβασης** («**ΑΝ** επ1**=**1 **ΤΟΤΕ»):**
  - Αρχικά, γίνεται έλεγχος αν το πλοίο είναι γεμάτο («**ΑΝ** τελος**=**250 **ΤΟΤΕ»**). Στην περίπτωση αυτή εμφανίζεται κατάλληλο μήνυμα («**ΓΡΑΨΕ** 'Το πλοίο είναι πλήρες και δε χωρά άλλα οχήματα'»)
  - Στη συνέχεια, θα πρέπει να ελεγχθεί αν υπάρχει όχημα για επιβίβαση. Ο έλεγχος πραγματοποιείται μέσω της μεταβλητής «**επ2**» («**ΔΙΑΒΑΣΕ** επ2, **ΑΝ** επ2 **<>**'Ν'…»).
  - Αν υπάρχει όχημα προς επιβίβαση, θα πρέπει να ελέγχεται αν είναι κενό το πλοίο/ουρά. Αν είναι κενό («**ΑΝ (**αρχ**=**0 **ΚΑΙ** τελος**=**0**)** **ΤΟΤΕ**»), αυξάνουμε τους δείκτες «**αρχ**» και «**τέλος**» για να δείχνουν στην πρώτη κενή θέση του πλοίου/ουράς («αρχ**<-**1, τελος**<-**1 ») και μετά γίνεται καταχώριση του αριθμού κυκλοφορίας του επιβιβαζόμενου οχήματος στη θέση αυτή (πρώτη θέση του πλοίου/ουράς) («π**[**τελος**]<-** αρ»).
    Στην περίπτωση που το πλοίο/ουρά δεν είναι κενό, αυξάνεται πρώτα η τιμή του πίσω δείκτη της ουράς («τελος**<-**τελος**+**1 »), για να δείχνει σε κενή θέση κι έπειτα καταχωρίζεται σε αυτή ο αριθμός κυκλοφορίας του επιβιβαζόμενου οχήματος. Η διαδικασία αυτή επαναλαμβάνεται όσο υπάρχουν οχήματα προς επιβίβαση και το πλοίο δεν έχει γεμίσει («**ΜΕΧΡΙΣ_ΟΤΟΥ** τελος **=**250 **Η** επ2**=**'0' **Η** επ2**=**'ο'»).
    Στην περίπτωση που διακοπεί η επαναληπτική διαδικασία της επιβίβασης λόγω «γεμίσματος» του πλοίου («**ΑΝ** τελος**=**250 **ΤΟΤΕ»),** εμφανίζεται κατάλληλο μήνυμα («**ΓΡΑΨΕ** 'Το πλοίο είναι πλήρες και δε χωρά άλλα οχήματα'»).
- Στην περίπτωση της **αποβίβασης** («**ΑΛΛΙΩΣ_ΑΝ** επ1**=**2 **ΤΟΤΕ**»):
  - Θα πρέπει πρώτα να ελεγχθεί αν το πλοίο/ουρά είναι κενό («**ΑΝ (**αρχ**=**0 **ΚΑΙ** τελος**=**0**)** **ΤΟΤΕ**») και στην περίπτωση αυτή να εμφανίζεται κατάλληλο μήνυμα («**ΓΡΑΨΕ** 'Το πλοίο είναι άδειο'»)
  - Έπειτα, θα πρέπει να γίνει έλεγχος της περίπτωσης που το πλοίο/ουρά έχει ένα μόνο όχημα (**ΑΛΛΙΩΣ_ΑΝ** αρχ**=**τελος **ΤΟΤΕ**»). Στην περίπτωση αυτή πρώτα αποβιβάζεται το όχημα από το πλοίο («εξάγεται» από την ουρά, «**ΓΡΑΨΕ** 'Αποβιβάζεται το μοναδικό όχημα:'**,** π**[**αρχ**]**») και μετά μηδενίζουμε τους δείκτες («αρχ**<-**0», «τέλος**<-**0»), αφού το πλοίο/ουρά είναι πλέον κενό.
    Σημειώνεται ότι για λόγους παρακολούθησης των τιμών του πίνακα, στη θέση του πίνακα που κατείχε το αποβιβαζόμενο όχημα καταχωρίζεται ο κενός χαρακτήρας («π**[**αρχ**]<-**' '»).
  - Αν υπάρχουν περισσότερα από ένα οχήματα για αποβίβαση, μέσα σε μια επαναληπτική διαδικασία πρώτα εμφανίζεται το όχημα που αποβιβάζεται («**ΓΡΑΨΕ** 'Αποβιβάζεται το όχημα: '**,** π**[**αρχ**]**») κι έπειτα ο μπροστινός δείκτης «**αρχ**» αυξάνεται κατά 1 («αρχ**<-**αρχ**+**1 ») και δείχνει στη θέση του επόμενου οχήματος προς αποβίβαση. Η επαναληπτική διαδικασία αποβίβασης ολοκληρώνεται, όταν πλέον δεν υπάρχει κανένα όχημα για αποβίβαση, δηλαδή όταν ο δείκτης «**αρχ**» γίνει μεγαλύτερος από τον πίσω δείκτη «**τέλος**» της ουράς («**ΜΕΧΡΙΣ_ΟΤΟΥ (**αρχ**>**τελος**)**»).
  - Τέλος, με την ολοκλήρωση της αποβίβασης όλων των οχημάτων, εκτυπώνεται με κατάλληλο μήνυμα το πλήθος αποβιβασθέντων οχημάτων στην ΑΙΓΙΝΑ («**ΓΡΑΨΕ** 'Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ: '**,** πλ»).

- Σε κάθε νέα εκκίνηση αποβίβασης οχημάτων, ο μετρητής «**πλ**» πρέπει να μηδενίζεται, για να είναι σωστό το σύνολο των αποβιβασθέντων οχημάτων της τρέχουσας αποβίβασης. Π.χ. εκτελείται το πρόγραμμα και επιβιβάζονται 15 οχήματα, τα οποία στη συνέχεια αποβιβάζονται. Στη συνέχεια, επιλέγεται, από το μενού, νέα επιβίβαση 30 οχημάτων, τα οποία στη συνέχεια αποβιβάζονται. Αν ο μετρητής «πλ» δεν μηδενίζεται σε κάθε νέα εκκίνηση αποβίβασης, στην πρώτη αποβίβαση θα είχε την τιμή «15» και στη δεύτερη την τιμή «45» και όχι την τιμή «30» που είναι η σωστή. Δηλαδή, στη δεύτερη αποβίβαση οχημάτων το πρόγραμμα θα τύπωνε λανθασμένα «Οχήματα που αποβιβάσθηκαν στην ΑΙΓΙΝΑ: 45» και όχι το σωστό «Οχήματα που αποβιβάσθηκαν στην ΑΙΓΙΝΑ: 30» («**ΓΡΑΨΕ** 'Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ: ', πλ»).
  Επίσης, αν επιλεχθεί νέα επιβίβαση μετά την ολοκλήρωση κάθε αποβίβασης, θα πρέπει η ουρά να γεμίζει από την αρχή. Κάτι τέτοιο επιτυγχάνεται με τον μηδενισμό των τιμών των δύο δεικτών, καθώς παίρνουν την ίδια τιμή («αρχ<-0», «τελος <- 0»).

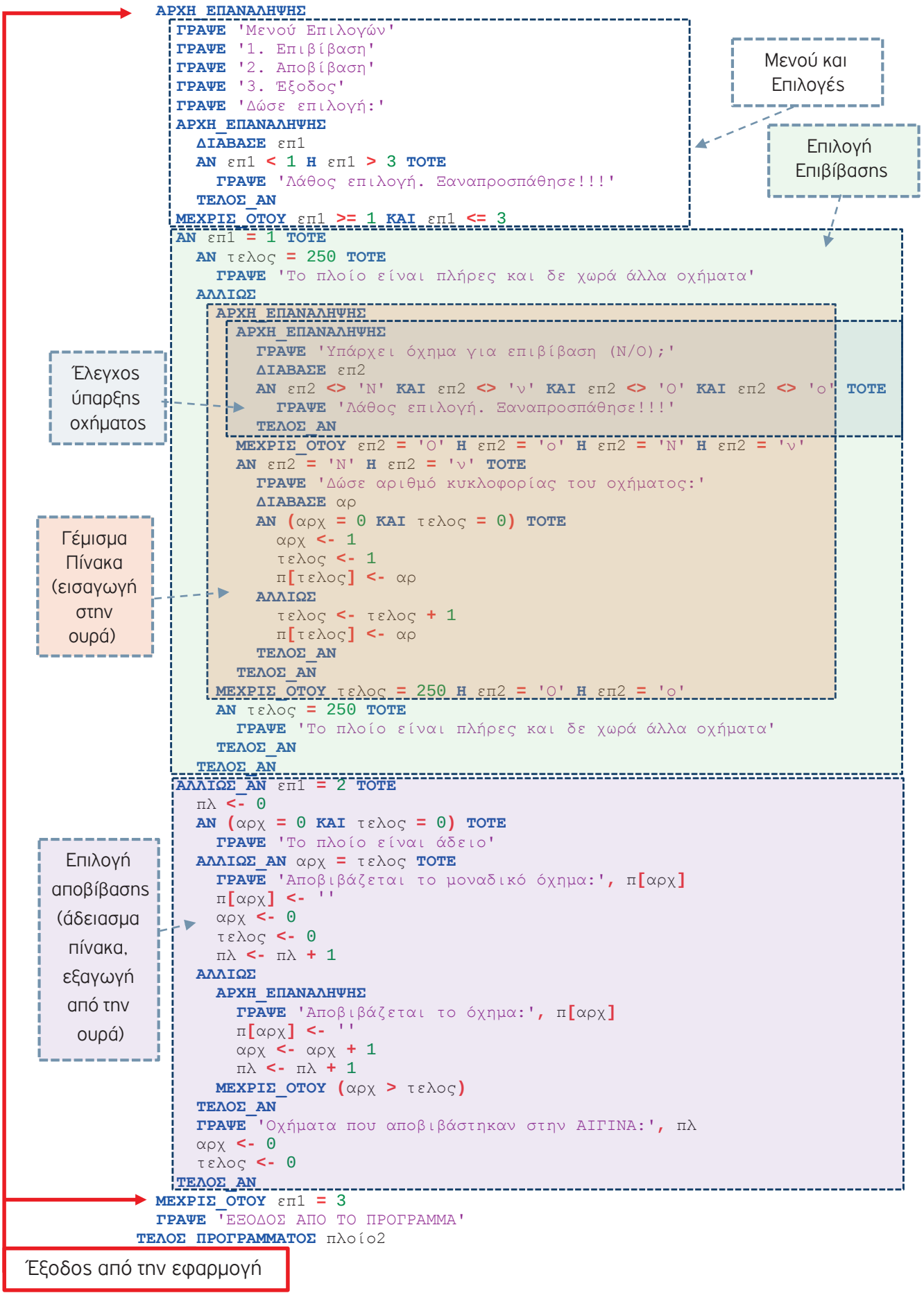
Ακολουθεί η υλοποίηση του προγράμματος σε ΓΛΩΣΣΑ.

Κώδικας σε ΓΛΩΣΣΑ [1.6]

```
ΠΡΟΓΡΑΜΜΑ πλοίο2
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: αρχ, τελος, επ1, πλ
  ΧΑΡΑΚΤΗΡΕΣ: επ2, αρ, π[250]
ΑΡΧΗ
  αρχ <- 0
  τελος <- 0
```

Δήλωση Μεταβλητών

Αρχικοποίηση Μεταβλητών


ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Μενού Επιλογών'
ΓΡΑΨΕ '1. Επιβίβαση'
ΓΡΑΨΕ '2. Αποβίβαση'
ΓΡΑΨΕ '3. Έξοδος'
ΓΡΑΨΕ 'Δώσε επιλογή:'
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ επ1
ΑΝ επ1 < 1 Η επ1 > 3 ΤΟΤΕ
ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ επ1 >= 1 ΚΑΙ επ1 <= 3
Μενού και Επιλογές
Επιλογή Επιβίβασης
ΑΝ επ1 = 1 ΤΟΤΕ
ΑΝ τελος = 250 ΤΟΤΕ
ΓΡΑΨΕ 'Το πλοίο είναι πλήρες και δε χωρά άλλα οχήματα'
ΑΛΛΙΩΣ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Υπάρχει όχημα για επιβίβαση (Ν/Ο);'
ΔΙΑΒΑΣΕ επ2
ΑΝ επ2 <> 'Ν' ΚΑΙ επ2 <> 'ν' ΚΑΙ επ2 <> 'Ο' ΚΑΙ επ2 <> 'ο' ΤΟΤΕ
ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'
ΤΕΛΟΣ_ΑΝ
Έλεγχος ύπαρξης οχήματος
ΜΕΧΡΙΣ_ΟΤΟΥ επ2 = 'Ο' Η επ2 = 'ο' Η επ2 = 'Ν' Η επ2 = 'ν'
ΑΝ επ2 = 'Ν' Η επ2 = 'ν' ΤΟΤΕ
ΓΡΑΨΕ 'Δώσε αριθμό κυκλοφορίας του οχήματος:'
ΔΙΑΒΑΣΕ αρ
ΑΝ (αρχ = 0 ΚΑΙ τελος = 0) ΤΟΤΕ
αρχ <- 1
τελος <- 1
π[τελος] <- αρ
Γέμισμα Πίνακα (εισαγωγή στην ουρά)
ΑΛΛΙΩΣ
τελος <- τελος + 1
π[τελος] <- αρ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ τελος = 250 Η επ2 = 'Ο' Η επ2 = 'ο'
ΑΝ τελος = 250 ΤΟΤΕ
ΓΡΑΨΕ 'Το πλοίο είναι πλήρες και δε χωρά άλλα οχήματα'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ_ΑΝ επ1 = 2 ΤΟΤΕ
πλ <- 0
ΑΝ (αρχ = 0 ΚΑΙ τελος = 0) ΤΟΤΕ
ΓΡΑΨΕ 'Το πλοίο είναι άδειο'
ΑΛΛΙΩΣ_ΑΝ αρχ = τελος ΤΟΤΕ
ΓΡΑΨΕ 'Αποβιβάζεται το μοναδικό όχημα:', π[αρχ]
π[αρχ] <- ''
αρχ <- 0
τελος <- 0
πλ <- πλ + 1
ΑΛΛΙΩΣ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Αποβιβάζεται το όχημα:', π[αρχ]
π[αρχ] <- ''
αρχ <- αρχ + 1
πλ <- πλ + 1
ΜΕΧΡΙΣ_ΟΤΟΥ (αρχ > τελος)
ΤΕΛΟΣ_ΑΝ
ΓΡΑΨΕ 'Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ:', πλ
αρχ <- 0
τελος <- 0
Επιλογή αποβίβασης (άδειασμα πίνακα, εξαγωγή από την ουρά)
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ επ1 = 3
ΓΡΑΨΕ 'ΕΞΟΔΟΣ ΑΠΟ ΤΟ ΠΡΟΓΡΑΜΜΑ'
ΤΕΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ πλοίο2
Έξοδος από την εφαρμογή

```
1   ΠΡΟΓΡΑΜΜΑ πλοίο2
2   ΜΕΤΑΒΛΗΤΕΣ
3     ΑΚΕΡΑΙΕΣ: αρχ, τελος, επ1, πλ
4     ΧΑΡΑΚΤΗΡΕΣ: επ2, αρ, π[250]
5   ΑΡΧΗ
6     αρχ <- 0
7     τελος <- 0
8     ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
9       ΓΡΑΨΕ 'Μενού Επιλογών'
10      ΓΡΑΨΕ '1. Επιβίβαση'
11      ΓΡΑΨΕ '2. Αποβίβαση'
12      ΓΡΑΨΕ '3. Έξοδος'
13      ΓΡΑΨΕ 'Δώσε επιλογή:'
14      ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
15        ΔΙΑΒΑΣΕ επ1
16        ΑΝ επ1 < 1 Η επ1 > 3 ΤΟΤΕ
17          ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'
18        ΤΕΛΟΣ_ΑΝ
19      ΜΕΧΡΙΣ_ΟΤΟΥ επ1 >= 1 ΚΑΙ επ1 <= 3
20      ΑΝ επ1 = 1 ΤΟΤΕ
21        ΑΝ τελος = 250 ΤΟΤΕ
22          ΓΡΑΨΕ 'Το πλοίο είναι πλήρες και δε χωρά άλλα οχήματα'
23        ΑΛΛΙΩΣ
24          ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
25            ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
26              ΓΡΑΨΕ 'Υπάρχει όχημα για επιβίβαση (Ν/Ο);'
27              ΔΙΑΒΑΣΕ επ2
28              ΑΝ επ2<>'Ν' ΚΑΙ επ2<>'ν' ΚΑΙ επ2<>'Ο' ΚΑΙ επ2<>'ο' ΤΟΤΕ
29                ΓΡΑΨΕ 'Λάθος επιλογή. Ξαναπροσπάθησε!!!'
30              ΤΕΛΟΣ_ΑΝ
31            ΜΕΧΡΙΣ_ΟΤΟΥ επ2='Ο' Η επ2='ο' Η επ2='Ν' Η επ2='ν'
32            ΑΝ επ2 = 'Ν' Η επ2 = 'ν' ΤΟΤΕ
33              ΓΡΑΨΕ 'Δώσε αριθμό κυκλοφορίας του οχήματος:'
34              ΔΙΑΒΑΣΕ αρ
35              ΑΝ (αρχ = 0 ΚΑΙ τελος = 0) ΤΟΤΕ
36                αρχ <- 1
37                τελος <- 1
38                π[τελος] <- αρ
39              ΑΛΛΙΩΣ
40                τελος <- τελος + 1
```

```
41                  π[τελος] <- αρ
42                ΤΕΛΟΣ_ΑΝ
43              ΤΕΛΟΣ_ΑΝ
44            ΜΕΧΡΙΣ_ΟΤΟΥ τελος = 250 Η επ2 = 'Ο' Η επ2 = 'ο'
45            ΑΝ τελος = 250 ΤΟΤΕ
46              ΓΡΑΨΕ 'Το πλοίο είναι πλήρες και δε χωρά άλλα οχήματα'
47            ΤΕΛΟΣ_ΑΝ
48          ΤΕΛΟΣ_ΑΝ
49        ΑΛΛΙΩΣ_ΑΝ επ1 = 2 ΤΟΤΕ
50          πλ <- 0
51          ΑΝ (αρχ = 0 ΚΑΙ τελος = 0) ΤΟΤΕ
52            ΓΡΑΨΕ 'Το πλοίο είναι άδειο'
53          ΑΛΛΙΩΣ_ΑΝ αρχ = τελος ΤΟΤΕ
54            ΓΡΑΨΕ 'Αποβιβάζεται το μοναδικό όχημα:', π[αρχ]
55            π[αρχ] <- ''
56            αρχ <- 0
57            τελος <- 0
58            πλ <- πλ + 1
59          ΑΛΛΙΩΣ
60            ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
61              ΓΡΑΨΕ 'Αποβιβάζεται το όχημα:', π[αρχ]
62              π[αρχ] <- ''
63              αρχ <- αρχ + 1
64              πλ <- πλ + 1
65            ΜΕΧΡΙΣ_ΟΤΟΥ (αρχ > τελος)
66          ΤΕΛΟΣ_ΑΝ
67          ΓΡΑΨΕ 'Οχήματα που αποβιβάστηκαν στην ΑΙΓΙΝΑ:', πλ
68          αρχ <- 0
69          τελος <- 0
70        ΤΕΛΟΣ_ΑΝ
71      ΜΕΧΡΙΣ_ΟΤΟΥ επ1 = 3
72      ΓΡΑΨΕ 'ΕΞΟΔΟΣ ΑΠΟ ΤΟ ΠΡΟΓΡΑΜΜΑ'
73    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ πλοίο2
```

1.2.2 Ερωτήσεις - Ασκήσεις

Ε. 1: Να δώσετε παραδείγματα ουράς από την καθημερινή ζωή.

Ε. 2: Δίνεται η επόμενη ακολουθία αριθμών : 4, 8, 2, 5, 9, 13.

1. Ποια λειτουργία θα χρησιμοποιήσετε για την τοποθέτηση των αριθμών σε ουρά;
2. Να σχεδιάσετε την ουρά έπειτα από την τοποθέτηση των αριθμών.
3. Ποια λειτουργία θα χρησιμοποιήσετε για την εξαγωγή των αριθμών από την ουρά;
4. Πόσες φορές θα πρέπει να εκτελεστεί η προηγούμενη λειτουργία στην ουρά για να εξαχθεί ο αριθμός 5;

Ε. 3:

1. Σε μια ουρά 10 θέσεων έχουν τοποθετηθεί διαδοχικά τα στοιχεία: Χ, Α, Β, Α, Ρ στην 1η, 2η, 3η, 4η και 5η θέση αντίστοιχα.
   i. Να προσδιορίσετε τις τιμές των δεικτών rear και front της παραπάνω ουράς και να τη σχεδιάσετε.
   ii. Αν εφαρμόσουμε τις ακόλουθες λειτουργίες: **Εξαγωγή**, **Εξαγωγή**, **Εξαγωγή**, **Εισαγωγή Χ , Εισαγωγή Δ** και **Εξαγωγή** ποιες είναι τις τιμές των δεικτών rear και front της παραπάνω ουράς και ποια η τελική μορφή της ουράς;
2. Σε μια κενή ουρά 10 θέσεων εισάγουμε τα στοιχεία Κ, Φ, Ι, Α,Ρ. Με ποιον τρόπο πρέπει να «εισαχθούν» και να «εξαχθούν» τα στοιχεία, ώστε να έχουμε ως έξοδο τα δεδομένα Α, Ρ, Χ, Η.

Ε. 4: Σε μια τράπεζα χρησιμοποιείται αυτόματο ηλεκτρονικό μηχάνημα που το χειρίζονται οι πελάτες, οι ταμίες και ο διευθυντής της τράπεζας. Κάθε ένας από τους χειριστές του μηχανήματος έχει δικαιώματα χρήσης συγκεκριμένων πλήκτρων του πληκτρολογίου.

Ο πελάτης το «**Π**», οι ταμίες το «**1**» ή το «**2**» ή το «**3**» ή το «**4**» αναλόγως της θέσης του ταμείου που εργάζονται και ο διευθυντής το πλήκτρο «**Δ**». Κατά την είσοδό του, ο κάθε πελάτης πατάει το πλήκτρο «**Π**» και εκτυπώνεται ένα χαρτί, στο οποίο αναγράφεται το νούμερο που έχει στην ουρά από την αρχή της ημέρας.
Η τράπεζα έχει 4 ταμεία, όπου όταν ο ταμίας εξυπηρετεί έναν πελάτη, πατάει το νούμερο του ταμείου του, «**1**» ή «**2**» ή «**3**» ή «**4**».
Ο διευθυντής της τράπεζας, πατώντας το κουμπί «**Δ**», σταματά τη διαδικασία εξυπηρέτησης των πελατών και μπορεί να δει το σύνολο των πελατών που έχουν ήδη εξυπηρετηθεί από το κάθε ταμείο. Ο μέγιστος αριθμός πελατών που μπορεί να εξυπηρετήσει η τράπεζα είναι 1.000 πελάτες.
Λαμβάνοντας υπόψη ότι ο πρώτος πελάτης εξυπηρετείται πρώτος και ο τελευταίος εξυπηρετείται τελευταίος, να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ, όπου:

1. Να υπάρχει μενού επιλογής:
   Π. Πελάτης **Τ.** Ταμίας **Δ.** Διευθυντής
2. Στην περίπτωση που επιλεχθεί από το μενού το **Π.** Πελάτης, το πρόγραμμα εκτυπώνει το νούμερο που έχει στην ουρά (από την αρχή της ημέρας).
3. Στην περίπτωση που επιλεχθεί από το μενού το **Τ.** Ταμίας, ο/η αρμόδιος/-α υπάλληλος επιλέγει το νούμερο του ταμείου που του/της αντιστοιχεί: «**1**» ή «**2**» ή «**3**» ή «**4**» και ο πελάτης διαγράφεται από την ουρά.
4. Στην περίπτωση που επιλεχθεί από το μενού το **Δ.** Διευθυντής, σταματά η διαδικασία εξυπηρέτησης και το πρόγραμμα τυπώνει το νούμερο του ταμείου που εξυπηρέτησε τους περισσότερους πελάτες.

Στο πρόγραμμα να γίνεται έλεγχος των δεδομένων εισόδου.

Ε. 5: 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». Το πρόγραμμα εμφανίζει το πλήθος των ατόμων που κατάφεραν να κάνουν κράτηση μέσα από την λίστα αναμονής, καθώς και το μέγιστο πλήθος των ατόμων που περίμεναν στην ουρά αναμονής.

Στο πρόγραμμα να γίνεται έλεγχος εγκυρότητας των τιμών που πληκτρολογούνται.

Ε. 6: Σε ένα ταχυδρομικό κατάστημα, οι πελάτες εξυπηρετούνται με βάση τη σειρά άφιξής τους σε αυτό. Το ταχυδρομικό κατάστημα έχει ένα ταμείο και ο μέσος χρόνος εξυπηρέτησης κάθε πελάτη είναι 3 λεπτά. Η ουρά αναμονής στο κατάστημα δεν μπορεί να ξεπερνά τα 30 άτομα.

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 ονοματεπώνυμο του πελάτη και αμέσως μετά να εμφανίζει το πλήθος των ατόμων που περιμένουν πριν από αυτόν, εκτός αν η ουρά αναμονής είναι γεμάτη, οπότε να εμφανίζει το μήνυμα «Το κατάστημα γέμισε. Παρακαλούμε ελάτε άλλη φορά».
3. Αν δοθεί η τιμή «**2.**ΕΠΟΜΕΝΟΣ», τότε το πρόγραμμα να εμφανίζει το ονοματεπώνυμο του 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.

Ε. 7: 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.

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

1. Να διαβάζει επαναληπτικά, με έλεγχο εγκυρότητας, το γράμμα "Ν" που καθορίζει την έλευση νέου αρχείου ή το γράμμα "Ε" που δηλώνει την προσπάθεια εκτύπωσης ενός αρχείου.
2. 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».
3. 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.
4. Η επαναληπτική διαδικασία ολοκληρώνεται, όταν εκτυπωθούν όλα τα αρχεία που έχουν τοποθετηθεί στην ουρά.
5. Μετά το τέλος της διαδικασίας, το πρόγραμμα εμφανίζει τον συνολικό αριθμό των αρχείων που εκτυπώθηκαν.

Ε. 8: Χαρακτηρίστε τις παρακάτω προτάσεις ως Σωστές ή Λάθος. Στην περίπτωση που πιστεύετε ότι είναι λανθασμένες δικαιολογήστε την επιλογή σας και σκεφτείτε ποια θα μπορούσε να είναι η αντίστοιχη σωστή πρόταση.

α/α	Προτάσεις	Σ	Λ
1	Για την υλοποίηση της ουράς μπορεί να χρησιμοποιηθεί πίνακας.	❒	❒
2	Κατά την εισαγωγή ενός στοιχείου σε ουρά, αυτό τοποθετείται στο μπροστινό άκρο της.	❒	❒
3	Σε μια ουρά κάθε στοιχείο της εξάγεται από το μπροστινό άκρο της.	❒	❒
4	Η απώθηση είναι μια από τις λειτουργίες της ουράς.	❒	❒
5	Η εισαγωγή και η εξαγωγή είναι οι δύο βασικές λειτουργίες της ουράς.	❒	❒
6	Στην ουρά το στοιχείο που μπαίνει πρώτο βγαίνει και πρώτο.	❒	❒
7	Η υλοποίηση της ουράς χρησιμοποιεί μία μεταβλητή-δείκτη για την εκτέλεση των δύο βασικών λειτουργιών της.	❒	❒
8	Η λειτουργία της εξαγωγής μπορεί να εκτελεστεί σε μια γεμάτη ουρά.	❒	❒

1.3 Άλλες δομές δεδομένων

Κοινό γνώρισμα των παραπάνω δομών που υλοποιήθηκαν με χρήση μονοδιάστατου πίνακα είναι ότι οι διαδοχικοί κόμβοι αποθηκεύονται σε συνεχόμενες θέσεις της κύριας μνήμης. Στην ενότητα αυτή παρουσιάζεται η περίπτωση τριών σημαντικών δομών δεδομένων, στις οποίες οι κόμβοι δεν είναι απαραίτητο να κατέχουν συνεχόμενες θέσεις μνήμης. Οι δομές αυτές ανήκουν στην κατηγορία των δυναμικών δομών δεδομένων και πρόκειται για τις λίστες, τα δένδρα και τους γράφους.

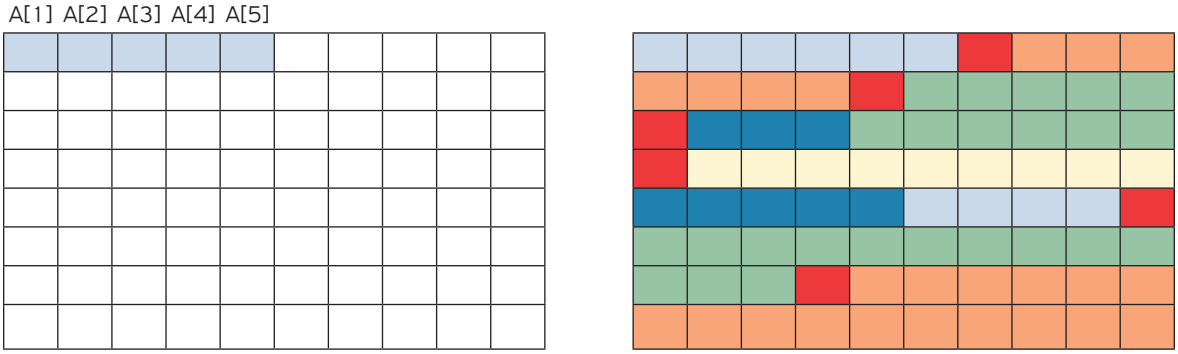
1.3.1 Λίστες

Η έννοια της λίστας συναντάται αρκετά συχνά στην καθημερινότητά μας. Πόσες φορές δεν έχουμε καταρτίσει λίστες με τα αγαπημένα μας τραγούδια, τις επαφές μας, τα ψώνια που θέλουμε να κάνουμε ή ακόμα και λίστες με επιθυμίες και όνειρα; Η λίστα δεν είναι τίποτα άλλο παρά μία συλλογή από αντικείμενα του ίδιου τύπου. Μπορούμε να έχουμε δηλαδή λίστες από λέξεις, από ονόματα αλλά και από αριθμούς.

Πώς όμως θα μπορούσατε να αποθηκεύσετε έναν συγκεκριμένο αριθμό στοιχείων του ίδιου τύπου, για παράδειγμα: τα μέρη που θα θέλατε να επισκεφτείτε; Μία από τις δομές δεδομένων που έχετε δει μέχρι τώρα και που θα μπορούσατε να χρησιμοποιήσετε για να αποθηκεύσετε τα στοιχεία αυτά είναι οι πίνακες. Πολύ απλά και εύκολα μπορείτε να τους δημιουργήσετε και να προσπελάστε άμεσα οποιοδήποτε στοιχείο τους.

Το γεγονός, όμως, ότι το μέγεθος ενός πίνακα παραμένει σταθερό (στατική δομή δεδομένων), δυσχεραίνει την προσθήκη και τη διαγραφή στοιχείων. Θεωρείστε την περίπτωση που θέλετε να προσθέσετε στον πίνακα και άλλα μέρη στα οποία θέλετε να ταξιδέψετε. Τι θα κάνετε αν ο πίνακας είναι γεμάτος; Μια πρώτη απάντηση θα ήταν να δημιουργήσουμε έναν νέο μεγαλύτερο πίνακα και να αντιγράψουμε σε αυτόν τα στοιχεία του προηγούμενου πίνακα. Έτσι, θα είχαμε μεγαλύτερο χώρο για να προσθέσουμε και νέα στοιχεία. Θεωρείτε ότι αυτή είναι η ιδανική λύση, όταν η προσθήκη και η διαγραφή στοιχείων αποτελεί συχνό φαινόμενο;

Ας μελετήσουμε μία άλλη λύση, δηλαδή μία νέα δυναμική δομή δεδομένων, τις **συνδεδεμένες λίστες,** που απαντάει σε αυτές τις προκλήσεις. Αρχικά, ας θεωρήσουμε το πλέγμα των κελιών που παρουσιάζεται στην Εικόνα 1.3.1.α ως τη μνήμη του υπολογιστή. Σε κάθε κελί αντιστοιχίζεται μία διεύθυνση. Για να προσπελάσουμε ένα οποιαδήποτε κελί θα πρέπει να γνωρίζουμε τη διεύθυνσή του. Στην Εικόνα 1.3.1.α παρουσιάζεται ο χώρος που καταλαμβάνει ο πίνακας Α με το γαλάζιο χρώμα. Ο πίνακας αυτός αποτελείται από έξι στοιχεία τα οποία είναι αποθηκευμένα σε συνεχόμενες θέσεις μνήμης.

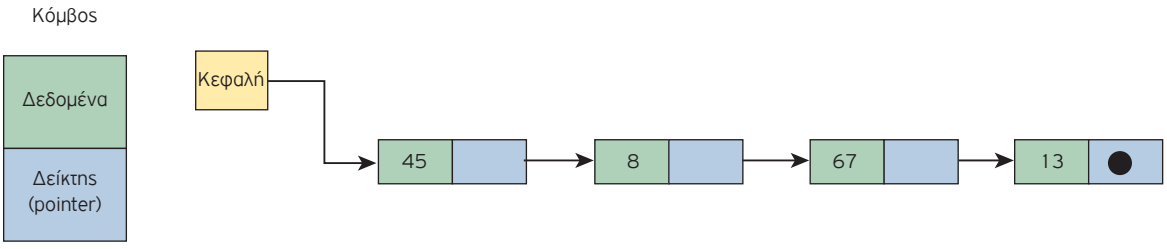


Εικόνα 1.3.1. **α: Ο πίνακας Α στη μνήμη** **β: Με το χρώμα κόκκινο αναπαριστάνουμε τα ελεύθερα κελιά**

Συνήθως όμως, η μνήμη του υπολογιστή δεν είναι τόσο καθαρή και τακτοποιημένη. Στην Εικόνα 1.3.1.β φαίνονται τα διάσπαρτα (μη συνεχόμενα) ελεύθερα κελιά με το κόκκινο χρώμα. Στην προκειμένη περίπτωση, δεν μπορείτε να «τοποθετήσετε» έναν πίνακα στη μνήμη, διότι δεν υπάρχουν συνεχόμενες ελεύθερες θέσεις. Μπορείτε, όμως, να χρησιμοποιήσετε τις διάσπαρτες αυτές ελεύθερες θέσεις για να αποθηκεύσετε τα δεδομένα σας, αν θεωρήσετε ότι αποτελούν «στοιχεία» μίας συνδεδεμένης λίστας.

Η συνδεδεμένη λίστα αποτελείται από μία σειρά από κόμβους, που συνήθως βρίσκονται σε απομακρυσμένες θέσεις μνήμης. Κάθε κόμβος αποτελείται από δύο κύρια τμήματα (Εικόνα 1.3.2.α). Το πρώτο τμήμα περιέχει τα δεδομένα και το δεύτερο τμήμα φιλοξενεί τη διεύθυνση του επόμενου κόμβου με τον οποίο συνδέεται ή όπως αλλιώς θα λέγαμε στη γλώσσα των δομών δεδομένων, το δεύτερο τμήμα περιέχει έναν δείκτη (pointer) που δείχνει στον επόμενο κόμβο.

Το πεδίο Δεδομένα μπορεί να περιέχει μία ή περισσότερες αλφαριθμητικές ή αριθμητικές πληροφορίες. Ο δείκτης (pointer) είναι ένας ιδιαίτερος τύπος δεδομένων που προσφέρεται από τις περισσότερες σύγχρονες γλώσσες προγραμματισμού. Ο δείκτης δε λαμβάνει αριθμητικές τιμές όπως ακέραιες, πραγματικές κ.ά., αλλά οι τιμές του είναι διευθύνσεις στην κύρια μνήμη και χρησιμοποιείται ακριβώς για τη σύνδεση των διαφόρων στοιχείων μιας δομής, που είναι αποθηκευμένα σε μη συνεχόμενες θέσεις μνήμης.



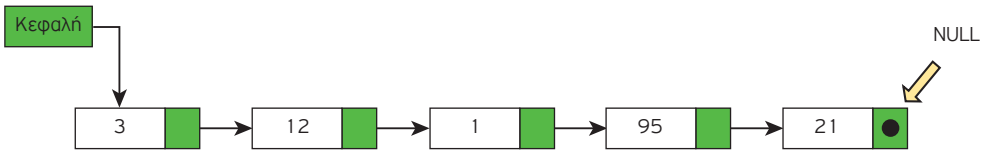
Εικόνα 1.3.2. α: Αναπαράσταση κόμβου **β: Αναπαράσταση απλά συνδεδεμένης λίστας με 3 κόμβους**

Ένα παράδειγμα απλά συνδεδεμένης λίστας με τρεις κόμβους, που περιέχουν ακεραίους, δίνεται στην Εικόνα 1.3.2.β. Ο δείκτης του τελευταίου κόμβου της λίστας έχει ως τιμή το NULL (κενό) και τον αναπαριστούμε συμβολικά με το σύμβολο «●». Οι δείκτες των υπολοίπων κόμβων περιέχουν τη διεύθυνση του επόμενου κόμβου αντίστοιχα και τους απεικονίζουμε συμβολικά με ένα βέλος για να υποδηλώσουμε τη σύνδεση μεταξύ του προηγούμενου και του επόμενου κόμβου. Κάθε λίστα συνοδεύεται από έναν δείκτη με το όνομα «Κεφαλή» (head), που δείχνει στον πρώτο κόμβο της λίστας, δηλαδή περιέχει τη διεύθυνση του πρώτου κόμβου της λίστας.

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**.

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)**.



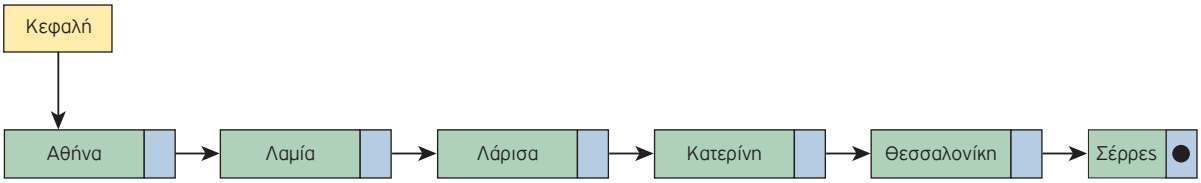
Εικόνα 1.3.3. Μία απλά συνδεδεμένη λίστα με 5 κόμβους.

Οι κόμβοι μιας λίστας δεν έχουν ονόματα. Γνωρίζουμε μόνο τις διευθύνσεις τους, που είναι αποθηκευμένες στους προηγούμενους κόμβους και αυτές θα αξιοποιήσουμε για να τους προσπελάσουμε. Επομένως, αν θελήσουμε να έχουμε πρόσβαση στον τέταρτο κόμβο μιας λίστας, για να επεξεργαστούμε τα δεδομένα που περιέχει, θα πρέπει να ξεκινήσουμε από τον πρώτο κόμβο της λίστας, η διεύθυνση του οποίου περιέχεται στον δείκτη **Κεφαλή**. Ξεκινώντας από τον πρώτο κόμβο της λίστας μπορούμε να έχουμε πρόσβαση στη διεύθυνση μνήμης του δεύτερου κόμβου. Από τον δεύτερο κόμβο μπορούμε να έχουμε πρόσβαση στη διεύθυνση μνήμης του τρίτου κόμβου. Και συνεχίζουμε έτσι μέχρι να φτάσουμε στον τέταρτο κόμβο. Αυτός είναι ο μόνος τρόπος για να διασχίσουμε μία συνδεδεμένη λίστα.

Πρόσβαση στους κόμβους μιας συνδεδεμένης λίστας

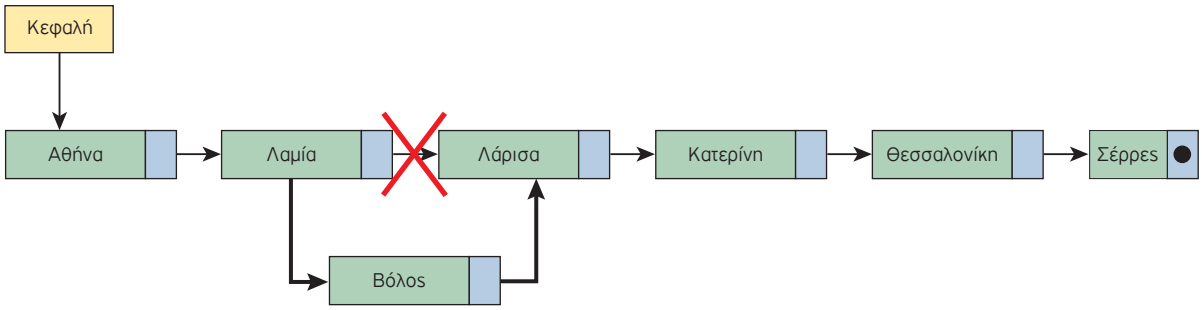
Οι κόμβοι μιας (απλά) συνδεδεμένης λίστας είναι διατεταγμένοι σε μια συγκεκριμένη σειρά, χωρίς αυτό να σημαίνει ότι αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη. Αντίθετα, είναι διασκορπισμένοι σε όλη τη μνήμη και η σύνδεση μεταξύ τους γίνεται μέσω των δεικτών. Έχουμε άμεση πρόσβαση μόνο στον πρώτο κόμβο της λίστας. 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.

Έστω ότι σχεδιάζουμε στη συνέχεια ένα ταξίδι στην Κεντρική Μακεδονία. Η αφετηρία μας είναι η Αθήνα και το τέρμα του ταξιδιού είναι οι Σέρρες. Έχουμε αποθηκεύσει σε μια συνδεδεμένη λίστα τις πόλεις που είναι πιθανό να επισκεφθούμε (Εικόνα 1.3.4).



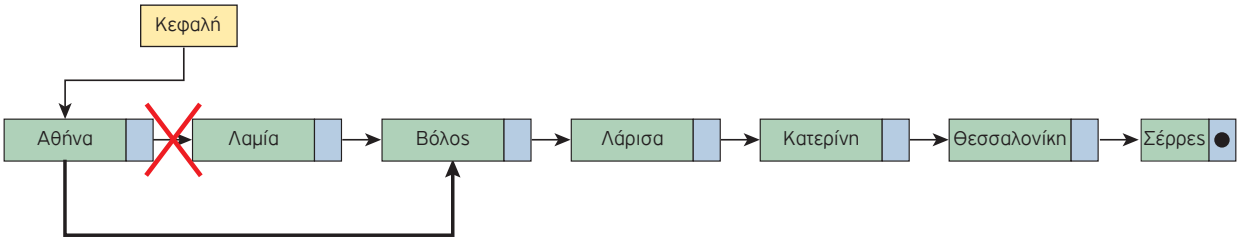
Εικόνα 1.3.4. Μια απλά συνδεδεμένη λίστα

Λίγο πριν το ταξίδι, αποφασίζουμε να επισκεφθούμε τον Βόλο, αμέσως μετά τη Λαμία και πριν φτάσουμε στη Λάρισα. Επομένως, χρειάζεται να εισαγάγουμε ένα νέο κόμβο στη λίστα μας, κάνοντας την κατάλληλη διευθέτηση των δεικτών (Εικόνα 1.3.5).



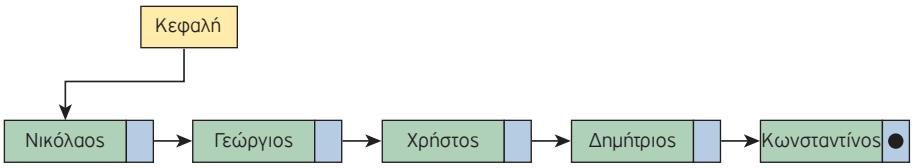
Εικόνα 1.3.5. Εισαγωγή νέου κόμβου στη λίστα

Όπως φαίνεται και στην Εικόνα 1.3.5, οι απαιτούμενες ενέργειες για την εισαγωγή (παρεμβολή) του νέου κόμβου είναι ο δείκτης του δεύτερου κόμβου να δείχνει τον νέο κόμβο και ο δείκτης του νέου κόμβου να δείχνει τον τέταρτο κόμβο (δηλαδή, να πάρει την τιμή που είχε πριν την εισαγωγή ο δείκτης του τρίτου κόμβου). Με αυτόν τον τρόπο, οι κόμβοι της λίστας διατηρούν τη λογική τους σειρά, αλλά οι φυσικές θέσεις στη μνήμη μπορεί να είναι τελείως διαφορετικές.
Αν, επιπλέον, αποφασίσουμε να μην επισκεφθούμε τη Λαμία, θα πρέπει να διαγράψουμε τον παραπάνω κόμβο από τη λίστα κάνοντας μια εκ νέου διευθέτηση των αντίστοιχων δεικτών (Εικόνα 1.3.6). Έτσι, για τη διαγραφή ενός κόμβου αρκεί ν' αλλάξει τιμή ο δείκτης του προηγούμενου κόμβου και να δείχνει πλέον στον επόμενο αυτού που διαγράφεται, όπως φαίνεται στην Εικόνα 1.3.6. Ο κόμβος που διαγράφηκε (ο δεύτερος) αποτελεί «άχρηστο δεδομένο» και ο χώρος μνήμης που καταλάμβανε, παραχωρείται για άλλη χρήση.



Εικόνα 1.3.6. Διαγραφή κόμβου από τη λίστα

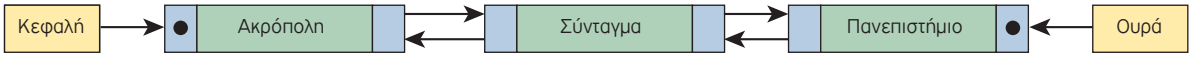
Ας δούμε ένα δεύτερο παράδειγμα με μία συνδεδεμένη λίστα. Ας φανταστούμε έναν αγώνα σκυταλοδρομίας (Εικόνα 1.3.7). Οι αθλητές που λαμβάνουν μέρος είναι τοποθετημένοι με μια συγκεκριμένη σειρά και ο κάθε αθλητής για να ξεκινήσει να τρέχει πρέπει να περιμένει τον προηγούμενο του. Σε περιπτώσεις όπως αυτή, που κάθε στοιχείο συνδέεται με το αμέσως επόμενο, χρησιμοποιούμε τη δομή δεδομένων της συνδεδεμένης λίστας για να αποθηκεύσουμε τις τιμές.



Εικόνα 1.3.7. Το παράδειγμα των αθλητών της σκυταλοδρομίας - Απλά συνδεδεμένη λίστα

Οι συνδεδεμένες λίστες αξιοποιούνται για την υλοποίηση της στοίβας και της ουράς, λόγω της δυνατότητάς αυξομείωσης του μεγέθους τους. Μπορείτε να εξηγήσετε γιατί;

Μια λίστα μπορεί να είναι **απλά συνδεδεμένη**, όπως στο παράδειγμα της σκυταλοδρομίας, δηλαδή να μπορούμε να κινηθούμε προς μία μόνο κατεύθυνση, ξεκινώντας από τον πρώτο κόμβο και μετακινούμενοι προς τον τελευταίο, όπως δείχνουν τα βέλη του σχήματος ή να είναι **διπλά συνδεδεμένη (doubly linked list)**, δηλαδή, να μπορούμε να τη διατρέξουμε και προς τις δύο κατευθύνσεις. Η χρήση του δεύτερου δείκτη προσφέρει τη δυνατότητα ξεκινώντας από οποιοδήποτε κόμβο της λίστας να μπορούμε να διαβάσουμε τη λίστα και προς τις δυο κατευθύνσεις. Ένα τέτοιο παράδειγμα είναι οι σταθμοί του μετρό.



Εικόνα 1.3.8. Οι σταθμοί του μετρό - Μια διπλά συνδεδεμένη λίστα

Όπως φαίνεται και στην Εικόνα 1.3.8., κάθε κόμβος της διπλά συνδεδεμένης λίστας, συνδέεται με τον αμέσως επόμενο και τον αμέσως προηγούμενο κόμβο της λίστας. Εκτός, βέβαια, από τον αρχικό και τον τελευταίο κόμβο της λίστας.

Σχεδιάστε οδικές μεταβάσεις

Παρατηρείστε τα σχήματα των απλών και διπλών συνδεδεμένων λιστών. Μπορείτε να εντοπίσετε τις διαφορές τους; Σχεδιάστε στη συνέχεια μία διπλά συνδεδεμένη λίστα που θα αποτυπώνει την οδική μετάβαση μεταξύ Αθηνών και Ιωαννίνων μέσω της νέας Ιονίας οδού.

Σε μια 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).

Διαφορές Λίστας σε σχέση με τον Πίνακα – Πλεονεκτήματα – Μειονεκτήματα

Συνοψίζοντας, θα μπορούσε κάποιος πρόχειρα να συμπεράνει ότι η λίστα δε διαφέρει από τον πίνακα. Παρατηρώντας όμως πιο προσεκτικά, είναι εύκολο να εντοπίσει σημαντικές **διαφορές** μεταξύ τους, οι οποίες παρατίθενται παρακάτω:

- 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.
- Ο 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.
- Οι κόμβοι της λίστας αποθηκεύονται σε μη συνεχόμενες θέσεις μνήμης σε αντιδιαστολή με τους πίνακες, όπου τα στοιχεία αποθηκεύονται σε συνεχόμενες θέσεις μνήμης.

Στα **πλεονεκτήματα** των λιστών (έναντι των πινάκων) συγκαταλέγονται τα εξής:

- Το δυναμικό τους μέγεθος,
- η ευκολία εισαγωγής και διαγραφής από οποιοδήποτε μέρος της λίστας, καθώς και
- η μη αναγκαιότητα δήλωσης του μεγέθους τους.

Στα **μειονεκτήματα** των λιστών (έναντι των πινάκων) περιλαμβάνονται τα εξής:

- 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.

Διαφορές στην προσπέλαση μεταξύ δομών δεδομένων

Εντοπίστε τις διαφορές στην προσπέλαση όσον αφορά τη λίστα, τη στοίβα και την ουρά. Για την απάντησή σας σκεφθείτε αν η εισαγωγή ενός στοιχείου σε μία στοίβα μπορεί να γίνει σε οποιαδήποτε θέση της.

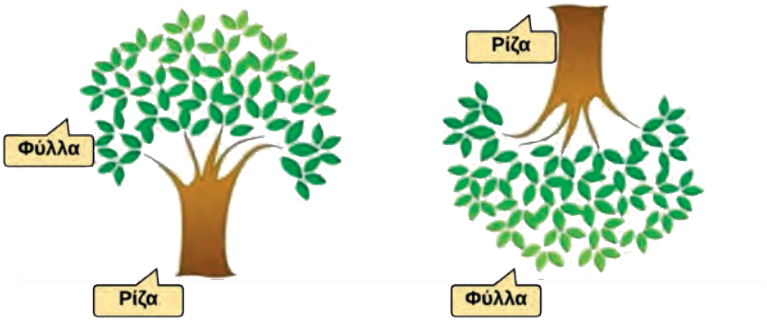
Βασικές πράξεις των συνδεδεμένων λιστών

Οι βασικές πράξεις των συνδεδεμένων λιστών είναι οι παρακάτω:

- 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).

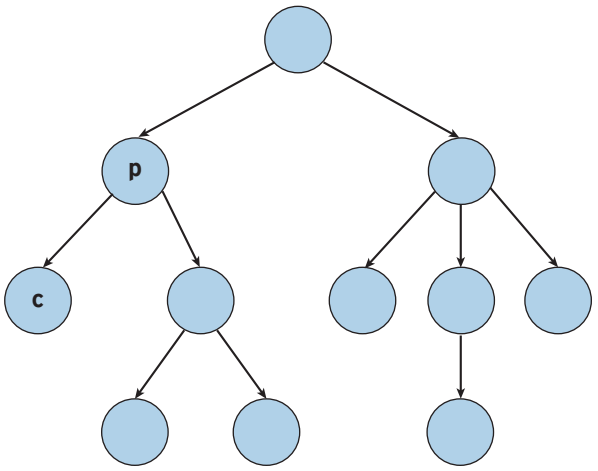
1.3.2 Δένδρα

Μερικές φορές τα πράγματα δεν είναι γραμμικά, όπως έχουμε δει μέχρι τώρα. Σε μία γραμμική δομή, μετά από κάθε στοιχείο ακολουθεί ένα άλλο στοιχείο εκτός και αν είναι το τελευταίο. Τι συμβαίνει όμως στις περιπτώσεις όπου μετά από ένα στοιχείο ακολουθεί όχι ένα, αλλά δύο, τρία ή και περισσότερα στοιχεία; Μπορείτε να φέρετε στον νου σας εικόνες, όπου εμφανίζεται αυτή η συγκεκριμένη συνδεσμολογία των στοιχείων; Μήπως μία από αυτές τις εικόνες είναι το δένδρο; Παρατηρείστε ένα δένδρο στη φύση. Έχει ρίζα και φύλλα, όπως φαίνεται στο αριστερό μέρος της Εικόνας 1.3.9. Η ρίζα είναι στο κάτω μέρος και τα φύλλα στο πάνω μέρος. Στην επιστήμη της Πληροφορικής, όμως, βλέπουμε τα δένδρα ανάποδα. Τα φύλλα είναι στο κάτω μέρος και η ρίζα στο επάνω μέρος, όπως φαίνεται στο δεξιό μέρος της Εικόνας 1.3.9.



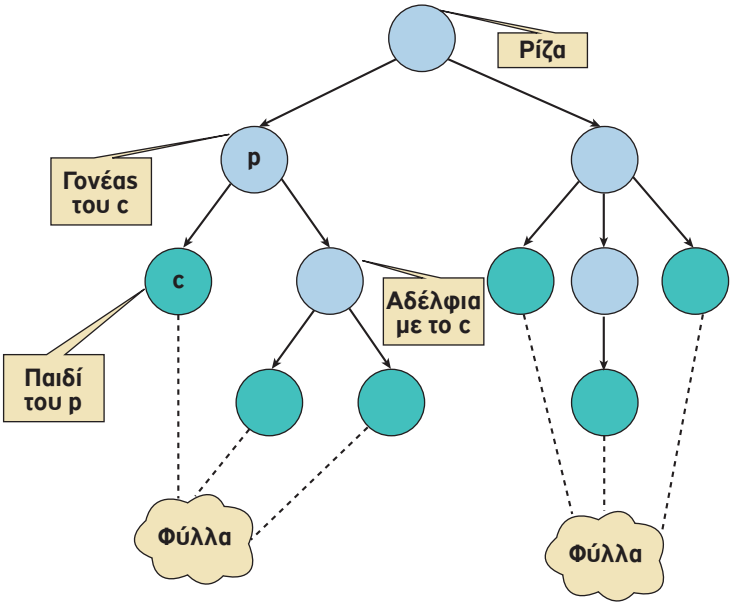
Εικόνα 1.3.9. Τα δένδρα στη φύση και στην Πληροφορική

Ας ξανασχεδιάσουμε το δένδρο που βρίσκεται στο δεξιό μέρος της Εικόνας 1.3.9 με έναν πιο αφαιρετικό τρόπο και στη συνέχεια ας προσπαθήσουμε να ορίσουμε αυτό το οποίο βλέπουμε σε ένα δένδρο, προσδιορίζοντας τη δομή του. Έτσι λοιπόν, θα λέγαμε ότι ένα **δένδρο** αποτελείται από κόμβους, οι οποίοι συνδέονται μεταξύ τους με ακμές. Στην Εικόνα 1.3.10 βλέπουμε τους κόμβους με γαλάζιο χρώμα και τις ακμές ως βέλη μεταξύ των κόμβων.



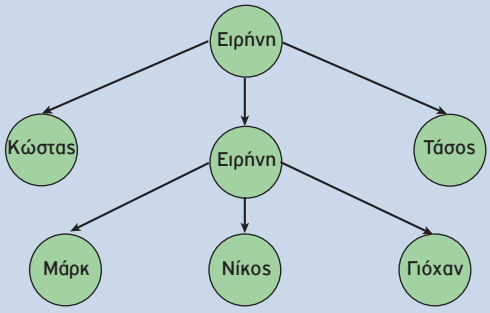
Εικόνα 1.3.10. Δομή ενός δένδρου

Όταν δύο κόμβοι συνδέονται μεταξύ τους με μία ακμή, τότε ονομάζουμε «**γονέα**» τον κόμβο από τον οποίο ξεκινάει η ακμή και «**παιδί**» τον κόμβο στον οποίο καταλήγει η ακμή. Στην Εικόνα 1.3.11 ο κόμβος p είναι γονέας του κόμβου c και ο κόμβος c είναι παιδί του κόμβου p. Ένας κόμβος μπορεί να έχει κανένα, ένα ή περισσότερα παιδιά. Όλοι οι κόμβοι, εκτός από έναν, έχουν ακριβώς έναν γονέα. Ο κόμβος χωρίς γονέα ονομάζεται «**ρίζα**» (root) και βρίσκεται στην κορυφή του δένδρου. Κόμβοι με τον ίδιο γονέα ονομάζονται «**αδέλφια**». Οι κόμβοι χωρίς παιδιά ονομάζονται «**φύλλα**».



Εικόνα 1.3.11. Σχέσεις μεταξύ των κόμβων ενός δένδρου

Οικογενειακές Σχέσεις



Εικόνα 1.3.12. Οικογενειακό δένδρο

Στο δένδρο της Εικόνας 1.3.12 βρείτε ποιος/ποιοι κόμβος/κόμβοι είναι:

- ο γονέας του Τάσου
- τα φύλλα του δένδρου
- η ρίζα του δένδρου
- τα παιδιά της Μαρίας
- ποια είναι αδέλφια

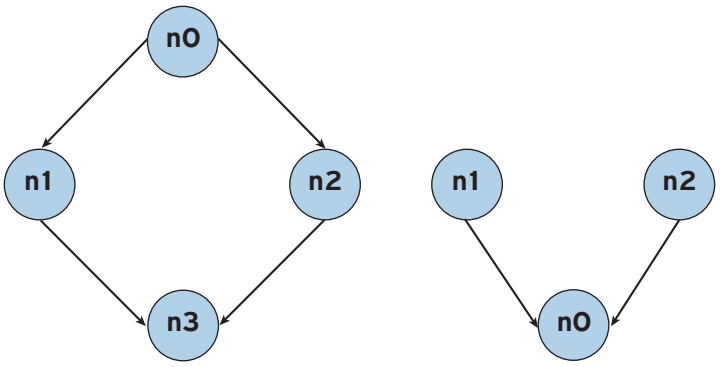
Μπορούμε να έχουμε ένα απλό δένδρο, το οποίο να απαρτίζεται από έναν μόνο κόμβο. Αυτός ο κόμβος είναι **και ρίζα** του απλού αυτού δένδρου, διότι δεν έχει γονέα **και φύλλο**, και διότι δεν έχει παιδιά.

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.

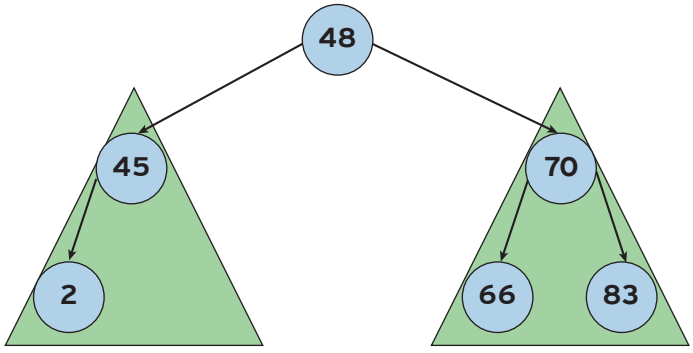
Δένδρο θεωρούμε και το κενό δένδρο, δηλαδή το δένδρο που δεν έχει ούτε κόμβους, ούτε ακμές. Το κενό δένδρο είναι το μόνο δένδρο χωρίς ρίζα.

Στην Εικόνα 1.3.13 παρουσιάζουμε παραδείγματα δομών που δεν είναι δένδρα. Η δομή στα αριστερά δεν είναι δένδρο επειδή ο κόμβος n3 έχει δύο γονείς, τους n1 και n2 και, όπως ξέρουμε, σε ένα δένδρο ένας κόμβος πρέπει να έχει ακριβώς έναν γονέα, με εξαίρεση τη ρίζα, που δεν έχει κανέναν. Επίσης, υπάρχουν δύο διαδρομές από την ρίζα n0 προς τον κόμβο n3, πράγμα που και αυτό απαγορεύεται σε ένα δένδρο. Η δομή στα δεξιά δεν είναι δένδρο διότι υπάρχουν δύο κόμβοι χωρίς γονέα, οι n1 και n2. Είναι σαν να λέμε ότι έχουμε δύο ρίζες στο δένδρο αυτό. Γνωρίζουμε όμως ότι η ρίζα είναι μοναδική σε ένα δένδρο.



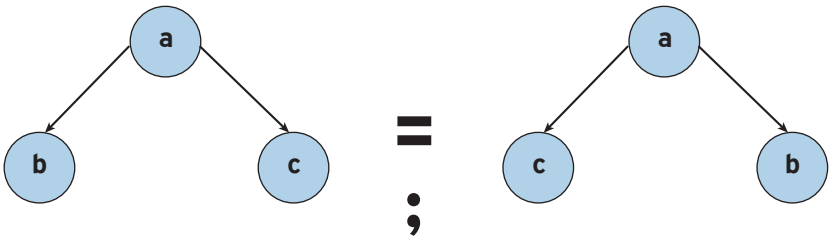
Εικόνα 1.3.13. Δομές που δεν είναι δένδρα

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. Στο δένδρο της Εικόνας 1.3.14, όπως φαίνεται και από το 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.



Εικόνα 1.3.14. Αριστερό και δεξιό υποδένδρο του κόμβου 48

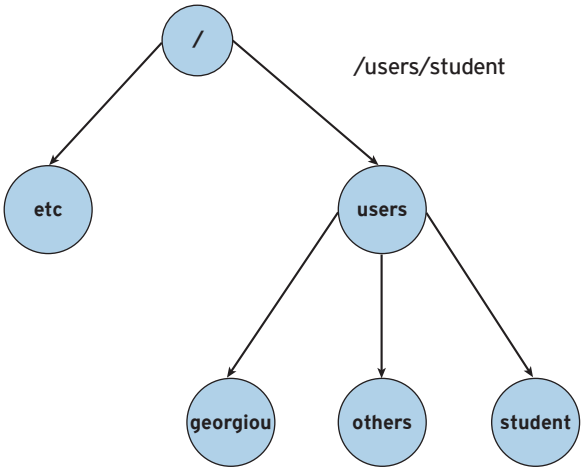
Τα δύο δένδρα της Εικόνας 1.3.15 είναι ίδια ή όχι; Μιλήσαμε για τη σχέση γονέα-παιδιού, αλλά τι γίνεται με τη σχέση μεταξύ των αδελφών; Έχει σημασία η σειρά των αδελφών b και c; Όχι πάντοτε. Για παράδειγμα, αν θέλουμε να μοντελοποιήσουμε την ιεραρχική σχέση των μελών μας οικογένειας και μας ενδιαφέρει 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.



Εικόνα 1.3.15. Πότε δύο δένδρα είναι ίδια

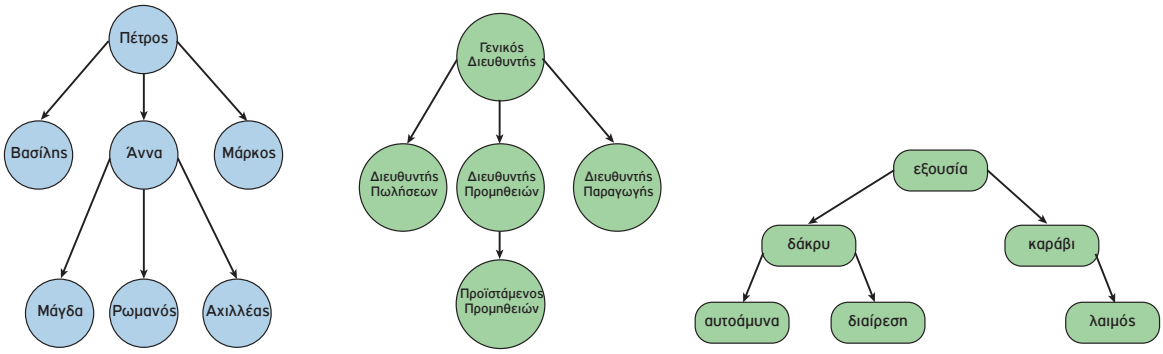
Τα δένδρα είναι μία μη-γραμμική ευέλικτη δομή δεδομένων που χρησιμοποιούνται σε πολλούς τομείς της επιστήμης των υπολογιστών, συμπεριλαμβανομένων των λειτουργικών συστημάτων, των γραφικών, των συστημάτων βάσεων δεδομένων, των παιχνιδιών, της τεχνητής νοημοσύνης και της δικτύωσης υπολογιστών.

Έχετε σκεφτεί γιατί υπάρχουν όμως τόσα δένδρα; Υπάρχουν δύο λόγοι για τους οποίους τα δένδρα είναι τόσο ισχυρά. Ο πρώτος λόγος αναφέρεται στη δυναμικότητα των δένδρων. Είναι πολύ εύκολο να προσθέσετε, να αφαιρέσετε ή να αναζητήσετε ένα στοιχείο σε ένα δένδρο, όπως θα δούμε στη συνέχεια. Ο δεύτερος βασικός λόγος είναι ότι η δομή των δένδρων μεταφέρει πληροφορίες. Για παράδειγμα, ας θεωρήσουμε το σύστημα αρχείων του υπολογιστή, όπως φαίνεται στην Εικόνα 1.3.16. Είναι πολύ εύκολο να προσθέσετε ένα νέο κατάλογο για τον καθηγητή “georgiou”. Επίσης, λόγω του ότι 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”.



Εικόνα 1.3.16. Μη-γραμμική ευέλικτη δομή δεδομένων

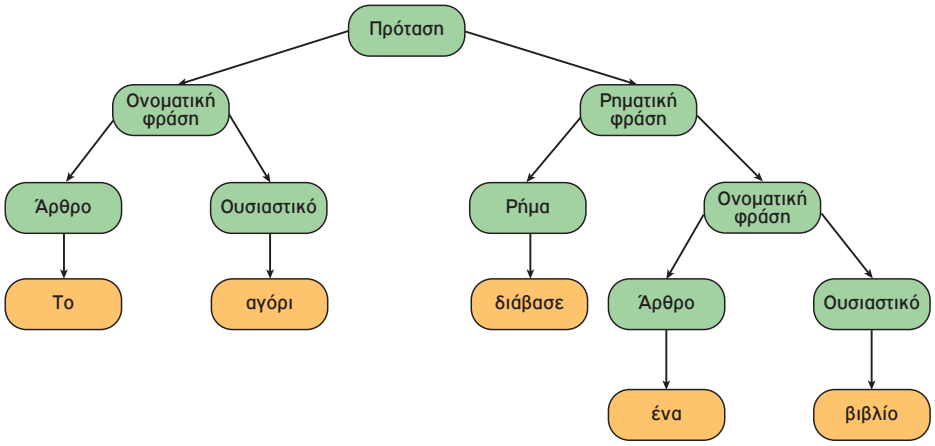
Συνήθως, όταν αναφερόμαστε στα δένδρα, μας έρχονται στο νου διάφορες αναπαραστάσεις δεδομένων του πραγματικού κόσμου αλλά και της Πληροφορικής που διέπονται από ένα είδος φυσικής ιεραρχίας, όπως είναι το οικογενειακό δένδρο (Εικόνα 1.3.17.α), η δομή ενός οργανισμού ή μιας εταιρείας (Εικόνα 1.3 17.β), ο πίνακας περιεχομένων ενός βιβλίου, τα αρχεία και οι φάκελοι ενός υπολογιστή, ένα λεξικό (Εικόνα 1.3.17.γ) (περισσότερα στην ενότητα των δυαδικών δένδρων αναζήτησης), τα μέρη που απαρτίζουν την μηχανή ενός αυτοκινήτου, τα συστατικά μιας πρότασης (Εικόνα 1.3.18) κ.ά.



Εικόνα 1.3.17.

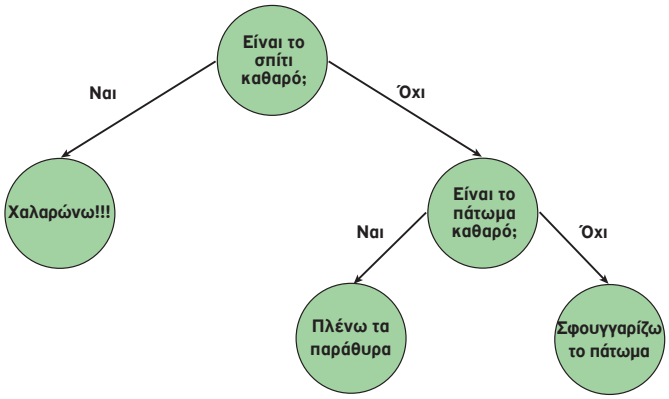
α. Οικογενειακό δένδρο **β. Οργανόγραμμα μιας εταιρείας** **γ. Οργάνωση ενός λεξικού**

Έχετε σκεφτεί, όμως, ότι τα δένδρα, πέρα από μια αποτελεσματική οργάνωση και διαχείριση δεδομένων, αποτελούν τη βάση αρκετών αλγορίθμων επίλυσης προβλήματος, όπως είναι για παράδειγμα η συμπίεση εικόνων, η ταξινόμηση, η αυτόματη συμπλήρωση λέξεων σε συσκευές κινητών τηλεφώνων, η μεταγλώττιση ενός προγράμματος και η λήψη αποφάσεων;
Για παράδειγμα, ένα απλό πρόβλημα λήψης απόφασης μπορεί να είναι αυτό της καθαριότητας του φοιτητικού σπιτιού σας. Κάποια στιγμή θα κληθείτε να πάρετε αποφάσεις λαμβάνοντας υπόψη κάποιους παράγοντες. Μήπως θα σας βοηθούσε ένα δένδρο απόφασης παρόμοιο με αυτό που εμφανίζεται στην Εικόνα 1.3.19; Ειδικότερα, η μεθοδολογία των δένδρων αποφάσεων έχει γίνει ιδιαίτερα δημοφιλής σε περιπτώσεις ιατρικών διαγνώσεων.



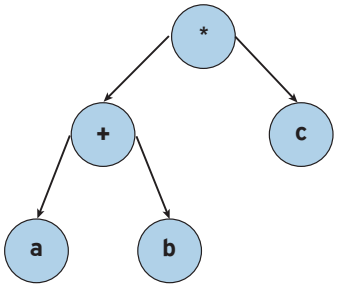
Εικόνα 1.3.18. Η δομή μιας πρότασης

Τα δένδρα απόφασης, όπως πολύ εύκολα μπορείτε να συμπεράνετε από την Εικόνα 1.3.19, είναι δένδρα στα οποία κάθε κόμβος αντιπροσωπεύει ένα χαρακτηριστικό (ιδιότητα), κάθε ακμή αντιπροσωπεύει μια απόφαση (κανόνα) και κάθε φύλλο αντιπροσωπεύει ένα αποτέλεσμα. Στους αλγορίθμους μηχανικής μάθησης (machine learning) τα δένδρα απόφασης έχουν πρωτεύοντα ρόλο.



Εικόνα 1.3.19. Δένδρο Απόφασης

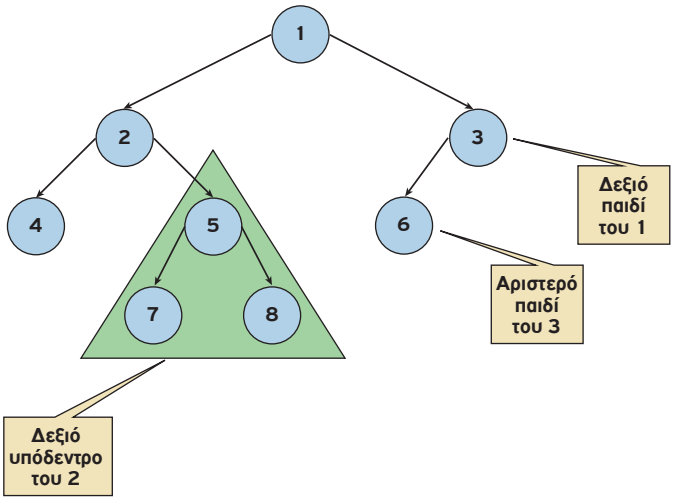
Από την άλλη πλευρά, γνωρίζετε ότι όταν παίζετε παιχνίδια στον υπολογιστή, όπως είναι το σκάκι, η τρίλιζα, το τάβλι και πολλά άλλα, ο υπολογιστής χρησιμοποιεί ένα ειδικό δένδρο, που ονομάζεται δένδρο του παιχνιδιού (game tree), το οποίο μοντελοποιεί όλες τις πιθανές κινήσεις των παικτών για να σας νικήσει; Θα μπορούσε κάθε κόμβος στο δένδρο, που αντιπροσωπεύει μία συγκεκριμένη κατάσταση παιχνιδιού, να περιέχει πληροφορίες σχετικά με το ποιος παίκτης έχει τη μεγαλύτερη πιθανότητα να κερδίσει από οποιαδήποτε πιθανή κίνηση;
Διαδεδομένα είναι επίσης τα δένδρα για την αναπαράσταση και κατ' επέκταση τον υπολογισμό αριθμητικών εκφράσεων, όπως αυτό της Εικόνας 1.3.20.



Εικόνα 1.3.20. (a+b)*c

Δυαδικά Δένδρα

Ένα δυαδικό δένδρο (binary tree) είναι ένα διατεταγμένο δένδρο, στο οποίο κάθε κόμβος έχει το πολύ δύο παιδιά, το αριστερό και το δεξί παιδί. Μπορούμε, συνεπώς, να μιλάμε για αριστερό και δεξιό υποδένδρο ενός κόμβου. Στο δυαδικό δένδρο της Εικόνας 1.3.21, ο κόμβος 3 έχει ως αριστερό υποδένδρο, το δένδρο με μοναδικό κόμβο το 6 και ως δεξιό υποδένδρο, το κενό δένδρο. Προφανώς, αν ανταλλάξουμε το αριστερό με το δεξιό υποδένδρο ενός κόμβου παίρνουμε ένα διαφορετικό δένδρο.



Εικόνα 1.3.21. Δυαδικό δένδρο

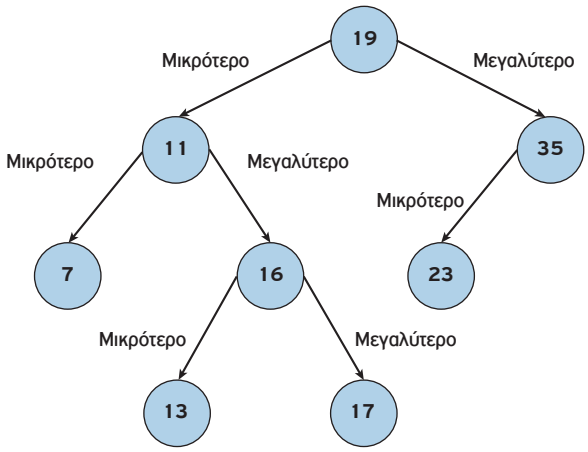
Στην Εικόνα 1.3.21 μπορείτε να βρείτε ποιοι κόμβοι έχουν ένα, δύο ή κανένα παιδί;.

Δυαδικά Δένδρα Αναζήτησης

Γνωρίζετε ότι οι αλγόριθμοι της αναζήτησης, για παράδειγμα όλων των email που έχετε λάβει ή στείλει σε μία συγκεκριμένη περίοδο ή της εύρεσης όλων των λέξεων, που αρχίζουν από την συμβολοσειρά «πληρ», αξιοποιούν μία ειδική κατηγορία δυαδικών δένδρων, αυτών των δυαδικών δένδρων αναζήτησης;

Θα χρησιμοποιήσουμε τώρα τα δυαδικά δένδρα με έναν συγκεκριμένο τρόπο. Θα αποθηκεύσουμε δεδομένα με έναν τρόπο που θα μας επιτρέψει να τα βρίσκουμε πιο εύκολα. Θα διαπιστώσετε και μόνοι σας στη συνέχεια ότι η ιδέα πίσω από ένα δυαδικό δένδρο αναζήτησης είναι παρόμοια με αυτήν της δυαδικής αναζήτησης σε έναν ταξινομημένο πίνακα.

Ένα δυαδικό δένδρο αναζήτησης (binary search tree) 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. Για λόγους απλούστευσης θεωρούμε ότι δεν υπάρχουν τιμές ίσες με την τιμή του κόμβου u. Στην Εικόνα 1.3.22 παρουσιάζεται το παράδειγμα ενός δυαδικού δένδρου αναζήτησης.

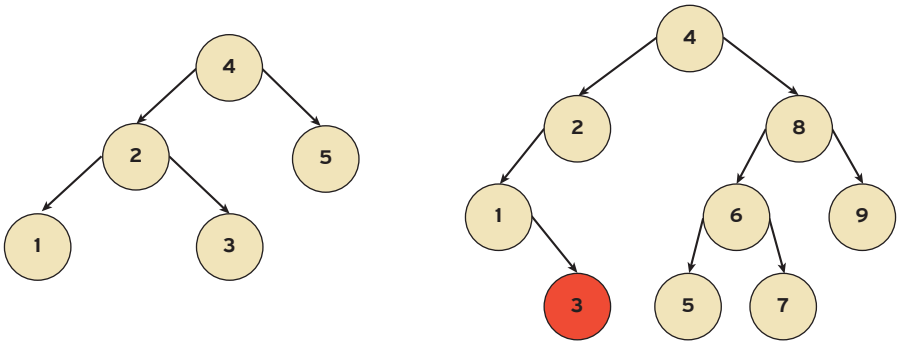


Εικόνα 1.3.22. Δυαδικό δένδρο αναζήτησης

Το δυαδικό δένδρο της Εικόνας 1.3.23.α πληροί τα κριτήρια του δυαδικού δένδρου αναζήτησης, σε αντίθεση με το δυαδικό δένδρο της Εικόνας 1.3.23.β.
Στο δυαδικό δένδρο, της Εικόνας 1.3.23.β, παρατηρούμε τα εξής, όσον αφορά τον κόμβο 3:

- Ας θεωρήσουμε το υποδένδρο με ρίζα το 1. Ο κόμβος 3 βρίσκεται στο δεξιό υποδένδρο αυτού και είναι μεγαλύτερος από τον κόμβο/ρίζα 1.
- Ας θεωρήσουμε το υποδένδρο με ρίζα το 2. Ο κόμβος 3 βρίσκεται στο αριστερό υποδένδρο αυτού αλλά δεν είναι μικρότερος από τον κόμβο/ρίζα 2.

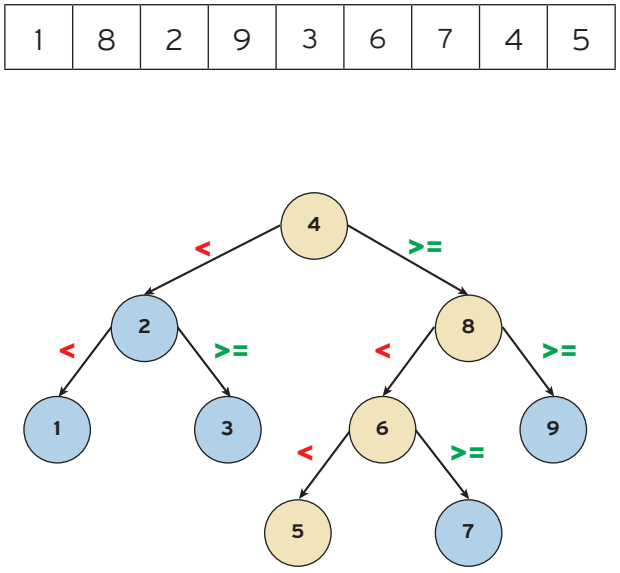
Εξαιτίας του τελευταίου γεγονότος, δηλαδή της ύπαρξης ενός τουλάχιστον κόμβου (στην περίπτωσή μας του κόμβου 3) που δεν πληροί το κριτήριο του δυαδικού δένδρου αναζήτησης, συμπεραίνουμε ότι δεν μπορούμε να κατατάξουμε το δυαδικό αυτό δένδρο στα δυαδικά δένδρα αναζήτησης.



Εικόνα 1.3.23. α: Δυαδικό δένδρο αναζήτησης **β: Μη δυαδικό δένδρο αναζήτησης**

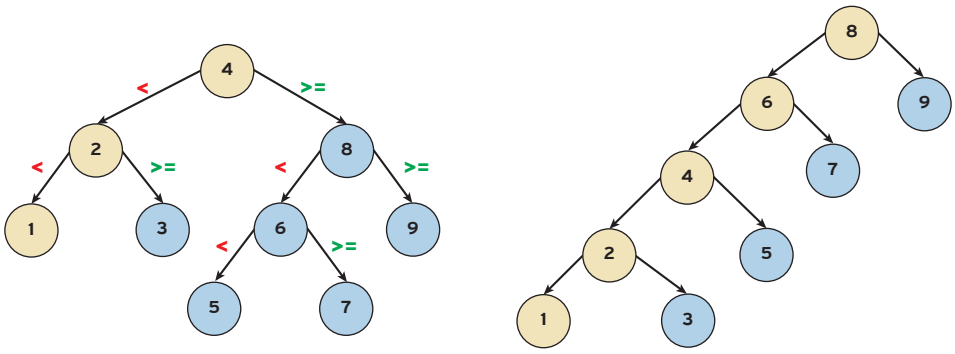
Στην Εικόνα 1.3.23.β, σε ποιο σημείο του δένδρου θα μπορούσε να τοποθετηθεί ο κόμβος 3, έτσι ώστε το δένδρο αυτό να είναι ένα δυαδικό δένδρο αναζήτησης;

Ποιο νομίζετε ότι είναι το πλεονέκτημα των δυαδικών δένδρων αναζήτησης; Το πλεονέκτημα βρίσκεται στο ίδιο το όνομα και συγκεκριμένα στη λέξη «αναζήτηση». Η αναζήτηση για μια συγκεκριμένη τιμή γίνεται ταχύτερα χάρη στον τρόπο αποθήκευσης των τιμών. Για παράδειγμα, ποια είναι η χειρότερη περίπτωση που μπορούμε να συναντήσουμε όταν ψάχνουμε μία τιμή σε μια γραμμική δομή, όπως είναι ο πίνακας; Θεωρείστε ότι ψάχνετε το στοιχείο 5 στον πίνακα της Εικόνας 1.3.24. Στην περίπτωση αυτή, θα χρειαζόταν να διασχίσουμε ολόκληρο τον πίνακα για να βρούμε το στοιχείο 5. Τώρα, θεωρείστε το δένδρο της Εικόνας 1.3.24. Μετά από πόσες συγκρίσεις θα βρείτε το στοιχείο 5;
Ας δούμε τον αλγόριθμο της αναζήτησης σε ένα δυαδικό δένδρο αναζήτησης μέσα από μερικά παραδείγματα. Αρχικά, συγκρίνετε τη ρίζα του δένδρου (δηλαδή το 4) με το υπό αναζήτηση στοιχείο (δηλαδή το 5). Ο αριθμός που αναζητάτε είναι μεγαλύτερος από τη ρίζα. Συνεπώς συνεχίζετε με το δεξιό υποδένδρο και αγνοείτε το αριστερό υποδένδρο. Στη συνέχεια, συγκρίνετε το 8 με το 5. Επειδή το 8 είναι μεγαλύτερο από το 5, συνεχίζετε με το αριστερό υποδένδρο. Κατόπιν, συναντάτε το 6 που είναι και αυτό μεγαλύτερο από το 5. Μεταβαίνετε, λοιπόν, στο αριστερό υποδένδρο του 6, όπου βρίσκετε το στοιχείο 5 που αναζητάτε. Στην περίπτωση που δεν υπάρχει το στοιχείο στο δένδρο (για παράδειγμα αν ψάχναμε το στοιχείο 12), η αναζήτηση καταλήγει σε ένα κενό υποδένδρο και εκεί τερματίζει η όλη διαδικασία. Με βάση τον αλγόριθμο αυτό, μειώνουμε δραματικά τον χρόνο για να βρούμε το στοιχείο που ψάχνουμε και αυτό διότι περιορίζουμε αισθητά τους κόμβους τους οποίους επισκεπτόμαστε. Κάθε φορά αφήνουμε στην άκρη ένα υποδένδρο και συνεχίζουμε με το άλλο.



Εικόνα 1.3.24. Αλγόριθμος αναζήτησης σε ένα δυαδικό δένδρο αναζήτησης

Θεωρείστε ότι αναζητούμε το στοιχείο 1 στα δυαδικά δένδρα αναζήτησης της Εικόνας 1.3.25. Σε ποιο από τα δύο δυαδικά δένδρα αναζήτησης θα βρούμε πιο γρήγορα την τιμή 1; Μήπως στο πρώτο, διότι σε κάθε βήμα απορρίπτουμε όσο περισσότερους κόμβους είναι δυνατόν; Οπτικά, θα μπορούσαμε να θεωρήσουμε ότι το πρώτο δένδρο είναι πιο «ισορροπημένο» σε σχέση με το δεύτερο; Παρατηρούμε, λοιπόν, χωρίς να αναφερθούμε διεξοδικά στα ισορροπημένα δένδρα, ότι αν θέλουμε να έχουμε γρήγορους αλγόριθμους αναζήτησης πρέπει να αποθηκεύουμε τις τιμές στα δυαδικά δένδρα αναζήτησης με έναν συγκεκριμένο τρόπο.

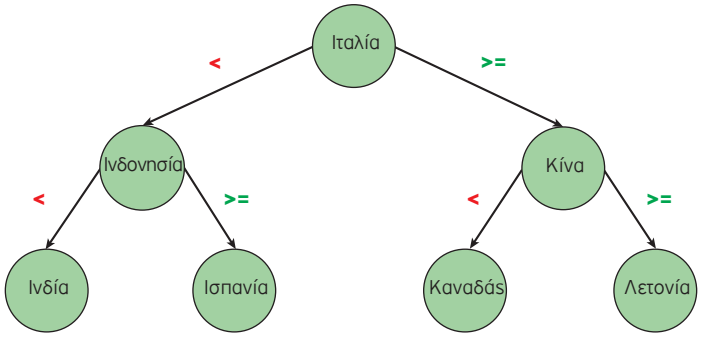


Εικόνα 1.3.25. Η σημασία της δομής ενός δυαδικού δένδρου αναζήτησης στην εύρεση στοιχείων

Σας θυμίζει ο αλγόριθμος αυτός τον αλγόριθμο της δυαδικής αναζήτησης που έχετε συναντήσει στους ταξινομημένους πίνακες; Ο αλγόριθμος της δυαδικής αναζήτησης «τρέχει» γρήγορα στους ταξινομημένους πίνακες σε αντίθεση με τους αλγορίθμους της εισαγωγής και της διαγραφής στοιχείου που είναι πιο χρονοβόροι και αυτό διότι πρέπει να διατηρηθεί η υπάρχουσα διάταξη των στοιχείων. Ας μην ξεχνάμε επίσης ότι οι πίνακες είναι στατικές δομές και επομένως, δεν έχουν δυνατότητα αυξομείωσης του μεγέθους τους. Αυτό σημαίνει ότι κάθε επόμενη εισαγωγή στοιχείου θα οδηγήσει ενδεχομένως σε αντιγραφή όλων των στοιχείων του πίνακα σε έναν μεγαλύτερο πίνακα.

Τα δυαδικά δένδρα αναζήτησης συνδυάζουν τα πλεονεκτήματα των λιστών, όσον αφορά τις πράξεις της εισαγωγής και της διαγραφής, αλλά και τα πλεονεκτήματα των ταξινομημένων πινάκων, όσον αφορά την πράξη της αναζήτησης. Φανταστείτε ότι μετατρέπετε έναν ταξινομημένο πίνακα σε ένα δυαδικό δένδρο, όπως παρουσιάζεται στην Εικόνα 1.3.26. Αυτό μας δίνει τη δυνατότητα να αναζητήσουμε ένα στοιχείο το ίδιο γρήγορα όσο και σε έναν ταξινομημένο πίνακα αλλά και να εισαγάγουμε και να διαγράψουμε εύκολα ένα στοιχείο ακριβώς επειδή δουλεύουμε με δένδρα και όχι με πίνακες.

Ινδία	Ινδονησία	Ισπανία	Ιταλία	Καναδάς	Κίνα	Λετονία



Εικόνα 1.3.26. Αναζήτηση σε ταξινομημένο πίνακα και σε δυαδικό δένδρο αναζήτησης

1.3.3 Γράφοι

Τώρα που μελετήσαμε τις δενδρικές δομές δεδομένων, προκύπτει το ερώτημα «από πού προέρχονται τα δένδρα;». Τα δένδρα είναι στην πραγματικότητα ένα υποσύνολο των γράφων. Αλλά προκειμένου να καταλάβετε πραγματικά γιατί χρησιμοποιούμε γράφους και τι είναι αυτοί, θα πρέπει να αναφερθούμε στη θεωρία γράφων.

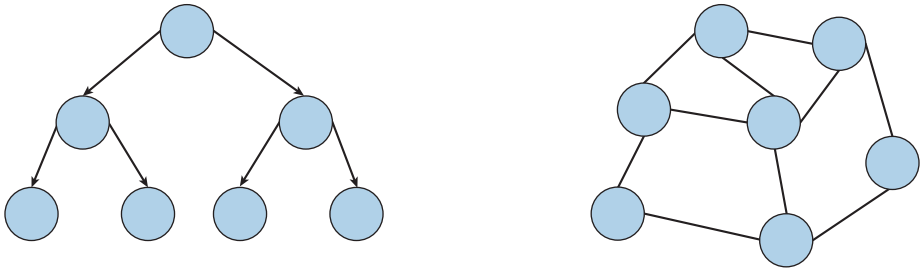
Είδαμε ότι το πιο θεμελιώδες χαρακτηριστικό των μη γραμμικών δομών είναι ότι τα δεδομένα τους δεν ακολουθούν μια σειρά - όπως στους πίνακες ή τις συνδεδεμένες λίστες. Τα δένδρα, όπως είδαμε, ξεκινούν με έναν κόμβο ρίζας και μπορεί να συνδεθούν με άλλους κόμβους, κάτι που σημαίνει ότι θα μπορούσαν να περιέχουν δευτερεύοντα δένδρα στο εσωτερικό τους. Τα δένδρα, γενικά, διέπονται από συγκεκριμένους κανόνες ενώ σε ορισμένους τύπους δένδρων ισχύουν ιδιαίτεροι κανόνες, όπως στα δυαδικά δένδρα αναζήτησης, στα οποία οι κόμβοι μπορεί να έχουν μόνο δύο συνδέσεις με δύο κόμβους ανά πάσα στιγμή.

Αλλά τι θα γίνει αν αγνοήσουμε αυτούς τους κανόνες; Τότε, δεν αναφερόμαστε σε δένδρα αλλά σε μία νέα δυναμική δομή δεδομένων, που ονομάζεται γράφος. Τα δένδρα δεν είναι παρά περιορισμένοι τύποι γράφων. Ένα δένδρο θα είναι πάντα ένα γράφος, αλλά δεν είναι όλοι οι γράφοι δένδρα.

Τι είναι αυτό που κάνει ένα δένδρο διαφορετικό από έναν γράφο;

Ένα δένδρο μπορεί μόνο να ρέει προς μία κατεύθυνση, από τον κόμβο ρίζας σε κόμβους φύλλων ή κόμβους παιδιών. Ένα δένδρο μπορεί να έχει μόνο μονόδρομες συνδέσεις - ένας κόμβος παιδιού μπορεί να έχει μόνο έναν γονέα και ένα δένδρο δεν μπορεί να έχει βρόχους ή κυκλικούς δεσμούς (Εικόνα 1.3.27.α).

Με τους γράφους, όλοι αυτοί οι περιορισμοί δεν υπάρχουν. Οι γράφοι δεν έχουν την έννοια ενός κόμβου «ρίζας». Οι κόμβοι μπορούν να συνδεθούν με οποιονδήποτε τρόπο. Για παράδειγμα, ένας κόμβος μπορεί να συνδεθεί με άλλους πέντε (Εικόνα 1.3.27.β)! Οι γράφοι, επίσης, δεν έχουν «μονοκατευθυντική» ροή - αντ' αυτού, μπορεί να έχουν κατεύθυνση ή να μην έχουν καμιά κατεύθυνση.



Εικόνα 1.3.27.

α: Δένδρο **β: Γράφος**

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.

Τύποι Γράφων

Ας εξετάσουμε τους δύο τύπους γράφων που είναι αρκετά εύκολο να εντοπιστούν και είναι αρκετά συνηθισμένοι στα προβλήματα θεωρίας γράφων: οι **κατευθυνόμενοι** γράφοι και οι **μη κατευθυνόμενοι** γράφοι.

Όπως έχουμε αναφέρει, σε έναν γράφο δεν υπάρχουν πραγματικοί κανόνες για τον τρόπο που ένας κόμβος συνδέεται με έναν άλλο κόμβο. Οι ακμές μπορούν να συνδέσουν τους κόμβους με οποιονδήποτε τρόπο. Οι διαφορετικοί τύποι ακμών είναι πολύ σημαντικοί όταν πρόκειται για την αναγνώριση και τον καθορισμό του τύπου των γράφων. Ως επί το πλείστον, οι γράφοι μπορούν να έχουν δύο τύπους ακμών: ακμές που έχουν κατεύθυνση (κατευθυνόμενες ακμές) και ακμές που δεν έχουν κατεύθυνση (μη κατευθυνόμενες ακμές). Σε μια κατευθυνόμενη ακμή, δύο κόμβοι συνδέονται με πολύ συγκεκριμένο τρόπο.

Στην εικόνα 1.3.28.α, όπου ο κόμβος Α συνδέεται με τον κόμβο Β, υπάρχει ένας μόνο τρόπος να ταξιδέψουμε μεταξύ αυτών των δύο κόμβων και αυτός είναι από τον κόμβο Α στον κόμβο Β. Είναι πολύ συνηθισμένο να αναφέρουμε τον κόμβο από τον οποίο ξεκινάμε ως προέλευση και τον κόμβο στον οποίο ταξιδεύουμε ως προορισμό. Σε μια κατευθυνόμενη ακμή, μπορούμε να ταξιδέψουμε μόνο από την προέλευση στον προορισμό, και ποτέ το αντίστροφο (Εικόνα 1.3.28.α). Ωστόσο, σε μια μη κατευθυνόμενη ακμή, η διαδρομή μεταξύ των δύο κόμβων είναι αμφίδρομη, πράγμα που σημαίνει ότι οι κόμβοι προέλευσης και προορισμού δεν είναι σταθεροί.

Αυτή η διαφοροποίηση είναι πραγματικά πολύ σημαντική, επειδή οι ακμές σε έναν γράφο καθορίζουν το πώς αποκαλείται ο γράφος.

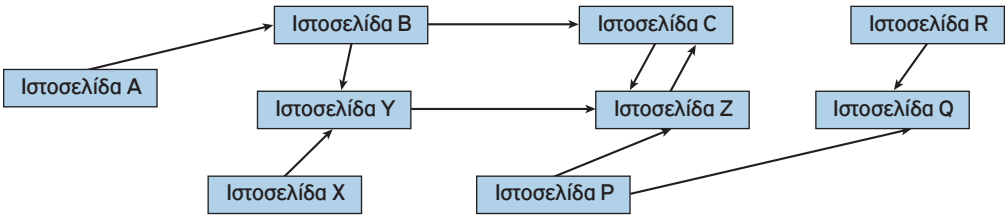
α: Κατευθυνόμενη ακμή **β: μη κατευθυνόμενη ακμή**

Εικόνα 1.3.28.

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).**

Οι γράφοι είναι γύρω μας, απλά δεν τους βλέπουμε. Ο Παγκόσμιος Ιστός (WWW) είναι ένας τεράστιος γράφος (Εικόνα 1.3.29)! Όταν κάνουμε κλικ ανάμεσα σε ιστότοπους και πλοηγούμαστε μεταξύ των διευθύνσεων URL, κάνουμε πραγματικά περιήγηση σε έναν γράφο. Μερικές φορές αυτές οι αναπαραστάσεις γράφων έχουν κόμβους με ακμές που είναι μη κατευθυνόμενες - μπορούμε να προχωρήσουμε από μια ιστοσελίδα σε μια άλλη και το αντίστροφο - και άλλες ακμές που κατευθύνονται – μπορούμε να πάμε μόνο από την ιστοσελίδα Α στην **ιστοσελίδα Β, και ποτέ το αντίστροφο.**



Εικόνα 1.3.29. Παγκόσμιος Ιστός – Η ιστοσελίδα R περιέχει σύνδεσμο προς την ιστοσελίδα Q

Κοινωνικά Δίκτυα - Facebook

Υπάρχει, όμως, ένα ακόμη καλύτερο παράδειγμα που απεικονίζει τις καθημερινές μας αλληλεπιδράσεις και η υλοποίησή του στηρίζεται στη δομή δεδομένων «γράφος»: τα κοινωνικά δίκτυα.

Το Facebook, ένα τεράστιο κοινωνικό δίκτυο, είναι ένας τύπος γράφου. Και αν σκεφτούμε περισσότερο το πώς λειτουργεί πραγματικά, αρχίζουμε να κατανοούμε καλύτερα τον τρόπο διασυνδέσεων των κόμβων καθώς και με ποιον τύπο γράφου αναπαριστάνεται. Στο Facebook, εάν σε προσθέσω σαν φίλο, πρέπει να αποδεχτείς το αίτημά μου. Δεν είναι δυνατόν να είμαι φίλος σου στο δίκτυο χωρίς να είσαι και δικός μου. Η σχέση μεταξύ δύο χρηστών (κόμβοι), είναι αμφίδρομη. Δεν υπάρχει η έννοια κόμβου «προέλευσης» και «προορισμού» - αντ' αυτού, είσαι φίλος μου και είμαι δικός σου. Ποιον τύπο γράφου θα χρησιμοποιήσουμε για την αναπαράσταση του κοινωνικού δικτύου Facebook;

Κοινωνικά Δίκτυα - Twitter

Α) Το Twitter, από την άλλη πλευρά, λειτουργεί πολύ διαφορετικά από το Facebook. Μπορώ να σε ακολουθήσω, αλλά δεν απαιτείται να με ακολουθήσεις. Με ποιον τύπο γράφου μπορούμε να αναπαραστήσουμε το Twitter;

Θα μπορούσαμε να αναπαραστήσουμε το Twitter ως κατευθυνόμενο γράφο; Κάθε ακμή που δημιουργούμε αντιπροσωπεύει μια μονόδρομη σχέση. Όταν με ακολουθείς στο Twitter, δημιουργείται μια ακμή στον γράφο με τον λογαριασμό σου ως κόμβο προέλευσης και τον λογαριασμό μου ως τον κόμβο προορισμού;

Β) Τι θα συμβεί εάν αποφασίσω να σε ακολουθήσω και εγώ; Αλλάζω την κατεύθυνση της ακμής που δημιουργήθηκε όταν με ακολούθησες; Ξαφνικά η ακμή γίνεται αμφίδρομη και επομένως μη κατευθυνόμενη;

Επτά γέφυρες του Königsberg

Υπάρχει μια ενδιαφέρουσα ιστορία πίσω από την προέλευσή του προβλήματος «οι Επτά γέφυρες του Königsberg». Η πρόκληση (ή απλά σπαζοκεφαλιά) με τις γέφυρες του Königsberg, ήταν να μπορεί κάποιος να διασχίσει και τις επτά γέφυρες μόνο μία φορά περπατώντας μέσα στην πόλη. Κοιτάξτε πρώτα την Εικόνα 1.3.30 για να αντιληφθείτε το πρόβλημα:

Εικόνα 1.3.30. Οι επτά γέφυρες του Königsberg

Προσπαθήστε και δείτε αν μπορείτε να περπατήσετε μέσα από την πόλη και να διασχίσετε όλες τις γέφυρες μόνο μία φορά. Θα πρέπει να λάβετε υπόψη δύο πράγματα ενώ προσπαθείτε να λύσετε το παραπάνω πρόβλημα-αίνιγμα:

- πρέπει να διασχίσετε όλες τις γέφυρες
- δεν πρέπει να διασχίσετε περισσότερο από μία φορά κάποια γέφυρα

Μπορείτε να δοκιμάσετε οποιονδήποτε αριθμό συνδυασμών θέλετε, αλλά θα δείτε ότι είναι αδύνατον. Δεν υπάρχει κανένας τρόπος με τον οποίο κάποιος μπορεί να περπατήσει μέσα στην πόλη και να διασχίσει κάθε γέφυρα μόνο μία φορά. Ο Leonhard Euler αναπαράστησε αυτό το πρόβλημα σε γράφο, όπου αναπαράστησε τα τμήματα γης (A,B,C,D) με κορυφές και τις γέφυρες με ακμές.
Αναζητήστε στο Διαδίκτυο πως ο Euler αναπαράστησε το πρόβλημα με γράφο και για ποιο λόγο είναι αδύνατο να λυθεί αυτό το πρόβλημα.

1.3.4 Ερωτήσεις - Ασκήσεις

Ε.1: Δώστε δύο παραδείγματα εφαρμογών από την καθημερινή ζωή

- απλά συνδεδεμένης λίστας
- διπλά συνδεδεμένης λίστας

Ε.2: Βρείτε το σωστό.

Χαρακτηρίστε τις παρακάτω προτάσεις ως Σωστές ή Λάθος. Στην περίπτωση που πιστεύετε ότι είναι λανθασμένες δικαιολογήστε την επιλογή σας.

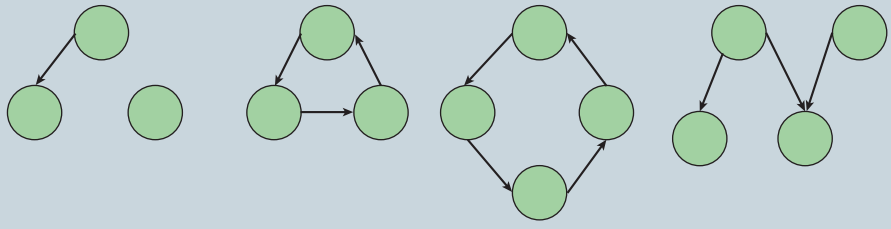
α/α	Προτάσεις	Σ	Λ
1	Μια απλά συνδεδεμένη λίστα μπορούμε να τη διατρέξουμε και προς τις δύο κατευθύνσεις.	❒	❒
2	Σε μία λίστα δε χρειάζεται να οριστεί ένα αρχικό μέγεθος.	❒	❒
3	Δεν είναι δυνατό να υπάρχει «τυχαία» πρόσβαση σε μια απλά συνδεδεμένη λίστα.	❒	❒
4	Σε μια λίστα, τα στοιχεία δεν μπορούν να προστεθούν ή να αφαιρεθούν από τη μέση της λίστας, παρά μόνο από την αρχή ή το τέλος της.	❒	❒
5	Στη διπλά συνδεδεμένη λίστα τα περιεχόμενα των κόμβων προσπελαύνονται και από τις δύο κατευθύνσεις.	❒	❒

Ε.3: Σχεδιάστε γεωγραφικές ιεραρχίες

Σχεδιάστε ένα δένδρο που θα αποτυπώνει την ιεραρχία μιας γεωγραφικής περιοχής σε επίπεδο χωρών, νομών και πόλεων.

Ε.4: Δένδρα ή Γράφοι;

Στην Εικόνα 1.3.31 ποιες από τις παρακάτω δομές είναι δένδρα και ποιες είναι γράφοι. Προσπαθήστε να εξηγήσετε το γιατί.



α. _____ β. _____ γ. _____ δ. _____

Εικόνα 1.3.31. Εντοπίστε τα δένδρα και τους γράφους

Ε.5: Επιλέξτε τη σωστή απάντηση.

1. Ποια από τις βασικές δομές δεδομένων είναι η πιο κατάλληλη για να αναπαραστήσετε τη δομή των καταλόγων, των υποκαταλόγων και των αρχείων στον σκληρό σας δίσκο:

- ❒ πίνακας
- ❒ λίστα
- ❒ δένδρο
- ❒ ουρά
- ❒ στοίβα

2. Ο κατάλογος των φοιτητών που εγγράφονται σε ένα μάθημα είναι ταξινομημένος αλφαβητικά με βάση το ονοματεπώνυμο και περιλαμβάνει ένα σύνολο πληροφοριών σχετικών με τον φοιτητή, όπως είναι ο κωδικός του φοιτητή, η ημερομηνία γέννησης, το φύλο, η διεύθυνση, ο αριθμός τηλεφώνου κ.λπ. Επιλέξτε ποια από τις παρακάτω δομές δεδομένων είναι καταλληλότερη για την αναπαράσταση αυτών των πληροφοριών:

- ❒ στοίβα
- ❒ δένδρο
- ❒ λίστα
- ❒ ουρά

Ε.6: Εγκυκλοπαίδεια

Οι πληροφορίες σε μια εγκυκλοπαίδεια μπορούν να οργανωθούν σε ένα σύνολο θεματικών κατηγοριών, όπως Θετικές Επιστήμες, Ιστορία, Τέχνες κ.λπ. Κάθε θεματική κατηγορία μπορεί με την σειρά της να υποδιαιρείται σε μια άλλη σειρά θεματικών υποκατηγοριών σε διάφορα επίπεδα και υποεπίπεδα *με τα άρθρα να βρίσκονται στη βάση της δομής*. Για παράδειγμα, η κατηγορία Θετικές Επιστήμες μπορεί να περιλαμβάνει τις υποκατηγορίες Μαθηματικά, Φυσική, Χημεία, Πληροφορική κ.λπ.

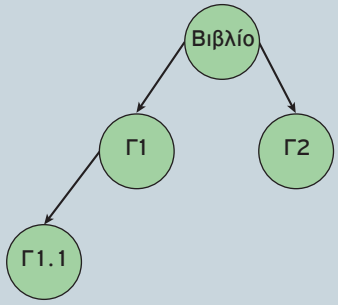
Επιλέξτε ποια από τις παρακάτω δομές δεδομένων είναι καταλληλότερη για την αναπαράσταση αυτών των πληροφοριών:

i) δένδρο **ii)** λίστα **iii)** πίνακας

Ε.7: Περιεχόμενα βιβλίου

Στην Εικόνα 1.3.32 παρουσιάζεται αριστερά η δομή των περιεχομένων ενός βιβλίου και δεξιά ένα ημιτελές δένδρο, το οποίο αναπαριστά εν μέρει την δομή του βιβλίου αυτού. Προσπαθήστε να συμπληρώσετε το δένδρο αυτό, ώστε να απεικονίζει την δομή του βιβλίου στο σύνολό της.

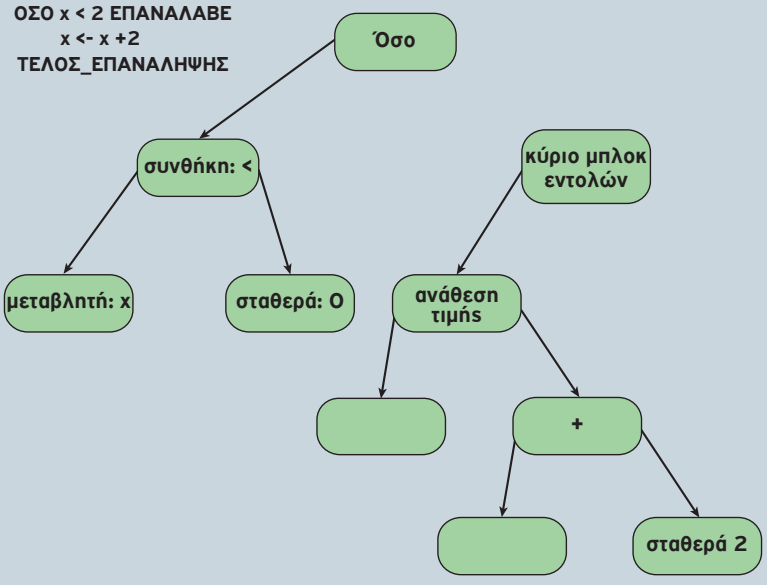
Βιβλίο
- Γ1
  - Γ1.1
  - Γ1.2
- Γ2
  - Γ2.1
    - Γ2.1.1
    - Γ2.1.2
  - Γ2.2
  - Γ2.3
- Γ3



Εικόνα 1.3.32. Περιεχόμενα βιβλίου **Διαγραμματική αναπαράσταση**

Ε.8: Δένδρο για την αναπαράσταση κώδικα

Προσπαθήστε να έρθετε στη θέση του μεταγλωττιστή του υπολογιστή σας και να μετατρέψετε το πρόγραμμα της Εικόνας 1.3.33 στο αντίστοιχο δένδρο. Εμείς το κάναμε για εσάς αλλά παραλείψαμε να δώσουμε περιεχόμενο σε κάποιους κόμβους. Μπορείτε να συμπληρώσετε το περιεχόμενο εσείς;



Εικόνα 1.3.33. Δομή τμήματος προγράμματος

Ε.9: Βρείτε το σωστό.

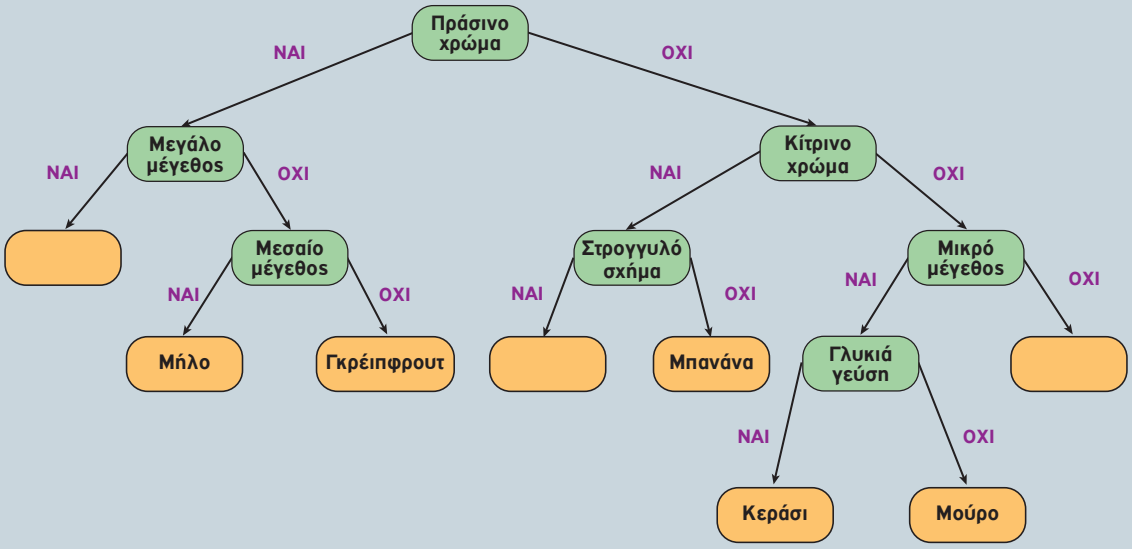
Χαρακτηρίστε τις παρακάτω προτάσεις ως Σωστές ή Λάθος. Στην περίπτωση που πιστεύετε ότι είναι λανθασμένες δικαιολογήστε την επιλογή σας.

α/α	Προτάσεις	Σ	Λ
1	Η ρίζα ενός δένδρου δεν μπορεί ποτέ να είναι φύλλο.	❒	❒
2	Σε ένα δυαδικό δένδρο, φύλλα συναντάμε μόνο στο αριστερό υποδένδρο.	❒	❒
3	Σε ένα δυαδικό δένδρο, κάθε κόμβος-γονέας μπορεί να έχει το πολύ δύο παιδιά	❒	❒
4	Δεν είναι δυνατό να υπάρχουν δύο διαφορετικές διαδρομές από την ρίζα προς έναν άλλον κόμβο ενός δένδρου.	❒	❒
5	Σε ένα δυαδικό δένδρο, κάθε κόμβος έχει μηδέν, ένα ή δύο υποδένδρα.	❒	❒
6	Η ρίζα ενός δένδρου είναι ο μόνος κόμβος ενός δένδρου που δεν έχει γονέα.	❒	❒
7	Τα φύλλα ενός δένδρου είναι απομονωμένοι κόμβοι που δε συνδέονται με άλλους κόμβους.	❒	❒
8	Σε ένα δένδρο, κάθε κόμβος-γονέας μπορεί να έχει οποιονδήποτε αριθμό παιδιών	❒	❒
9	Μπορούν να υπάρχουν διαφορετικές δομές δυαδικών δένδρων αναζήτησης που αποθηκεύουν τα ίδια στοιχεία.	❒	❒
10	Κάθε δένδρο είναι γράφος	❒	❒

Ε.10: Δένδρο απόφασης

Το δένδρο απόφασης της Εικόνας 1.3.34 αποτελεί μια προσπάθεια κατηγοριοποίησης μιας σειράς φρούτων, όπως το καρπούζι, το μήλο, το γκρέιπφρουτ, το λεμόνι, η μπανάνα, το κεράσι, το μούρο και το πορτοκάλι, με βάση τα χαρακτηριστικά τους, όπως το χρώμα, το μέγεθος, το σχήμα και η γεύση.

Κάποια φρούτα, όμως, όπως το καρπούζι, το πορτοκάλι και το λεμόνι δεν έχουν κατηγοριοποιηθεί ακόμα. Προσπαθήστε να τα ταξινομήσετε ονοματίζοντας τα κενά φύλλα με τις λέξεις καρπούζι, πορτοκάλι και λεμόνι.

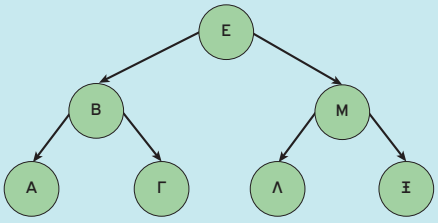


Εικόνα 1.3.34. Κατηγοριοποίηση φρούτων

Ε.12: Δεξιά ή αριστερά;

Στο δυαδικό δένδρο αναζήτησης της Εικόνας 1.3.35, αφού συγκρίνετε το στοιχείο Π (το στοιχείο που προσπαθείτε να βρείτε) με το στοιχείο Ε (τη ρίζα) και βρείτε ότι δεν είναι ίσα, σε ποια κατεύθυνση θα το αναζητήσετε στη συνέχεια;

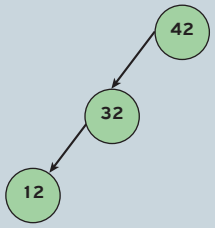
- ❒ Στα αριστερά του Ε
- ❒ Στα δεξιά του Ε
- ❒ Δεν μπορείτε να πείτε και πρέπει να δοκιμάσετε και προς τις δύο κατευθύνσεις.



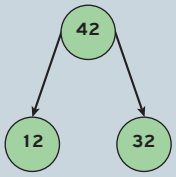
Εικόνα 1.3.35. Δυαδική αναζήτηση

Ε.11: Χαμένοι στο δάσος των δυαδικών δένδρων

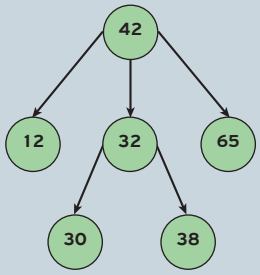
Ποια από τα παρακάτω δένδρα είναι δυαδικά δένδρα αναζήτησης. Εξηγείστε το γιατί:



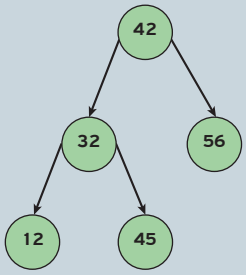
α. _____



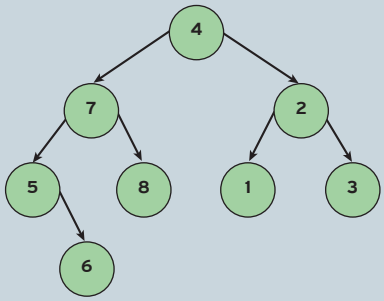
β. _____



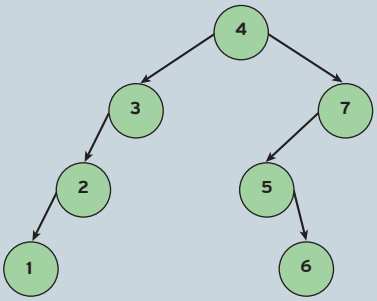
γ. _____



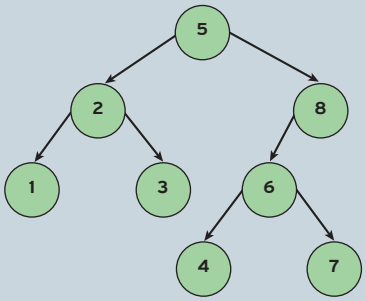
δ. _____



α. _____



β. _____



γ. _____

Ε.13: Τηλεφωνικό Κέντρο Ταξί

Ας υποθέσουμε ότι έχετε τηλεφωνήσει για να έρθει να σας πάρει ένα ταξί. Ένα από τα πιο σημαντικά πράγματα, που είναι κρίσιμα για τη λειτουργία του κέντρου που καλείτε, είναι η ικανότητά του να συνδυάζει τους οδηγούς με τους επιβάτες με αποτελεσματικό τρόπο. Σκεφτείτε ότι υπάρχουν 6 πιθανά ταξί τα οποία μπορούν να έρθουν εγκαίρως να σας παραλάβουν. Πώς θα επιλέξει το κέντρο να σας παραχωρήσει ένα ταξί; Μπορούμε να χρησιμοποιήσουμε γράφο για να δούμε πώς μπορεί να πραγματοποιηθεί η διαδικασία επιλογής του κατάλληλου ταξί. Το πρώτο ταξί (Α) απέχει 12, το δεύτερο (Β) 5, το τρίτο (Γ) 10, το τέταρτο (Δ) 7, το πέμπτο (Ε) 13 και το έκτο (ΣΤ) 20 χιλιόμετρα.

i) Να σχεδιάσετε τον γράφο που αναπαριστά το πρόβλημα (πάνω στις ακμές γράψτε τα χιλιόμετρα).

ii) Ποιον τύπο γράφου χρησιμοποιήσατε;

iii) Ποιο ταξί θα επιλέξει το κέντρο; Θεωρείστε ότι το μόνο κριτήριο για την επιλογή είναι η χιλιομετρική απόσταση.

Ενότητα 2

Τεχνικες Σχεδιασης Αλγοριθμων

Μέθοδος Διαίρει και Βασίλευε

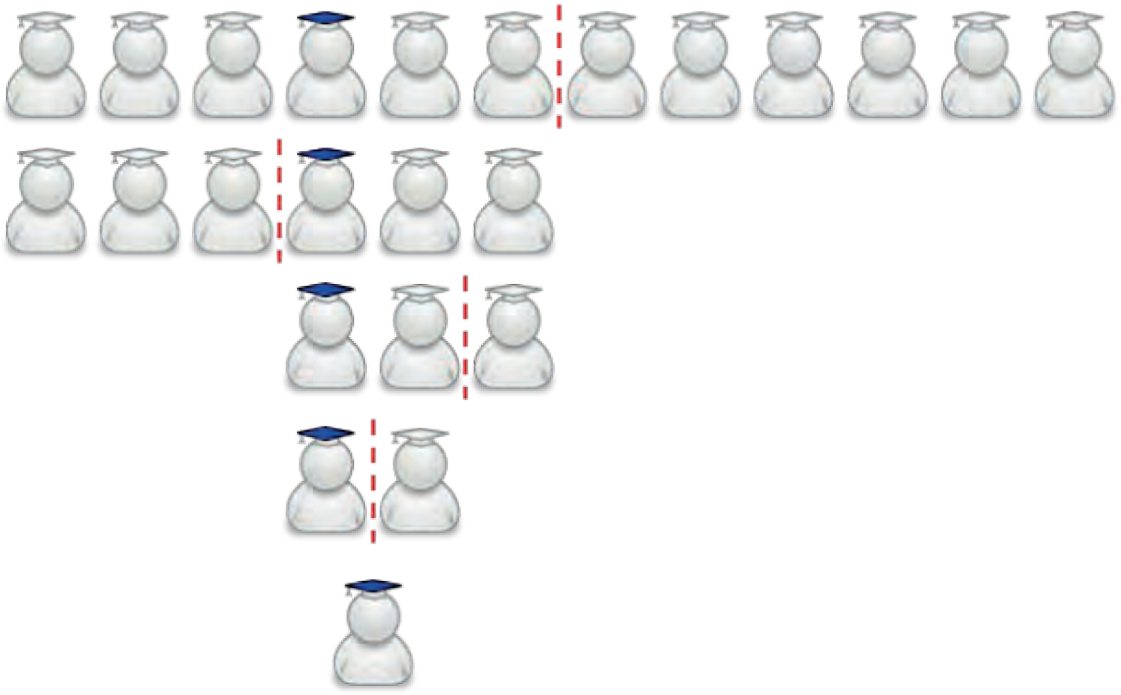
Ενότητα 2. Τεχνικές Σχεδίασης Αλγορίθμων

2.1 Μέθοδος Διαίρει και Βασίλευε

Σε μια κατασκήνωση, μια ομάδα 13 παιδιών αποφασίζει να παίξει το παιχνίδι «βρες το δαχτυλίδι». Σύμφωνα με τους κανόνες, ένα από τα παιδιά (στο εξής Παιδί-1) δίνει ένα δαχτυλίδι στα υπόλοιπα 12 και γυρνάει το σώμα του, δίχως να έχει οπτική επαφή με τα υπόλοιπα. Ένα παιδί κρύβει το δαχτυλίδι στην τσέπη του και όλοι μαζί κάθονται σε μια γραμμή γνωρίζοντας ποιος έχει κρύψει το δαχτυλίδι. Το Παιδί-1 που δε βλέπει τα υπόλοιπα, ισχυρίζεται ότι θα μαντέψει πάρα πολύ γρήγορα πού βρίσκεται το δαχτυλίδι με την προϋπόθεση ότι κάποιο παιδί (εκπρόσωπος) θα απαντά στις ερωτήσεις που θα του κάνει με ένα ναι ή ένα όχι.

Το Παιδί-1 ζητά από τα μισά παιδιά της γραμμής να κάνουν ένα βήμα μπροστά. Στη συνέχεια, ρωτά τον εκπρόσωπο εάν το δαχτυλίδι βρίσκεται στο δεύτερο «μισό» της γραμμής, που είναι ένα βήμα πιο πίσω. Όταν μάθει σε ποιο από τα δύο «μισά» βρίσκεται το δαχτυλίδι, ζητά από τα παιδιά που βρίσκονται στο άλλο «μισό» να βγουν εκτός παιχνιδιού. Η ίδια διαδικασία επαναλαμβάνεται με το «μισό» που βρίσκεται ακόμη στο παιχνίδι, μέχρι να μείνουν μόνο δύο παιδιά, όπου θα γίνει και η τελευταία ερώτηση από το Παιδί-1. Επομένως με τέσσερις ερωτήσεις, το Παιδί-1 εντόπισε το παιδί που έχει κρυμμένο το δαχτυλίδι.

Στο παρακάτω σχήμα παρουσιάζεται η περιγραφόμενη διαδικασία κατά την οποία το παιδί με το μπλε καπέλο έχει το δαχτυλίδι και η διακεκομμένη κόκκινη γραμμή σηματοδοτεί το σημείο, όπου κάθε φορά (σε κάθε επανάληψη) μειώνεται η γραμμή των παιδιών στο μισό.



Εικόνα 1. Σχηματική αναπαράσταση του παιχνιδιού βρες το δαχτυλίδι

Όλα τα παιδιά εντυπωσιάστηκαν από την ταχύτητα με την οποία το Παιδί-1 βρήκε το δαχτυλίδι και προσπαθούν να ανακαλύψουν τον συστηματικό τρόπο με τον οποίο σκέφτηκε.
Το μόνο που ήξεραν ήταν ότι το Παιδί-1 χώρισε αρχικά το σύνολο των παιδιών σε δύο «μισά» (ίσες ομάδες) και όταν ανακάλυπτε σε ποιο μισό βρισκόταν το δαχτυλίδι, επαναλάμβανε την ίδια διαδικασία μέχρι τελικά να απομείνουν δύο παιδιά που υποχρεωτικά το ένα είχε το δακτυλίδι. Για να μάθουν περισσότερα έψαξαν στο Διαδίκτυο και εντόπισαν ότι το παιχνίδι στο οποίο συμμετείχαν, αποτελεί συγκεκριμένη μέθοδο επίλυσης προβλημάτων, που εφαρμόζεται για τη γρήγορη αναζήτηση δεδομένων σε μια υπάρχουσα δομή δεδομένων και ονομάζεται «Διαίρει και Βασίλευε».

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).

Προσπαθήστε να εφαρμόσετε τη μέθοδο «Διαίρει και Βασίλευε» στην παρακάτω δραστηριότητα.

Το κάλπικο νόμισμα

Είστε εκτιμητής χρυσών νομισμάτων. Κάποιος σας φέρνει 128 χρυσά νομίσματα και σας λέει ότι ένα από αυτά είναι κάλπικο. Το κάλπικο νόμισμα είναι πανομοιότυπο στην εμφάνιση με τα υπόλοιπα, αλλά επειδή περιέχει λιγότερη ποσότητα χρυσού είναι λίγο πιο ελαφρύ. Έχετε στη διάθεσή σας μια ζυγαριά ακριβείας με δύο δίσκους. Πώς θα εντοπίσετε το κάλπικο νόμισμα με όσο το δυνατό λιγότερα ζυγίσματα;

Η μέθοδος σχεδίασης αλγορίθμων «Διαίρει και Βασίλευε» μπορεί να αποδοθεί με τα επόμενα βήματα:

1. Δίνεται για επίλυση ένα στιγμιότυπο ενός προβλήματος.
2. Το στιγμιότυπο του προβλήματος υποδιαιρείται σε υπο-στιγμιότυπα του ίδιου προβλήματος.
3. Δίνεται ανεξάρτητη λύση σε κάθε ένα υπο-στιγμιότυπο.
4. Συνδυάζονται όλες οι μερικές λύσεις που βρέθηκαν για τα υπο-στιγμιότυπα, έτσι ώστε να δοθεί η συνολική λύση του προβλήματος.

Χρησιμοποιώντας τη μέθοδο «Διαίρει και Βασίλευε» δοκιμάστε να παίξετε με έναν συμμαθητή ή μία συμμαθήτρια σας το παιχνίδι «Μάντεψε τον αριθμό», ως εξής: Σκεφθείτε έναν αριθμό από το 1 έως το 100 και ζητήστε από τον συμμαθητή ή τη συμμαθήτρια σας (παίκτης-1) να μαντέψει τον αριθμό αυτόν με όσο το δυνατόν λιγότερες προσπάθειες.

- Ποια διαδικασία πρέπει να ακολουθήσει ο παίκτης-1 για να βρει τον αριθμό; Τι είδους ερωτήσεις πρέπει να κάνει, ώστε να ελαχιστοποιήσει τις προσπάθειες που θα χρειαστούν;
- Ποιος είναι ο μέγιστος αριθμός προσπαθειών που θα χρειαστεί ο παίκτης-1 για να μαντέψει τον αριθμό;

Επαναλάβετε το ίδιο παιχνίδι με έναν αριθμό από το 1 έως το 1000.

- Ποιος είναι τώρα ο μέγιστος αριθμός προσπαθειών που θα χρειαστεί ο παίκτης-1 για να μαντέψει τον αριθμό;
- Τι παρατηρείτε σχετικά με την αύξηση του αριθμού των προσπαθειών σε σχέση με την αύξηση του πλήθους των δεδομένων;
- Με βάση τις παρατηρήσεις σας, θεωρείτε ότι η τεχνική «Διαίρει και Βασίλευε» είναι αποτελεσματική για μεγάλο όγκο δεδομένων;

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 $[\log_2(n)+1]$ (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 $\log_2(n)$.
Για παράδειγμα, σε ένα σύνολο 100 ταξινομημένων στοιχείων (n=100), ο μέγιστος αριθμός συγκρίσεων (επαναλήψεων) είναι: $[\log_2(100)+1]=[6,643856+1]=[7,643856]=7$.

Ένας κλασικός αλγόριθμος που ακολουθεί τη φιλοσοφία της μεθόδου «Διαίρει και Βασίλευε» είναι η «Δυαδική αναζήτηση», η οποία εφαρμόζεται μόνο στην περίπτωση ταξινομημένου συνόλου στοιχείων (π.χ. οι αριθμοί από το 1 έως το 100, τα στοιχεία ενός ταξινομημένου μονοδιάστατου πίνακα κ.ά.) και θα αξιοποιηθεί στο επόμενο παράδειγμα.

Παράδειγμα 1 – Μάντεψε τον αριθμό (παιχνίδι με αντίπαλο τον υπολογιστή)

Ένα παιδί παίζει με τον υπολογιστή το παιχνίδι «Μάντεψε τον αριθμό». Οι κανόνες του παιχνιδιού είναι οι εξής:

- Το παιδί αποτυπώνει στο μυαλό του έναν αριθμό από το 1 έως το 100.
- Ο υπολογιστής προσπαθεί να μαντέψει τον αριθμό το πολύ σε 7 προσπάθειες $[\log_2(100)+1]=[6,643856+1]=[7,643856]=7$
- Κάθε φορά που ο υπολογιστής προτείνει έναν αριθμό, με κατάλληλο μήνυμα στην οθόνη, ρωτά το παιδί να του απαντήσει, μέσω του πληκτρολογίου, αν ο αριθμός που μάντεψε ο υπολογιστής, είναι αυτός που έχει βάλει το παιδί στο μυαλό του ή αν είναι μεγαλύτερος ή μικρότερος.

Να αναπτύξετε πρόγραμμα που να υλοποιεί το παραπάνω παιχνίδι:

1. Ο υπολογιστής με κατάλληλο μήνυμα σας ζητάει να σκεφτείτε έναν ακέραιο αριθμό από το 1 μέχρι το 100.
2. Ο υπολογιστής εμφανίζει κατάλληλο μήνυμα που σας πληροφορεί ότι θα βρει τον αριθμό το πολύ με 7 προσπάθειες.
3. Ο υπολογιστής με κατάλληλο μήνυμα προτείνει έναν ακέραιο αριθμό και στη συνέχεια (μέσω κατάλληλου μηνύματος) ζητάει να πληκτρολογήσετε αν ο αριθμός αυτός είναι ίδιος, μεγαλύτερος ή μικρότερος από τον αριθμό που είχατε σκεφτεί.
4. Όταν ο υπολογιστής μαντέψει τον αριθμό που σκεφθήκατε, εμφανίζει στην οθόνη κατάλληλο μήνυμα με τον αριθμό αυτό, καθώς και τον αριθμό των προσπαθειών που έκανε μέχρι να τον βρει.

Απάντηση:

Ανάλυση

- Το πρόγραμμα ζητάει από το παιδί να σκεφθεί έναν ακέραιο αριθμό από το 1 έως το 100 προκειμένου να τον μαντέψει το πολύ σε 7 προσπάθειες (« **ΓΡΑΨΕ** 'Σκέψου έναν ακέραιο αριθμό από το 1 μέχρι το 100 και θα τον μαντέψω το πολύ σε 7 προσπάθειες αρκεί να απαντάς ειλικρινά στις ερωτήσεις μου: '»).
- Αναζήτηση του αριθμού: Ο πρώτος αριθμός που «μαντεύει» το πρόγραμμα (ο υπολογιστής) είναι το 50 (ο μεσαίος αριθμός από το 1-100, «μεση **<-** (αρχη_ **+** τελος**)** **DIV** **2**») και το παιδί καλείται να πληκτρολογήσει ως απάντηση στο αντίστοιχο ερώτημα που εμφανίζεται στην οθόνη («**ΓΡΑΨΕ** ' Είναι ο αριθμός ', μεση, '; '»), («**ΓΡΑΨΕ** 'Δώσε Ν(ΝΑΙ) ή Ο(ΟΧΙ): '»), αν είναι ο αριθμός που σκέφτηκε.
- Το παιδί πληκτρολογεί «Ν» ή «Ο». Η τιμή αυτή καταχωρείται στη μεταβλητή «απ» («**ΔΙΑΒΑΣΕ** απ»).
- Στην περίπτωση που ο υπολογιστής βρήκε τον αριθμό («**ΑΝ** απ **=** 'Ν' **Η** απ **=** 'ν' **ΤΟΤΕ ...**»), «βρέθηκε **<-** **ΑΛΗΘΗΣ ...**»), τυπώνεται στην οθόνη, ως απάντηση του υπολογιστή, κατάλληλο μήνυμα («**ΓΡΑΨΕ** 'Τον βρήκα σε ', προσ, ' προσπάθεια/ες...'»).

- Στην περίπτωση που ο υπολογιστής δε βρήκε τον αριθμό («**ΑΛΛΙΩΣ ...**»), το παιδί καλείται, με αντίστοιχο μήνυμα που εμφανίζεται στην οθόνη, να δηλώσει αν ο αριθμός που σκέφτηκε είναι μεγαλύτερος ή μικρότερος από τον αριθμό που του εμφάνισε ο υπολογιστής («**ΓΡΑΨΕ** 'Ο αριθμός που έβαλες είναι (1) μεγαλύτερος ή (2) μικρότερος. Δώσε απάντηση 1 ή 2: '») («**ΔΙΑΒΑΣΕ** απαντηση»). Με αυτόν τον τρόπο ο υπολογιστής λαμβάνει την πληροφορία ότι ο αριθμός που ψάχνει είναι στο «κάτω μισό» του πλήθους των αριθμών [1-49] (αν το παιδί πληκτρολογήσει το 1) ή στο «πάνω μισό» του πλήθους των αριθμών [51-100] (αν το παιδί πληκτρολογήσει το 2). Ο υπολογιστής σε κάθε προσπάθεια αφαιρεί το μισό πλήθος των αριθμών (είτε το «πάνω μισό» είτε το «κάτω μισό») και στην επόμενη προσπάθεια (επόμενη επανάληψη), θα ψάξει να βρει τον αριθμό, μεταξύ αυτών που έχουν απομείνει στο μισό πλήθος αριθμών κάθε φορά. Για παράδειγμα, αν το παιδί πληκτρολογήσει 1 (δηλ. ο αριθμός που έβαλε είναι μικρότερος του 50), ο επόμενος αριθμός που θα μαντέψει ο υπολογιστής θα είναι το 25. Αν το παιδί πληκτρολογήσει 2 (δηλ. ο αριθμός που έβαλε είναι μεγαλύτερος του 50), ο επόμενος αριθμός που θα μαντέψει ο υπολογιστής θα είναι το 75, κ.ο.κ.
  Επομένως, ο υπολογιστής χρησιμοποιεί τη μέθοδο «Διαίρει και Βασίλευε», αξιοποιώντας έναν σίγουρο και σύντομο τρόπο για την εύρεση του αριθμού, μέσω αμφίδρομης επικοινωνίας (διαδοχικές ερωταπαντήσεις) με το παιδί.

Ακολουθεί η υλοποίηση του προγράμματος σε ΓΛΩΣΣΑ.

Κώδικας σε ΓΛΩΣΣΑ [2.1]

```
1   ΠΡΟΓΡΑΜΜΑ μάντεψε_τον_αριθμό
2   ΜΕΤΑΒΛΗΤΕΣ
3     ΑΚΕΡΑΙΕΣ: προσ, αρχη_, τελος, μεση, απαντηση
4     ΛΟΓΙΚΕΣ: βρεθηκε
5     ΧΑΡΑΚΤΗΡΕΣ: απ
6   ΑΡΧΗ
7     ΓΡΑΨΕ 'Σκέψου έναν ακέραιο αριθμό από το 1 μέχρι το 100'
8     ΓΡΑΨΕ ' και θα τον μαντέψω το πολύ σε 7 προσπάθειες'
9     ΓΡΑΨΕ ' αρκεί να απαντάς ειλικρινά στις ερωτήσεις μου: '
10    ΓΡΑΨΕ
11    αρχη_<- 1
12    τελος <- 100
13    προσ <- 0
14    βρεθηκε <- ΨΕΥΔΗΣ
```

```
15  ΟΣΟ αρχη_<=τελος ΚΑΙ βρεθηκε=ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
16      προσ <- προσ + 1
17      μεση <- (αρχη_ + τελος) div 2
18      ΓΡΑΨΕ 'Προσπάθεια ', προσ, 'η'
19      ΓΡΑΨΕ ' Είναι ο αριθμός ', μεση, '; '
20      ΓΡΑΨΕ 'Δώσε Ν(ΝΑΙ) ή Ο(ΟΧΙ):'
21      ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
22        ΔΙΑΒΑΣΕ απ
23        ΑΝ απ<>'Ν' ΚΑΙ απ<>'ν' ΚΑΙ απ<>'Ο' ΚΑΙ απ<>'ο' ΤΟΤΕ
24          ΓΡΑΨΕ 'Λάθος απάντηση. Ξαναπροσπάθησε....'
25        ΤΕΛΟΣ_ΑΝ
26      ΜΕΧΡΙΣ_ΟΤΟΥ απ='Ν' Η απ='ν' Η απ='Ο' Η απ='ο'
27      ΑΝ απ='Ν' Η απ ='ν' ΤΟΤΕ
28        βρεθηκε <- ΑΛΗΘΗΣ
29        ΓΡΑΨΕ 'Τον βρήκα σε ',προσ,' προσπάθεια/ες...'
30      ΑΛΛΙΩΣ
31        ΓΡΑΨΕ 'Ο αριθμός που έβαλες είναι '
32        ΓΡΑΨΕ '(1)μεγαλύτερος ή (2)μικρότερος...'
33        ΓΡΑΨΕ 'Δώσε απάντηση 1 ή 2: '
34        ΔΙΑΒΑΣΕ απαντηση
35        ΑΝ απαντηση = 1 ΤΟΤΕ
36          αρχη_<- μεση + 1
37        ΑΛΛΙΩΣ
38          τελος <- μεση - 1
39        ΤΕΛΟΣ_ΑΝ
40      ΤΕΛΟΣ_ΑΝ
41      ΓΡΑΨΕ '_____'
42    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
43    ΑΝ προσ > 7 Η αρχη_ > τελος ΤΟΤΕ
44      ΓΡΑΨΕ 'Δε βρήκα τον αριθμό σε 7 προσπάθειες'
45      ΓΡΑΨΕ 'γιατί δεν είσαι ειλικρινής ή  '
46      ΓΡΑΨΕ 'έκανες κάτι λάθος στη διαδικασία '
47      ΓΡΑΨΕ ' που συμφωνήσαμε'
48    ΤΕΛΟΣ_ΑΝ
49  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ μάντεψε_τον_αριθμό
```

Ενότητα 3

Επιλογη και Επαναληψη

Εντολή ΕΠΙΛΕΞΕ

Ενότητα 3. Επιλογή και επανάληψη

3.1 Εντολή ΕΠΙΛΕΞΕ

Υπάρχουν προβλήματα με πολλές εναλλακτικές περιπτώσεις επιλογής. Στις περιπτώσεις αυτές μπορούμε να χρησιμοποιήσουμε την εντολή **ΕΠΙΛΕΞΕ**, η οποία εκφράζει τη δομή της πολλαπλής επιλογής.

Γενική μορφή της εντολής ΕΠΙΛΕΞΕ

```
ΕΠΙΛΕΞΕ <έκφραση>
 ΠΕΡΙΠΤΩΣΗ <λίστα_τιμών_1>
  <εντολές_1>
  ΠΕΡΙΠΤΩΣΗ <λίστα_τιμών_2>
  <εντολές_2>
  .....
  ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
  <εντολές_αλλιώς>
ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
```

Όπου:

- **<έκφραση> :**
  είναι μια μεταβλητή, η τιμή της οποίας θα ελεγχθεί με τις τιμές που δίνονται στις ΠΕΡΙΠΤΩΣΕΙΣ και ανάλογα σε ποια ΠΕΡΙΠΤΩΣΗ ανήκει θα εκτελεστούν οι αντίστοιχες εντολές **ή** η πράξη, που υπολογίζει την τιμή της.
  Δηλαδή, η **<έκφραση>** μπορεί να είναι:
  - Μεταβλητή
  - Αριθμητική πράξη
  - Συγκριτική πράξη

- **<λίστα_τιμών_Ν>:**
  οι τιμές που μπορεί να πάρει μια έκφραση. Οι τιμές αυτές μπορεί να είναι διακριτές τιμές, περιοχή τιμών από...έως ή να υπακούν σε μια συνθήκη.

Τρόπος εκτέλεσης

Κατά την εκτέλεση της εντολής υπολογίζεται η τιμή της έκφρασης και στη συνέχεια εκτελούνται οι εντολές που ανήκουν στην αντίστοιχη περίπτωση τιμών. Στην περίπτωση που η τιμή 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.

3.1.1 Παραδείγματα με χρήση της εντολής ΕΠΙΛΕΞΕ

Παράδειγμα 1 – Κωδικός καταστημάτων

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει τον κωδικό ενός καταστήματος και να εμφανίζει την πόλη στην οποία ανήκει. Τα καταστήματα της Αθήνας έχουν τους κωδικούς 1, 2, 3, 4 και τα καταστήματα της Θεσσαλονίκης έχουν τους κωδικούς 5 και 6. Αν δώσετε κάποιον άλλον αριθμό, να εμφανίζεται το μήνυμα «Δεν υπάρχει αυτός ο κωδικός καταστήματος».

Απάντηση:

Ανάλυση

- Δίνεται ακέραιος μονοψήφιος αριθμός από το πληκτρολόγιο.
- Γίνεται έλεγχος εάν ο αριθμός αντιστοιχεί σε κωδικό καταστήματος της Αθήνας (1, 2, 3, 4) ή της Θεσσαλονίκης (5, 6). Είναι προτιμότερο να χρησιμοποιηθεί η εντολή ΕΠΙΛΕΞΕ αντί της ΑΝ...ΑΛΛΙΩΣ_ΑΝ... λόγω της συμπαγούς δομής της.
- Το πρόγραμμα τυπώνει κατάλληλο μήνυμα.

Ακολουθεί η υλοποίηση του προγράμματος σε ΓΛΩΣΣΑ.

Κώδικας σε ΓΛΩΣΣΑ [3.1]

```
1   ΠΡΟΓΡΑΜΜΑ Πόλη_καταστήματος
2   ΜΕΤΑΒΛΗΤΕΣ
3     ΑΚΕΡΑΙΕΣ: κωδικός
4   ΑΡΧΗ
5     ΓΡΑΨΕ ' Δώσε τον κωδικό του καταστήματος:'
6     ΔΙΑΒΑΣΕ κωδικός
7     ΕΠΙΛΕΞΕ κωδικός
8         ΠΕΡΙΠΤΩΣΗ 1, 2, 3, 4
9             ΓΡΑΨΕ 'Αθήνα'
10      ΠΕΡΙΠΤΩΣΗ 5, 6
11            ΓΡΑΨΕ 'Θεσσαλονίκη'
12        ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
13            ΓΡΑΨΕ ' Δεν υπάρχει αυτός ο κωδικός καταστήματος.
14    ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
15  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Πόλη_καταστήματος
```

Παράδειγμα 2 – Άρτιος ή περιττός αριθμός

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει έναν ακέραιο αριθμό και στη συνέχεια να τυπώνει αν ο αριθμός είναι άρτιος ή περιττός.

Απάντηση:

Ανάλυση

- Δίνεται ένας ακέραιος αριθμός από το 1 μέχρι το 1000 (από το πληκτρολόγιο).
- Γίνεται έλεγχος αν ο αριθμός είναι άρτιος χρησιμοποιώντας τη πράξη x MOD 2 (είναι πολλές οι διακριτές τιμές για να ελεγχθούν). Αν το αποτέλεσμα είναι μηδέν, τότε ο αριθμός είναι άρτιος, ενώ αν είναι ένα, τότε ο αριθμός είναι περιττός. Είναι προτιμότερο να χρησιμοποιηθεί η εντολή ΕΠΙΛΕΞΕ αντί της ΑΝ...ΑΛΛΙΩΣ_ΑΝ... λόγω της συμπαγούς δομής της στον προγραμματισμό.
- Το πρόγραμμα τυπώνει κατάλληλο μήνυμα.

Ακολουθεί η υλοποίηση του προγράμματος σε ΓΛΩΣΣΑ.

Κώδικας σε ΓΛΩΣΣΑ [3.2]

```
1   ΠΡΟΓΡΑΜΜΑ άρτιος_περιττός
2   ΜΕΤΑΒΛΗΤΕΣ
3     ΑΚΕΡΑΙΕΣ: χ
4   ΑΡΧΗ
5     ΓΡΑΨΕ 'Δώσε ένα αριθμό από το 1 μέχρι το 1000: '
6     ΔΙΑΒΑΣΕ χ
7     ΕΠΙΛΕΞΕ χ MOD 2
8         ΠΕΡΙΠΤΩΣΗ 0
9             ΓΡΑΨΕ 'Άρτιος'
10        ΠΕΡΙΠΤΩΣΗ 1
11            ΓΡΑΨΕ 'Περιττός'
12    ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
13  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ άρτιος_περιττός
```

Παράδειγμα 3 – Τέλη κυκλοφορίας αυτοκινήτων

Η εφορία κάθε τέλος του έτους φορολογεί τα αυτοκίνητα ανάλογα με τον κυβισμό τους, σύμφωνα με τον παρακάτω πίνακα:

Κυβισμός	Φόρος
0 έως 1000	100€
1001 έως 1299	120€
1300 έως 1800	250€
1801 και άνω	600€

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει τον κυβισμό του αυτοκινήτου, να υπολογίζει τον φόρο που του αναλογεί και να τυπώνει το αντίστοιχο ποσό.

Απάντηση:

Ανάλυση

- Δίνεται από το πληκτρολόγιο ακέραιος αριθμός, που αντιστοιχεί στα κυβικά ενός οχήματος.
- Γίνεται έλεγχος των κυβικών ανάλογα με την κατηγορία στην οποία ανήκουν, σύμφωνα με τον παραπάνω πίνακα. Είναι καλύτερα να χρησιμοποιηθεί η εντολή ΕΠΙΛΕΞΕ αντί της ΑΝ... ΑΛΛΙΩΣ_ΑΝ... λόγω της συμπαγούς δομής της στον προγραμματισμό.
- Το πρόγραμμα τυπώνει κατάλληλο μήνυμα.

Ακολουθεί η υλοποίηση του προγράμματος σε ΓΛΩΣΣΑ.

Κώδικας σε ΓΛΩΣΣΑ [3.3]

```
1  ΠΡΟΓΡΑΜΜΑ τέλη_κυκλοφορίας1
2  ΜΕΤΑΒΛΗΤΕΣ
3    ΑΚΕΡΑΙΕΣ: κ
4  ΑΡΧΗ
5    ΓΡΑΨΕ 'Δώσε κυβισμό αυτοκινήτου:.'
6    ΔΙΑΒΑΣΕ κ
```

```
7     ΕΠΙΛΕΞΕ κ
8       ΠΕΡΙΠΤΩΣΗ <= 1000
9         ΓΡΑΨΕ 'ΤΕΛΗ = 100€'
10      ΠΕΡΙΠΤΩΣΗ <= 1299
11        ΓΡΑΨΕ 'ΤΕΛΗ = 120€'
12      ΠΕΡΙΠΤΩΣΗ <= 1800
13        ΓΡΑΨΕ 'ΤΕΛΗ = 250€'
14      ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
15        ΓΡΑΨΕ 'ΤΕΛΗ = 600€'
16    ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
17  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ τέλη_κυκλοφορίας1
```

Παράδειγμα 4 – Μετατροπή ΕΠΙΛΕΞΕ σε ΑΝ...ΑΛΛΙΩΣ_ΑΝ...

Να κάνετε τη μετατροπή του παραπάνω προγράμματος (Κώδικας σε ΓΛΩΣΣΑ [3.3]), με χρήση της εντολής ΑΝ...ΑΛΛΙΩΣ_ΑΝ...

Απάντηση:

Κώδικας σε ΓΛΩΣΣΑ [3.4]

```
1   ΠΡΟΓΡΑΜΜΑ τέλη_κυκλοφορίας2
2   ΜΕΤΑΒΛΗΤΕΣ
3     ΑΚΕΡΑΙΕΣ: κ
4   ΑΡΧΗ
5     ΓΡΑΨΕ 'Δώσε κυβισμό αυτοκινήτου: '
6     ΔΙΑΒΑΣΕ κ
7     ΑΝ κ <= 1000 ΤΟΤΕ
8       ΓΡΑΨΕ 'ΤΕΛΗ = 100€'
9     ΑΛΛΙΩΣ_ΑΝ κ <= 1299 ΤΟΤΕ
10      ΓΡΑΨΕ 'ΤΕΛΗ = 120€'
11    ΑΛΛΙΩΣ_ΑΝ κ <= 1800 ΤΟΤΕ
12      ΓΡΑΨΕ 'ΤΕΛΗ = 250€'
13    ΑΛΛΙΩΣ
14      ΓΡΑΨΕ 'ΤΕΛΗ = 600€'
15    ΤΕΛΟΣ_ΑΝ
16  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ τέλη_κυκλοφορίας2
```

Παράδειγμα 5 – Μετατροπή ΑΝ...ΑΛΛΙΩΣ_ΑΝ... σε ΕΠΙΛΕΞΕ

Δίνεται το παρακάτω πρόγραμμα σε ΓΛΩΣΣΑ. Να κάνετε τη μετατροπή του προγράμματος χρησιμοποιώντας την εντολή πολλαπλής επιλογής ΕΠΙΛΕΞΕ.

Κώδικας σε ΓΛΩΣΣΑ [3.5]

```
ΠΡΟΓΡΑΜΜΑ άρτιος_περιττός_μονοψήφιος_2
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: χ
ΑΡΧΗ
  ΓΡΑΨΕ 'Δώσε μονοψήφιο αριθμό:.'
  ΔΙΑΒΑΣΕ χ
  ΑΝ  (χ=2)Η (χ=4) Η (χ=6) Η (χ=8) ΤΟΤΕ
          ΓΡΑΨΕ 'Άρτιος'
  ΑΛΛΙΩΣ_ΑΝ   (χ=1) Η (χ=3)Η (χ=5) Η (χ=7) Η (χ=9) ΤΟΤΕ
          ΓΡΑΨΕ 'Περιττός'
  ΑΛΛΙΩΣ_ΑΝ χ=0 ΤΟΤΕ
          ΓΡΑΨΕ 'Μηδέν'
  ΑΛΛΙΩΣ
          ΓΡΑΨΕ 'ο αριθμός δεν είναι μονοψήφιος...'
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ άρτιος_περιττός_μονοψήφιος_2
```

Απάντηση:

Κώδικας σε ΓΛΩΣΣΑ [3.6]

```
ΠΡΟΓΡΑΜΜΑ άρτιος_περιττός_μονοψήφιος_1
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: χ
ΑΡΧΗ
  ΓΡΑΨΕ 'Δώσε μονοψήφιο αριθμό:.'
  ΔΙΑΒΑΣΕ χ
  ΕΠΙΛΕΞΕ χ
      ΠΕΡΙΠΤΩΣΗ 2, 4, 6, 8
          ΓΡΑΨΕ 'Άρτιος'
    ΠΕΡΙΠΤΩΣΗ 1, 3, 5, 7, 9
          ΓΡΑΨΕ 'Περιττός'
      ΠΕΡΙΠΤΩΣΗ 0
          ΓΡΑΨΕ 'Μηδέν'
      ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
          ΓΡΑΨΕ 'ο αριθμός δεν είναι θετικός μονοψήφιος...'
  ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ άρτιος_περιττός_μονοψήφιος_1
```

3.1.2 Ερωτήσεις - Ασκήσεις

Ε.1: Ένα πρατήριο βενζίνης παρέχει τους εξής τύπους καυσίμων:

- Απλή αμόλυβδη με τιμή 1,395 €/λίτρο
- Super αμόλυβδη με τιμή 1,654 €/λίτρο
- Υγραέριο κίνησης με τιμή 0,698 €/λίτρο

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ, το οποίο να διαβάζει τον τύπο καυσίμου που θα βάλει ένας πελάτης στο όχημα του και τα χρήματα του πελάτη και να εμφανίζει πόσα λίτρα καυσίμου θα βάλει.

Υπόδειξη: Λίτρα = χρήματα/τιμή ανά λίτρο

Η παραπάνω άσκηση να λυθεί με χρήση της εντολής ΕΠΙΛΕΞΕ.

Ε.2: Να αντικαταστήσετε στο παρακάτω πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [3.7]) την εντολή πολλαπλής επιλογής ΕΠΙΛΕΞΕ με την εντολή ΑΝ... ΑΛΛΙΩΣ_ΑΝ..., έτσι ώστε να προκύπτουν τα ίδια αποτελέσματα.

Κώδικας σε ΓΛΩΣΣΑ [3.7]

```
1   ΠΡΟΓΡΑΜΜΑ Βαθμολογία
2   ΜΕΤΑΒΛΗΤΕΣ
3       ΠΡΑΓΜΑΤΙΚΕΣ: βαθμός
4   ΑΡΧΗ
5       ΓΡΑΨΕ 'Δώσε βαθμό: '
6       ΔΙΑΒΑΣΕ βαθμός
7      ΕΠΙΛΕΞΕ βαθμός
8           ΠΕΡΙΠΤΩΣΗ >= 17.5
9               ΓΡΑΨΕ 'Άριστα'
10          ΠΕΡΙΠΤΩΣΗ >= 15.5
11              ΓΡΑΨΕ 'Αρκετά καλά'
12            ΠΕΡΙΠΤΩΣΗ >= 13.5
13              ΓΡΑΨΕ 'Καλά'
14            ΠΕΡΙΠΤΩΣΗ >= 9.5
15              ΓΡΑΨΕ 'Μέτρια'
16            ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
17              ΓΡΑΨΕ 'Απορρίπτεται'
18     ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
19  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ Βαθμολογία
```

Ε.3: Η ΔΕΗ χρεώνει την ηλεκτρική κατανάλωση σύμφωνα με τον παρακάτω πίνακα:

Κιλοβατώρες (Kwh)	Τιμή μονάδας κιλοβατώρας
0 έως 2000	1,52€
2001 έως 3200	2,03€
3201 και άνω	4,65€

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει τις δύο τελευταίες μετρήσεις από το ρολόι της ΔΕΗ. Στη συνέχεια να υπολογίζει και να τυπώνει τα παρακάτω:

1. Το πλήθος των κιλοβατώρων που καταναλώθηκαν.
2. Την αξία του ρεύματος που καταναλώθηκε τη συγκεκριμένη περίοδο, σύμφωνα με τον παραπάνω πίνακα.
3. Το τελικό ποσό πληρωμής αν ο ΦΠΑ είναι 24% επί της αξίας του ρεύματος.

Υπόδειξη: Η χρέωση γίνεται κλιμακωτά.

Το πρόγραμμα να επιλυθεί με χρήση της εντολής πολλαπλής επιλογής ΕΠΙΛΕΞΕ.

Ενότητα 4

Σύγχρονα Προγραμματιστικά Περιβάλλοντα

Αντικειμενοστραφής Προγραμματισμός

Ενότητα 4. Σύγχρονα Προγραμματιστικά Περιβάλλοντα

4.1 Αντικειμενοστραφής Προγραμματισμός: ένας φυσικός τρόπος επίλυσης προβλημάτων

Από τις πρώτες γραμμές του βιβλίου σας, ο υπολογιστής σας παρουσιάστηκε ως μια μηχανή επίλυσης προβλημάτων, η λύση των οποίων περιγράφεται με τη μορφή ακολουθιών διακριτών βημάτων που εκτελούνται διαδοχικά και τα καλούμε αλγορίθμους. Πράγματι, η στρατηγική αυτή αποτελεί έναν πολύ χρήσιμο και πρακτικό τρόπο σκέψης για την επίλυση πολλών προβλημάτων (αν όχι όλων) με γνωστά ή διακριτά βήματα.

Αυτό είναι «φυσιολογικό». Εάν γνωρίζουμε τα βήματα, ή εάν έχουμε τον πλήρη έλεγχο των ενεργειών και των αποφάσεων που απαιτούνται, τότε μπορούμε να βάλουμε τα πράγματα σε μια σειρά και να λύσουμε οποιοδήποτε πρόβλημα.

Όμως, πόσες φορές έχετε βρεθεί μπροστά σε μια κατάσταση, για την οποία, ενώ γνωρίζετε, έστω και αόριστα, τη λύση, δεν μπορείτε να την εφαρμόσετε; Για παράδειγμα, τότε που η μαμά σας ήθελε να στείλει λουλούδια στη φίλη της την Άννα που γιόρταζε αλλά το τελευταίο διάστημα κατοικούσε στη Ρώμη! Ή τότε που οι γονείς σας αποφάσισαν να αλλάξουν τη διαρρύθμιση του σαλονιού και της κουζίνας του σπιτιού; Την επόμενη κιόλας μέρα επικοινώνησαν με τον κ. Πέτρο τον αρχιτέκτονα και μετά... για δύο μήνες κυκλοφορούσαν στο σπίτι χτίστες, σοβατζήδες, ελαιοχρωματιστές, ηλεκτρολόγοι, διάφοροι βοηθοί τους και άλλοι άγνωστοι. Όλοι αυτοί φώναζαν οδηγίες (που εσείς δεν καταλαβαίνατε) ο ένας στον άλλο. Πολλές φορές, η μια ομάδα περίμενε την άλλη να τελειώσει και συνέχιζε από το σημείο εκείνο. Τελικά, σε δύο μήνες είχατε ένα ανανεωμένο και πολύ πιο λειτουργικό σπίτι!

Πρόβλημα «Αποστολή λουλουδιών»

Ας προσπαθήσουμε να περιγράψουμε λίγο πιο τυπικά τη διαδικασία αποστολής λουλουδιών, την οποία θα θεωρήσουμε ως διαδικασία επίλυσης ενός προβλήματος: Η μητέρα σας, η οποία ήταν αδύνατο να πάει αυτοπροσώπως τα λουλούδια στην Άννα, 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 και τέλος! Ήταν σίγουρη ότι η ανθοδέσμη με τις ευχές της θα έφταναν έγκαιρα στη φίλη της.

Αυτό, βέβαια, έκρυβε ένα ολόκληρο σχέδιο ενεργειών για τον κ. Γιώργο, 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.

Το εντυπωσιακό είναι ότι κανείς από τους εμπλεκόμενους στο παραπάνω εγχείρημα δε γνώριζε τις

λεπτομέρειες του πώς υλοποιείται η κάθε επιμέρους εργασία, γνώριζε όμως από ποιον θα ζητούσε να την εκτελέσει και ποιες πληροφορίες θα του έδινε αναφορικά με το αναμενόμενο αποτέλεσμα. Ενδεχομένως, ο ειδικός να έφερε μαζί του και άλλους σαν κι αυτόν. Στο τέλος, όλες οι ομάδες ειδικών συνεργάστηκαν, άμεσα ή έμμεσα, για να υλοποιήσουν το σχέδιο της μαμάς.
Πρόκειται για τον πιο «φυσικό» τρόπο επίλυσης σύνθετων, άγνωστων προβλημάτων: όταν δεν ξέρεις πώς, ρωτάς αυτούς που ξέρουν! Το μόνο που χρειάζεται είναι η συγκρότηση ομάδας από ανθρώπους που γνωρίζουν να φέρουν εις πέρας καλά ορισμένες δραστηριότητες και η επίβλεψή της, ώστε να διασφαλιστεί η επικοινωνία μεταξύ των μελών.

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. Τα 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).

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.
Οι θεμελιώδεις αρχές του αντικειμενοστραφούς προγραμματισμού πηγάζουν από τον καθημερινό μας φυσικό κόσμο, καθώς είναι πολύ κοντά στον τρόπο που σκεφτόμαστε για να επιλύσουμε προβλήματα της καθημερινότητάς μας. Στηρίζονται 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.

Πρόβλημα «Παραγγελία Πίτσας»

Ας θεωρήσουμε ένα ακόμη καθημερινό πρόβλημα, το πρόβλημα της παραγγελίας και της παράδοσης μιας πίτσας στο σπίτι. Ας υποθέσουμε ότι είστε με παρέα στο σπίτι, παρακολουθείτε μία ταινία και θέλετε να φάτε πίτσα. Σας αρέσει μία συγκεκριμένη πίτσα. Δεν έχετε όμως ούτε την απαραίτητη «τεχνογνωσία», αλλά ούτε και τα υλικά για να την φτιάξετε. Τι κάνετε λοιπόν για να μην μείνετε νηστικοί; Μεταβιβάζετε τη διεκπεραίωση αυτού του έργου σε κάποιον άλλον που έχει τις κατάλληλες ικανότητες. Τηλεφωνείτε στην τοπική πιτσαρία και δίνετε στον ιδιοκτήτη, τον κ. Αλέξανδρο, την παραγγελία και τη διεύθυνση του σπιτιού σας. Έχετε εμπιστοσύνη στον κ. Αλέξανδρο ότι θα αναλάβει την ευθύνη να εκτελέσει το έργο που του αναθέσατε. Όταν οι πίτσες ετοιμαστούν, ο κ. Αλέξανδρος ζητάει από τον κ. Πέτρο, να σας παραδώσει την παραγγελία σας στη διεύθυνση που δώσατε.
Σημειώστε ότι στο σενάριο αυτό υπάρχουν τρεις βασικοί πρωταγωνιστές, τρία βασικά αντικείμενα, όπως θα λέγαμε στη γλώσσα του αντικειμενοστραφούς προγραμματισμού, τα οποία αλληλεπιδρούν μεταξύ τους με μηνύματα. Τα αντικείμενα αυτά είστε εσείς, ο κ. Αλέξανδρος και ο κ. Πέτρος.

Δραστηριότητα 1 – Παραγγελία Πίτσας: Μηνύματα και ρόλοι αντικειμένων

Μπορείτε να σκεφθείτε ποιο είναι το μήνυμα που απευθύνετε εσείς στον κ. Αλέξανδρο και ο κ. Αλέξανδρος με τη σειρά του στον κ. Πέτρο; Κάθε αντικείμενο παρέχει μία υπηρεσία που χρησιμοποιείται από τα άλλα αντικείμενα. Ποια είναι η υπηρεσία που παρέχει ή ο ρόλος του κ. Πέτρου και του κ. Αλέξανδρου; Ποιος είναι ο δικός σας ρόλος στο σενάριο αυτό;

Παρατηρήστε ότι το παραπάνω πρόβλημα, να φάτε πίτσα, επιλύθηκε με την εξεύρεση ενός κατάλληλου ατόμου, του κ. Αλέξανδρου, στον οποίο διαβιβάσατε ένα μήνυμα με το αίτημά σας. Ο κ. Αλέξανδρος είχε την ευθύνη να διεκπεραιώσει την αίτησή σας. Υπάρχει μια σειρά διαδικασιών ή μεθόδων που χρησιμοποίησε ο κ. Αλέξανδρος για την κατάλληλη προετοιμασία της πίτσας. Η προετοιμασία καθώς και όλες οι λεπτομέρειες της παρασκευής της πίτσας δεν χρειάζεται αλλά ούτε και σας ενδιαφέρει να γνωρίζετε. Είναι «κρυμμένες» από εσάς. Όπως θα δούμε και στη συνέχεια, η απόκρυψη των λεπτομερειών είναι μία από τις σημαντικές αρχές που χαρακτηρίζουν τον αντικειμενοστραφή προγραμματισμό.

Παρόμοια, όταν ο κ. Αλέξανδρος δίνει τις πίτσες στον κ. Πέτρο με το μήνυμα να τις παραδώσει, δεν χρειάζεται να γνωρίζει όλες τις λεπτομέρειες για το πώς θα επιτευχθεί η παράδοση της παραγγελίας. Οι λεπτομέρειες αυτές είναι «κρυμμένες» από αυτόν. Έχει αναθέσει αυτό το καθήκον στον κ. Πέτρο, τον οποίον και εμπιστεύεται και ο κ. Πέτρος ξέρει πώς θα παραδώσει την παραγγελία. Ο κ. Πέτρος μπορεί να συμβουλευτεί έναν χάρτη ή να χρησιμοποιήσει ένα σύστημα πλοήγησης για να βρει τη διεύθυνση, αλλά αυτό είναι κάτι που δεν ενδιαφέρει τον κ. Αλέξανδρο. Ο κ. Πέτρος κατέχει, γνωρίζει την κατάλληλη μέθοδο, όπως θα λέγαμε στη γλώσσα του αντικειμενοστραφούς προγραμματισμού για την εκτέλεση της παράδοσης.

4.2 Χτίζοντας Αντικειμενοστραφή Προγράμματα

Το «χτίσιμο» μιας αντικειμενοστραφούς εφαρμογής επιτυγχάνεται με τη δημιουργία και τον χειρισμό αντικειμένων τα οποία πρέπει να συνεργαστούν για την επίτευξη του κοινού στόχου που είναι η επίλυση του προβλήματος. Με ποιον όμως τρόπο εργαζόμαστε, ώστε να εντοπίσουμε τα απαραίτητα δομικά στοιχεία της εφαρμογής;

4.2.1 Μεθοδολογία

Το μόνο που έχουμε να κάνουμε είναι να αναλύσουμε το πρόβλημα το οποίο θέλουμε να επιλύσουμε, δηλαδή να αναγνωρίσουμε και να καταγράψουμε τα βασικά συστατικά στοιχεία της διαδικασίας επίλυσής του που είναι:

1. τα **αντικείμενα** που συμμετέχουν με βάση τον ρόλο τους στο συγκεκριμένο σενάριο,
2. οι **ιδιότητες** κάθε αντικειμένου, δηλ. τα σχετικά με το συγκεκριμένο πρόβλημα χαρακτηριστικά του, και
3. 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.

Πρόβλημα «Αποστολή λουλουδιών» (συστατικά επίλυσης προβλήματος)

Με βάση την παραπάνω μεθοδολογία είμαστε ήδη σε θέση να αρχίσουμε να αποτυπώνουμε το αντικειμενοστραφές περιβάλλον και τα συστατικά επίλυσης του προβλήματος «Αποστολή λουλουδιών» που αποτελεί τη βάση ανάπτυξης μιας πραγματικής αντικειμενοστραφούς εφαρμογής. Η εφαρμογή αυτή θα έδινε στη μαμά τη δυνατότητα ηλεκτρονικής παραγγελίας και αποστολής λουλουδιών από τα ηλεκτρονικά καταστήματα των ανθοπωλών μας!

1. Αντικείμενα

Διαβάζοντας προσεκτικά το σενάριο του προβλήματος «Αποστολή λουλουδιών» που παρουσιάστηκε εντοπίζουμε τη συμμετοχή των εξής αντικειμένων. Δίπλα σε κάθε αντικείμενο σημειώνουμε το ρόλο του στο συγκεκριμένο σενάριο. Θα μας χρειαστεί!

- Μαμά (Πελάτης),
- Γιώργος (Ανθοπώλης),
- Τζιοβάνι (Ανθοπώλης),
- Αντόνιο (Ανθοδέτης),
- Πέπε (Ταχυμεταφορέας),
- Άννα (Πελάτης)

2. Ιδιότητες

Διαβάζοντας ξανά το σενάριό μας προσπαθούμε να εντοπίσουμε τα σχετικά με το πρόβλημα χαρακτηριστικά κάθε αντικειμένου , δηλαδή τις ιδιότητές του. Τόσο για τη μαμά όσο και για την Άννα απαιτούνται προσωπικά στοιχεία (Όνομα, Επώνυμο) και στοιχεία επικοινωνίας (Διεύθυνση, Τηλέφωνο, Email).

- Μαμά (Πελάτης): Όνομα, Επώνυμο, Διεύθυνση, Τηλέφωνο, Email
- Άννα (Πελάτης): Όνομα, Επώνυμο, Διεύθυνση, Τηλέφωνο, Email

Παρατηρούμε ότι στην ιστορία μας δεν αναφέρονται συγκεκριμένα χαρακτηριστικά για τους ανθοπώλες, εν τούτοις, για την ανάπτυξη μιας αντικειμενοστραφούς εφαρμογής είναι απαραίτητο να γνωρίζουμε τις ιδιότητες όλων των αντικειμένων.

Αν θεωρήσουμε λοιπόν ότι για έναν ανθοπώλη είναι απαραίτητο να γνωρίζουμε εταιρικά στοιχεία, στοιχεία επικοινωνίας, στοιχεία πληρωμής και στοιχεία που αφορούν το δίκτυο συνεργασίας στο οποίο ανήκουν τότε μπορούμε να καταγράψουμε τις εξής ιδιότητες για τους ανθοπώλες της ιστορίας μας:

- Γιώργος (Ανθοπώλης): Επωνυμία εταιρείας, Όνομα Ιδιοκτήτη, Επώνυμο Ιδιοκτήτη, Διεύθυνση, ΑΦΜ, Τηλέφωνο, Email, Τραπεζικός Λογαριασμός, Κωδικός Δικτύου Συνεργασίας
- Τζιοβάνι (Ανθοπώλης): Επωνυμία εταιρείας, Όνομα Ιδιοκτήτη, Επώνυμο Ιδιοκτήτη, Διεύθυνση, ΑΦΜ, Τηλέφωνο, Email, Τραπεζικός Λογαριασμός, Κωδικός Δικτύου Συνεργασίας

Αντίστοιχα θα θεωρήσουμε ότι για τους ανθοδέτες και τους ταχυμεταφορείς πρέπει να γνωρίζουμε τα εξής:

- Αντόνιο (Ανθοδέτης): Επωνυμία εταιρείας, Όνομα Ιδιοκτήτη, Επώνυμο Ιδιοκτήτη, Διεύθυνση, ΑΦΜ, Τηλέφωνο, Email, Ειδικότητα, Ωριαία αμοιβή
- Πέπε (Ταχυμεταφορέας): Επωνυμία εταιρείας, Όνομα Ιδιοκτήτη, Επώνυμο Ιδιοκτήτη, Διεύθυνση, ΑΦΜ, Τηλέφωνο, Email, Τύπος

3. Ενέργειες/Υπηρεσίες και Είδος Συνεργασίας

Αναφέρθηκε ότι στην αντικειμενοστραφή προσέγγιση η επίλυση των προβλημάτων επιτυγχάνεται με τις συνεργασίες που αναπτύσσονται μεταξύ των αντικειμένων. Ας ξαναδούμε την ιστορία μας εστιάζοντας στις ενέργειες ή τις υπηρεσίες (μεθόδους) που παρέχει κάθε αντικείμενο και στο είδος της συνεργασίας που πρέπει να αναπτυχθεί μεταξύ των αντικειμένων για την παράδοση της ανθοδέσμης στην Άννα.

Ενέργειες/Υπηρεσίες

- Μαμά (Πελάτης): ΚάνειΠαραγγελία()
- Γιώργος (Ανθοπώλης): ΔέχεταιΠαραγγελία(), ΖητάΣυνεργασία()
- Τζιοβάνι (Ανθοπώλης): ΑποδέχεταιΣυνεργασία(), ΑναθέτειΑνθοδεσία(), ΑναθέτειΠαράδοση()
- Αντόνιο (Ανθοδέτης): ΕτοιμάζειΑνθοδέσμη()
- Πέπε (Ταχυμεταφορέας): ΠαραδίδειΑνθοδέσμη()
- Άννα (Πελάτης): ΠαραλαμβάνειΑνθοδέσμη()

Παρατηρήστε ότι οι μέθοδοι παρουσιάζονται με βάση την ονοματοδοσία των υποπρογραμμάτων, ώστε να διαφοροποιείται ο ρόλος τους ως λειτουργικών στοιχείων των αντικειμένων (ενέργειες ή υπηρεσίες που εφαρμόζονται πάνω στα δεδομένα) έναντι των ιδιοτήτων, οι οποίες αναπαριστούν τα δεδομένα που προσδιορίζουν την υπόστασή τους.

Είδος Συνεργασίας

- 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)

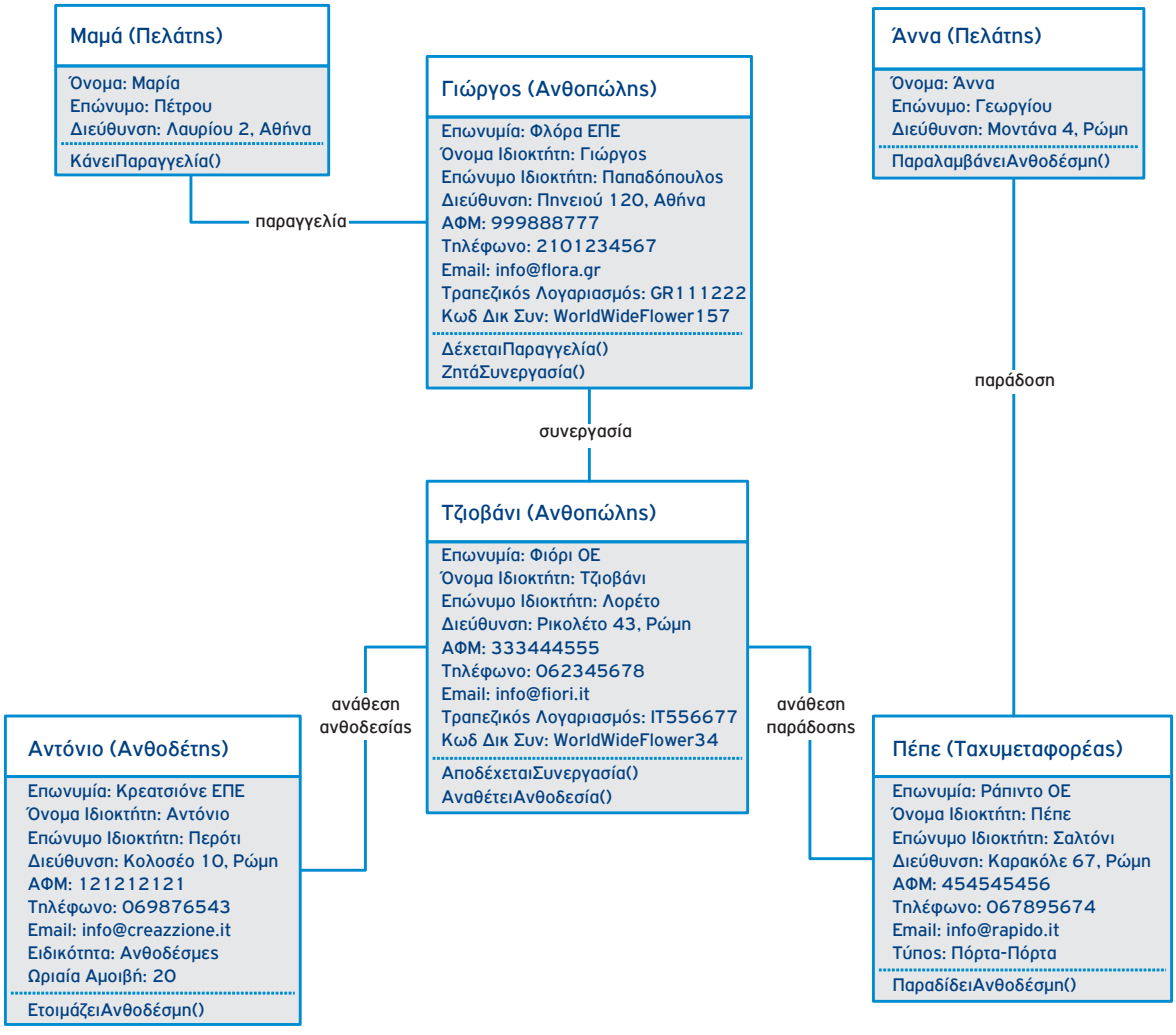
4.2.2 Διαγραμματική αναπαράσταση

Αφού εντοπίσουμε τα συστατικά επίλυσης του προβλήματος, μπορούμε να τα οργανώσουμε σε μια απλή διαγραμματική αναπαράσταση χρησιμοποιώντας **παραλληλόγραμμα** για την αποτύπωση των αντικειμένων, των ιδιοτήτων και των μεθόδων τους και **γραμμές σύνδεσης** για την περιγραφή του είδους της μεταξύ τους συνεργασίας (Εικόνα 4.1).
Η αναπαράσταση αυτή είναι ιδιαίτερα σημαντική, διότι μας δίνει την εποπτική εικόνα των συνεργαζόμενων οντοτήτων του προβλήματός μας και ουσιαστικά αποτελεί το σχέδιο επίλυσής του με βάση την αντικειμενοστραφή προσέγγιση.

Εικόνα 4.1 Διαγραμματική αναπαράσταση συστατικών επίλυσης προβλημάτων

Πρόβλημα «Αποστολή λουλουδιών» (διαγραμματική αναπαράσταση)

Με δεδομένα τα συστατικά επίλυσης που εντοπίσαμε για το πρόβλημα της αποστολής λουλουδιών κατασκευάζουμε τη σχετική διαγραμματική αναπαράσταση της Εικόνας 4.2.



Εικόνα 4.2 Διαγραμματική αναπαράσταση αντικειμένων του προβλήματος «Αποστολή λουλουδιών»

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.

Δραστηριότητα 2 – Παραγγελία Πίτσας: Μεθοδολογία και διαγραμματική αναπαράσταση

Ας επιστρέψουμε στο πρόβλημα «Παραγγελία πίτσας» της υποενότητας 4.1 για το οποίο θέλουμε να αναπτύξουμε την ηλεκτρονική εφαρμογή «e-pizza».
Εφαρμόστε τη μεθοδολογία ανάλυσης που παρουσιάστηκε στην ενότητα και δώστε τη διαγραμματική αναπαράσταση του σεναρίου της παραγγελίας σας στον κ. Αλέξανδρο και της παράδοσης της πίτσας από τον κ. Πέτρο.

4.3 Ομαδοποίηση Αντικειμένων σε Κλάσεις: Αφαιρετικότητα και Ενθυλάκωση

Κατά τη διαδικασία οργάνωσης και διαγραμματικής αναπαράστασης του προβλήματος της αποστολής λουλουδιών είδαμε ότι κάθε αντικείμενο περιέχει ένα σύνολο ιδιοτήτων και μεθόδων που ενεργούν πάνω στο αντικείμενο. Όπως ήδη σχολιάσαμε, στόχος αυτής της δόμησης των αντικειμένων είναι η απόκρυψη των λεπτομερειών υλοποίησης και λειτουργίας τους από τον υπόλοιπο κόσμο. Ουσιαστικά το αντικείμενο αποτελεί έναν «θύλακα», δηλαδή ένα σακούλι στο οποίο αποθηκεύει και συνδυάζει τα δεδομένα (ιδιότητες) και τις λειτουργίες (μεθόδους) του. Λέμε λοιπόν ότι τα αντικείμενα παρέχουν έναν τρόπο ενθυλάκωσης δεδομένων και λειτουργιών σε αυτά.

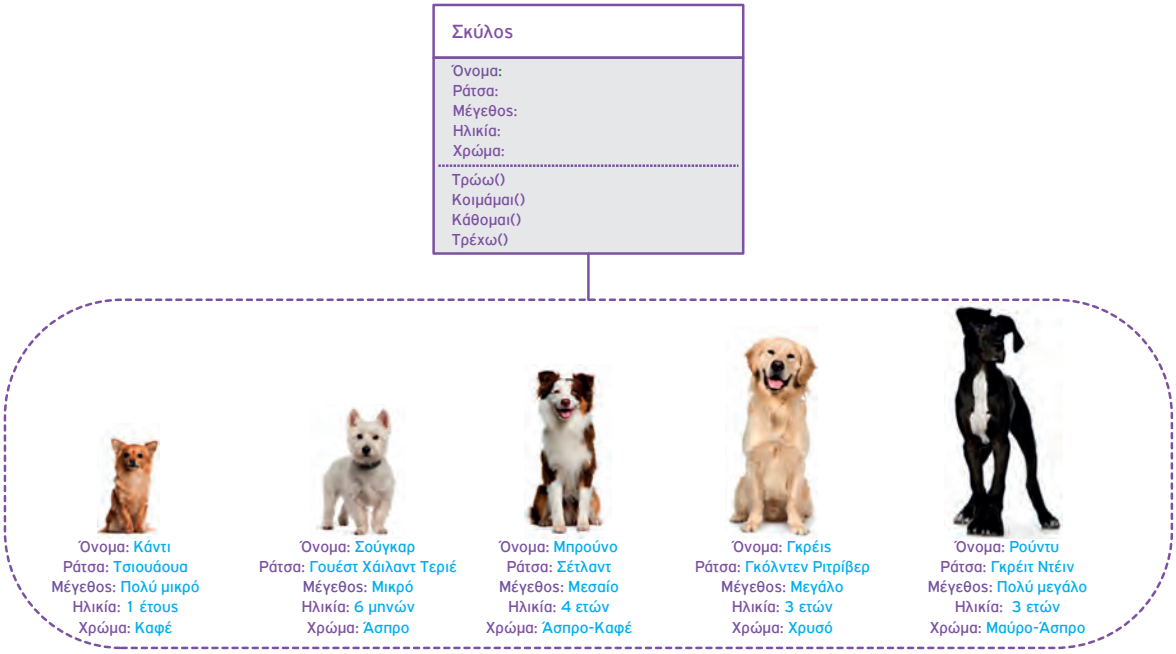
Σε μια αντικειμενοστραφή εφαρμογή κάθε αντικείμενο αποτελεί ξεχωριστή οντότητα και περιέχει ενσωματωμένες τις ιδιότητες (δεδομένα) και τους κανόνες συμπεριφοράς του (μεθόδους). Η δυνατότητα ενός αντικειμένου να συνδυάζει εσωτερικά τα δεδομένα και τις μεθόδους χειρισμού του καλείται **ενθυλάκωση** (encapsulation).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

Για οικονομία στην ανάπτυξη αντικειμενοστραφών προγραμμάτων ομαδοποιούμε τα αντικείμενα σε κλάσεις.

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.

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 διαφο-

ρετικές 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. Αντίστοιχα, στο ηλεκτρονικό παιχνίδι «Η φάρμα των ζώων» ο σκύλος μπορεί να αποτελέσει μια κλάση με βάση την οποία έχουμε τη δυνατότητα να δημιουργούμε απεριόριστα διαφορετικά σκυλάκια για καθένα από τα οποία θα ορίζουμε τη ράτσα, το μέγεθος, την ηλικία και το χρώμα (ιδιότητες), όπως επίσης και τις δραστηριότητες που θα μπορεί να έχει, π.χ. Τρώω(), Κοιμάμαι(), Κάθομαι(), Τρέχω() και ότι άλλο θέλουμε να μπορούν να κάνουν οι ψηφιακοί μας φίλοι μέσα στο παιχνίδι (μέθοδοι)!!!



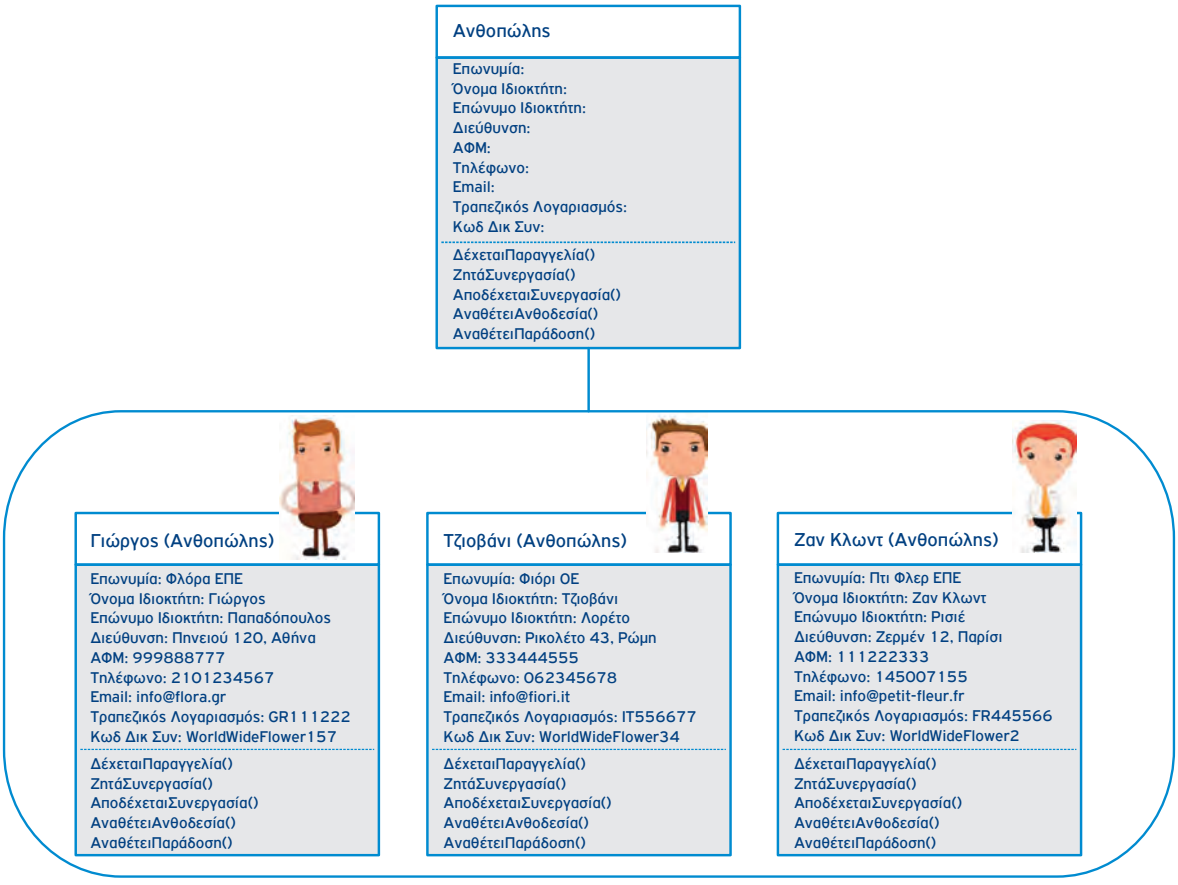
Εικόνα 4.3. Η κλάση «Σκύλος» και μερικά αντικείμενά της

Πρόβλημα «Αποστολή λουλουδιών» (ομαδοποίηση αντικειμένων σε κλάσεις)

Για να δούμε λοιπόν, μήπως και στην ιστορία της αποστολής λουλουδιών μπορούμε να κάνουμε μια αντίστοιχη «οικονομία», δηλαδή ομαδοποίηση των αντικειμένων σε κλάσεις, ώστε και το πρόγραμμά μας να γίνει πιο απλό και κομψό.

Πράγματι, είναι προφανές ότι όλοι οι ανθοπώλες (δηλαδή, ο κ. Γιώργος, ο κ. Τζιοβάνι και όσοι είναι μέλη του δικτύου συνεργασίας) έχουν τις ίδιες ιδιότητες (επωνυμία εταιρείας, όνομα ιδιοκτήτη, επώνυμο ιδιοκτήτη, διεύθυνση, κ.λπ.) και μπορούν να κάνουν τις ίδιες ενέργειες (δέχονται παραγγελίες, ζητούν συνεργασία, αποδέχονται συνεργασία, αναθέτουν ανθοδεσίες, αναθέτουν παραδόσεις), αλλά ο καθένας τους έχει διαφορετική υπόσταση και ταυτότητα: έχει τη δική του εταιρεία, βρίσκεται σε διαφορετική χώρα, κ.λπ. Άρα, ο «Ανθοπώλης» μπορεί να αποτελέσει μια κλάση με αντικείμενα τον κ. Γιώργο, τον κ. Τζιοβάνι και άλλους συναδέλφους τους από όλο τον κόσμο!

Ας δούμε μια απλή αναπαράσταση της κλάσης «Ανθοπώλης» και των αντικειμένων της, του κ. Γιώργου, του κ. Τζιοβάνι και ενός νέου μέλους, του κ. Ζαν Κλωντ από το Παρίσι.

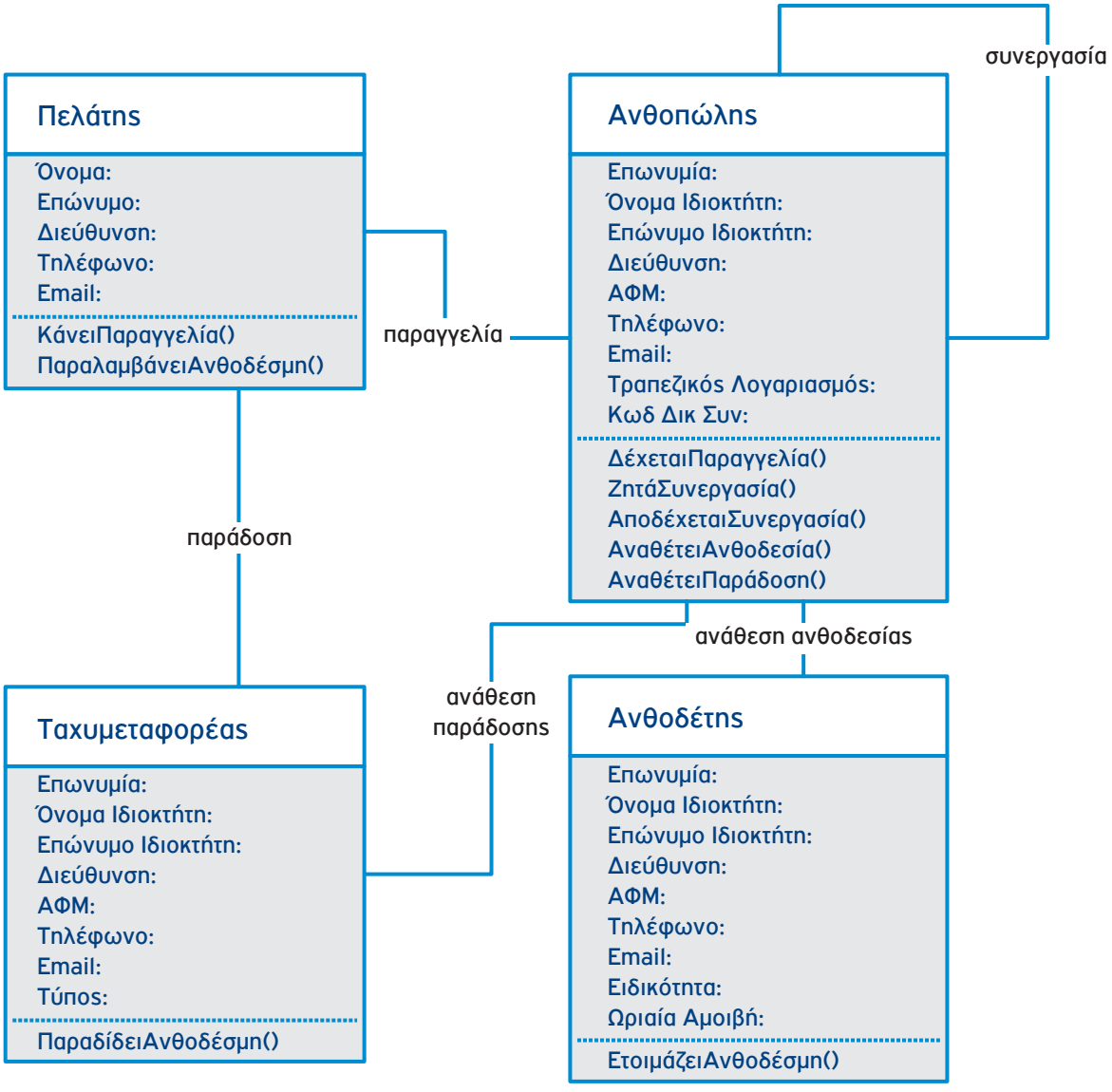


Εικόνα 4.4. Η κλάση «Ανθοπώλης» και τα αντικείμενα της ιστορίας «Αποστολή λουλουδιών»

Θα ήταν λογικό βέβαια να αναρωτηθείτε: «μα εγώ διάβασα στην ιστορία ότι ο κ. Γιώργος μόνο δέχεται παραγγελίες και ζητά συνεργασίες, ενώ ο κ. Τζιοβάνι αποδέχεται συνεργασίες και αναθέτει ανθοδεσίες και παραδόσεις! Δεν κάνουν τα ίδια πράγματα!». Σωστά ως εδώ, όμως σκεφτείτε τι θα γίνει όταν στα γενέθλια της μαμάς αποφασίσει και η Άννα να στείλει λουλούδια από τη Ρώμη.
Μπορείτε να δείτε στην ιστορία μας ποια επιπλέον αντικείμενα μπορούν να ομαδοποιηθούν σε κλάσεις; Οι πελάτες των ανθοπωλών (η μαμά και η Άννα) μπορούν να ομαδοποιηθούν στην κλάση «Πελάτης». Αλλά και ο κ. Πέπε και ο κ. Αντόνιο είναι αντικείμενα των κλάσεων «Ταχυμεταφορέας» και «Ανθοδέτης».

Τελικά οι ρόλοι που ορίσαμε για το κάθε αντικείμενο στην αρχή της ενασχόλησής μας με το πρόβλημα μας φάνηκαν ιδιαίτερα χρήσιμοι, αφού από αυτούς μπορούν άμεσα να προκύψουν τα ονόματα των κλάσεων μας!

Με τις ομαδοποιήσεις θα προσπαθήσουμε να φτιάξουμε μια νέα αναπαράσταση, η οποία εκτός του ότι «συμμαζεύει» και «τακτοποιεί» το διάγραμμα της επίλυσής μας, έχει την ιδιότητα να μπορεί να εκφράσει οποιαδήποτε ιστορία αποστολής λουλουδιών: Η μαμά στην Άννα στη Ρώμη, η Άννα στη μαμά στην Αθήνα, ο μπαμπάς στον συνεργάτη του στο Παρίσι, κ.λπ.



Εικόνα 4.5. Διαγραμματική αναπαράσταση κλάσεων του προβλήματος «Αποστολή λουλουδιών»

Δραστηριότητα 3 – Παραγγελία Πίτσας: Κλάσεις και διαγραμματική αναπαράσταση

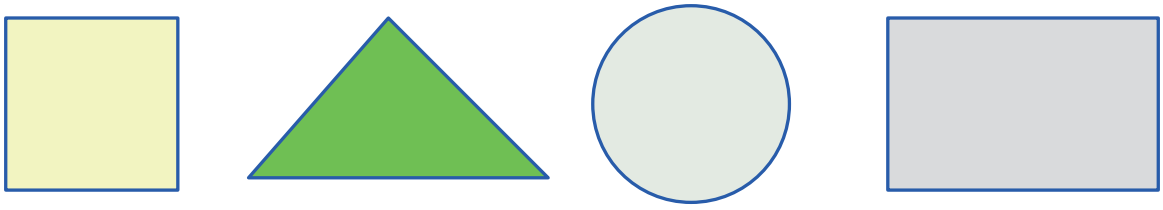
Ας επιστρέψουμε στο πρόβλημα «Παραγγελία Πίτσας» της υποενότητας 4.1. Όταν σκεφτόσασταν να παραγγείλετε πίτσα, γνωρίζατε τον τρόπο που λειτουργούν οι πιτσαρίες και τις υπηρεσίες που παρέχουν. Όλες οι πιτσαρίες έχουν κάποια συγκεκριμένα κοινά χαρακτηριστικά, που σας επιτρέπουν να τις ομαδοποιήσετε και να τις εντάξετε σε μία ομάδα ή αλλιώς σε μία κλάση με το όνομα πιτσαρία. Η δικιά σας, η τοπική πιτσαρία από την οποία παραγγείλατε, έχει μία συγκεκριμένη διεύθυνση που την κάνει να ξεχωρίζει από τις άλλες πιτσαρίες. Η δικιά σας η τοπική πιτσαρία αλλά και όλες οι άλλες πιτσαρίες αποτελούν αντικείμενα της κλάσης πιτσαρία.
Με βάση την ανάλυση του σεναρίου παραγγελίας που αναπτύξατε στην προηγούμενη υποενότητα, προσπαθήστε να εντοπίσετε τις κατάλληλες κλάσεις για την εφαρμογή σας και να τις αναπαραστήσετε σε ένα νέο διάγραμμα κλάσεων.

4.3.1 Παραδείγματα Διαγραμματικής Αναπαράστασης Κλάσεων

Παρακάτω παρουσιάζονται αντιπροσωπευτικά παραδείγματα διαγραμματικής αναπαράστασης κλάσεων που διευκολύνουν την επίλυση εφαρμογών με την αντικειμενοστραφή προσέγγιση.

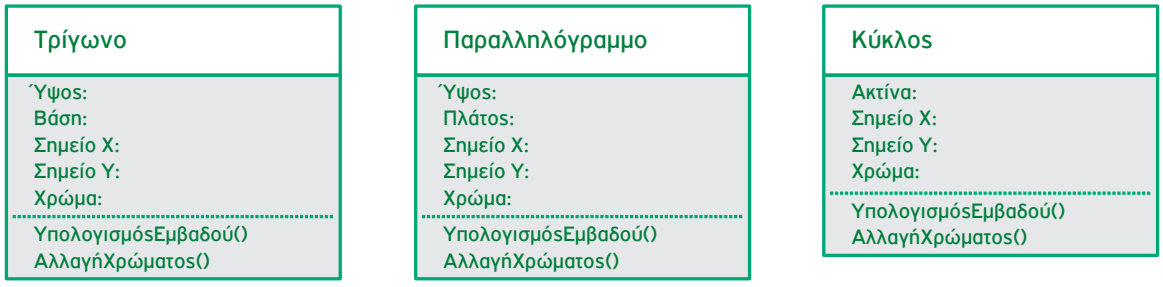
Παράδειγμα: «Σχεδιασμός Εικόνων με Γεωμετρικά Σχήματα»

Μας ζητείται να κατασκευάσουμε μια εφαρμογή που υποστηρίζει τον σχεδιασμό σύνθετων εικόνων που περιέχουν γεωμετρικά σχήματα σαν αυτά της Εικόνας 4.6.
Σύμφωνα με την αντικειμενοστραφή προσέγγιση, κάθε φορά που ο χρήστης επιλέγει να σχεδιάσει ένα σχήμα το πρόγραμμα θα πρέπει να ενεργοποιεί ένα αντικείμενο του σχήματος αυτού.



Εικόνα 4.6. Αντικείμενα γεωμετρικών σχημάτων της εφαρμογής σχεδιασμού εικόνων

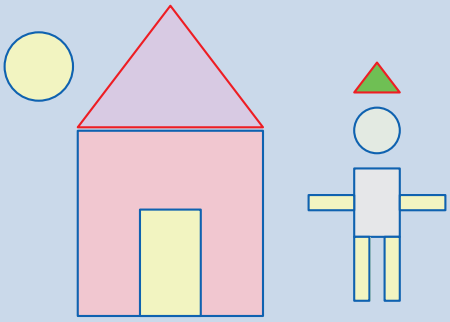
Αν θεωρήσουμε ότι τα αντικείμενα σχημάτων που επιθυμούμε να υποστηρίζονται από την εφαρμογή είναι αυτά της Εικόνας 4.6, πρέπει να ορίσουμε την κλάση αντικειμένων που αντιστοιχεί σε καθένα από τα γεωμετρικά αυτά σχήματα. Επομένως, απαιτείται ο ορισμός των κλάσεων «Τρίγωνο», «Παραλληλόγραμμο» και «Κύκλος». Το τετράγωνο αποτελεί μια ειδική περίπτωση παραλληλόγραμμου και συνεπώς μπορούμε να μην ορίσουμε ξεχωριστή κλάση για το σχήμα αυτό.
Από τη μορφή των αντικειμένων της Εικόνας 4.6 μπορούν να προκύψουν οι βασικές ιδιότητες που πρέπει να συμπεριλάβουμε στον ορισμό κάθε κλάσης. Οι ιδιότητες αυτές αφορούν στις διαστάσεις κάθε σχήματος, στο χρώμα του, αλλά και στο σημείο τοποθέτησης του σχήματος στην εικόνα. Επίσης, πρέπει να ορίσουμε τις επιθυμητές μεθόδους, μία που όταν κληθεί θα υπολογίζει το εμβαδόν του αντίστοιχου σχήματος και μία που θα δίνει τη δυνατότητα αλλαγής του χρώματος του σχήματος. Με βάση τα στοιχεία αυτά ο ορισμός των τριών κλάσεων γεωμετρικών σχημάτων δίνεται στην Εικόνα 4.7.



Εικόνα 4.7. Κλάσεις γεωμετρικών σχημάτων

Δραστηριότητα 4 – Διαγραμματική αναπαράσταση εικόνας γεωμετρικών σχημάτων

Θεωρείστε ότι στην εφαρμογή σχεδιασμού εικόνων η σύνδεση μεταξύ αντικειμένων γεωμετρικών σχημάτων είναι επιτρεπτή μόνο για κάποια είδη σχημάτων.
Με βάση την Εικόνα 4.8 ποια είναι τα σχήματα αυτά;
Δώστε τη διαγραμματική αναπαράσταση που να περιγράφει την εικόνα συνδέοντας κατάλληλα τις τρεις κλάσεις που αναφέρθηκαν στο παράδειγμα: «Σχεδιασμός Εικόνων με Γεωμετρικά Σχήματα».



Εικόνα 4.8. Εικόνα εφαρμογής σχεδιασμού γεωμετρικών σχημάτων

Παράδειγμα: «Ενοικίαση ταινιών»

Ας δούμε ένα ακόμη παράδειγμα. Αποφασίζετε με τους φίλους σας να ενοικιάσετε μια ταινία από ένα κατάστημα ενοικίασης ταινιών της περιοχής σας. Για τον σκοπό αυτό χρησιμοποιείτε σχετική διαδικτυακή εφαρμογή, στην οποία διαθέτετε κωδικό πελάτη. Το σενάριο ενοικίασης της ταινίας που επιλέξατε είναι το εξής:
«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».

Ας προσπαθήσουμε με βάση το σενάριο αυτό να εντοπίσουμε τις κλάσεις που εμπλέκονται στην ανάπτυξη της εφαρμογής. Παρατηρούμε ότι τα βασικά **αντικείμενα** που συμμετέχουν στο σενάριο είναι:

1. Γιώργος Πέτρου (Πελάτης)
2. ΒιντεοΔράση (Κατάστημα)
3. Πόλεμος των Άστρων (Ταινία)

Συνεπώς, οι απαιτούμενες κλάσεις είναι Πελάτης, Κατάστημα και Ταινία.

Στη συνέχεια, έστω ότι μας ζητείται η διαγραμματική αναπαράσταση των κλάσεων της εφαρμογής και των μεταξύ τους συνεργασιών. Για την υλοποίηση της διαγραμματικής αναπαράστασης απαιτείται η πλήρης προδιαγραφή καθεμιάς από τις συμμετέχουσες κλάσεις, δηλαδή ο εντοπισμός των ιδιοτήτων, των μεθόδων και του είδους της μεταξύ τους συνεργασίας.

Οι **ιδιότητες** των κλάσεων που απορρέουν από τις πληροφορίες του σεναρίου είναι:

1. Πελάτης: Όνομα, Επώνυμο, Κωδικός, Διεύθυνση, Τηλέφωνο
2. Κατάστημα: Ονομασία, Διεύθυνση
3. Ταινία: Τίτλος, Αριθμός

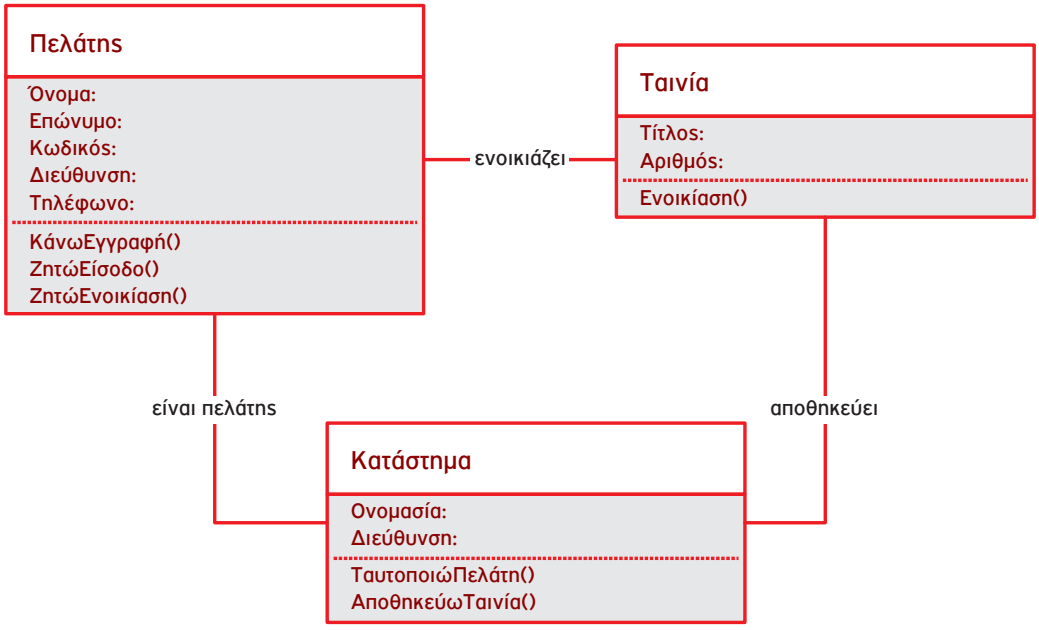
Τα **είδη συνεργασίας** που περιγράφονται στο σενάριο μεταξύ των συμμετεχόντων αντικειμένων είναι:

1. Είναι πελάτης: Γιώργος Πέτρου (Πελάτης) - ΒιντεοΔράση (Κατάστημα)
2. Ενοικιάζει: Γιώργος Πέτρου (Πελάτης) - Πόλεμος των Άστρων (Ταινία)
3. Αποθηκεύει: ΒιντεοΔράση (Κατάστημα) - Πόλεμος των Άστρων (Ταινία)

Με βάση τα παραπάνω είδη συνεργασίας, οι ελάχιστες ενέργειες που πρέπει να υποστηρίζουν τα αντικείμενα, δηλαδή οι **μέθοδοι** που πρέπει να υλοποιούν είναι:

1. Γιώργος Πέτρου (Πελάτης): ΖητώΕίσοδο(), ΖητώΕνοικίαση()
2. ΒιντεοΔράση (Κατάστημα): ΤαυτοποιώΠελάτη(), ΑποθηκεύωΤαινία()
3. Πόλεμος των Άστρων (Ταινία): Ενοικίαση()

Στην Εικόνα 4.9 δίνεται η διαγραμματική αναπαράσταση ορισμού και συνεργασίας των κλάσεων της εφαρμογής ενοικίασης ταινιών.



Εικόνα 4.9. Διαγραμματική αναπαράσταση κλάσεων της εφαρμογής «Ενοικίαση ταινιών»

Δραστηριότητα 5 – Αγορά εισιτηρίων μέσω Διαδικτύου

Τα τελευταία χρόνια υπάρχουν πολλές ιστοσελίδες που δίνουν τη δυνατότητα αγοράς εισιτηρίων για διάφορα θεάματα (θεατρικές παραστάσεις, συναυλίες, κ.λπ.). Οι εταιρείες που προσφέρουν τέτοιες υπηρεσίες έχουν αναπτύξει κατάλληλο πληροφοριακό σύστημα. Οι βασικές λειτουργίες του πληροφοριακού συστήματος είναι: εγγραφή και διαγραφή χρήστη, σύνδεση και αποσύνδεση χρήστη, καταχώριση και απενεργοποίηση θεάματος, παραγγελία εισιτηρίου και ακύρωση παραγγελίας, εξόφληση παραγγελίας.
Σύμφωνα με την παραπάνω περιγραφή, εντοπίστε τις κλάσεις που πρέπει να δημιουργηθούν στο πληροφοριακό σύστημα αγοράς εισιτηρίων. Καθορίστε τις ιδιότητες και μεθόδους κάθε κλάσης και αποτυπώστε τις σχέσεις μεταξύ των κλάσεων, ώστε να υλοποιηθεί το σενάριο.

4.4 Η Αντικειμενοστραφής «Οικογένεια»: Κλάσεις - Πρόγονοι, Κλάσεις - Απόγονοι

Είναι γνωστό ότι για κάθε πράγμα που αποφασίζει να κάνει η μαμά πρέπει να κινητοποιηθεί μια σειρά ειδικών και μη, ώστε να γίνουν όλα στην εντέλεια και με απόλυτη ακρίβεια. Τα λουλούδια της Άννας, για παράδειγμα, ενέπλεξαν τρεις (τουλάχιστον) κατηγορίες επαγγελματιών ενεργοποιώντας ακόμα και διακρατικές συνεργασίες;) Οι κ. Γιώργος, Τζιοβάνι, Αντόνιο και Πέπε, αν και αντικείμενα διαφορετικών κλάσεων (Ανθοπώλης, Ανθοδέτης, Ταχυμεταφορέας), παρατηρούμε ότι έχουν πολλά κοινά στοιχεία (ιδιότητες) μεταξύ τους. Έχουν όμως και διαφορές. Μήπως θα μπορούσαμε να ομαδοποιήσουμε τα κοινά στοιχεία των τριών κλάσεων, ώστε η αποτύπωση του σεναρίου μας να είναι ακόμη περισσότερο εποπτική;
Φυσικά και μπορούμε! Η αντικειμενοστραφής προσέγγιση μας παρέχει τη δυνατότητα να συνδέσουμε ιεραρχικά δύο ή περισσότερες κλάσεις με κοινές ιδιότητες και μεθόδους.

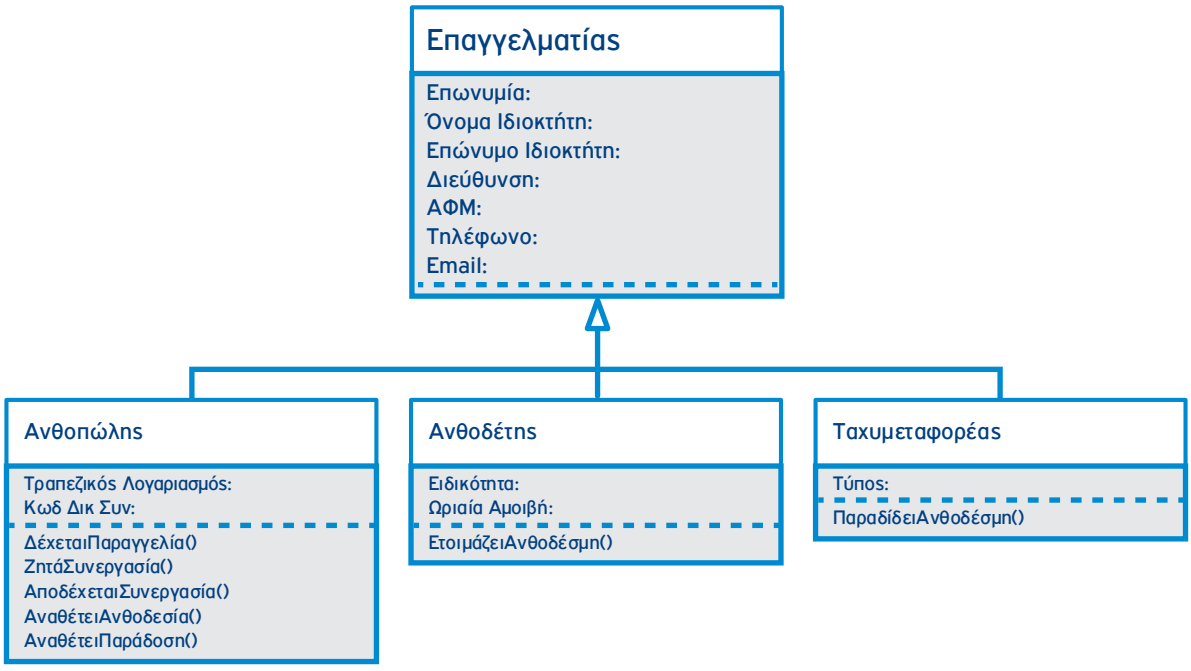
Η δυνατότητα δημιουργίας ιεραρχιών αντικειμένων καλείται **κληρονομικότητα** (inheritance). Με βάση την κληρονομικότητα, μια κλάση μπορεί να περιγραφεί γενικά και στη συνέχεια μέσω αυτής της κλάσης να οριστούν υποκλάσεις αντικειμένων. Η κλάση απόγονος (υποκλάση) κληρονομεί και μπορεί να χρησιμοποιήσει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

Πρόβλημα «Αποστολή λουλουδιών» (ιεραρχία κλάσεων)

Μπορούμε να διαπιστώσουμε ότι η κλάση «Επαγγελματίας» διαθέτει τις κοινές ιδιότητες όλων των εμπλεκόμενων ειδικοτήτων «Ανθοπώλης», «Ανθοδέτης» και «Ταχυμεταφορέας» και τις κληροδοτεί σε αυτές. Αντίστοιχα, οι τρεις τελευταίες κλάσεις κληρονομούν όλες τις ιδιότητες της κλάσης «Επαγγελματίας», στα οποία η καθεμία προσθέτει και ορισμένες δικές της ιδιότητες.
Ας προετοιμαστούμε να αποτυπώσουμε τη σχέση κληρονομικότητας μεταξύ της κλάσης Επαγγελματίας και των κλάσεων Ανθοπώλης, Ανθοδέτης και Ταχυμεταφορέας με βάση τον εξής κανόνα:

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 ↑ όπως παρουσιάζεται στην Εικόνα 4.10.



Εικόνα 4.10. Ιεραρχία κλάσεων με σχέση κληρονομικότητας επαγγελματιών στην «Αποστολή λουλουδιών»

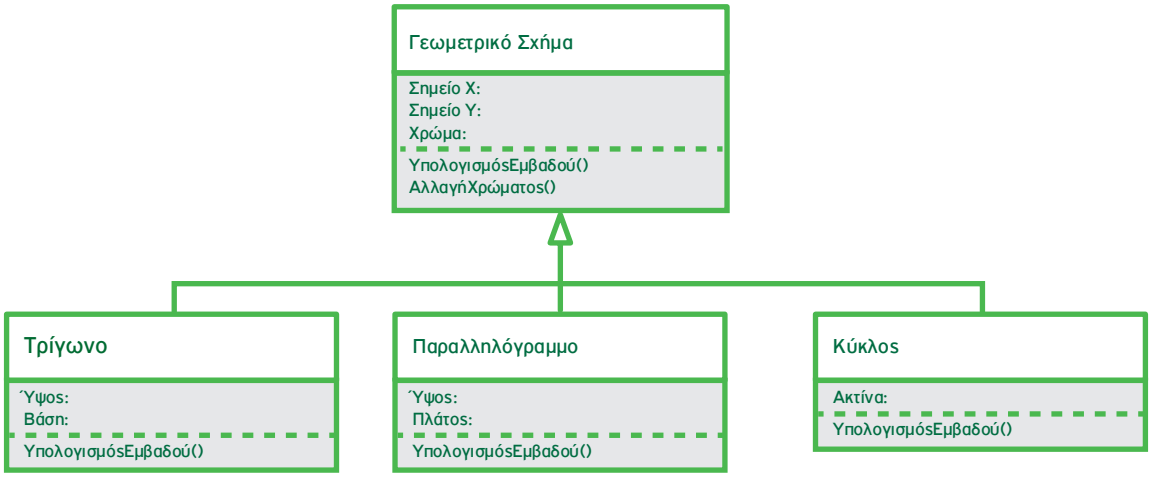
Αν επιχειρήσουμε να διαβάσουμε το διάγραμμα θα πούμε ότι «ο Ανθοπώλης είναι (is_a) Επαγγελματίας». Εκμεταλλευόμενοι τη σχέση της κληρονομικότητας και τη σημασία της, δεν επαναλαμβάνουμε τις ιδιότητες της κλάσης Επαγγελματίας στην κλάση Ανθοπώλης, γνωρίζουμε όμως ότι κάθε ένα από τα αντικείμενα που δημιουργούνται με βάση την κλάση αυτή (κ. Γιώργος, κ. Τζιοβάνι, κ. Ζαν Κλωντ) θα διαθέτει το σύνολο των χαρακτηριστικών τόσο του Επαγγελματία όσο και του Ανθοπώλη.

Παράδειγμα: «Σχεδιασμός Εικόνων με Γεωμετρικά Σχήματα»

Θυμηθείτε την εφαρμογή σχεδιασμού εικόνων με τα γεωμετρικά σχήματα που συναντήσαμε στην υποενότητα 4.3. Ας προσπαθήσουμε να ορίσουμε μια κατάλληλη κλάση-πρόγονο (υπερκλάση) των τριών σχημάτων και να δώσουμε τη διαγραμματική αναπαράσταση της σχέσης κληρονομικότητας που δημιουργείται.

Αν κοιτάξουμε προσεκτικά τον ορισμό των τριών κλάσεων, θα εντοπίσουμε κοινά στοιχεία τα οποία επαναλαμβάνονται σε κάθε κλάση. Ορίζουμε λοιπόν την αφαιρετική κλάση «Γεωμετρικό Σχήμα» στην οποία θα αποδώσουμε όλα τα κοινά στοιχεία (ιδιότητες και μεθόδους), των τριών κλάσεων.

Στη συνέχεια θα ορίσουμε σχέση κληρονομικότητας ανάμεσα στην κλάση «Γεωμετρικό Σχήμα» και τις τρεις κλάσεις σχημάτων, ώστε καθεμία από αυτές να κληρονομεί τα κοινά στοιχεία. Για το διάγραμμα κλάσεων θα χρησιμοποιήσουμε τη σχέση γενίκευσης προς την κλάση με όνομα «Γεωμετρικό Σχήμα». Στην κλάση αυτή θα συλλέξουμε την πληροφορία που είναι κοινή σε κάθε κλάση γεωμετρικού σχήματος και δεν θα εμφανίζεται πλέον στις υποκλάσεις. Έτσι δεν καταγράφουμε τις ιδιότητες «Χρώμα», «Σημείο Χ» και «Σημείο Υ» στις κλάσεις «Τρίγωνο», «Παραλληλόγραμμο» και «Κύκλος», αφού αποτελούν πλέον ιδιότητες της κλάσης «Γεωμετρικό Σχήμα» (υπερκλάση), ωστόσο, οι ιδιότητες αυτές κληρονομούνται σε καθεμία από τις κλάσεις σχημάτων (υποκλάσεις).



Εικόνα 4.11. Σχέση κληρονομικότητας γεωμετρικών σχημάτων

Οι μέθοδοι «ΥπολογισμόςΕμβαδού()» και «ΑλλαγήΧρώματος()» θα πρέπει επίσης να εμφανίζονται στην υπερκλάση Γεωμετρικό Σχήμα, εφόσον αποτελούν κοινές μεθόδους των τριών σχημάτων. Προσέξτε εδώ ένα ενδιαφέρον σημείο, το οποίο θα εξηγήσουμε αναλυτικά στην επόμενη υποενότητα: Επειδή ο υπολογισμός του εμβαδού κάθε σχήματος γίνεται με διαφορετικό τρόπο, η μέθοδος αυτή θα εμφανιστεί και στις υποκλάσεις «Τρίγωνο», «Παραλληλόγραμμο», «Κύκλος». Η διαγραμματική αναπαράσταση της σχέσης κληρονομικότητας που δημιουργείται παρουσιάζεται στην Εικόνα 4.11.

Δραστηριότητα 6: Δεν μπορούμε να ξεφύγουμε από την κληρονομικότητα

Ξεκάστε προς στιγμή κλάσεις και αντικείμενα και μελετήστε τις παρακάτω προτάσεις:

- ❒ Το αυτοκίνητο είναι ένα μέσο μεταφοράς.
- ❒ Το λεωφορείο είναι ένα μέσο μεταφοράς.
- ❒ Το αυτοκίνητο είναι ένα λεωφορείο.
- ❒ Ο υπάλληλος είναι ένα πρόσωπο.
- ❒ Ο πελάτης είναι ένα πρόσωπο.
- ❒ Ο πελάτης είναι μία πιστωτική κάρτα.
- ❒ Ο τρεχούμενος λογαριασμός είναι ένα είδος τραπεζικού λογαριασμού.
- ❒ Ο λογαριασμός ταμιευτηρίου είναι ένα είδος τραπεζικού λογαριασμού.

Σε ποιες από τις παραπάνω φράσεις βγαίνει κάποιο νόημα και σε ποιες όχι; Μήπως σε αυτές που βγαίνει νόημα και περιέχουν την λέξη «είναι» ή «είναι ένα είδος» εμπεριέχουν την έννοια της κληρονομικότητας; Μήπως, στην καθημερινότητά μας, όταν μιλάμε, χρησιμοποιούμε την έννοια της κληρονομικότητας χωρίς να το αντιλαμβανόμαστε;

Πράγματι, η οργάνωση των εννοιών του κόσμου μας στο πλαίσιο ιεραρχιών κληρονομικότητας αποτελεί βασική νοητική λειτουργία κάθε ανθρώπου αλλά και απαραίτητη δεξιότητα στην αντικειμενοστραφή σχεδίαση και τον προγραμματισμό. Ας την αξιοποιήσουμε, λοιπόν, για να ξεκινήσουμε να χτίζουμε τις πρώτες μας ιεραρχίες!
Ένας χρήσιμος κανόνας που μας βοηθά στο έργο μας αυτό είναι:

Μια κλάση Α μπορεί να είναι έγκυρη υποκλάση της Β αν έχει νόημα να πούμε «ένα Α είναι ένα (is_a) Β»

Δραστηριότητα 7: Βρείτε τα έγκυρα ζεύγη

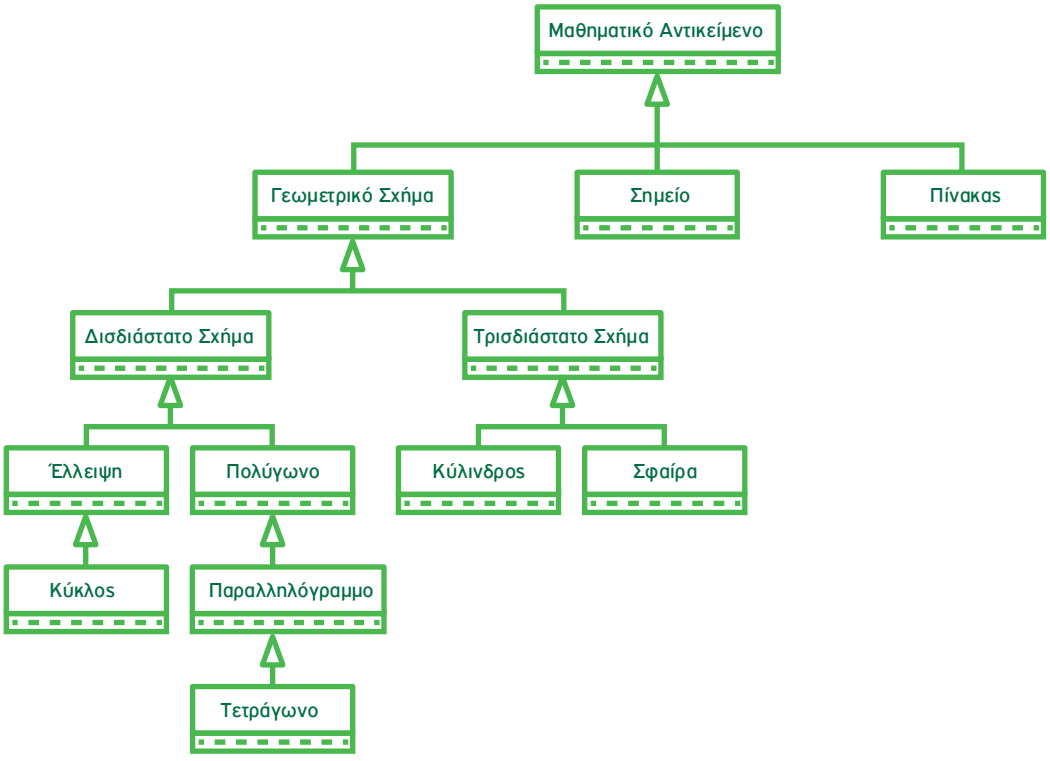
Ποια από τα παρακάτω δε σχηματίζουν έγκυρα ζεύγη υπερκλάσης-υποκλάσης και γιατί;

- ❒ Νόμισμα - Ευρώ
- ❒ Τράπεζα - Λογαριασμός
- ❒ Οργανωτική Μονάδα - Τμήμα
- ❒ Λογαριασμός - Λογαριασμός_23456
- ❒ Άνθρωπος - Πελάτης
- ❒ Φοιτητής - Προπτυχιακός φοιτητής
- ❒ Ήπειρος – Χώρα
- ❒ Δήμος - Συνοικία

Παράδειγμα: «Επεκτείνοντας την ιεραρχία των Γεωμετρικών Σχημάτων»

Ας επιστρέψουμε στην ιεραρχία κλάσεων γεωμετρικών σχημάτων που κατασκευάσαμε στην υποενότητα 4.4 και ας προσπαθήσουμε να την επεκτείνουμε με το παρακάτω σύνολο κλάσεων το οποίο μας ζητείται να οργανώσουμε σε μία νέα ιεραρχία: Σημείο, Πολύγωνο, Πίνακας, Κύκλος, Δισδιάστατο Σχήμα, Τετράγωνο, Έλλειψη, Κύλινδρος, Μαθηματικό Αντικείμενο, Σφαίρα, Παραλληλόγραμμο. Στο παράδειγμα αυτό δε θα εστιάσουμε στις ιδιότητες ή τις μεθόδους των αναφερόμενων κλάσεων.
Μελετώντας τις έννοιες προσπαθούμε να εντοπίσουμε τα επίπεδα ιεραρχίας από το γενικό στο ειδικό και να κατατάξουμε κάθε κλάση στο αντίστοιχο επίπεδο ορίζοντας τις κατάλληλες σχέσεις κληρονομικότητας. Παρατηρούμε ότι η πιο γενική έννοια του συνόλου μας, και συνεπώς η ανωτέρου επιπέδου υπερκλάση, είναι το «Μαθηματικό Αντικείμενο». Τα είδη των γενικών μαθηματικών αντικειμένων που περιλαμβάνονται στο σύνολο είναι ο «Πίνακας», το «Σημείο» και το «Δισδιάστατο Σχήμα», του οποίου

υποκλάσεις είναι το «Πολύγωνο», ο «Κύκλος», το «Τετράγωνο», η «Έλλειψη», το «Παραλληλόγραμμο» και το «Τρίγωνο». Μας απομένουν δύο κλάσεις οι οποίες είναι μεν γεωμετρικά σχήματα, αλλά όχι δισδιάστατα: ο «Κύλινδρος» και η «Σφαίρα». Μπορούμε λοιπόν να εισαγάγουμε μια επιπλέον κλάση «Τρισδιάστατο Σχήμα», υποκλάσεις της οποίας θα είναι τα σχήματα αυτά. Επιπλέον, το «Δισδιάστατο Σχήμα» και το «Τρισδιάστατο Σχήμα» μπορούν να οργανωθούν υπό την γενικότερη κλάση «Γεωμετρικό Σχήμα», η οποία και αποτελεί ένα Μαθηματικό Αντικείμενο. Με βάση τις παραπάνω παρατηρήσεις προκύπτει η ιεραρχία κληρονομικότητας Μαθηματικών Αντικειμένων της Εικόνας 4.12 ως επέκταση της αρχικής μας ιεραρχίας.



Εικόνα 4.12. Ιεραρχία κλάσεων Μαθηματικών Αντικειμένων

Δραστηριότητα 8: Οργάνωση αντικειμένων σε ιεραρχία κληρονομικότητας

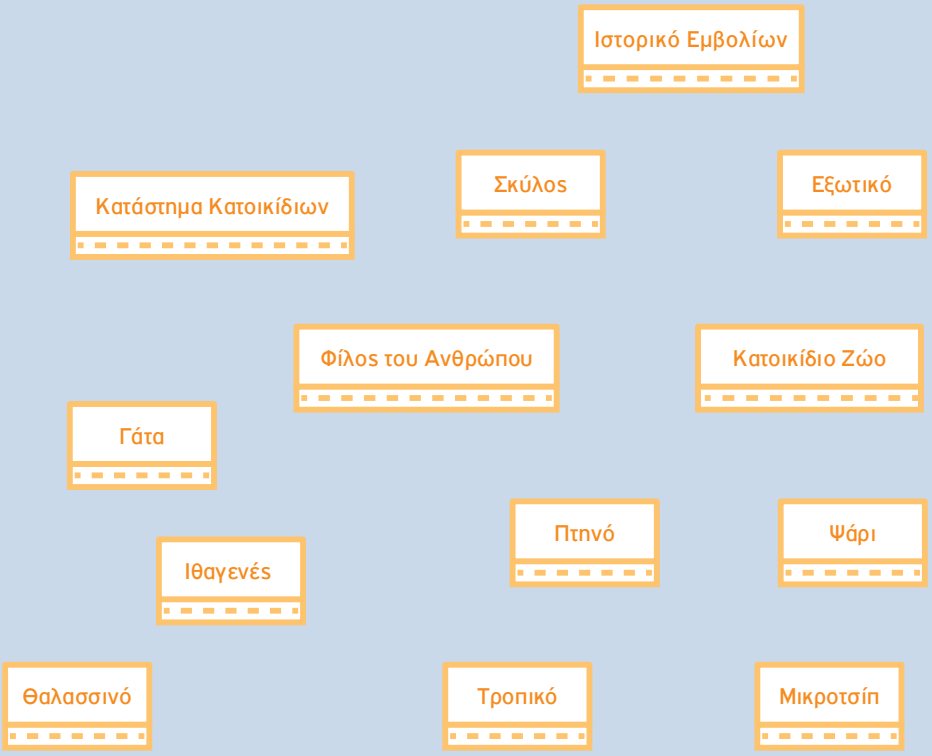
Να οργανώσετε το παρακάτω σύνολο αντικειμένων σε ιεραρχία κληρονομικότητας κλάσεων. «Τηλέφωνο», «Τηλεφωνική γραμμή», «Τηλεφωνική κλήση», «Ψηφιακή γραμμή», «Καλούμενος», «Υπηρεσία», «Συνδιάσκεψη», «Αναμονή κλήσης», «Προώθηση κλήσης», «Φωνητικό ταχυδρομείο», «Καλών», «Τηλεφωνικός Αριθμός».

Υποδείξεις:

- Για κάθε σύνολο στοιχείων μπορεί να προκύπτουν περισσότερες από μία διαφορετικές ιεραρχίες
- Σε κάποιες περιπτώσεις θα χρειαστεί να προσθέσετε επιπλέον κλάσεις, ως υπερκλάσεις
- Θυμηθείτε να εφαρμόζετε τον κανόνα “είναι ένα” (is_a)

Δραστηριότητα 9: Σχεδίαση διαγράμματος κλάσεων με ιεραρχία κληρονομικότητας

Έστω οι κλάσεις αντικειμένων της εικόνας 4.13 που αναφέρονται σε μια εφαρμογή για καταστήματα κατοικίδιων ζώων: «σκύλους», «γάτες», «πτηνά (ιθαγενή και εξωτικά)» και «ψάρια (τροπικά και θαλασσινά)». Για κάθε κατοικίδιο ζώο το κατάστημα κρατάει ένα ιστορικό εμβολίων. Θεωρείστε ότι οι φίλοι του ανθρώπου είναι οι σκύλοι και οι γάτες. Σε καθένα από αυτά τα ζώα είναι εμφυτευμένο ένα μικροτσίπ με σκοπό την ανεύρεσή του σε περίπτωση που χαθεί.
Να συνδέσετε τις κλάσεις αντικειμένων μεταξύ τους χρησιμοποιώντας τις κατάλληλες σχέσεις.



Εικόνα 4.13. Κλάσεις της εφαρμογής «Κατάστημα Κατοικίδιων Ζώων»

4.5 Ορίζοντας την Κατάλληλη Συμπεριφορά: Πολυμορφισμός

Πολυμορφισμός είναι η ικανότητα να συμπεριφερόμαστε διαφορετικά ανάλογα με το αντίστοιχο πλαίσιο.

Δραστηριότητα 10: Προσαρμογή συμπεριφοράς

Εντοπίστε τις διαφορές στη συμπεριφορά σας ανάλογα με την περίπτωση. Ας υποθέσουμε ότι βρίσκεστε στο σχολείο, στην αίθουσα διδασκαλίας. Εκείνη τη στιγμή συμπεριφέρεστε ως μαθητής/μαθήτρια. Στο διάλειμμα, με την παρέα σας, συμπεριφέρεστε ως φίλος/φίλη. Αν πάτε να ψωνίσετε, θα συμπεριφερθείτε ως πελάτης. Στο σπίτι σας, μπροστά στους γονείς σας συμπεριφέρεστε ως γιός/κόρη. Μπορείτε να αναγνωρίσετε ανάλογη συμπεριφορά στους εκπαιδευτικούς σας;

Σε ένα αυτοκίνητο τα πεντάλ του γκαζιού και του φρένου υποστηρίζουν τη συμπεριφορά «πάτησε» του οδηγού. Πατώντας πεντάλ ο οδηγός δίνει μια εντολή για την εκτέλεση μιας εργασίας, αλλά η εφαρμογή της ενέργειας είναι διαφορετική σε κάθε πεντάλ. Όταν ο οδηγός πατήσει το γκάζι, το αυτοκίνητο αναπτύσσει ταχύτητα, ενώ το πάτημα του φρένου το επιβραδύνει. Η ίδια συμπεριφορά του οδηγού επιφέρει διαφορετικό αποτέλεσμα.
Μήπως και το κινητό σας παρουσιάζει συμπτώματα πολυμορφισμού; Άλλοτε συμπεριφέρεται ως τηλέφωνο, άλλοτε ως κάμερα, άλλοτε ως mp3-player και άλλοτε ως ραδιόφωνο. Μήπως πατάτε το ίδιο κουμπί για να ανοίξει και να κλείσει; Είναι και το κουμπί πολυμορφικό!

Παρατηρούμε λοιπόν ότι πολυμορφισμός σημαίνει πολλές διαφορετικές μορφές ή πολλές διαφορετικές συνθήκες. 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.
Μια άλλη, ευρέως διαδεδομένη μορφή πολυμορφισμού βιώνετε στις οικογένειές σας και γενικότερα σε καταστάσεις που εμπλέκουν στοιχεία ιεραρχίας και κληρονομικότητας. Για παράδειγμα, του χρόνου θα μάθετε οδήγηση από τον πατέρα σας ή τον θείο σας. Στο τέλος όμως θα οδηγείτε το ίδιο όχημα με ένα διαφορετικό τρόπο από εκείνους! Μάθαμε από τους γονείς μας να κάνουμε κάτι, το οποίο προσαρμόζουμε στη δική μας προσωπικότητα, στη δική μας προσέγγιση. Κάτι αντίστοιχο συμβαίνει με τις συνταγές μαγειρικής που μεταφέρονται από γενιά σε γενιά. Το ίδιο φαγητό, με τα ίδια υλικά, έχει άλλη γεύση όταν το μαγειρεύει η γιαγιά σας και άλλη γεύση όταν το μαγειρεύει η μητέρα σας. Αυτό είναι πολυμορφισμός! Ίδια συνάρτηση (μαγείρεμα), ίδιες παράμετροι (υλικά), αλλά διαφορετικοί αλγόριθμοι (στυλ μαγειρέματος).
Αντίστοιχα με τις συμπεριφορές και τις λειτουργίες των οντοτήτων στον φυσικό κόσμο, οι συμπεριφορές των αντικειμένων στον αντικειμενοστραφή προγραμματισμό μπορούν να είναι επίσης πολυμορ-

φικές. Τα αντικείμενα μπορούν δηλαδή να υποστηρίζουν συμπεριφορές (μεθόδους) με κοινό όνομα και τον ίδιο βασικό σκοπό αλλά με διαφορετική λειτουργική υλοποίηση. Κάθε φορά που καλείται μια πολυμορφική λειτουργία, το πρόγραμμα αποφασίζει ποια από τις διαφορετικές μεθόδους με την ίδια ονομασία θα ενεργοποιηθεί, με βάση την κλάση του αντικειμένου στην οποία απευθύνεται η εφαρμογή της λειτουργίας.

Πολυμορφισμός (polymorphism) είναι μια ιδιότητα του αντικειμενοστραφούς προγραμματισμού με την οποία μια λειτουργία μπορεί να υλοποιείται με πολλούς διαφορετικούς τρόπους.

Παράδειγμα: «Πολυμορφισμός αριθμητικού τελεστή»

Στον προγραμματισμό πιθανόν να έχετε ήδη συναντήσει το φαινόμενο του πολυμορφισμού χωρίς να το αντιληφθείτε! Για παράδειγμα, όταν χρησιμοποιείτε το σύμβολο «+». Σε ορισμένες γλώσσες προγραμματισμού, ο αριθμητικός τελεστής «+» προσαρμόζεται ή συμπεριφέρεται διαφορετικά, σύμφωνα με τον τύπο των δεδομένων που καλείται να «προσθέσει». Αν του δοθούν δύο αριθμοί, το 20 και το 40, τότε θα τους προσθέσει και θα βγάλει ως αποτέλεσμα το 60. Αν όμως του δοθούν δύο συμβολοσειρές, όπως το «Γεια » και το «σας», τότε ο αριθμητικός τελεστής «+» θα προσαρμοστεί στα νέα δεδομένα, θα αλλάξει δηλαδή μορφή και αντί για την κλασική πρόσθεση θα εκτελέσει μία άλλη πράξη, που έχει νόημα στον χώρο των συμβολοσειρών, και είναι αυτή της συνένωσης. Έτσι, η πρόσθεση των συμβολοσειρών «Γεια » και «σας» θα έχει ως αποτέλεσμα τη συμβολοσειρά «Γεια σας».
Όπως παρατηρείτε, δε χρησιμοποιούμε διαφορετικό τελεστή όταν τα δεδομένα μας είναι συμβολοσειρές και όχι αριθμοί. Απλώς αλλάζει η συμπεριφορά του τελεστή. Παρόμοια, αν στον τελεστή «+» δοθεί η συμβολοσειρά «Καλώς ήρθες » και ο αριθμός 2020, τότε θα μετατρέψει τον αριθμό 2020 στη συμβολοσειρά «2020» και θα πραγματοποιήσει στη συνέχεια την συνένωση των συμβολοσειρών «Καλώς ήρθες » και «2020». Το αποτέλεσμα της «πρόσθεσης» αυτής θα είναι η συμβολοσειρά «Καλώς ήρθες 2020».
Το παραπάνω παράδειγμα αποτελεί ένα απλό παράδειγμα πολυμορφισμού, δηλαδή αλλαγής συμπεριφοράς.
Αν υποστηρίζονταν από τη γλώσσα προγραμματισμού, θα μπορούσαμε να ορίσουμε και μία νέα συνάρτηση με το όνομα «Πρόσθεση» και συμπεριφορά ανάλογη με τα δεδομένα εισόδου, όπως παρουσιάζονται στον Πίνακα 4.1.

Πίνακας 4.1. Περιπτώσεις κλήσης και αποτελέσματα της συνάρτησης «Πρόσθεση()»

Συνάρτηση Πρόσθεση (a, b) ΑΡΧΗ ... ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ	
Είσοδος	**Έξοδος**
Πρόσθεση(20,40)	60
Πρόσθεση(«Γεια », «σας!»)	«Γεια σας!»
Πρόσθεση(«Καλώς ήρθες », 2019)	«Καλώς ήρθες 2019»

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

Παράδειγμα: «Σχεδιασμός Εικόνων με Γεωμετρικά Σχήματα»

Στην υποενότητα 4.4 σχεδιάσαμε το διάγραμμα κλάσεων των γεωμετρικών σχημάτων και συναντήσαμε τη μέθοδο «ΥπολογισμόςΕμβαδού()» στα διάφορα σχήματα. Η κλάση καθενός από τα σχήματα διαθέτει την ίδια ακριβώς μέθοδο, ο υπολογισμός όμως του εμβαδού για κάθε σχήμα γίνεται με διαφορετικό μαθηματικό τύπο (Πίνακας 4.2).

Πίνακας 4.2. Υπολογισμός εμβαδού γεωμετρικών σχημάτων

Τρίγωνο	Παραλληλόγραμμο	Κύκλος
Εμ <- Βάση*Υψος/2	Εμ <- Μήκος*Πλάτος	Εμ <- 3.14*Ακτίνα*Ακτίνα

Η λειτουργία αυτή είναι συνεπώς πολυμορφική και, κατά την εκτέλεση της εφαρμογής, ενεργοποιείται η κατάλληλη μέθοδος υπολογισμού του εμβαδού ανάλογα με το γεωμετρικό σχήμα.
Με αφορμή το παράδειγμα του υπολογισμού εμβαδού είναι ενδιαφέρον να δούμε πώς γράφονται οι μέθοδοι. Ουσιαστικά πρόκειται για υποπρογράμματα, όπως αυτά που έχετε ήδη μάθει να γράφετε. 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 στον Πίνακα 4.3.

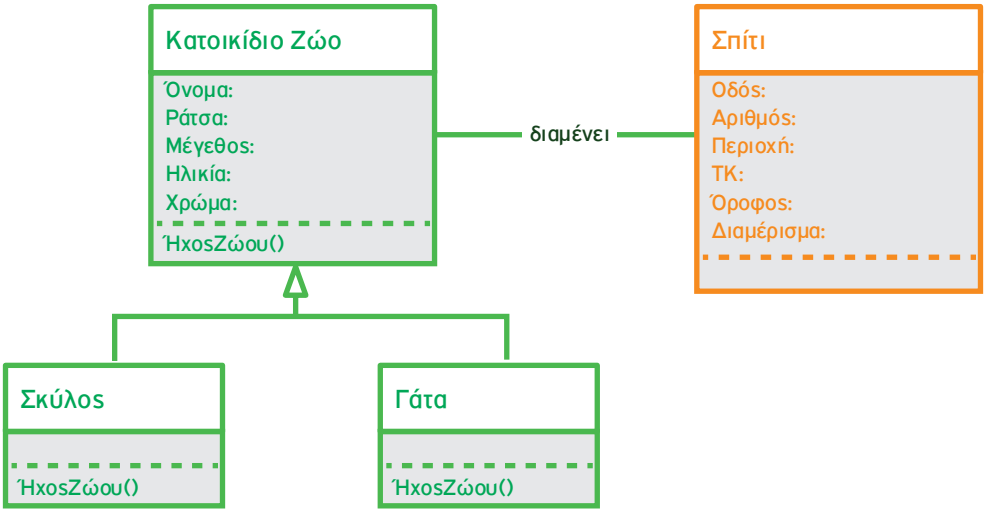
Πίνακας 4.3. Κώδικας υλοποίησης της μεθόδου «ΥπολογισμόςΕμβαδού()» των τριών γεωμετρικών σχημάτων

Τρίγωνο	Παραλληλόγραμμο	Κύκλος

ΣΥΝΑΡΤΗΣΗ
ΥπολογισμόςΕμβαδού():
ΠΡΑΓΜΑΤΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ
  ΠΡΑΓΜΑΤΙΚΕΣ:Εμ
ΑΡΧΗ
Εμ <- Βάση*Ύψος/2
ΥπολογισμόςΕμβαδού <- Εμ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
	ΣΥΝΑΡΤΗΣΗ
	ΥπολογισμόςΕμβαδού():
	ΠΡΑΓΜΑΤΙΚΗ
	ΜΕΤΑΒΛΗΤΕΣ
	  ΠΡΑΓΜΑΤΙΚΕΣ:Εμ
	ΑΡΧΗ
	Εμ <- Μήκος*Πλάτος
	ΥπολογισμόςΕμβαδού<-Εμ
	ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
		ΣΥΝΑΡΤΗΣΗ
		ΥπολογισμόςΕμβαδού():
		ΠΡΑΓΜΑΤΙΚΗ
		ΣΤΑΘΕΡΕΣ
		  Π=3.14
		ΜΕΤΑΒΛΗΤΕΣ
		  ΠΡΑΓΜΑΤΙΚΕΣ:Εμ
		ΑΡΧΗ
		Εμ <- Π*Ακτίνα*Ακτίνα
		ΥπολογισμόςΕμβαδού <- Εμ
		ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

Παράδειγμα: «Η Φάρμα των Ζώων»

Έστω ότι στο ηλεκτρονικό παιχνίδι «Η Φάρμα των Ζώων» της υποενότητας 4.3 θέλουμε να τηρούμε δεδομένα για τα σπίτια (Οδός, Αριθμός, Περιοχή, ΤΚ, Όροφος, Διαμέρισμα), στα οποία διαμένουν τα κατοικίδια ζώα (σκύλοι και γάτες) που έχουμε δημιουργήσει. Υπενθυμίζουμε ότι κάθε κατοικίδιο ζώο ταυτοποιείται με βάση το όνομα, τη ράτσα, το μέγεθος, την ηλικία και το χρώμα του ενώ, επιπλέον των υπολοίπων, διαθέτει τη λειτουργία «ΉχοςΖώου()», η οποία ενεργοποιεί τον κατάλληλο ήχο για κάθε είδος κατοικίδιου.
Για την υποστήριξη της παραπάνω λειτουργικότητας απαιτείται ο ορισμός των κλάσεων «Σπίτι», «Κατοικίδιο Ζώο», «Σκύλος» και «Γάτα» συμπεριλαμβάνοντας ως ιδιότητες τα δεδομένα κάθε κλάσης και τη μέθοδο «ΉχοςΖώου()». Η διαγραμματική αναπαράσταση των παραπάνω κλάσεων του ηλεκτρονικού παιχνιδιού και των μεταξύ τους σχέσεων δίνεται στην Εικόνα 4.14.



Εικόνα 4.14. Διάγραμμα κλάσεων ηλεκτρονικού παιχνιδιού «Η Φάρμα των Ζώων»

Παρατηρούμε ότι η μέθοδος «ΉχοςΖώου()» είναι πολυμορφική, διότι κάθε ζώο κάνει διαφορετικό ήχο και συνεπώς η λειτουργία αυτή πρέπει να αναπαράγει διαφορετικό αρχείο ήχου ανά περίπτωση. Εάν υποτεθεί ότι έχουμε στη διάθεσή μας τη συνάρτηση «ΑναπαραγωγήΉχου("όνομα_αρχείου_ήχου.mp4")» και τα αρχεία «νιάου.mp4» και «γαβ.mp4», τότε ο Πίνακας 4.4 αποτυπώνει τον κώδικα υλοποίησης της πολυμορφικής συνάρτησης για τις περιπτώσεις του σκύλου και της γάτας.

Πίνακας 4.4. Κώδικας υλοποίησης της μεθόδου «ΉχοςΖώου()» για τον σκύλο και τη γάτα

Σκύλος	Γάτα
ΔΙΑΔΙΚΑΣΙΑ ΉχοςΖώου() **ΑΡΧΗ** **ΚΑΛΕΣΕ** ΑναπαραγωγήΉχου("γαβ.mp4") **ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ**	**ΔΙΑΔΙΚΑΣΙΑ** ΉχοςΖώου() **ΑΡΧΗ** **ΚΑΛΕΣΕ** ΑναπαραγωγήΉχου("νιάου.mp4") **ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ**

4.6 Ερωτήσεις - Ασκήσεις

Ε.1: Απαντήστε στις ακόλουθες ερωτήσεις πολλαπλής επιλογής

1. Η διαδικασία με την οποία ένα αντικείμενο αποκτά χαρακτηριστικά από άλλο αντικείμενο ονομάζεται:
   - ❒ Ενθυλάκωση
   - ❒ Κληρονομικότητα
   - ❒ Πολυμορφισμός
2. Η διαδικασία απόκρυψης λεπτομερειών στην γλώσσα του αντικειμενοστραφούς προγραμματισμού αναφέρεται και ως:
   - ❒ Ενθυλάκωση
   - ❒ Πολυμορφισμός
   - ❒ Κληρονομικότητα
3. Η ενθυλάκωση υποδηλώνει ότι οι εσωτερικές λειτουργίες ενός αντικειμένου είναι ορατές στον έξω κόσμο.
   - ❒ Σωστό
   - ❒ Λάθος

Ε.2: Αντιστοιχίστε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β

Στήλη Α				**Στήλη Β**
Ενθυλάκωση	1	.....	Α	Δυνατότητα αντικειμένων να διαθέτουν μεθόδους με ίδιο όνομα αλλά διαφορετική υλοποίηση
Κληρονομικότητα	2	.....	Β	Δυνατότητα αντικειμένου να συνδυάζει εσωτερικά δεδομένα και μεθόδους
Πολυμορφισμός	3	.....	Γ	Δυνατότητα δημιουργίας ιεραρχίας αντικειμένων

Ε.3: Σε κάθε μία από τις παρακάτω προτάσεις, να συμπληρώσετε τα κενά με τις λέξεις που λείπουν:

1. Ένα αντικείμενο περιέχει ..... και .....
2. Ένα αντικείμενο εκτελεί ενέργειες μέσω των .....
3. Ο γενικός τύπος ενός αντικειμένου καλείται .....
4. Ένα αντικειμενοστραφές πρόγραμμα δομείται ως ένα δίκτυο συνεργαζόμενων οντοτήτων που είναι τα .....
5. ..... ονομάζεται η δυνατότητα ενός αντικειμένου να συνδυάζει εσωτερικά δεδομένα και μεθόδους χειρισμού του αντικειμένου.

Ε.4: Ποια από τις παρακάτω προτάσεις περιγράφει με ακρίβεια τη σχέση μεταξύ ενός αντικειμένου προγόνου και απόγονου;

- ❒ Ένα αντικείμενο πρόγονος περιέχει τα ίδια ακριβώς χαρακτηριστικά με το παιδί του.
- ❒ Ένα αντικείμενο απόγονος δεν σχετίζεται με τον γονέα του.
- ❒ Ένα αντικείμενο πρόγονος κληρονομεί χαρακτηριστικά και συμπεριφορά από το παιδί του.
- ❒ Ένα αντικείμενο απόγονος περιέχει χαρακτηριστικά από τον γονέα του αλλά μπορεί να οριστούν και πρόσθετα χαρακτηριστικά.

Ε.5: Να οργανώσετε καθένα από τα παρακάτω σύνολα αντικειμένων σε ιεραρχίες κληρονομικότητας κλάσεων.

1. «Όχημα», «Αυτοκίνητο», «Αγωνιστικό αυτοκίνητο», «Αεροπλάνο», «Αμφίβιο Όχημα», «Μηχανή», «Μηχανή αεροπλάνου», «Ηλεκτρικός κινητήρας», «Τροχός», «Μεταφορά», «Φορτηγό», «Ποδήλατο»
2. «Φοιτητής», «Μάθημα», «Καθηγητής», «Τελειόφοιτος», «Ενότητα Μαθήματος», «Βοηθός διδάσκοντα», «Βοηθός διοίκησης», «Τεχνικός», «Τάξη», «Κτήριο», «Γυμναστήριο», «Φροντιστηριακό μάθημα», «Εξέταση», «Εργαστήριο», «Αίθουσα Συνεδριάσεων»

Υποδείξεις:

- Δεν απαιτείται αποτύπωση ιδιοτήτων και μεθόδων.
- Για κάθε σύνολο στοιχείων προκύπτουν περισσότερες της μιας διαφορετικές ιεραρχίες.
- Σε κάποιες περιπτώσεις θα χρειαστεί να προσθέσετε επιπλέον κλάσεις, ως υπερκλάσεις.
- Θυμηθείτε να εφαρμόζετε τον κανόνα “είναι ένα” (is_a).

Ε.6: Εκπαιδευτικοί και Μαθητές Σχολικής Μονάδας

Για την υποστήριξη της λειτουργίας των σχολείων έχει αναπτυχθεί κεντρικό πληροφοριακό σύστημα από το Υπουργείο Παιδείας. Στην έναρξη κάθε σχολικής χρονιάς, οι εκπαιδευτικοί αναλαμβάνουν υπηρεσία στο σχολείο τοποθέτησής τους. Αντίστοιχα, οι μαθητές εγγράφονται ή ανανεώνουν την εγγραφή τους στο σχολείο φοίτησής τους.

Με βάση το παραπάνω σενάριο, αναγνωρίστε τις κλάσεις που πρέπει να υλοποιηθούν στο κεντρικό πληροφοριακό σύστημα του Υπουργείου Παιδείας. Για κάθε κλάση καθορίστε τις ιδιότητες και μεθόδους που απαιτούνται για την υλοποίηση του παραπάνω σεναρίου. Επίσης, καθορίστε τις σχέσεις μεταξύ των κλάσεων και δημιουργήστε το αντίστοιχο διάγραμμα.

Ε.7: Ανάρτηση σε Κοινωνικό Δίκτυο

Ένα συνηθισμένο σενάριο χρήσης ενός κοινωνικού δικτύου είναι το ακόλουθο: ο χρήστης συνδέεται, δημιουργεί μια νέα ανάρτηση, αποδέχεται ή απορρίπτει αιτήματα άλλων χρηστών, σχολιάζει αναρτήσεις άλλων χρηστών και τέλος αποσυνδέεται.

Καθορίστε τις κλάσεις που απαιτούνται για την υποστήριξη του παραπάνω σεναρίου από το πληροφοριακό σύστημα του κοινωνικού δικτύου. Για κάθε κλάση ορίστε τις ιδιότητες και τις μεθόδους που απαιτούνται. Επίσης, εντοπίστε τις σχέσεις μεταξύ των κλάσεων και δημιουργήστε το κατάλληλο διάγραμμα.

Ε.8: Εφαρμογή Διαχείρισης Εισιτηρίων Κινηματογράφου

Σχεδιάσετε τη διαγραμματική αναπαράσταση κλάσεων για ένα σύστημα διαχείρισης των εισιτηρίων ενός κινηματογράφου που διαθέτει αίθουσες προβολής. Κάθε αίθουσα προβάλλει ταινίες. Για κάθε ταινία ο θεατής μπορεί να εκδώσει κανονικό ή μειωμένο εισιτήριο. Εντοπίστε τις κλάσεις και τις μεταξύ τους σχέσεις χωρίς να αναφερθείτε σε ιδιότητες και μεθόδους των κλάσεων.

Ε.9: Πληροφοριακό Σύστημα Σχολικής Μονάδας

Το Υπουργείο Παιδείας διαθέτει κεντρικό πληροφοριακό σύστημα για την υποστήριξη της λειτουργίας των σχολικών μονάδων. Μια από τις βασικές λειτουργίες του συστήματος είναι η τήρηση πληροφοριών για τους εκπαιδευτικούς και τους μαθητές κάθε σχολείου. Τα σημαντικότερα στοιχεία που καταχωρίζονται στο πληροφοριακό σύστημα είναι τα ακόλουθα:

Εκπαιδευτικός: όνομα, επώνυμο, όνομα πατέρα, όνομα μητέρας, ημ/νία γέννησης, διεύθυνση, τηλέφωνο, αριθμός δελτίου ταυτότητας, αριθμός μητρώου εκπαιδευτικού, ειδικότητα, ημ/νία διορισμού.

Μαθητής: όνομα, επώνυμο, όνομα πατέρα, όνομα μητέρας, ημ/νία γέννησης, διεύθυνση, τηλέφωνο, αριθμός μητρώου μαθητή, αριθμός δημοτολογίου, ημ/νία εγγραφής, τάξη εγγραφής.

Σύμφωνα με τα παραπάνω, εντοπίστε τις κλάσεις που πρέπει να υλοποιηθούν στο πληροφοριακό σύστημα, καθορίζοντας τις ιδιότητες κάθε κλάσης. Στη συνέχεια οργανώστε τις κλάσεις σε μια ιεραρχία, μεταφέροντας τις κοινές ιδιότητες στην υπερκλάση.

Ε.10: Πληροφοριακό Σύστημα Πανεπιστημιακής Βιβλιοθήκης

Μια πανεπιστημιακή βιβλιοθήκη χρησιμοποιεί ένα πληροφοριακό σύστημα για την υποστήριξη της λειτουργίας της. Στο πληροφοριακό σύστημα καταχωρίζονται τα στοιχεία των τεκμηρίων που διαθέτει η βιβλιοθήκη. Επίσης, μέσω του πληροφοριακού συστήματος γίνεται η διαχείριση του δανεισμού των τεκμηρίων. Οι βασικές κατηγορίες τεκμηρίων και οι πληροφορίες που τηρούνται για καθένα από αυτά περιγράφονται στη συνέχεια:

- **Βιβλίο**: Κωδικός, Συγγραφείς, Τίτλος, Έτος έκδοσης, Εκδοτικός οίκος, Τόπος έκδοσης, Γλώσσα, Πλήθος σελίδων, ISBN
- **Συλλογικός Τόμος**: Κωδικός, Επιμελητές έκδοσης, Τίτλος, Έτος έκδοσης, Εκδοτικός οίκος, Τόπος έκδοσης, Γλώσσα, Πλήθος σελίδων, ISBN
- **Επιστημονικό Περιοδικό**: Κωδικός, Τίτλος, Έτος έκδοσης, Αριθμός τεύχους, Εκδοτικός οίκος, Τόπος έκδοσης, Γλώσσα, Πλήθος σελίδων, ISSN

Κάθε τεκμήριο μπορεί να ζητηθεί για δανεισμό (αν δεν είναι διαθέσιμο), να δανειστεί, να επιστραφεί από δανεισμό.

Με βάση την παραπάνω περιγραφή, καταγράψτε τις κλάσεις αντικειμένων καθώς και τις ιδιότητες και μεθόδους κάθε κλάσης. Στη συνέχεια οργανώστε τις κλάσεις σε μια ιεραρχία.

Ε.11: Υπολογισμός Τελικής Βαθμολογίας Μαθήματος

Στο Λύκειο τα μαθήματα χωρίζονται σε δύο κατηγορίες: εκείνα που δεν εξετάζονται γραπτώς στις ενδοσχολικές εξετάσεις στο τέλος της σχολικής χρονιάς και εκείνα που εξετάζονται. Στα μαθήματα που δεν εξετάζονται γραπτώς, ο τελικός βαθμός είναι ο προφορικός βαθμός του μαθήματος, δηλαδή ο μέσος όρος της βαθμολογίας των δύο τετραμήνων. Στα μαθήματα που εξετάζονται γραπτώς, ο τελικός βαθμός προκύπτει από τον μέσο όρο του προφορικού βαθμού (μ.ο. της βαθμολογίας των δύο τετραμήνων) με το βαθμό της γραπτής εξέτασης.

Επιλέξετε τη σωστή απάντηση.

Το παραπάνω σενάριο αποτελεί παράδειγμα...

- ❒ ενθυλάκωσης
- ❒ κληρονομικότητας
- ❒ πολυμορφισμού
- ❒ αφηρημένου τύπου

Ενότητα 5

Εκσφαλματωση Προγραμματος

Κατηγορίες Λαθών

Εκσφαλμάτωση

Ενότητα 5. Εκσφαλμάτωση Προγράμματος

5.1 Κατηγορίες Λαθών

Μέχρι στιγμής έχετε αποκτήσει κάποια εμπειρία ανάπτυξης προγραμμάτων. Χρειάστηκε να έρθετε αντιμέτωποι με ένα πρόβλημα, να προσπαθήσετε να το κατανοήσετε και να το αναλύσετε, να διακρίνετε τα δεδομένα και τα ζητούμενα, να προσδιορίσετε τον αλγόριθμο επίλυσής του και να συντάξετε το πρόγραμμα που υλοποιεί τον αλγόριθμο επίλυσης. Όσες φορές προσπαθήσατε να εκτελέσετε το πρόγραμμά σας σε ένα προγραμματιστικό περιβάλλον, η εκτέλεση ήταν πάντα επιτυχής; Το μεταφραστικό πρόγραμμα δεν εντόπισε κάποιο λάθος; Δε συνέβη κάποιες φορές το πρόγραμμά σας να εκτελείται μεν, αλλά να μη βγάζει σωστά αποτελέσματα για όλες τις περιπτώσεις δεδομένων εισόδου; Πώς διαχειριστήκατε την κατάσταση αυτή;
Ακόμη και πολύ ικανοί επαγγελματίες προγραμματιστές κάνουν λάθη στην ανάπτυξη των προγραμμάτων. Μπορούμε να διακρίνουμε τις εξής κατηγορίες λαθών:

- Συντακτικά λάθη
- Λάθη που οδηγούν σε αντικανονικό τερματισμό του προγράμματος
- Λογικά λάθη που παράγουν λανθασμένα αποτελέσματα

5.1.1 Συντακτικά λάθη

Κάποιες φορές ένα πρόγραμμα δεν μπορεί να εκτελεστεί, επειδή κατά τη μετάφραση εντοπίζονται συντακτικά λάθη. Π.χ. δε γράψαμε σωστά μια δεσμευμένη λέξη, παραλείψαμε μια δεσμευμένη λέξη ή παραλείψαμε να δηλώσουμε μια μεταβλητή.

Παράδειγμα 1 – Παράδειγμα εντοπισμού συντακτικών λαθών

Προσπαθήστε να εντοπίσετε τα συντακτικά λάθη στο παρακάτω πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.1]) και να προτείνετε διορθώσεις. Στη συνέχεια διασταυρώστε την απάντησή σας με αυτή που ακολουθεί.

Κώδικας σε ΓΛΩΣΣΑ [5. 1]. Πρόγραμμα με συντακτικά λάθη

```
1    ΠΡΟΓΡΑΜΜΑ Άθροισμα_θετικών_αριθμών
2    ΜΕΤΑΒΛΗΤΕΣ
3      ΑΚΕΡΑΙΕΣ: Σ,Χ,Ι
4      Σ <- 0
5      ΓΙΑ Ι ΑΠΟ 1 ΕΩΣ 10
6        ΓΡΑΨΕ 'Δώσε έναν ακέραιο αριθμό'
7        ΔΙΑΒΑΣΕ Χ
8        ΑΝ Χ > 0
9          Σ <- Σ + Χ
```

```
10        ΤΕΛΟΣ_ΑΝ
11      ΓΡΑΨΕ 'Σ=', Σ
12    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Απάντηση:

Ακολουθεί το πρόγραμμα με διορθωμένα τα συντακτικά λάθη (κώδικας σε ΓΛΩΣΣΑ [5.2]).

Κώδικας σε ΓΛΩΣΣΑ [5. 2]. Το πρόγραμμα με διορθωμένα τα συντακτικά λάθη.

```
1     ΠΡΟΓΡΑΜΜΑ Άθροισμα_θετικών_αριθμών
2     ΜΕΤΑΒΛΗΤΕΣ
3       ΑΚΕΡΑΙΕΣ: Σ,Χ,Ι
4     ΑΡΧΗ
5       Σ <- 0
6       ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
7         ΓΡΑΨΕ 'Δώσε έναν ακέραιο αριθμό'
8         ΔΙΑΒΑΣΕ Χ
9         ΑΝ Χ > 0 ΤΟΤΕ
10          Σ <- Σ + Χ
11        ΤΕΛΟΣ_ΑΝ
12      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
13      ΓΡΑΨΕ 'Σ=', Σ
14    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Δραστηριότητα 1 – Εντοπισμός συντακτικών λαθών

Προσπαθήστε να εντοπίσετε τα συντακτικά λάθη στο παρακάτω πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.3]) και να προτείνετε διορθώσεις.

Κώδικας σε ΓΛΩΣΣΑ [5.3]. Πρόγραμμα με συντακτικά λάθη

```
1     ΠΡΟΓΡΑΜΜΑ Άθροισμα_θετικών_αριθμών
2     ΜΕΤΑΒΛΗΤΕΣ
3       ΠΡΑΓΜΑΤΙΚΕΣ: Χ
4     ΑΡΧΗ
5       ΓΡΑΨΕ 'Χ='
```

```
6       ΔΙΑΒΑΣΕ Χ
7       Σ <- 0
8       ΟΣΟ Χ > 0
9         Σ <- Σ + Χ
10        ΓΡΑΨΕ 'Χ='
11        ΔΙΑΒΑΣΕ Χ
12      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
13      ΕΑΝ Σ > 0 ΤΟΤΕ
14        ΓΡΑΨΕ 'Σ=', Σ
15      ΑΛΛΙΩΣ
16        ΓΡΑΨΕ 'Δεν δόθηκαν αυστηρά θετικοί αριθμοί'
17      ΤΕΛΟΣ_ΑΝ
18    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

5.1.2 Λάθη που οδηγούν σε αντικανονικό τερματισμό του προγράμματος

Ένα πρόγραμμα μπορεί να τερματίσει αντικανονικά λόγω διαφόρων λαθών. Για παράδειγμα, αν επιχειρήσουμε να διαιρέσουμε με το μηδέν ή αν κατά την ανάγνωση ενός ακεραίου αριθμού εισαχθεί ένα γράμμα.

Παράδειγμα 2 – Παράδειγμα εντοπισμού λαθών που μπορεί να οδηγήσουν σε αντικανονικό τερματισμό

Ακολουθεί η εκφώνηση για την ανάπτυξη ενός προγράμματος:
«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει το ύψος των μελών ενός χορευτικού ομίλου και να υπολογίζει τον μέσο όρο τους. Το ύψος κάθε μέλους δίνεται σε μέτρα. Για να σταματήσει η εισαγωγή των δεδομένων ο χρήστης πρέπει να εισάγει μια αρνητική τιμή».
Δίνεται το πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.4]) και καλείστε να εντοπίσετε τυχόν λάθη που θα μπορούσαν να οδηγήσουν σε αντικανονικό τερματισμό του προγράμματος και να προτείνετε διορθώσεις.
Τι θα συμβεί αν ο χρήστης κατά την πρώτη ανάγνωση του ύψους δώσει την τιμή -1;
Διασταυρώστε την απάντησή σας με αυτή που ακολουθεί.

Κώδικας σε ΓΛΩΣΣΑ [5.4]. Παράδειγμα λάθους πιθανού αντικανονικού τερματισμού

```
1   ΠΡΟΓΡΑΜΜΑ Μέσος_όρος_ύψους
2   ΜΕΤΑΒΛΗΤΕΣ
3     ΑΚΕΡΑΙΕΣ: πλήθος
4     ΠΡΑΓΜΑΤΙΚΕΣ: ύψος, Σ, ΜΟ
```

```
5   ΑΡΧΗ
6     Σ <- 0
7     πλήθος <- 0
8     ΓΡΑΨΕ 'Δώσε ύψος:'
9     ΔΙΑΒΑΣΕ ύψος
10    ΟΣΟ ύψος > 0 ΕΠΑΝΑΛΑΒΕ
11      Σ <- Σ + ύψος
12      πλήθος <- πλήθος + 1
13      ΓΡΑΨΕ 'Δώσε ύψος:'
14      ΔΙΑΒΑΣΕ ύψος
15    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
16    ΜΟ <- Σ/πλήθος
17    ΓΡΑΨΕ 'Μέσος όρος ύψους:', ΜΟ
18  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Απάντηση:
Αν ο χρήστης κατά την πρώτη ανάγνωση του ύψους δώσει την τιμή -1, κατά την εκτέλεση της εντολής «*ΜΟ<- Σ/πλήθος*», θα επιχειρηθεί διαίρεση με το μηδέν και το πρόγραμμα θα τερματίσει αντικανονικά. Το πρόβλημα θα δημιουργηθεί, επειδή πριν τη διαίρεση δεν έγινε ο απαραίτητος έλεγχος της τιμής του παρονομαστή και παραβιάστηκε το κριτήριο της καθοριστικότητας.
Το λάθος αυτό θα μπορούσε να αντιμετωπιστεί με τη χρήση των παρακάτω εντολών (κώδικας σε ΓΛΩΣΣΑ [5.5]) στη θέση των εντολών των γραμμών 16 και 17.

Κώδικας σε ΓΛΩΣΣΑ [5.5]. Προτεινόμενη λύση για λάθος πιθανού αντικανονικού τερματισμού

```
16    ΑΝ πλήθος = 0 ΤΟΤΕ
17      ΓΡΑΨΕ 'Δεν δόθηκαν στοιχεία μελών'
18    ΑΛΛΙΩΣ
19      ΜΟ <- Σ/πλήθος
20      ΓΡΑΨΕ 'Μέσος όρος ύψους:', ΜΟ
21    ΤΕΛΟΣ_ΑΝ
```

Αν κατά την ανάγνωση του ύψους, ο χρήστης αντί να δώσει μια αριθμητική τιμή, εισαγάγει ένα γράμμα, τότε το πρόγραμμα θα τερματίσει αντικανονικά λόγω λάθους του χρήστη. Στις σύγχρονες γλώσσες προγραμματισμού υπάρχουν τρόποι διαχείρισης τέτοιων λαθών, οι οποίοι δε θα μας απασχολήσουν στο πλαίσιο του μαθήματος αυτού.

5.1.3 Λογικά λάθη

Ακόμη κι αν το πρόγραμμά μας δεν περιέχει συντακτικά λάθη και μπορεί να εκτελεστεί, πρέπει οπωσδήποτε να ελεγχθεί, ώστε να διαπιστώσουμε αν κατά την εκτέλεσή του εμφανίζονται λογικά λάθη. Τα λογικά λάθη έχουν ως συνέπεια το πρόγραμμα σε κάποιες περιπτώσεις να εξάγει λανθασμένα αποτελέσματα. Για να εντοπίσουμε τα λογικά λάθη μπορούμε να κάνουμε δοκιμαστικές εκτελέσεις του προγράμματός μας και να ελέγξουμε αν για συγκεκριμένες τιμές εισόδου, το πρόγραμμά μας εξάγει σωστά αποτελέσματα.

Παράδειγμα 3 – Παράδειγμα εντοπισμού λογικού λάθους

Ακολουθεί η εκφώνηση για την ανάπτυξη ενός προγράμματος:
«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει την τιμή ενός τετραδίου, το πλήθος των τετραδίων που θέλει να αγοράσει ένας μαθητής και το χρηματικό ποσό που έχει διαθέσιμο. Στη συνέχεια, να υπολογίζει το συνολικό ποσό που οφείλει να πληρώσει για να αγοράσει τα τετράδια και ανάλογα με το ποσό που διαθέτει να εμφανίζει ένα από τα παρακάτω μηνύματα: "Η αγορά είναι εφικτή", "Η αγορά δεν είναι εφικτή"».
Δίνεται το πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.6]). Προσπαθήστε να εντοπίσετε τυχόν λογικά λάθη που οδηγούν σε λανθασμένα αποτελέσματα και να προτείνετε διορθώσεις.
Προκειμένου να ελέγξετε την ορθότητα του προγράμματος πραγματοποιήστε δοκιμαστική εκτέλεση με τα παρακάτω δεδομένα.

Τιμές εισόδου:

τιμή=1,2
πλήθος=2
διαθέσιμο ποσό=3

Ερωτήματα

α) Με βάση την εκφώνηση είναι η παραπάνω αγορά εφικτή;
β) Με βάση το πρόγραμμα είναι η παραπάνω αγορά εφικτή;

Διασταυρώστε την απάντησή σας με αυτή που ακολουθεί.

Κώδικας σε ΓΛΩΣΣΑ [5.6]. Παράδειγμα προγράμματος με λογικό λάθος

```
1   ΠΡΟΓΡΑΜΜΑ Αγορά_τετραδίων
2   ΜΕΤΑΒΛΗΤΕΣ
3     ΠΡΑΓΜΑΤΙΚΕΣ: τιμή, πλήθος, διαθέσιμο_ποσό, οφειλόμενο
4   ΑΡΧΗ
5     ΓΡΑΨΕ 'Δώσε την τιμή του τετραδίου:'
6     ΔΙΑΒΑΣΕ τιμή
7     ΓΡΑΨΕ 'Δώσε το πλήθος των τετραδίων'
8     ΔΙΑΒΑΣΕ πλήθος
```

```
 9      ΓΡΑΨΕ 'Δώσε το διαθέσιμο ποσό'
10      ΔΙΑΒΑΣΕ διαθέσιμο_ποσό
11      οφειλόμενο_ποσό <- τιμή + πλήθος
12      ΑΝ οφειλόμενο_ποσό <= διαθέσιμο_ποσό ΤΟΤΕ
13        ΓΡΑΨΕ 'Η αγορά είναι εφικτή'
14      ΑΛΛΙΩΣ
15        ΓΡΑΨΕ 'Η αγορά δεν είναι εφικτή'
16      ΤΕΛΟΣ_ΑΝ
17    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Απάντηση:
α) Σύμφωνα με την εκφώνηση, τα δύο τετράδια κοστίζουν 1,2x2=2,4€. Ο μαθητής διαθέτει 3€, δηλαδή ποσό μεγαλύτερο από το οφειλόμενο. Άρα **η αγορά είναι εφικτή**.
β) Σύμφωνα με το πρόγραμμα, το οφειλόμενο ποσό υπολογίζεται ως εξής: 1,2+2=3,2€. Ο μαθητής διαθέτει 3€, δηλαδή ποσό μικρότερο από το οφειλόμενο. Άρα **η αγορά δεν είναι εφικτή**.

Συμπεράσματα - προτάσεις διορθώσεων:
Το λογικό λάθος δε σχετίζεται με τη συνθήκη ή τις εντολές που εκτελούνται, όταν η συνθήκη είναι αληθής ή ψευδής. Το λογικό λάθος υπάρχει στην εντολή

```
«οφειλόμενο_ποσό <- τιμή + πλήθος»
```

που πρέπει να αντικατασταθεί με την εντολή

```
«οφειλόμενο_ποσό <- τιμή * πλήθος»
```

Μερικές φορές το λογικό λάθος δεν υπάρχει στην εντολή που εμφανίζεται το λανθασμένο αποτέλεσμα, αλλά σε προηγούμενη εντολή.

5.2 Εκσφαλμάτωση

Στην ενότητα αυτή θα ασχοληθούμε με τον εντοπισμό των λογικών λαθών ενός προγράμματος και την εκσφαλμάτωσή του.

5.2.1 Εκσφαλμάτωση λογικών λαθών στις δομές επιλογής

Παρακάτω θα προσπαθήσουμε να διερευνήσουμε μεθοδικά τα λογικά λάθη που εμφανίζονται στις δομές επιλογής.

Σε μια δομή επιλογής μπορεί να εμφανιστούν λογικά λάθη που σχετίζονται με:

- τη συνθήκη ή τις συνθήκες
- τις ομάδες εντολών που εκτελούνται όταν μια συνθήκη είναι αληθής ή ψευδής.

Όπως είδαμε παραπάνω, μερικές φορές κατά την εκτέλεση της δομής επιλογής εμφανίζονται λανθασμένα αποτελέσματα, τα οποία σχετίζονται με λογικά λάθη σε προηγούμενες εντολές, που επηρεάζουν την τιμή που λαμβάνει η συνθήκη.

Παράδειγμα 4 – Παράδειγμα εκσφαλμάτωσης λογικών λαθών σε δομή επιλογής

Ακολουθεί η εκφώνηση για την ανάπτυξη ενός προγράμματος:
«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει την ηλικία ενός επιβάτη αστικού λεωφορείου σε έτη και ανάλογα με την τιμή της ηλικίας του, να υπολογίζει το αντίτιμο του εισιτηρίου που πρέπει να πληρώσει. Εάν έχει συμπληρώσει το 18ο έτος της ηλικίας του, ο επιβάτης πληρώνει κανονικό εισιτήριο 1€. Διαφορετικά, πληρώνει μειωμένο εισιτήριο που αντιστοιχεί στο 50% του κανονικού εισιτηρίου. Το αντίτιμο του εισιτηρίου να εμφανίζεται στην οθόνη».
Δίνεται το πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.7]). Προσπαθήστε να εντοπίσετε τυχόν λογικά λάθη που οδηγούν σε λανθασμένα αποτελέσματα και να προτείνετε διορθώσεις.
Διασταυρώστε την απάντησή σας με αυτή που ακολουθεί.

Κώδικας σε ΓΛΩΣΣΑ [5.7]. Παράδειγμα προγράμματος με λογικό λάθος σε δομή επιλογής

```
1   ΠΡΟΓΡΑΜΜΑ Αντίτιμο_εισιτηρίου
2   ΣΤΑΘΕΡΕΣ
3     Κανονικό_εισιτήριο = 1
4   ΜΕΤΑΒΛΗΤΕΣ
5     ΑΚΕΡΑΙΕΣ: ηλικία
6     ΠΡΑΓΜΑΤΙΚΕΣ: Αντίτιμο
7   ΑΡΧΗ
8     ΓΡΑΨΕ 'Δώσε ηλικία'
9     ΔΙΑΒΑΣΕ ηλικία
10    ΑΝ ηλικία > 18 ΤΟΤΕ
11      Αντίτιμο <- Κανονικό_εισιτήριο
12    ΑΛΛΙΩΣ
13      Αντίτιμο <- 0.5*Κανονικό_εισιτήριο
14    ΤΕΛΟΣ_ΑΝ
15    ΓΡΑΨΕ 'Το αντίτιμο του εισιτηρίου είναι:', Αντίτιμο
16  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Απάντηση:

Τιμές εισόδου που ενδείκνυται να χρησιμοποιηθούν στον έλεγχο

Με βάση τη συνθήκη **ηλικία>18** ενδείκνυται να χρησιμοποιηθούν οι παρακάτω τιμές εισόδου στον έλεγχο ορθότητας:

1η περίπτωση: **ηλικία<18**. Π.χ. 17
2η περίπτωση: **ηλικία=18**
3η περίπτωση: **ηλικία>18.** Π.χ. 19

Τα αποτελέσματα της εκτέλεσης των σεναρίων ελέγχου αποτυπώνονται στον πίνακα 5.1:

Πίνακας [5. 1]. Αποτελέσματα εκτέλεσης σεναρίων ελέγχου

Α/Α	Τιμή εισόδου (*Ηλικία*)	Αναμενόμενο αποτέλεσμα με βάση την εκφώνηση (*Αντίτιμο*)	Αποτέλεσμα συνθήκης ηλικία > 18	Τιμή μεταβλητής *Αντίτιμο*	Ορθότητα αποτελέσματος προγράμματος
1	17	0,5	ψευδής	0,5	Σωστό
2	18	1	ψευδής	0,5	Λάθος
3	19	1	αληθής	1	Σωστό

Συμπεράσματα - προτάσεις διορθώσεων:

Όταν η «ηλικία» είναι 18, το αποτέλεσμα του προγράμματος είναι λανθασμένο. Η συνθήκη «***ηλικία>18***» πρέπει να γίνει «***ηλικία>=18***».

Παράδειγμα 5 – Παράδειγμα εκσφαλμάτωσης λογικών λαθών σε δομή επιλογής

Ακολουθεί η εκφώνηση για την ανάπτυξη ενός προγράμματος:

«Ένας οργανισμός ύδρευσης υπολογίζει την οφειλή ενός καταναλωτή για ένα τετράμηνο ως εξής: Χρεώνει το πάγιο ποσό 8€, τα πρώτα δέκα κυβικά μέτρα χρεώνονται με 0,4€/κυβικό. Τα δέκα επόμενα κυβικά επιπλέον των δέκα πρώτων χρεώνονται με 0,5€/κυβικό. Κάθε κυβικό επιπλέον των είκοσι πρώτων χρεώνεται με 0,6€/κυβικό. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει τα κυβικά που κατανάλωσε ένα νοικοκυριό σε ένα τετράμηνο και να υπολογίζει και να εμφανίζει το ποσό της οφειλής του νοικοκυριού».

Δίνεται το πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.8]). Προσπαθήστε να εντοπίσετε τυχόν λογικά λάθη που οδηγούν σε λανθασμένα αποτελέσματα και να προτείνετε διορθώσεις.

Προκειμένου να ελέγξετε την ορθότητα του προγράμματος πραγματοποιήστε δοκιμαστική εκτέλεση με τα παρακάτω δεδομένα.

1η περίπτωση: κυβικά<0. Π.χ. -1.
2η περίπτωση: κυβικά=0.
3η περίπτωση: 0<κυβικά<10. Π.χ. κυβικά=2.
4η περίπτωση: κυβικά=10.
5η περίπτωση: 10<κυβικά<20. Π.χ. κυβικά=11.
6η περίπτωση: κυβικά=20.
7η περίπτωση: κυβικά>20. Π.χ. κυβικά=21.

Διασταυρώστε την απάντησή σας με αυτή που ακολουθεί.

Κώδικας σε ΓΛΩΣΣΑ [5.8]. Παράδειγμα προγράμματος με λογικό λάθος σε δομή επιλογής

```
1   ΠΡΟΓΡΑΜΜΑ Χρέωση_ύδρευσης
2   ΣΤΑΘΕΡΕΣ
3     πάγιο = 8
4   ΜΕΤΑΒΛΗΤΕΣ
5     ΠΡΑΓΜΑΤΙΚΕΣ: κυβικά, οφειλή
6   ΑΡΧΗ
7   ! Ανάγνωση δεδομένων
8     ΓΡΑΨΕ 'Δώσε τα κυβικά που καταναλώθηκαν'
9     ΔΙΑΒΑΣΕ κυβικά
10  ! Υπολογισμός κλιμακωτής χρέωσης
11    ΑΝ κυβικά < 0 ΤΟΤΕ
12      ΓΡΑΨΕ 'Μη αποδεκτή τιμή.'
13    ΑΛΛΙΩΣ_ΑΝ κυβικά < 10 ΤΟΤΕ
14      οφειλή <- 8 + κυβικά*0.4
15    ΑΛΛΙΩΣ_ΑΝ κυβικά <= 20 ΤΟΤΕ
16      οφειλή <- 8 + 10*0.4 + (κυβικά - 10)*0.5
17    ΑΛΛΙΩΣ
18      οφειλή <- 10*0.4 + 10*0.5 + (κυβικά - 20)*0.6
19    ΤΕΛΟΣ_ΑΝ
20  ! Εμφάνιση αποτελέσματος
21    ΓΡΑΨΕ 'Οφειλή=', οφειλή
22  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Απάντηση:
Τα αποτελέσματα της εκτέλεσης των σεναρίων ελέγχου αποτυπώνονται στον πίνακα 5.2:

Πίνακας [5. 2]. Αποτελέσματα εκτέλεσης σεναρίων ελέγχου

Α/Α	Τιμή εισόδου (κυβικά)	Αληθής συνθήκη (1η στη σειρά)	Αναμενόμενο αποτέλεσμα με βάση την εκφώνηση (Οφειλή)	Έξοδος προγράμματος	Ορθότητα αποτελέσματος προγράμματος
1	-1	ΑΝ κυβικά < 0	-	Μη αποδεκτή τιμή	Σωστό
2	0	ΑΛΛΙΩΣ_ΑΝ κυβικά < 10	8	8	Σωστό
3	2	ΑΛΛΙΩΣ_ΑΝ κυβικά < 10	8,8	8,8	Σωστό
4	10	ΑΛΛΙΩΣ_ΑΝ κυβικά <= 20	12	12	Σωστό
5	11	ΑΛΛΙΩΣ_ΑΝ κυβικά <= 20	12,5	12,5	Σωστό
6	20	ΑΛΛΙΩΣ_ΑΝ κυβικά <= 20	17	17	Σωστό
7	21	ΑΛΛΙΩΣ	17,6	9,6	Λάθος

Συμπεράσματα - προτάσεις διορθώσεων:
Όταν τα «κυβικά» είναι 21, το αποτέλεσμα του προγράμματος είναι λανθασμένο.
Η εντολή της 18ης γραμμής χρειάζεται να διορθωθεί (βλ. Κώδικας σε ΓΛΩΣΣΑ 5.9).

Κώδικας σε ΓΛΩΣΣΑ [5.9]. Προτεινόμενη διόρθωση εντολής λόγω λογικού λάθους

```
18      οφειλή <- 8 + 10*0.4 + 10*0.5 + (κυβικά - 20)*0.6
```

Παρατήρηση: Όταν η μεταβλητή «κυβικά» έχει την τιμή 10, η συνθήκη «**κυβικά<10**» και η συνθήκη **«κυβικά<=10»** έχουν το ίδιο αποτέλεσμα.

Στην ανίχνευση ενός λογικού λάθους στις δομές επιλογής δεν αρκεί η μεμονωμένη μελέτη των συνθηκών και των ομάδων εντολών που εκτελούνται όταν μια συνθήκη είναι αληθής ή ψευδής, αλλά χρειάζεται να μελετηθεί το αποτέλεσμα που παράγει ο συνδυασμός των συνθηκών και των ομάδων εντολών.

Δραστηριότητα 2 – Εκσφαλμάτωση λογικών λαθών σε δομή επιλογής

Ακολουθεί η εκφώνηση για την ανάπτυξη ενός προγράμματος:
«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει το πλήθος των εισιτηρίων που θέλουμε να εκδώσουμε και την κατηγορία τους. Στη συνέχεια να υπολογίζει και να εμφανίζει το κόστος των εισιτηρίων. Υπάρχουν οι εξής κατηγορίες εισιτηρίων:

1. Κανονικό εισιτήριο (1€)
2. Μειωμένο εισιτήριο (0,5€)

Σε μία εκτέλεση του προγράμματος μπορούν να ληφθούν υπόψη μόνο εισιτήρια μίας κατηγορίας. Θεωρείστε ότι ο χρήστης δίνει μη αρνητικές τιμές για το πλήθος των εισιτηρίων».
Δίνεται το πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.10]). Προσπαθήστε να εντοπίσετε τυχόν λογικά λάθη που οδηγούν σε λανθασμένα αποτελέσματα και να προτείνετε διορθώσεις.

Κώδικας σε ΓΛΩΣΣΑ [5.10]. Παράδειγμα προγράμματος με λογικό λάθος σε δομή επιλογής

```
1   ΠΡΟΓΡΑΜΜΑ Υπολογισμός_αντιτίμου_εισιτηρίου
2   ΣΤΑΘΕΡΕΣ
3     Κανονικό_εισιτήριο = 1
4     Μειωμένο_εισιτήριο = 0.5
5   ΜΕΤΑΒΛΗΤΕΣ
6     ΑΚΕΡΑΙΕΣ: κατηγορία, πλήθος
7     ΠΡΑΓΜΑΤΙΚΕΣ: Αντίτιμο
8   ΑΡΧΗ
```

```
 9      ΓΡΑΨΕ '1. Κανονικό εισιτήριο (1 ευρώ)'
10      ΓΡΑΨΕ '2. Μειωμένο εισιτήριο (0.5 ευρώ)'
11      ΓΡΑΨΕ 'Δώσε τον αριθμό της κατηγορίας του εισιτηρίου:'
12      ΔΙΑΒΑΣΕ κατηγορία
13      ΓΡΑΨΕ 'Δώσε το πλήθος των εισιτηρίων'
14      ΔΙΑΒΑΣΕ πλήθος
15      ΑΝ κατηγορία = 1 ΤΟΤΕ
16        Αντίτιμο <- πλήθος*Κανονικό_εισιτήριο
17        ΓΡΑΨΕ 'Αντίτιμο=', Αντίτιμο
18      ΑΛΛΙΩΣ_ΑΝ κατηγορία = 2 ΤΟΤΕ
19        Αντίτιμο <- Μειωμένο_εισιτήριο
20        ΓΡΑΨΕ 'Αντίτιμο=', Αντίτιμο
21      ΑΛΛΙΩΣ
22        ΓΡΑΨΕ 'Μη αποδεκτή κατηγορία εισιτηρίου'
23      ΤΕΛΟΣ_ΑΝ
24    ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

5.2.2 Εκσφαλμάτωση λογικών λαθών στις δομές επανάληψης

Στην ενότητα αυτή θα ασχοληθούμε με την εκσφαλμάτωση κάποιων συνηθισμένων λαθών στις δομές επανάληψης. Σε μια δομή επανάληψης μπορεί να εμφανιστούν λογικά λάθη που σχετίζονται με:

- τη συνθήκη επανάληψης ή τερματισμού,
- την αρχικοποίηση της συνθήκης,
- την ενημέρωση της συνθήκης εντός του βρόχου επανάληψης,
- τις εντολές που περιλαμβάνονται εντός του βρόχου.

Πολλές φορές τα πιθανά σενάρια ελέγχου ενός προγράμματος είναι πάρα πολλά. Για τον λόγο αυτό, στις δραστηριότητες που ακολουθούν δε θα ασχοληθούμε με την εξαντλητική μελέτη όλων των πιθανών σεναρίων ελέγχου. Θα μελετήσουμε κάποια συνηθισμένα λάθη και θα ασχοληθούμε με την ανάδειξη καλών πρακτικών για την εκσφαλμάτωση των προγραμμάτων.

Παράδειγμα 6 – Παράδειγμα εκσφαλμάτωσης λογικού λάθους σε δομή επανάληψης

Ακολουθεί η εκφώνηση για την ανάπτυξη ενός προγράμματος:
«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει έναν βαθμό τετραμήνου στην εικοσάβαθμη κλίκαμα. Να εκτελείται έλεγχος αποδεκτής τιμής.»

Δίνεται το πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.11]). Αν δοθεί η τιμή -9 από το πληκτρολόγιο τι θα εμφανιστεί στην οθόνη; Προσπαθήστε να εντοπίσετε τυχόν λογικά λάθη που οδηγούν σε λανθασμένα αποτελέσματα και να προτείνετε διορθώσεις.
Διασταυρώστε την απάντησή σας με αυτή που ακολουθεί.

Κώδικας σε ΓΛΩΣΣΑ [5.11]. Παράδειγμα προγράμματος με λογικό λάθος σε δομή επανάληψης

```
1   ΠΡΟΓΡΑΜΜΑ Ανάγνωση_βαθμολογίας
2   ΜΕΤΑΒΛΗΤΕΣ
3     ΑΚΕΡΑΙΕΣ: Βαθμός
4   ΑΡΧΗ
5     ΓΡΑΨΕ "Δώσε βαθμό στην εικοσάβαθμη κλίμακα"
6     ΔΙΑΒΑΣΕ Βαθμός
7     ΟΣΟ Βαθμός < 0 ΚΑΙ Βαθμός > 20 ΕΠΑΝΑΛΑΒΕ
8       ΓΡΑΨΕ "Μη αποδεκτή τιμή"
9       ΓΡΑΨΕ "Δώσε βαθμό στην εικοσάβαθμη κλίμακα"
10      ΔΙΑΒΑΣΕ Βαθμός
11    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
12    ΓΡΑΨΕ "Καταχωρήθηκε ο βαθμός της εικοσάβαθμης κλίμακας ", Βαθμός
13  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Απάντηση:

Δοκιμαστική εκτέλεση

Αν δοθεί η τιμή -9 από το πληκτρολόγιο, θα εμφανιστεί το μήνυμα «Καταχωρήθηκε ο βαθμός της εικοσάβαθμης κλίμακας -9» και θα τερματιστεί η εκτέλεση του προγράμματος.

Έλεγχος ορθότητας αποτελέσματος

Με βάση την εκφώνηση η τιμή -9 δεν είναι αποδεκτή και το μήνυμα που εμφανίστηκε είναι λανθασμένο.
Τα λάθος εντοπίζεται στη συνθήκη.
Με τη συνθήκη επανάληψης «*Βαθμός<0 ΚΑΙ Βαθμός>20*»
ο βρόχος επανάληψης δε θα εκτελεστεί για καμία τιμή της μεταβλητής «Βαθμός», επειδή πολύ απλά κανείς αριθμός μικρότερος του μηδενός δεν είναι μεγαλύτερος του είκοσι.

Προτεινόμενη διόρθωση

Η συνθήκη «*Βαθμός<0 ΚΑΙ Βαθμός>20*» πρέπει να αντικατασταθεί με τη συνθήκη «*Βαθμός<0 Η Βαθμός>20*».

Παράδειγμα 7 – Παράδειγμα εκσφαλμάτωσης λογικού λάθους σε δομή επανάληψης

Ακολουθεί η εκφώνηση για την ανάπτυξη ενός προγράμματος:
«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει από το πληκτρολόγιο αριθμούς διάφορους του μηδενός, να υπολογίζει το γινόμενό τους και στο τέλος να το εμφανίζει. Αν δε δοθούν αριθμοί, να εμφανίζει τον αριθμό 1».
Δίνεται το πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.12]). Αν δοθούν από το πληκτρολόγιο οι τιμές 4, 3 και 0 ποια θα είναι η τιμή της μεταβλητής «Γινόμενο» που θα εμφανιστεί στην οθόνη; Προσπαθήστε να εντοπίσετε τυχόν λογικά λάθη που οδηγούν σε λανθασμένα αποτελέσματα και να προτείνετε διορθώσεις.
Διασταυρώστε την απάντησή σας με αυτή που ακολουθεί.

Κώδικας σε ΓΛΩΣΣΑ [5.12]. Παράδειγμα προγράμματος με λογικό λάθος σε δομή επανάληψης

```
1   ΠΡΟΓΡΑΜΜΑ Γινόμενο_μη_μηδενικών_αριθμών
2   ΜΕΤΑΒΛΗΤΕΣ
3     ΠΡΑΓΜΑΤΙΚΕΣ: Χ, Γινόμενο
4   ΑΡΧΗ
5     Γινόμενο <- 1
6     ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
7       ΓΡΑΨΕ "Δώσε μη μηδενική τιμή"
8       ΔΙΑΒΑΣΕ Χ
9       Γινόμενο <- Χ*Γινόμενο
10    ΜΕΧΡΙΣ_ΟΤΟΥ Χ = 0
11    ΓΡΑΨΕ "Γινόμενο=", Γινόμενο
12  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Απάντηση:

Δοκιμαστική εκτέλεση

Αν δοθούν από το πληκτρολόγιο οι τιμές 4, 3 και 0, θα εμφανιστεί η τιμή μηδέν.

Έλεγχος ορθότητας αποτελέσματος

Με βάση την εκφώνηση θα έπρεπε να εμφανιστεί η τιμή 12 για το γινόμενο των αριθμών. Όμως, στον υπολογισμό του γινομένου συμπεριλήφθηκε και το 0 και ως εκ τούτου η τιμή της μεταβλητής «Γινόμενο» μηδενίστηκε.

Μελέτη πιθανών διορθώσεων

Το λογικό λάθος θα μπορούσε να διορθωθεί εύκολα με τη σωστή χρήση της δομής επανάληψης ΟΣΟ...ΕΠΑΝΑΛΑΒΕ (βλ. Κώδικας σε ΓΛΩΣΣΑ [5.13]).

Κώδικας σε ΓΛΩΣΣΑ [5.13]. Χρήση δομής επανάληψης ΟΣΟ...ΕΠΑΝΑΛΑΒΕ που διορθώνει το λογικό λάθος

```
6       ΓΡΑΨΕ "Δώσε μη μηδενική τιμή"
7       ΔΙΑΒΑΣΕ Χ
8       ΟΣΟ Χ <> 0 ΕΠΑΝΑΛΑΒΕ
9         Γινόμενο <- Χ*Γινόμενο
10        ΓΡΑΨΕ "Δώσε μη μηδενική τιμή"
11        ΔΙΑΒΑΣΕ Χ
12      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Προσοχή: Η επιλογή της δομής επανάληψης ΟΣΟ...ΕΠΑΝΑΛΑΒΕ δε λύνει από μόνη της το πρόβλημα (π.χ. κώδικας σε ΓΛΩΣΣΑ [5.14]), αλλά χρειάζεται να δοθεί προσοχή στη σωστή αξιοποίησή της, ώστε στην τελευταία επανάληψη ο αριθμός μηδέν να μη ληφθεί υπόψη στον υπολογισμό του γινομένου.

Κώδικας σε ΓΛΩΣΣΑ [5.14]. Χρήση δομής επανάληψης ΟΣΟ που δεν διορθώνει το λογικό λάθος

```
6       Χ <- 1
7       ΟΣΟ Χ <> 0 ΕΠΑΝΑΛΑΒΕ
8         ΓΡΑΨΕ "Δώσε μη μηδενική τιμή"
9         ΔΙΑΒΑΣΕ Χ
10        Γινόμενο <- Χ*Γινόμενο
11      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Το λογικό λάθος θα μπορούσε να διορθωθεί και με τη χρήση της δομής επανάληψης ΜΕΧΡΙΣ_ΟΤΟΥ αν χρησιμοποιούνταν και μια εμφωλευμένη δομή επιλογής (π.χ. κώδικας σε ΓΛΩΣΣΑ [5.15]).

Κώδικας σε ΓΛΩΣΣΑ [5.15]. Χρήση δομής επανάληψης ΜΕΧΡΙΣ_ΟΤΟΥ που διορθώνει το λογικό λάθος

```
6       ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
7         ΓΡΑΨΕ "Δώσε μη μηδενική τιμή"
8         ΔΙΑΒΑΣΕ Χ
9         ΑΝ Χ <> 0 ΤΟΤΕ
10          Γινόμενο <- Χ*Γινόμενο
11        ΤΕΛΟΣ_ΑΝ
12      ΜΕΧΡΙΣ_ΟΤΟΥ Χ = 0
```

Στους υπολογισμούς που γίνονται εντός των δομών επανάληψης χρειάζεται να δοθεί ιδιαίτερη προσοχή στο αν θα συμπεριλάβουμε στον υπολογισμό την τιμή που λαμβάνει κάποια μεταβλητή στην τελευταία επανάληψη.

Δραστηριότητα 3 – Εκσφαλμάτωση λογικού λάθους σε δομή επανάληψης

Ακολουθεί η εκφώνηση για την ανάπτυξη ενός προγράμματος:
«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει βαθμούς μαθητών, να υπολογίζει τον μέσο όρο τους και στο τέλος να τον εμφανίζει. Το πρόγραμμα να αποδέχεται μόνο τιμές μεγαλύτερες ή ίσες του μηδενός για τους βαθμούς. Μόλις διαβάσει κάποιον αριθμό μικρότερο του μηδενός, να σταματήσει την ανάγνωση των βαθμών. Θεωρούμε ότι από το πληκτρολόγιο δε δίνονται τιμές μεγαλύτερες από τον μέγιστο επιτρεπτό βαθμό».
Δίνεται το πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.16]).
Α) Εκτελέστε το πρόγραμμα για τις τιμές εισόδου 15, 16, 17 και -1 και καταγράψτε τις τιμές των μεταβλητών στον παρακάτω πίνακα τιμών. Στον πίνακα χρησιμοποιήστε όσες γραμμές χρειάζεστε.

Πίνακας [5. 3]. Πίνακας τιμών προς συμπλήρωση

Επανάληψη	Άθροισμα	Πλήθος	Βαθμός	ΜΟ	Οθόνη

Β) Ποια λάθη εντοπίσατε κατά την εκτέλεση του προγράμματος; Προτείνετε διορθώσεις.

Κώδικας σε ΓΛΩΣΣΑ [5.16]. Πρόγραμμα με λογικό λάθος σε δομή επανάληψης

```
1   ΠΡΟΓΡΑΜΜΑ ΜΟ_Βαθμολογίας
2   ΜΕΤΑΒΛΗΤΕΣ
3     ΑΚΕΡΑΙΕΣ: Βαθμός, Πλήθος, Άθροισμα
4     ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ
5   ΑΡΧΗ
6     Άθροισμα <- 0
7     Πλήθος <- 0
8     ΓΡΑΨΕ "Βαθμός:"
9     ΔΙΑΒΑΣΕ Βαθμός
10    ΟΣΟ Βαθμός >= 0 ΕΠΑΝΑΛΑΒΕ
11      ΓΡΑΨΕ "Βαθμός:"
12      ΔΙΑΒΑΣΕ Βαθμός
13      Άθροισμα <- Άθροισμα + Βαθμός
14    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
15    ΜΟ <- Άθροισμα/Πλήθος
16    ΓΡΑΨΕ "Μέσος όρος βαθμών=", ΜΟ
17  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Συμβουλή: Κατά την εκσφαλμάτωση των δομών επανάληψης χρειάζεται να δίνετε προσοχή στα εξής:

- στους συγκριτικούς και τους λογικούς τελεστές των συνθηκών επανάληψης ή τερματισμού
- στην αρχικοποίηση της συνθήκης
- στην ενημέρωση της συνθήκης εντός του βρόχου
- στην αλληλουχία των εντολών του βρόχου και στη σειρά εκτέλεσής τους
- στο κριτήριο της περατότητας
- στην πρώτη επανάληψη και στην περίπτωση που ο βρόχος επανάληψης δεν πρέπει να εκτελεστεί ούτε μία φορά
- στην τελευταία επανάληψη

5.2.3 Εκσφαλμάτωση λογικών λαθών σε πίνακες

Στην ενότητα αυτή θα ασχοληθούμε με την εκσφαλμάτωση κάποιων συνηθισμένων λαθών στα προγράμματα που χρησιμοποιούν πίνακες ως στατικές δομές δεδομένων.

Παράδειγμα 8 – Παράδειγμα εκσφαλμάτωσης λογικού λάθους σε πρόγραμμα που χρησιμοποιεί πίνακα

Ακολουθεί η εκφώνηση για την ανάπτυξη ενός προγράμματος:
«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει από το πληκτρολόγιο τα μηνιαία έσοδα ενός καταστήματος για το πρώτο εξάμηνο και να τα καταχωρεί σε πίνακα. Στη συνέχεια για τους μήνες Φεβρουάριο, Μάρτιο, Απρίλιο, Μάιο και Ιούνιο να ελέγχει αν είχαν περισσότερα έσοδα από τον ακριβώς προηγούμενο μήνα και να εμφανίζει κατάλληλο μήνυμα που θα δηλώνει την ύπαρξη αύξησης».
Α) Δίνεται το πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.17]). Να εκτελέσετε το πρόγραμμα για τις τιμές εισόδου 2000, 1800, 2100, 2100, 2000 και 2000 και καταγράψτε την εκτέλεση της επανάληψης ελέγχου αυξητικής τάσης στον παρακάτω πίνακα τιμών. Στον πίνακα χρησιμοποιήστε όσες γραμμές χρειάζεστε.
Β) Ποια λάθη εντοπίσατε κατά την εκτέλεση του προγράμματος; Προτείνετε διορθώσεις.
Διασταυρώστε την απάντησή σας με αυτή που ακολουθεί.

Πίνακας [5. 4]. Πίνακας τιμών και ελέγχου ορθότητας προς συμπλήρωση

Ι	ΕΣΟΔΑ[Ι]	ΕΣΟΔΑ[Ι+1]	ΕΣΟΔΑ[Ι]<= ΕΣΟΔΑ[Ι+1]	Έξοδος προγράμματος	Αναμενόμενο αποτέλεσμα	Ορθότητα εξόδου

Κώδικας σε ΓΛΩΣΣΑ [5.17]. Παράδειγμα προγράμματος που χρησιμοποιεί πίνακα και έχει λογικό λάθος

```
1   ΠΡΟΓΡΑΜΜΑ ΕΣΟΔΑ_ΚΑΤΑΣΤΗΜΑΤΟΣ
2   ΜΕΤΑΒΛΗΤΕΣ
3     ΠΡΑΓΜΑΤΙΚΕΣ: ΕΣΟΔΑ[6]
4     ΑΚΕΡΑΙΕΣ: Ι
5   ΑΡΧΗ
6   ! Επανάληψη ανάγνωσης εσόδων
7     ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 6
8       ΓΡΑΨΕ "Δώσε τα έσοδα του ", Ι, "ου μήνα"
9       ΔΙΑΒΑΣΕ ΕΣΟΔΑ[Ι]
10    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
11  ! Επανάληψη ελέγχου αυξητικής τάσης
12    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 6
13      ΑΝ ΕΣΟΔΑ[Ι] <= ΕΣΟΔΑ[Ι + 1] ΤΟΤΕ
14        ΓΡΑΨΕ "Ο ", Ι, "ος μήνας ΑΥΞΗΣΗ"
15      ΤΕΛΟΣ_ΑΝ
16    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
17  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Απάντηση:

Δοκιμαστική εκτέλεση

Πίνακας [5. 5]. Συμπληρωμένος πίνακας τιμών

Ι	ΕΣΟΔΑ[Ι]	ΕΣΟΔΑ[Ι+1]	ΕΣΟΔΑ[Ι]<= ΕΣΟΔΑ[Ι+1]	Έξοδος προγράμματος	Αναμενόμενο αποτέλεσμα	Ορθότητα εξόδου
1	2000	1800	Ψευδής			Σωστό
2	1800	2100	Αληθής	Ο 2ος μήνας ΑΥΞΗΣΗ	Ο 3ος μήνας ΑΥΞΗΣΗ	Λάθος
3	2100	2100	Αληθής	Ο 3ος μήνας ΑΥΞΗΣΗ		Λάθος
4	2100	2000	Ψευδής			Λάθος
5	2000	2000	Αληθής	Ο 5ος μήνας ΑΥΞΗΣΗ		Λάθος
6	2000	?				Αντικανονικός τερματισμός

Προτεινόμενες διορθώσεις

- Στη δεύτερη επανάληψη το μήνυμα που εμφανίζεται αναφέρεται σε λανθασμένο μήνα. Για να διορθωθεί αυτό πρέπει η εντολή «**ΓΡΑΨΕ** "Ο ", Ι, "ος μήνας ΑΥΞΗΣΗ"» να γίνει «**ΓΡΑΨΕ** "Ο ", Ι+1, "ος μήνας ΑΥΞΗΣΗ"».
- Στην τρίτη και την πέμπτη επανάληψη εμφανίζεται η ύπαρξη αύξησης, ενώ τα έσοδα των δύο μηνών είναι ίσα. Για να διορθωθεί αυτό πρέπει η συνθήκη «ΕΣΟΔΑ[Ι] **<=** ΕΣΟΔΑ[Ι **+** 1]» να γίνει «ΕΣΟΔΑ[Ι] **<** ΕΣΟΔΑ[Ι **+** 1]».
- Στην έκτη επανάληψη το πρόγραμμα προσπαθεί να προσπελάσει την τιμή του στοιχείου ΕΣΟΔΑ[7], το οποίο είναι εκτός των ορίων του πίνακα. Για να διορθωθεί αυτό πρέπει η εντολή «**ΓΙΑ** Ι **ΑΠΟ** 1 **ΜΕΧΡΙ** 6» στην επανάληψη ελέγχου να γίνει «**ΓΙΑ** Ι **ΑΠΟ** 1 **ΜΕΧΡΙ** 5».

Συμβουλή: Κατά την εκσφαλμάτωση προγραμμάτων που χρησιμοποιούν πίνακες χρειάζεται να δίνετε ιδιαίτερη προσοχή:

- στο μέγεθος των πινάκων κατά τη δήλωσή τους,
- στους δείκτες των πινάκων κατά την προσπέλασή τους,
- στη μη υπέρβαση των ορίων του πίνακα.

Δραστηριότητα 4 – Εκσφαλμάτωση λογικών λαθών σε πρόγραμμα που χρησιμοποιεί πίνακα

Ακολουθεί η εκφώνηση για την ανάπτυξη ενός προγράμματος:
«Μια επιχείρηση έχει τρία υποκαταστήματα. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει από το πληκτρολόγιο τα έσοδα κάθε υποκαταστήματος ανά τρίμηνο ενός έτους και να τα καταχωρεί σε πίνακα. Για κάθε υποκατάστημα να υπολογίζει και να εμφανίζει τα ετήσια έσοδα».
i) Δίνεται το πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.18]). Εκτελέστε το πρόγραμμα για τις τιμές εισόδου του πίνακα 5.6. και συμπληρώστε τον πίνακα τιμών 5.7. Στον πίνακα χρησιμοποιήστε όσες γραμμές χρειάζεστε.
ii) Ποια λάθη εντοπίσατε κατά την εκτέλεση του προγράμματος; Προτείνετε διορθώσεις.

Πίνακας [5.6]. Έσοδα τριμήνων

	1ο τρίμηνο	2ο τρίμηνο	3ο τρίμηνο	4ο τρίμηνο
1ο υποκατάστημα	6000	7000	7500	6500
2ο υποκατάστημα	5000	4000	5000	6000
3ο υποκατάστημα	5000	6000	4000	5000

Πίνακας [5.7]. Πίνακας τιμών προς συμπλήρωση

Ι	Κ	Άθροισμα

Κώδικας σε ΓΛΩΣΣΑ [5.18]. Πρόγραμμα που χρησιμοποιεί πίνακα και έχει λογικό λάθος

```
1   ΠΡΟΓΡΑΜΜΑ Έσοδα_υποκαταστημάτων
2   ΜΕΤΑΒΛΗΤΕΣ
3     ΠΡΑΓΜΑΤΙΚΕΣ: ΕΣΟΔΑ[3, 4], Άθροισμα
4     ΑΚΕΡΑΙΕΣ: Ι, Κ
5   ΑΡΧΗ
6     Άθροισμα <- 0
7     ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 3
8       ΓΡΑΨΕ 'Δώσε τα έσοδα των τριμήνων του ', Ι, 'ου υποκαταστήματος:'
9       ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 4
10        ΔΙΑΒΑΣΕ ΕΣΟΔΑ[Ι, Κ]
11        Άθροισμα <- Άθροισμα + ΕΣΟΔΑ[Ι, Κ]
12      ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
13      ΓΡΑΨΕ 'Ετήσια έσοδα:', Άθροισμα
14    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
15  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

5.2.4 Εκσφαλμάτωση λογικών λαθών στα υποπρογράμματα

Στην ενότητα αυτή θα ασχοληθούμε με την εκσφαλμάτωση λογικών λαθών στα υποπρογράμματα.

Παράδειγμα 9 – Παράδειγμα εκσφαλμάτωσης λογικών λαθών σε πρόγραμμα που χρησιμοποιεί υποπρόγραμμα

Ακολουθεί η εκφώνηση για την ανάπτυξη ενός προγράμματος:
«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει τα μηνιαία έσοδα και τα μηνιαία έξοδα μιας επιχείρησης για το πρώτο εξάμηνο του έτους και να τα καταχωρεί σε πίνακες. Μέσω συνάρτησης να υπολογίζει το πλήθος των μηνών που είχαν ζημία, δηλαδή τα έσοδα ήταν λιγότερα από τα έξοδα. Τέλος, να εμφανίζει το πλήθος των μηνών που είχαν ζημία».
i) Δίνεται το πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.19]). Εκτελέστε το πρόγραμμα για τις τιμές εισόδου των πινάκων 5.8 και 5.9 και συμπληρώστε τον πίνακα τιμών 5.10. Στον πίνακα χρησιμοποιήστε όσες γραμμές χρειάζεστε. Ποια τιμή θα πάρει η μεταβλητή «*Μήνες_με_ζημία*» μετά την ολοκλήρωση της εκτέλεσης της συνάρτησης; Με βάση την εκφώνηση πόσοι μήνες είχαν ζημία;
ii) Ποια λάθη εντοπίσατε κατά την εκτέλεση του προγράμματος; Προτείνετε διορθώσεις.
iii) Διασταυρώστε την απάντησή σας με αυτή που ακολουθεί.

Πίνακας [5. 8]. Μηνιαία έσοδα

1ος μήνας	2ος μήνας	3ος μήνας	4ος μήνας	5ος μήνας	6ος μήνας
2000	2000	2300	2500	1800	2100

Πίνακας [5. 9]. Μηνιαία έξοδα

1ος μήνας	2ος μήνας	3ος μήνας	4ος μήνας	5ος μήνας	6ος μήνας
2000	2200	2300	2100	1500	2000

Πίνακας [5. 10]. Πίνακας τιμών προς συμπλήρωση

I	Πίνακας1[I]	Πίνακας2[I]	Πλήθος

Κώδικας σε ΓΛΩΣΣΑ [5.19]. Παράδειγμα προγράμματος που χρησιμοποιεί υποπρόγραμμα και έχει λογικό λάθος

```
ΠΡΟΓΡΑΜΜΑ ΕΛΕΓΧΟΣ_ΖΗΜΙΑΣ
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: Ι, Πλήθος, Μήνες_με_ζημία
  ΠΡΑΓΜΑΤΙΚΕΣ: ΕΣΟΔΑ[6], ΕΞΟΔΑ[6]
ΑΡΧΗ
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 6
    ΓΡΑΨΕ 'Δώσε τα έσοδα του ', Ι, 'ου μήνα:'
    ΔΙΑΒΑΣΕ ΕΣΟΔΑ[Ι]
    ΓΡΑΨΕ 'Δώσε τα έξοδα του ', Ι, 'ου μήνα:'
    ΔΙΑΒΑΣΕ ΕΞΟΔΑ[Ι]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  Μήνες_με_ζημία <- Υπολογισμός(ΕΣΟΔΑ, ΕΞΟΔΑ)
  ΓΡΑΨΕ 'Μήνες που είχαν ζημία:', Μήνες_με_ζημία
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΝΑΡΤΗΣΗ Υπολογισμός(Πίνακας1, Πίνακας2): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ
  ΠΡΑΓΜΑΤΙΚΕΣ: Πίνακας1[6], Πίνακας2[6]
  ΑΚΕΡΑΙΕΣ: Ι, Πλήθος
ΑΡΧΗ
  Πλήθος <- 0
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 6
    ΑΝ Πίνακας2[Ι] < Πίνακας1[Ι] ΤΟΤΕ
      Πλήθος <- Πλήθος + 1
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  Υπολογισμός <- Πλήθος
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

Απάντηση:

i) Δοκιμαστική εκτέλεση

Πίνακας [5. 11]. Συμπληρωμένος πίνακας τιμών

I	Πίνακας1[I]	Πίνακας2[I]	Πλήθος
-			0
1	2000	2000	0
2	2000	2200	0
3	2300	2300	0
4	2500	2100	1
5	1800	1500	2
6	2100	2000	3

Η μεταβλητή «*Μήνες_με_ζημία*» μετά την ολοκλήρωση της εκτέλεσης της συνάρτησης θα λάβει την τιμή 3. Με βάση την εκφώνηση, μόνο ένας μήνας είχε ζημία.
ii) Υπάρχει λογικό λάθος κατά την κλήση της συνάρτησης. Η συνάρτηση υπολογίζει πόσα στοιχεία του Πίνακα2 είναι μικρότερα από την τιμή του αντίστοιχου στοιχείου του Πίνακα1. Εμείς θέλουμε να βρούμε πόσα στοιχεία του πίνακα «ΕΣΟΔΑ» είναι μικρότερα από την τιμή του αντίστοιχου στοιχείου του πίνακα «ΕΞΟΔΑ». Άρα, ο Πίνακας «ΕΣΟΔΑ» πρέπει να αντιστοιχιστεί στον Πίνακα2 και ο πίνακας «ΕΞΟΔΑ» πρέπει να αντιστοιχιστεί στον Πίνακα1.

Συμπεραίνουμε λοιπόν πως η εντολή της γραμμής 12 πρέπει να αντικατασταθεί με την εντολή:

```
«Μήνες_με_ζημία <- Υπολογισμός(ΕΞΟΔΑ, ΕΣΟΔΑ)»
```

Εναλλακτικά, η δήλωση της συνάρτησης στην γραμμή 16 θα μπορούσε να γίνει:

```
«ΣΥΝΑΡΤΗΣΗ Υπολογισμός(Πίνακας2, Πίνακας1): ΑΚΕΡΑΙΑ»
```

Μια άλλη λύση θα ήταν η εντολή της γραμμής 23 να γίνει:

```
«ΑΝ Πίνακας1[I] < Πίνακας2[I] ΤΟΤΕ»
```

Συμβουλή: Κατά την εκσφαλμάτωση προγραμμάτων που χρησιμοποιούν υποπρογράμματα χρειάζεται να δίνεται προσοχή στον εντοπισμό λογικών λαθών που σχετίζονται με:

- την κλήση του υποπρογράμματος και το πέρασμα των παραμέτρων
- τα λοιπά λογικά λάθη που εμφανίζονται και στα προγράμματα.

Δραστηριότητα 5 – Εκσφαλμάτωση λογικού λάθους σε πρόγραμμα που χρησιμοποιεί υποπρόγραμμα

Ακολουθεί η εκφώνηση για την ανάπτυξη ενός προγράμματος:
«Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει έναν ακέραιο αριθμό ν, να υπολογίζει μέσω συνάρτησης το παραγοντικό του και να το εμφανίζει. Το παραγοντικό του αριθμού ν συμβολίζεται με ν! και υπολογίζεται ως εξής:

ν!=1x2x . . . x ν.

Π.χ. 4!=1x2x3x4=24».

Δίνεται το πρόγραμμα (κώδικας σε ΓΛΩΣΣΑ [5.20]).

i) Να εκτελέσετε το πρόγραμμα για την τιμή εισόδου 4 και να καταγράψετε τις τιμές των μεταβλητών κατά την εκτέλεση της συνάρτησης στον πίνακα τιμών 5.12. Στον πίνακα μπορείτε να χρησιμοποιήσετε όσες γραμμές χρειάζεστε.

Πίνακας [5.12]. Πίνακας τιμών προς συμπλήρωση

Π	Ι

ii) Ποια λάθη εντοπίσατε κατά την εκτέλεση του προγράμματος; Προτείνετε διορθώσεις.

Κώδικας σε ΓΛΩΣΣΑ [5.20]. Πρόγραμμα που χρησιμοποιεί υποπρόγραμμα και έχει λογικό λάθος

```
1  ΠΡΟΓΡΑΜΜΑ Παραγοντικό
2  ΜΕΤΑΒΛΗΤΕΣ
3    ΑΚΕΡΑΙΕΣ: Ν,Παρ
4  ΑΡΧΗ
5    ΓΡΑΨΕ "Δώσε έναν ακέραιο θετικό αριθμό"
6    ΔΙΑΒΑΣΕ Ν
7    Παρ<-Υπολογισμός(Ν)
8    ΓΡΑΨΕ "Παραγοντικό=", Παρ
9  ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
10
11 ΣΥΝΑΡΤΗΣΗ Υπολογισμός(Χ):ΑΚΕΡΑΙΑ
12 ΜΕΤΑΒΛΗΤΕΣ
13 ΑΚΕΡΑΙΕΣ: Ι,Π,Χ
14 ΑΡΧΗ
15   Π <- 0
16   ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Χ
17     Π <- Π*Ι
18   ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
19   Υπολογισμός<-Π
20 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

5.2.5 Μέθοδος ελέγχου «Μαύρο Κουτί»

Κάθε πρόγραμμα πρέπει να ελεγχθεί για τη σωστή λειτουργία του, δηλαδή να επιβεβαιωθεί ότι παράγει τα αποτελέσματα που πρέπει. Είναι συνηθισμένη πρακτική, κατά το στάδιο της υλοποίησης να πραγματοποιούνται ανασκοπήσεις κώδικα από ομάδες προγραμματιστών προκειμένου να εντοπιστούν συνηθισμένα λάθη (όπως αυτά που παρουσιάστηκαν στις προηγούμενες ενότητες). Αυτή η τεχνική δεν εξασφαλίζει τον εντοπισμό όλων των λαθών και γι' αυτό ακολουθείται πιο συστηματική προσέγγιση με τη χρήση σεναρίων ελέγχου.

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.

Είναι σημαντικό να δημιουργούνται διαστήματα και για τις μη έγκυρες τιμές εισόδου, καθώς δεν μπορούμε να είμαστε σίγουροι ότι ένα πρόγραμμα θα τροφοδοτείται μόνο με έγκυρες τιμές.

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 >=).

Παράδειγμα 10 – Βαθμολογία γραπτής εξέτασης

Η βαθμολογία στις γραπτές δοκιμασίες τετραμήνου στο Λύκειο δίνεται με ακέραιους αριθμούς στην κλίμακα από 0 έως και 20. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ που να διαβάζει τη βαθμολογία σε μια γραπτή δοκιμασία και στη συνέχεια να εμφανίζει μήνυμα «Επιτυχής εξέταση», αν η βαθμολογία είναι τουλάχιστον 10, και μήνυμα «Ανεπιτυχής εξέταση» αν η βαθμολογία είναι μικρότερη από 10. Σε περίπτωση που δοθεί τιμή εκτός του διαστήματος 0-20 να εμφανίζεται μήνυμα λάθους «Μη έγκυρη βαθμολογία».
Με βάση τις παραπάνω προδιαγραφές, προσπαθήστε να δημιουργήσετε κατάλληλα σενάρια για να πραγματοποιήσετε έλεγχο ακραίων τιμών.

Απάντηση:

Βήμα 1ο: Δημιουργία ισοδύναμων διαστημάτων
Από την εκφώνηση είναι προφανές ότι υπάρχουν δύο διαστήματα για την είσοδο:

- 0 <= βαθμός < 10
- 10 <= βαθμός <= 20

Επίσης υπάρχουν δύο διαστήματα μη έγκυρων τιμών εισόδου:

- βαθμός < 0
- βαθμός > 20

---------->	**0** <---------->	**10** <----------> **20**	<----------
Μη έγκυρη βαθμολογία	Ανεπιτυχής εξέταση	Επιτυχής εξέταση	Μη έγκυρη βαθμολογία

Διάγραμμα [5.1]. Διαστήματα δεδομένων εισόδου

Βήμα 2ο: Καθορισμός ακραίων τιμών διαστημάτων
Τα διαστήματα των δεδομένων εισόδου απεικονίζονται στο Διάγραμμα 5.1, όπου φαίνεται ότι λείπουν κάποια άκρα. Για να τα υπολογίσουμε αρκεί να προσθέσουμε ή να αφαιρέσουμε 1 από το άκρο του προηγούμενου ή επόμενου διαστήματος αντίστοιχα, αφού σύμφωνα με την εκφώνηση η είσοδος είναι ένας ακέραιος αριθμός. Με αυτό τον τρόπο καταλήγουμε στο Διάγραμμα 5.2.

----------> **-1**	**0** <----------> **9**	**10** <----------> **20**	**21** <----------
Μη έγκυρη βαθμολογία	Ανεπιτυχής εξέταση	Επιτυχής εξέταση	Μη έγκυρη βαθμολογία

Διάγραμμα [5.2]. Ακραίες τιμές διαστημάτων εισόδου

Βήμα 3ο: Δημιουργία σεναρίων ελέγχου
Το τελευταία βήμα είναι να δημιουργήσουμε ένα σενάριο ελέγχου για κάθε ακραία τιμή. Κάθε σενάριο πρέπει κατ' ελάχιστο να περιλαμβάνει την τιμή εισόδου, το αναμενόμενο αποτέλεσμα (σύμφωνα με την εκφώνηση του προβλήματος) και περιγραφή της περίπτωσης που ελέγχεται. Έτσι καταλήγουμε στα σενάρια ελέγχου του Πίνακα 5.13

Πίνακας [5.13]. Σενάρια ελέγχου

Α/Α	Είσοδος	Αναμενόμενο αποτέλεσμα	Περίπτωση που ελέγχεται
1	-1	Μη έγκυρη βαθμολογία	Άνω άκρο διαστήματος βαθμός < 0
2	0	Ανεπιτυχής εξέταση	Κάτω άκρο διαστήματος 0 <= βαθμός < 10
3	9	Ανεπιτυχής εξέταση	Άνω άκρο διαστήματος 0 <= βαθμός < 10
4	10	Επιτυχής εξέταση	Κάτω άκρο διαστήματος 10 <= βαθμός <= 20
5	20	Επιτυχής εξέταση	Άνω άκρο διαστήματος 10 <= βαθμός <= 20
6	21	Μη έγκυρη βαθμολογία	Κάτω άκρο διαστήματος βαθμός > 20

Βλέπουμε λοιπόν, ότι ακόμα και ένα πολύ απλό πρόγραμμα με μία μόνο είσοδο, η οποία δεν υφίσταται καμία επεξεργασία, απαιτεί έξι σενάρια ελέγχου. Είναι γεγονός ότι ο έλεγχος ενός προγράμματος είναι διαδικασία που απαιτεί χρόνο, τόσο για τον σχεδιασμό όσο και για την υλοποίηση και εκτέλεση των σεναρίων ελέγχου, ενώ αν προκύψουν λάθη θα απαιτηθεί επιπλέον χρόνος για τον εντοπισμό και τη διόρθωση τους. Γι' αυτό και η φάση του ελέγχου και της εκσφαλμάτωσης αποτελούν σημαντικό ποσοστό του συνολικού χρόνου ανάπτυξης ενός προγράμματος. Σε μεγάλες εταιρείες λογισμικού, μάλιστα, υπάρχουν εξειδικευμένες ομάδες προγραμματιστών που ασχολούνται αποκλειστικά με τον έλεγχο και την εκσφαλμάτωση των προγραμμάτων.

Δραστηριότητα 6 – Χώρος στάθμευσης οχημάτων

Ένα πάρκινγκ στο κέντρο της πόλης έχει την ακόλουθη τιμολογιακή πολιτική: για στάθμευση έως και 3 ώρες σταθερή χρέωση 6 ευρώ, κάθε επιπλέον ώρα χρεώνεται 1,5 € με μέγιστο συνολικό χρόνο παραμονής τις 8 ώρες. Η χρέωση γίνεται για ολόκληρες ώρες. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ, το οποίο να διαβάζει έναν ακέραιο αριθμό που αντιστοιχεί στις ώρες στάθμευσης ενός οχήματος. Στη συνέχεια να υπολογίζει και να εμφανίζει τη συνολική χρέωση. Αν δοθεί ως χρόνος στάθμευσης τιμή εκτός του διαστήματος 1-8, να εμφανίζεται μήνυμα λάθους «Μη έγκυρος χρόνος».
Με βάση τις παραπάνω προδιαγραφές, να δημιουργήσετε κατάλληλα σενάρια για να πραγματοποιήσετε έλεγχο ακραίων τιμών.

Η τεχνική που περιγράφηκε ανωτέρω, μπορεί να εφαρμοστεί και σε υποπρογράμματα. Αυτό είναι συνηθισμένο σε μεγάλα προγράμματα που αποτελούνται από χιλιάδες γραμμές κώδικα τα οποία είναι οργανωμένα σε υποπρογράμματα. Σε αυτές τις περιπτώσεις πρώτα ελέγχεται κάθε υποπρόγραμμα μεμονωμένα. Αφού διαπιστωθεί η ορθή λειτουργία του καθενός, μόνο τότε πραγματοποιείται έλεγχος ολόκληρου του προγράμματος.
Στα προηγούμενα παραδείγματα, η είσοδος ήταν ακέραιος αριθμός οπότε ο υπολογισμός των ακραίων τιμών που έλειπαν ήταν εύκολη διαδικασία προσθέτοντας ή αφαιρώντας 1. Αν η είσοδος λαμβάνει πραγματικές τιμές, τότε για τον υπολογισμό των ακραίων τιμών θα ληφθεί υπόψη η ακρίβεια δεκαδι-

κών ψηφίων με την οποία λειτουργεί το πρόγραμμα. Αν για παράδειγμα στην εκφώνηση αναφέρεται απαίτηση για ακρίβεια 2 δεκαδικών ψηφίων, τότε θα προσθέσουμε ή θα αφαιρέσουμε 0,01 από το άκρο του προηγούμενου ή επόμενου διαστήματος αντίστοιχα. Αν δεν υπάρχει σχετική αναφορά στην εκφώνηση, τότε θα αποφασίσουμε εμείς, π.χ. αν επιλέξουμε ακρίβεια 1 δεκαδικού ψηφίου θα προσθέσουμε ή θα αφαιρέσουμε 0,1.

Δραστηριότητα 7 – Σχετική υγρασία αέρα

Η σχετική υγρασία του αέρα είναι ένας δείκτης της ποσότητας υδρατμών που περιέχει ο αέρας και εκφράζεται ως ποσοστό. Για εσωτερικούς χώρους, το ιδανικό επίπεδο σχετικής υγρασίας για τον άνθρωπο είναι από 30% έως 60%, με τιμές εκτός αυτών των ορίων να προκαλούν δυσφορία. Να αναπτύξετε διαδικασία σε ΓΛΩΣΣΑ, η οποία να δέχεται ως είσοδο μια πραγματική τιμή από 0 έως και 1 που αντιστοιχεί στη σχετική υγρασία του αέρα. Στη συνέχεια να εμφανίζει μήνυμα «Ιδανική υγρασία» αν η σχετική υγρασία είναι από 0,3 έως και 0,6. Αν η σχετική υγρασία είναι χαμηλότερη από 0,3 να εμφανίζει μήνυμα «Ξηρός αέρας», ενώ αν είναι μεγαλύτερη από 0,6 να εμφανίζει μήνυμα «Υγρός αέρας». Σε περίπτωση που δοθεί τιμή εκτός του διαστήματος 0-1, να εμφανίζεται μήνυμα λάθους «Μη έγκυρη τιμή». Ο έλεγχος της σχετικής υγρασίας να γίνει με ακρίβεια 2 δεκαδικών ψηφίων.
Με βάση τις παραπάνω προδιαγραφές, να δημιουργήσετε κατάλληλα σενάρια για να πραγματοποιήσετε έλεγχο ακραίων τιμών.

Τα παραδείγματα που παρουσιάστηκαν αφορούσαν σε περιπτώσεις με μία μόνο είσοδο. Αν οι είσοδοι είναι περισσότερες, τότε κάθε μία πρέπει να ελεγχθεί ανεξάρτητα από τις άλλες. Για να γίνει αυτό, δημιουργούνται σενάρια ελέγχου όπου μία από τις εισόδους λαμβάνει όλες τις ακραίες τιμές ενώ οι υπόλοιπες διατηρούνται σταθερές δίνοντας μια οποιαδήποτε έγκυρη τιμή. Αυτό επαναλαμβάνεται για όλες τις εισόδους. Γίνεται κατανοητό ότι η αύξηση των δεδομένων εισόδου οδηγεί σε μεγάλη αύξηση των σεναρίων ελέγχου, ενώ η διαδικασία αρχίζει να γίνεται περίπλοκη. Γι' αυτό και δεν ασχοληθήκαμε με τέτοιες περιπτώσεις.

5. 3 Ερωτήσεις - Ασκήσεις

Ε.1: Περιγράψτε τις τρεις βασικές κατηγορίες λαθών και δώστε ένα παράδειγμα για κάθε μία από αυτές.

Ε.2: Χαρακτηρίστε τις παρακάτω προτάσεις ως Σωστές ή Λάθος. Στην περίπτωση που πιστεύετε ότι είναι λανθασμένες δικαιολογήστε την επιλογή σας και σκεφτείτε ποια θα μπορούσε να είναι η αντίστοιχη σωστή πρόταση.

α/α	Προτάσεις	Σ	Λ
1	Στον έλεγχο «μαύρου κουτιού» τα σενάρια ελέγχου βασίζονται στον κώδικα του προγράμματος.	❒	❒
2	Τα σενάρια ελέγχου περιλαμβάνουν και μη έγκυρες τιμές εισόδου.	❒	❒
3	Κατά τον έλεγχο ακραίων τιμών ελέγχονται τυχαίες τιμές από κάθε διάστημα της εισόδου.	❒	❒
4	Ο έλεγχος «μαύρου κουτιού» μπορεί να εφαρμοστεί και σε υποπρογράμματα.	❒	❒

Ε.3: Στο Λύκειο, για την ετήσια επίδοση των μαθητών και μαθητριών χρησιμοποιείται ο γενικός μέσος όρος (Γ.Μ.Ο.) που είναι πραγματικός αριθμός από 0 μέχρι και 20 με ακρίβεια ενός δεκαδικού ψηφίου. Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ, το οποίο να διαβάζει έναν πραγματικό αριθμό που να αντιστοιχεί στον Γ.Μ.Ο. ενός μαθητή ή μιας μαθήτριας. Αν ο Γ.Μ.Ο. είναι τουλάχιστον 9,5 να εμφανίζεται μήνυμα «Προάγεται», διαφορετικά να εμφανίζεται μήνυμα «Παραπέμπεται σε επανεξέταση». Αν δοθεί τιμή εκτός του διαστήματος 0-20, να εμφανίζεται μήνυμα «Μη έγκυρος Γ.Μ.Ο.».
Σύμφωνα με τις παραπάνω προδιαγραφές, να πραγματοποιήσετε έλεγχο ακραίων τιμών δημιουργώντας τα κατάλληλα σενάρια.

Ε.4: Μια τουριστική επιχείρηση διαθέτει διαμερίσματα για βραχυχρόνια μίσθωση σύμφωνα με την ακόλουθη τιμολογιακή πολιτική: για διαμονή έως και 3 ημέρες 50€/ημέρα, για διαμονή έως και 7 ημέρες 47€/ημέρα, για διαμονή έως και 20 ημέρες 42€/ημέρα. Ο μέγιστος χρόνος μίσθωσης κάθε διαμερίσματος είναι 20 ημέρες. Να αναπτύξετε συνάρτηση σε ΓΛΩΣΣΑ, η οποία να δέχεται ως είσοδο το πλήθος των ημερών διαμονής και να επιστρέφει τη συνολική χρέωση. Σε περίπτωση που δοθεί είσοδος εκτός του διαστήματος 1-20 η συνάρτηση να επιστρέφει την τιμή -1.
Να δημιουργήσετε κατάλληλα σενάρια με βάση τις παραπάνω προδιαγραφές, για να πραγματοποιήσετε έλεγχο ακραίων τιμών.

Βιβλιογραφία

Ελληνική

Καμέας, Α. (2008). *Τεχνικές Προγραμματισμού*, Ελληνικό Ανοικτό Πανεπιστήμιο.

Κοίλιας, Χ. (2004). *Δομές Δεδομένων και Οργανώσεις Αρχείων*. Αθήνα: Εκδόσεις Νέων Τεχνολογιών.

Κωτσάκης, Σ. & Ταταράκη, Α. (2006). *Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον*. Αθήνα: Εκδόσεις Λιβάνη.

Μποζάνης, Π. Δ. (2006). *Δομές Δεδομένων*. Εκδόσεις Τζιόλα.

Παπουτσής, Ι. (2010). *Εισαγωγή στις Δομές Δεδομένων και στους Αλγόριθμους – Υλοποίηση σε C.* Εκδόσεις Σταμούλης.

Χατζηγιαννάκης, Ν. (2012*). Η Γλώσσα C σε Βάθος*, 4η Έκδοση. Αθήνα: Εκδόσεις Κλειδάριθμος.

Barnes, D.K., & Kolling, M. (2008). *Αντικειμενοστρεφής Προγραμματισμός σε Java*, Εκδόσεις Κλειδάριθμος.

Cormen, T.H., Leiserson. C.E., Rivest, R.L., & Stein, C. (2012). *Εισαγωγή στους αλγορίθμους*. USA/Ελλάδα, MIT Press/Ίδρυμα Τεχνολογίας & Έρευνας - Πανεπιστημιακές Εκδόσεις Κρήτης.

Knuth, D.E. (2009). *Η τέχνη του προγραμματισμού: θεμελιώδεις αλγόριθμοι*, Τόμος Α. 3η Έκδοση. Ελλάδα, Εκδόσεις Τζιόλα.

Lethbridge, T., & Laganiére, R. (2017). *Μηχανική Αντικειμενοστραφούς Λογισμικού, Πρακτική Ανάπτυξη Λογισμικού με τις γλώσσες UML και Java*, Εκδόσεις Τζιόλα.

Wirth, N. (2004). *Αλγόριθμοι και Δομές Δεδομένων.* Αθήνα: Εκδόσεις Κλειδάριθμος.

Ξενόγλωσση

British Computer Society Specialist Interest Group in Software Testing (2001). *Standard for software component testing*, Working Draft 3.4, 27 April 2001.

Budd, T. (2002). *An Introduction to Object-Oriented Programming* (3rd ed.), Addison Wesley.

Carrano, F.M., & Henry, T. (2013). *Data abstraction and problem solving with C++: walls and mirrors*. 6th ed. UK: Pearson Education Limited.

Kalicharan, N. (2013). *Advanced Topics in C – Core Topics in Data Structures*. USA: Apress.

Martin, R. C. (2009). *Clean code: a handbook of agile software craftsmanship*. Pearson Education.

McConnell, S. (2004). *Code complete: A practical handbook of software construction* (2nd ed.). Redmond, WA: Microsoft Press.

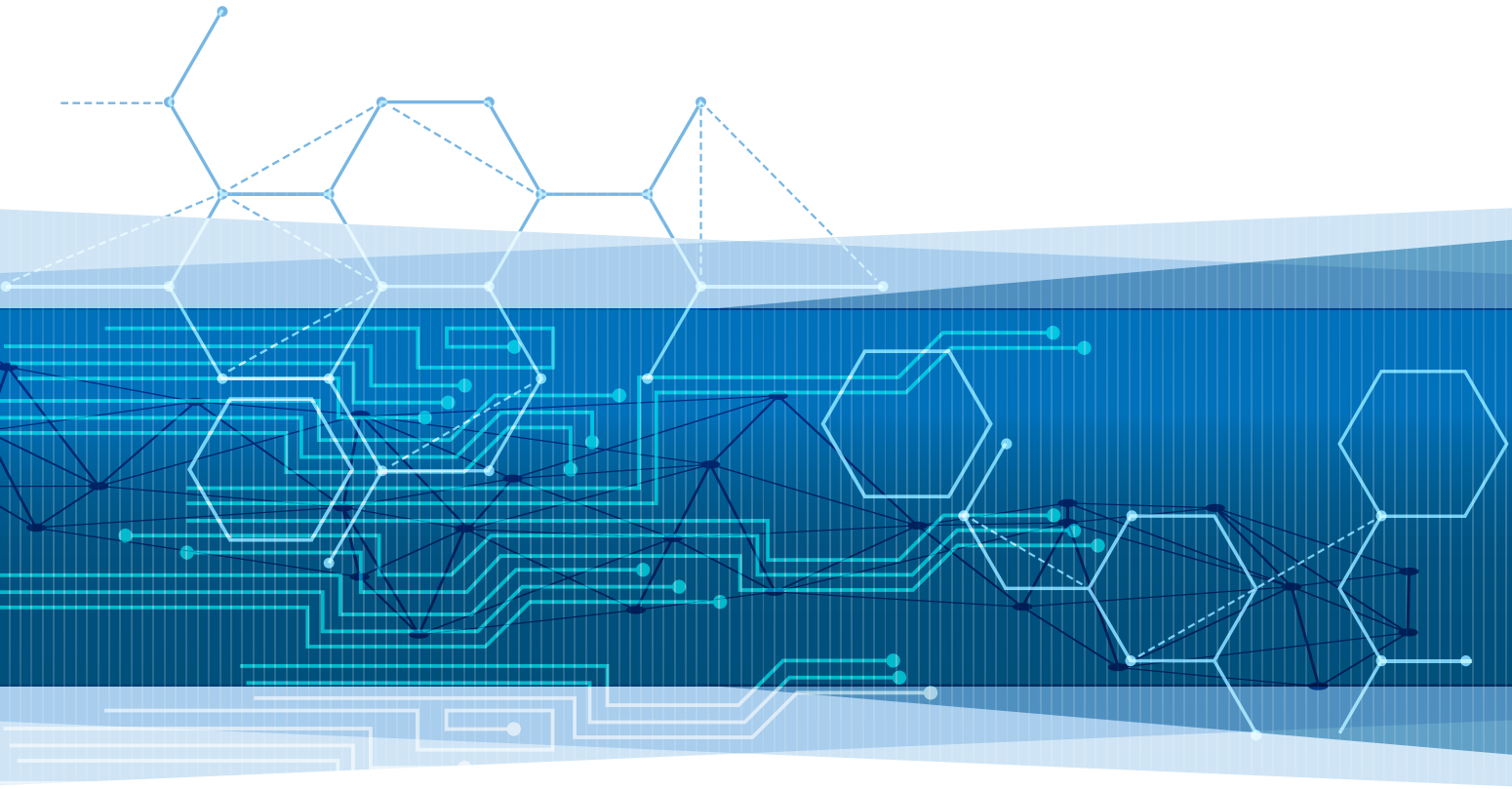
Mehlhorn, K., & Sanders, P. (2008*). Algorithms and Data Structures: The Basic Toolbox*. Springer-Verlag.

Myers, G. J., Sandler, C., & Badgett, T. (2012). *The art of software testing* (3rd ed.). Hoboken, NJ: John Wiley & Sons.

Sedgewick, R., & Wayne, K. (2011). *Algorithms* (4th ed). Addison-Wesley.

Thompson, K. (2015). *Zero bugs and program faster*. Kate Thompson Books.

Βάσει του ν. 3966/2011 τα διδακτικά βιβλία του Δημοτικού, του Γυμνασίου, του Λυκείου, των ΕΠΑ.Λ. και των ΕΠΑ.Σ. τυπώνονται από το ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ και διανέμονται δωρεάν στα Δημόσια Σχολεία. Τα βιβλία μπορεί να διατίθενται προς πώληση, όταν φέρουν στη δεξιά κάτω γωνία του εμπροσθόφυλλου ένδειξη «ΔΙΑΤΙΘΕΤΑΙ ΜΕ ΤΙΜΗ ΠΩΛΗΣΗΣ». Κάθε αντίτυπο που διατίθεται προς πώληση και δεν φέρει την παραπάνω ένδειξη θεωρείται κλεψίτυπο και ο παραβάτης διώκεται σύμφωνα με τις διατάξεις του άρθρου 7 του νόμου 1129 της 15/21 Μαρτίου 1946 (ΦΕΚ 1946,108, Α').



Κωδικός Βιβλίου: 0-22-0260
ISBN 978-960-06-5997-9

ΙΤΥΕ
"ΔΙΟΦΑΝΤΟΣ"
ινστιτουτο
τεχνολογιας
υπολογιστων & εκδοσεων

(01) 000000 0 22 0260 0