

Δομές Προγραμματισμού

Βήμα 1: Εισαγωγή στον Αλγοριθμικό Προγραμματισμό

Ο προγραμματισμός στηρίζεται σε **αλγορίθμους** – μια σειρά από βήματα που οδηγούν στην επίλυση ενός προβλήματος. Για να εκφράσουμε έναν αλγόριθμο χρησιμοποιούμε μια **ψευδογλώσσα**, η οποία είναι ανεξάρτητη από μια συγκεκριμένη γλώσσα προγραμματισμού αλλά ακολουθεί συγκεκριμένες δομές και εντολές.

Βήμα 2: Δομή Ακολουθίας

Η **δομή ακολουθίας** σημαίνει ότι οι εντολές εκτελούνται με τη σειρά που γράφονται, από πάνω προς τα κάτω. Είναι η βασικότερη δομή.

Θεωρία:

- Η σειριακή δομή δεν περιέχει αποφάσεις ή επαναλήψεις.
- Χρησιμοποιείται για απλά προβλήματα όπου κάθε βήμα ακολουθεί το προηγούμενο.

Παράδειγμα 1: Άθροισμα δύο αριθμών

Αλγόριθμος Άθροισμα

Διάβασε a, b

$\Sigma \leftarrow a + b$

Εμφάνισε Σ

Τέλος Άθροισμα

Εξήγηση:

1. Διαβάζουμε δύο τιμές (a, b).
 2. Υπολογίζουμε το άθροισμά τους και το αποθηκεύουμε στη μεταβλητή Σ .
 3. Εμφανίζουμε το αποτέλεσμα.
-

Βήμα 3: Δομή Επιλογής

Η **δομή επιλογής** χρησιμοποιείται όταν η ροή του προγράμματος εξαρτάται από μια **συνθήκη**.

Απλή Επιλογή (Αν-Τότε)

- Εκτελείται μια ομάδα εντολών **μόνο αν** η συνθήκη είναι αληθής.

Παράδειγμα 2: Απόλυτη τιμή

Αλγόριθμος Απόλυτη_Τιμή

Διάβασε a

Αν $a < 0$ τότε

$a \leftarrow a * (-1)$

Τέλος_αν

Εμφάνισε a

Τέλος Απόλυτη_Τιμή

Σύνθετη Επιλογή (Αν-Αλλιώς)

- Αν η συνθήκη είναι αληθής, εκτελείται η πρώτη ομάδα εντολών, αλλιώς η δεύτερη.

Παράδειγμα 3: Ζυγός ή περιττός

Αλγόριθμος Ζυγός_Περιττός
Διάβασε α
Αν $a \bmod 2 = 0$ τότε
 Εμφάνισε "Ζυγός"
αλλιώς
 Εμφάνισε "Περιττός"
Τέλος_αν
Τέλος Ζυγός_Περιττός

Πολλαπλή Επιλογή (Αν-Αλλιώς_Αν)

- Έλεγχος πολλαπλών συνθηκών με τη σειρά.

Παράδειγμα 4: Επίδομα ανά ηλικία

Αλγόριθμος Επίδομα
Διάβασε ηλικία
Αν ηλικία < 18 τότε
 Επίδομα ← 50
αλλιώς αν ηλικία ≤ 65 τότε
 Επίδομα ← 30
αλλιώς
 Επίδομα ← 100
Τέλος_αν
Εμφάνισε Επίδομα
Τέλος Επίδομα

Βήμα 4: Εμφωλευμένες Εντολές Επιλογής

Μπορούμε να βάλουμε μια εντολή επιλογής **μέσα** σε μια άλλη.

Παράδειγμα 5: Τρίγωνο και τύπος Ήρωνα

Αλγόριθμος Εμβαδό_Τριγώνου
Διάβασε α, β, γ
Αν $\alpha + \beta > \gamma$ και $\beta + \gamma > \alpha$ και $\gamma + \alpha > \beta$ τότε
 $\tau \leftarrow (\alpha + \beta + \gamma) / 2$
 $E \leftarrow T_P(\tau * (\tau - \alpha) * (\tau - \beta) * (\tau - \gamma))$
 Εμφάνισε E
αλλιώς
 Εμφάνισε "Δεν σχηματίζεται τρίγωνο"
Τέλος_αν
Τέλος Εμβαδό_Τριγώνου

Βήμα 5: Δομή Επανάληψης

Χρησιμοποιείται όταν θέλουμε να επαναλάβουμε μια ομάδα εντολών πολλές φορές.

1. Όσο ... επανάλαβε

- Εκτελείται όσο η συνθήκη είναι **αληθής**.
- Ο έλεγχος γίνεται **στην αρχή**.

Παράδειγμα 6: Εκτύπωση αριθμών 1 έως N

Αλγόριθμος Σειρά_Αριθμών
Διάβασε N
 $i \leftarrow 1$
Όσο $i \leq N$ επανάλαβε
 Εμφάνισε i
 $i \leftarrow i + 1$
Τέλος_επανάληψης
Τέλος Σειρά_Αριθμών

2. Επανάλαβε ... Μέχρις_ότου

- Οι εντολές εκτελούνται τουλάχιστον **μία φορά**.
- Ο έλεγχος γίνεται **στο τέλος**.

Παράδειγμα 7: Επιβεβαίωση κωδικού

Αλγόριθμος Εισαγωγή_Κωδικού
Επανάλαβε
 Εμφάνισε "Δώσε κωδικό:"
 Διάβασε κωδικός
 Μέχρις_ότου κωδικός = "1234"
 Εμφάνισε "Σωστός κωδικός!"
Τέλος Εισαγωγή_Κωδικού

3. Για ... από ... μέχρι

- Χρησιμοποιείται όταν γνωρίζουμε **ακριβώς πόσες φορές** θα επαναληφθεί.
- Η μεταβλητή-μετρητής αυξάνεται αυτόματα.

Παράδειγμα 8: Άθροισμα 10 αριθμών

Αλγόριθμος Άθροισμα_10
 $\Sigma \leftarrow 0$
Για i από 1 μέχρι 10
 Διάβασε α
 $\Sigma \leftarrow \Sigma + \alpha$
Τέλος_επανάληψης
Εμφάνισε Σ
Τέλος Άθροισμα_10

Βήμα 6: Κατάλογος Επιλογών (Menu)

Χρησιμοποιείται για να δίνουμε στο χρήστη πολλαπλές επιλογές, συνήθως με βρόχο έως ότου επιλεγεί μια έγκυρη επιλογή.

Παράδειγμα 9: Απλό μενού

Αλγόριθμος Μενού
Επανάλαβε
 Εμφάνισε "1. Εισαγωγή δεδομένων"
 Εμφάνισε "2. Επεξεργασία"
 Εμφάνισε "3. Έξοδος"
 Εμφάνισε "Επίλεξε:"

Διάβασε επιλογή
Μέχρις_ότου επιλογή ≥ 1 και επιλογή ≤ 3
Εμφάνισε "Επιλέχθηκε: ", επιλογή
Τέλος Μενού

Βήμα 7: Εμφωλευμένες Εντολές Επανάληψης

Μπορούμε να βάλουμε έναν βρόχο μέσα σε έναν άλλο.

Παράδειγμα 10: Πίνακας πολλαπλασιασμού

Αλγόριθμος Πίνακας_Πολλαπλασιασμού

Για i από 1 μέχρι 10

 Για j από 1 μέχρι 10

 Εμφάνισε i, " * ", j, " = ", i * j

 Τέλος_επανάληψης

Τέλος_επανάληψης

Τέλος Πίνακας_Πολλαπλασιασμού

Βήμα 8: Τμηματικός Προγραμματισμός

Ο τμηματικός προγραμματισμός σημαίνει ότι σπάμε ένα μεγάλο πρόγραμμα σε μικρότερα, επαναχρησιμοποιήσιμα τμήματα (υποαλγόριθμους). Χρησιμοποιούμε την εντολή Κάλεσε.

Πλεονεκτήματα:

1. **Ευανάγνωστος κώδικας** – κάθε τμήμα εκτελεί μια ξεχωριστή λειτουργία.
2. **Επαναχρησιμοποίηση** – το ίδιο τμήμα καλείται πολλές φορές.
3. **Ευκολότερος έλεγχος** – ελέγχουμε κάθε τμήμα ξεχωριστά.
4. **Συνεργασία** – πολλοί προγραμματιστές μπορούν να δουλέψουν σε διαφορετικά τμήματα.

Παράδειγμα 11: Τμηματικός αλγόριθμος

Αλγόριθμος Κύριος

 Κάλεσε Εισαγωγή_Δεδομένων

 Κάλεσε Επεξεργασία

 Κάλεσε Εκτύπωση_Αποτελεσμάτων

Τέλος Κύριος

Αλγόριθμος Εισαγωγή_Δεδομένων

 Διάβασε α, β

Τέλος Εισαγωγή_Δεδομένων

Αλγόριθμος Επεξεργασία

$\Sigma \leftarrow \alpha + \beta$

Τέλος Επεξεργασία

Αλγόριθμος Εκτύπωση_Αποτελεσμάτων

 Εμφάνισε Σ

Τέλος Εκτύπωση_Αποτελεσμάτων

Βήμα 9: Συνοπτικός Πίνακας Δομών

Δομή	Σύνταξη (ψευδογλώσσα)	Χρήση
Ακολουθία	Διάβασε, ←, Εμφάνισε	Γραμμική εκτέλεση
Απλή επιλογή	Αν συνθήκη τότε ... Τέλος_αν	1 πιθανότητα
Σύνθετη επιλογή	Αν ... τότε ... αλλιώς ... Τέλος_αν	2 πιθανότητες
Πολλαπλή επιλογή	Αν ... αλλιώς_αν ... αλλιώς ...	>2 πιθανότητες
Όσο επανάλαβε	Όσο συνθήκη επανάλαβε ...	Εκτέλεση όσο ισχύει η συνθήκη (έλεγχος στην αρχή)
Επανάλαβε μέχρις_ότου	Επανάλαβε ... Μέχρις_ότου συνθήκη	Εκτέλεση τουλάχιστον 1 φορά (έλεγχος στο τέλος)
Για από μέχρι	Για i από τ1 μέχρι τ2 ...	Εκτέλεση για γνωστό πλήθος επαναλήψεων
Κλήση αλγορίθμου	Κάλεσε Όνομα_Αλγορίθμου	Τμηματικός προγραμματισμός

Βήμα 10: Συμβουλές για Μαθητές

1. Σχεδιάζετε πρώτα σε χαρτί τη λογική του προβλήματος (διάγραμμα ροής).
2. Χρησιμοποιείτε σχόλια (όπως ! Αυτό είναι σχόλιο) για να εξηγείτε τα βήματα.
3. Δοκιμάζετε με απλά δεδομένα πριν προχωρήσετε σε πολύπλοκα.
4. Προσέχετε την ιεραρχία πράξεων και τις παρενθέσεις.
5. Επαναχρησιμοποιείτε κώδικα με υποαλγόριθμους για να εξοικονομείτε χρόνο.

1. Δομή Ακολουθίας: Βασική και Αναλυτική Θεωρία

Θεωρητική Ανάλυση:

Η δομή ακολουθίας είναι η πιο θεμελιώδης δομή στον προγραμματισμό. Σε αυτήν, οι εντολές εκτελούνται με **αυστηρά σειριακή σειρά**, χωρίς αποκλίσεις ή επαναλήψεις. Κάθε εντολή είναι ένα ανεξάρτητο βήμα που εξαρτάται μόνο από τα αποτελέσματα των προηγούμενων εντολών.

Βασικές Εντολές Ακολουθίας:

1. Εισαγωγή Δεδομένων:

- Διάβασε μεταβλητή1, μεταβλητή2, ...
- Δεδομένα // μεταβλητή1, μεταβλητή2 //

2. Εκχώρηση Τιμών:

- μεταβλητή ← έκφραση
- Η έκφραση μπορεί να περιλαμβάνει σταθερές, μεταβλητές, συναρτήσεις και τελεστές

3. Έξοδος Αποτελεσμάτων:

- Εμφάνισε "κείμενο", μεταβλητή
- Γράψε μεταβλητή
- Αποτελέσματα // μεταβλητή //

Λεπτομερή Παραδείγματα:

Παράδειγμα 1.1: Υπολογισμός εμβαδού ορθογωνίου

Αλγόριθμος Εμβαδό_Ορθογωνίου

! Δήλωση μεταβλητών

Πραγματικές: μήκος, πλάτος, εμβαδό

! Εισαγωγή δεδομένων

Εμφάνισε "Δώσε το μήκος:"

Διάβασε μήκος

Εμφάνισε "Δώσε το πλάτος:"

Διάβασε πλάτος

! Υπολογισμοί

εμβαδό ← μήκος * πλάτος

! Έξοδος αποτελεσμάτων

Εμφάνισε "Το εμβαδόν του ορθογωνίου είναι: ", εμβαδό

Τέλος Εμβαδό_Ορθογωνίου

Παράδειγμα 1.2: Μετατροπή θερμοκρασίας

Αλγόριθμος Μετατροπή_Θερμοκρασίας

! Μετατροπή από Κελσίου σε Φαρενάιτ

Πραγματικές: Κελσίου, Φαρενάιτ

Εμφάνισε "Δώσε τη θερμοκρασία σε βαθμούς Κελσίου:"

Διάβασε Κελσίου

Φαρενάιτ ← (Κελσίου * 9/5) + 32

Εμφάνισε Κελσίου, "°C = ", Φαρενάιτ, "°F"
Τέλος Μετατροπή_Θερμοκρασίας

Παράδειγμα 1.3: Χρήση Δεδομένα/Αποτελέσματα

Αλγόριθμος Υπολογισμός_ΦΠΑ

Δεδομένα // καθαρή_αξία, ποσοστό_ΦΠΑ //

! Οι μεταβλητές έχουν ήδη τιμές

τελική ← καθαρή_αξία + (καθαρή_αξία * ποσοστό_ΦΠΑ / 100)

Αποτελέσματα // τελική //

! Η τελική είναι η μεταβλητή εξόδου

Τέλος Υπολογισμός_ΦΠΑ

2. Δομή Επιλογής: Αναλυτική Θεωρία και Τύποι

Βασική Θεωρία:

Η δομή επιλογής επιτρέπει στο πρόγραμμα να παίρνει αποφάσεις. Βασίζεται στην αξιολόγηση λογικών εκφράσεων που έχουν τιμή Αληθής ή Ψευδής.

Τύποι Λογικών Εκφράσεων:

1. Αριθμητικές Συνθήκες:

$\alpha > \beta$

$\chi \leq 100$

$\alpha + \beta = \gamma$

2. Συνδυασμένες Συνθήκες:

$(\alpha > 0)$ και $(\alpha < 10)$

$(\beta = 0)$ ή $(\gamma \neq 5)$

όχι (τελικό ≥ 50)

3. Αλφαριθμητικές Συνθήκες:

όνομα = "ΜΑΡΙΑ"

επώνυμο > "ΚΑΡΑ"

Τύποι Δομών Επιλογής:

A) Απλή Επιλογή (If-Then)

Σύνταξη:

Αν συνθήκη τότε

 εντολή1

 εντολή2

...

Τέλος_αν

Λεπτομερές Παράδειγμα 2.1: Έλεγχος ενηλικίωσης

Αλγόριθμος Έλεγχος_Ενηλικίωσης

Ακέραιος: ηλικία

Εμφάνισε "Πόσων χρονών είσαι;"

Διάβασε ηλικία

Αν ηλικία \geq 18 τότε
Εμφάνισε "Είσαι ενήλικας"
Εμφάνισε "Έχεις πλήρη πολιτικά δικαιώματα"
Τέλος_αν

! Αυτές οι εντολές εκτελούνται πάντα
Εμφάνισε "Η διαδικασία ολοκληρώθηκε"
Τέλος Έλεγχος_Ενηλικίωσης

B) Επιλογή (If-Then-Else)

Σύνταξη:

Αν συνθήκη τότε
 εντολές_για_αληθές
αλλιώς
 εντολές_για_ψευδές
Τέλος_αν

Λεπτομερές Παράδειγμα 2.2: Έλεγχος αριθμού

Αλγόριθμος Έλεγχος_Αριθμού

Πραγματικός: αριθμός

Εμφάνισε "Δώσε έναν αριθμό:"

Διάβασε αριθμός

Αν αριθμός \geq 0 τότε
 Εμφάνισε "Ο αριθμός είναι θετικός ή μηδέν"
 ! Υπολογίζουμε την τετραγωνική ρίζα μόνο για μη αρνητικούς
 ρίζα \leftarrow T_P(αριθμός)
 Εμφάνισε "Η τετραγωνική του ρίζα είναι: ", ρίζα
αλλιώς
 Εμφάνισε "Ο αριθμός είναι αρνητικός"
 Εμφάνισε "Δεν έχει πραγματική τετραγωνική ρίζα"
Τέλος_αν
Τέλος Έλεγχος_Αριθμού

Γ) Πολλαπλή Επιλογή (If-Else If-Else)

Σύνταξη:

Αν συνθήκη1 τότε
 εντολές1
αλλιώς_αν συνθήκη2 τότε
 εντολές2
αλλιώς_αν συνθήκη3 τότε
 εντολές3
...
αλλιώς
 εντολές_αλλιώς
Τέλος_αν

Λεπτομερές Παράδειγμα 2.3: Συστήματα βαθμολογίας

Αλγόριθμος Βαθμολόγηση

Πραγματικός: βαθμός

Χαρακτήρας: γράμμα

Εμφάνισε "Δώσε τον βαθμό (0-100):"

Διάβασε βαθμός

Αν βαθμός ≥ 90 τότε

 γράμμα \leftarrow "Α"

 Εμφάνισε "Άριστα!"

αλλιώς_αν βαθμός ≥ 80 τότε

 γράμμα \leftarrow "Β"

 Εμφάνισε "Πολύ Καλά"

αλλιώς_αν βαθμός ≥ 70 τότε

 γράμμα \leftarrow "Γ"

 Εμφάνισε "Καλά"

αλλιώς_αν βαθμός ≥ 60 τότε

 γράμμα \leftarrow "Δ"

 Εμφάνισε "Μέτρια"

αλλιώς

 γράμμα \leftarrow "Ε"

 Εμφάνισε "Αποτυχία"

Τέλος_αν

Εμφάνισε "Η αντίστοιχη βαθμολογία σε γράμμα είναι: ", γράμμα

Τέλος Βαθμολόγηση

3. Εμφωλευμένες Εντολές Επιλογής: Σύνθετες Δομές

Θεωρία:

Οι εμφωλευμένες εντολές επιλογής δημιουργούν **πολυεπίπεδους ελέγχους**. Κάθε επίπεδο ελέγχει μια διαφορετική συνθήκη, επιτρέποντας πολύπλοκη λογική απόφασης.

Προσοχή: Η σωστή χρήση εσοχών (indentation) είναι **απαραίτητη** για την αναγνωσιμότητα.

Λεπτομερή Παράδειγματα:

Παράδειγμα 3.1: Σύστημα ταξινόμησης προϊόντων

Αλγόριθμος Ταξινόμηση_Προϊόντων

Χαρακτήρας: κατηγορία, υπόκατηγορία

Πραγματικός: τιμή

Εμφάνισε "Δώσε την κατηγορία (Η/Β/Π): "

Διάβασε κατηγορία

Αν κατηγορία = "Η" τότε ! Ηλεκτρονικά

 Εμφάνισε "Δώσε την τιμή: "

 Διάβασε τιμή

Αν τιμή > 1000 τότε

Εμφάνισε "Ακριβά ηλεκτρονικά"

υπόκατηγορία ← "Α"

αλλιώς_αν τιμή > 500 τότε

Εμφάνισε "Μεσαία ηλεκτρονικά"

υπόκατηγορία ← "Β"

αλλιώς

Εμφάνισε "Οικονομικά ηλεκτρονικά"

υπόκατηγορία ← "Γ"

Τέλος_αν

αλλιώς_αν κατηγορία = "Β" τότε ! Βιβλία

Εμφάνισε "Δώσε την τιμή: "

Διάβασε τιμή

Αν τιμή > 50 τότε

Εμφάνισε "Εκπαιδευτικά βιβλία"

υπόκατηγορία ← "Α"

αλλιώς

Εμφάνισε "Λογοτεχνία"

υπόκατηγορία ← "Β"

Τέλος_αν

αλλιώς_αν κατηγορία = "Π" τότε ! Προσφορά

Εμφάνισε "ΕΙΔΙΚΗ ΠΡΟΣΦΟΡΑ"

υπόκατηγορία ← "Π"

αλλιώς

Εμφάνισε "Λάθος κατηγορία"

υπόκατηγορία ← "Άγνωστη"

Τέλος_αν

Εμφάνισε "Κατηγορία: ", κατηγορία, ", Υποκατηγορία: ", υπόκατηγορία

Τέλος Ταξινόμηση_Προϊόντων

Παράδειγμα 3.2: Σύστημα πρόσβασης

Αλγόριθμος Σύστημα_Πρόσβασης

Χαρακτήρας: όνομα, κωδικός, ρόλος

Εμφάνισε "Όνομα χρήστη: "

Διάβασε όνομα

Εμφάνισε "Κωδικός: "

Διάβασε κωδικός

Αν όνομα = "ΔΙΑΧΕΙΡΙΣΤΗΣ" τότε

Αν κωδικός = "ADMIN123" τότε

ρόλος ← "ΔΙΑΧΕΙΡΙΣΤΗΣ"

Εμφάνισε "Πρόσβαση σε όλες τις λειτουργίες"

αλλιώς
Εμφάνισε "Λάθος κωδικός διαχειριστή"
ρόλος ← "ΑΝΕΥ ΠΡΟΣΒΑΣΗΣ"
Τέλος_αν

αλλιώς_αν όνομα = "ΧΡΗΣΤΗΣ" τότε
Αν κωδικός = "USER456" τότε
ρόλος ← "ΧΡΗΣΤΗΣ"
Εμφάνισε "Πρόσβαση σε βασικές λειτουργίες"
αλλιώς
Εμφάνισε "Λάθος κωδικός χρήστη"
ρόλος ← "ΑΝΕΥ ΠΡΟΣΒΑΣΗΣ"
Τέλος_αν

αλλιώς
Εμφάνισε "Άγνωστος χρήστης"
ρόλος ← "ΑΝΕΥ ΠΡΟΣΒΑΣΗΣ"
Τέλος_αν

Εμφάνισε "Τελικός ρόλος: ", ρόλος
Τέλος Σύστημα_Πρόσβασης

4. Δομή Επανάληψης:

Θεωρητική Ανάλυση: Οι δομές επανάληψης επιτρέπουν την **αυτόματη επανάληψη** μιας ομάδας εντολών. Χρησιμοποιούνται όταν μία διαδικασία πρέπει να επαναληφθεί πολλές φορές.

Τύποι Δομών Επανάληψης:

A) Όσο...Επανάλαβε (While Loop)

Χαρακτηριστικά:

- Ο έλεγχος γίνεται **στην αρχή**
- Μπορεί να μην εκτελεστεί **ποτέ** αν η συνθήκη είναι ψευδής από την αρχή
- Χρησιμοποιείται όταν δεν γνωρίζουμε το ακριβές πλήθος επαναλήψεων

Σύνταξη:

Όσο συνθήκη επανάλαβε
εντολή1
εντολή2

...

Τέλος_επανάληψης

Λεπτομερές Παράδειγμα 4.1: Υπολογισμός μέσου όρου με απροσδιόριστο πλήθος

Αλγόριθμος Μέσος_Όρος_Απροσδιόριστος
Πραγματικός: αριθμός, άθροισμα, μέσος_όρος
Ακέραιος: πλήθος
άθροισμα ← 0
πλήθος ← 0

Εμφάνισε "Δώσε αριθμούς (0 για τερματισμό):"

Διάβασε αριθμός

Όσο αριθμός $\neq 0$ επανάλαβε
 άθροισμα \leftarrow άθροισμα + αριθμός
 πλήθος \leftarrow πλήθος + 1

Εμφάνισε "Δώσε επόμενο αριθμό (0 για τερματισμό):"
Διάβασε αριθμός
Τέλος_επανάληψης

Αν πλήθος > 0 τότε
 μέσος_όρος \leftarrow άθροισμα / πλήθος
 Εμφάνισε "Μέσος όρος των ", πλήθος, " αριθμών: ", μέσος_όρος
αλλιώς
 Εμφάνισε "Δεν δόθηκαν αριθμοί"
Τέλος_αν
Τέλος Μέσος_Όρος_Απροσδιόριστος

B) Επανάλαβε...Μέχρις_ότου (Do-While Loop)

Χαρακτηριστικά:

- Ο έλεγχος γίνεται **στο τέλος**
- Εκτελείται **τουλάχιστον μία φορά**
- Χρησιμοποιείται για **επιβεβαίωση εισόδου ή μενού επιλογών**

Σύνταξη:

Επανάλαβε
 εντολή1
 εντολή2
 ...
Μέχρις_ότου συνθήκη

Λεπτομερές Παράδειγμα 4.2: Έλεγχος εγκυρότητας εισαγωγής

Αλγόριθμος Εισαγωγή_Ηλικίας

Ακέραιος: ηλικία

Λογική: έγκυρη

Επανάλαβε
 Εμφάνισε "Δώσε την ηλικία σου (1-120): "
 Διάβασε ηλικία

 έγκυρη \leftarrow (ηλικία ≥ 1) και (ηλικία ≤ 120)

Αν NOT(έγκυρη) τότε
 Εμφάνισε "Λάθος ηλικία! Δώσε τιμή μεταξύ 1 και 120."
Τέλος_αν

Μέχρις_ότου έγκυρη
 Εμφάνισε "Ηλικία αποδεκτή: ", ηλικία

! Ταξινόμηση ανά ηλικία
Αν ηλικία < 13 τότε
 Εμφάνισε "Παιδί"
αλλιώς_αν ηλικία < 20 τότε
 Εμφάνισε "Έφηβος"
αλλιώς_αν ηλικία < 65 τότε
 Εμφάνισε "Ενήλικας"
αλλιώς
 Εμφάνισε "Συνταξιούχος"
Τέλος_αν
Τέλος Εισαγωγή_Ηλικίας

Γ) Για...Από...Μέχρι (For Loop)

Χαρακτηριστικά:

- Χρησιμοποιείται όταν **γνωρίζουμε ακριβώς** πόσες φορές θα επαναληφθεί
- Η μεταβλητή-μετρητής αυξάνεται αυτόματα
- Μπορεί να έχει προσαρμοσμένο βήμα

Σύνταξη:

Για μεταβλητή από τιμή1 μέχρι τιμή2 [με_βήμα βήμα]
 εντολή1
 εντολή2
 ...
Τέλος_επανάληψης

Λεπτομερές Παράδειγμα 4.3: Πολλαπλάσια αριθμού

Αλγόριθμος Πολλαπλάσια

Ακέραιος: αριθμός, πολλαπλάσιο, πλήθος

Εμφάνισε "Δώσε έναν αριθμό: "
Διάβασε αριθμός
Εμφάνισε "Πόσα πολλαπλάσια θέλεις; "
Διάβασε πλήθος

Εμφάνισε "Τα πρώτα ", πλήθος, " πολλαπλάσια του ", αριθμός, " είναι:"

Για i από 1 μέχρι πλήθος
 πολλαπλάσιο ← αριθμός * i
 Εμφάνισε i, " x ", αριθμός, " = ", πολλαπλάσιο
Τέλος_επανάληψης

! Άλλος τρόπος με διαφορετικό βήμα
Εμφάνισε "Πολλαπλάσια με βήμα 2:"

Για i από αριθμός μέχρι αριθμός * πλήθος με_βήμα αριθμός
 Εμφάνισε i
Τέλος_επανάληψης
Τέλος Πολλαπλάσια

5. Εμφωλευμένες Επανάληψεις: Σύνθετες Δομές

Θεωρία:

Οι εμφωλευμένες επαναλήψεις δημιουργούν **πολυδιάστατες επαναλήψεις**. Κάθε βρόχος λειτουργεί ανεξάρτητα, αλλά το εσωτερικό βρόχο εκτελείται πλήρως για κάθε επανάληψη του εξωτερικού.

Σπουδαίος Κανόνας: Για κάθε 2 εμφωλευμένους βρόχους, οι συνολικές επαναλήψεις είναι: Εξωτερικές_Επαναλήψεις × Εσωτερικές_Επαναλήψεις

Λεπτομερή Παράδειγματα:

Παράδειγμα 5.1: Πίνακας πολλαπλασιασμού (πλήρης)

Αλγόριθμος Πίνακας_Πολλαπλασιασμού

Ακέραιος: i, j, γινόμενο

Εμφάνισε "Πίνακας Πολλαπλασιασμού 1-10"

Εμφάνισε "=====

! Εκτύπωση επικεφαλίδων

Εμφάνισε " "

Για i από 1 μέχρι 10

Εμφάνισε i:4 ! :4 σημαίνει πλάτος 4 χαρακτήρων

Τέλος_επανάληψης

Εμφάνισε "" ! Νέα γραμμή

! Κύριο μέρος πίνακα

Για i από 1 μέχρι 10

Εμφάνισε i:3, "|" ! Στήλη με τον πολλαπλασιαστή

Για j από 1 μέχρι 10

γινόμενο ← i * j

Εμφάνισε γινόμενο:4

Τέλος_επανάληψης

Εμφάνισε "" ! Νέα γραμμή μετά από κάθε σειρά

Τέλος_επανάληψης

Τέλος Πίνακας_Πολλαπλασιασμού

Παράδειγμα 5.2: Αναζήτηση σε δισδιάστατο πίνακα

Αλγόριθμος Αναζήτηση_Πίνακα

Ακέραιος: πίνακας[3,4], στόχος, i, j

Λογική: βρέθηκε

Ακέραιος: θέση_γραμμή, θέση_στήλη

! Αρχικοποίηση πίνακα 3x4

Εμφάνισε "Αρχικοποίηση πίνακα 3x4:"

Για i από 1 μέχρι 3

Για j από 1 μέχρι 4

πίνακας[i,j] ← (i * 10) + j

Εμφάνισε πίνακας[i,j]:4

Τέλος_επανάληψης
Εμφάνισε ""
Τέλος_επανάληψης

! Αναζήτηση στοιχείου
Εμφάνισε "Δώσε αριθμό για αναζήτηση: "
Διάβασε στόχος

βρέθηκε ← Ψευδής
θέση_γραμμή ← 0
θέση_στήλη ← 0

! Αναζήτηση στο πίνακα
Για i από 1 μέχρι 3
 Για j από 1 μέχρι 4
 Αν πίνακας[i,j] = στόχος τότε
 βρέθηκε ← Αληθής
 θέση_γραμμή ← i
 θέση_στήλη ← j
 ! Βγαίνουμε από τον εσωτερικό βρόχο
 j ← 5 ! Τρόπος για να τερματίσουμε νωρίς τον βρόχο
 Τέλος_αν
 Τέλος_επανάληψη

Αν βρέθηκε τότε
 ! Βγαίνουμε και από τον εξωτερικό βρόχο
 i ← 4
 Τέλος_αν
Τέλος_επανάληψης

! Αποτελέσματα αναζήτησης
Αν βρέθηκε τότε
 Εμφάνισε "Ο αριθμός ", στόχος, " βρέθηκε στη θέση [", θέση_γραμμή, ",", θέση_στήλη, "]"
 αλλιώς
 Εμφάνισε "Ο αριθμός ", στόχος, " δεν βρέθηκε στον πίνακα"
 Τέλος_αν
Τέλος Αναζήτηση_Πίνακα

6. Κατάλογοι Επιλογών (Menus): Πρακτική Εφαρμογή

Θεωρία: Οι κατάλογοι επιλογών παρέχουν στον χρήστη **διάλογο με το σύστημα**. Συνδυάζουν **δομές επανάληψης** και **επιλογής** για να δημιουργήσουν φιλικά προς το χρήστη περιβάλλοντα.

Λεπτομερή Παράδειγματα:

Παράδειγμα 6.1: Πλήρες σύστημα διαχείρισης μαθητών

Αλγόριθμος Σύστημα_Διαχείρισης_Μαθητών

Ακέραιος: επιλογή

Ακέραιος: πλήθος_μαθητών
Πραγματικός: βαθμός
Λογική: έγκυρη_επιλογή
πλήθος_μαθητών ← 0

Επανάλαβε

! Εμφάνιση μενού
Εμφάνισε "=====
Εμφάνισε "ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΜΑΘΗΤΩΝ"
Εμφάνισε "=====
Εμφάνισε "1. Εισαγωγή νέου μαθητή"
Εμφάνισε "2. Υπολογισμός μέσου όρου"
Εμφάνισε "3. Εμφάνιση στατιστικών"
Εμφάνισε "4. Εξαγωγή αναφοράς"
Εμφάνισε "5. Έξοδος από το σύστημα"
Εμφάνισε "=====
Εμφάνισε "Επιλογή (1-5): "

! Εισαγωγή και έλεγχος επιλογής
Διάβασε επιλογή

έγκυρη_επιλογή ← (επιλογή >= 1) και (επιλογή <= 5)

Αν NOT(έγκυρη_επιλογή) τότε

Εμφάνισε "Λάθος επιλογή! Παρακαλώ δώστε αριθμό 1-5."
Εμφάνισε "Πατήστε Enter για συνέχεια..."
Διάβασε ! Αναμονή για Enter

αλλιώς

! Επεξεργασία επιλογής
Αν επιλογή = 1 τότε
Εμφάνισε "--- ΕΙΣΑΓΩΓΗ ΝΕΟΥ ΜΑΘΗΤΗ ---"
πλήθος_μαθητών ← πλήθος_μαθητών + 1
Εμφάνισε "Μαθητής ", πλήθος_μαθητών, " καταχωρήθηκε."

αλλιώς_αν επιλογή = 2 τότε

Εμφάνισε "--- ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΣΟΥ ΟΡΟΥ ---"
Αν πλήθος_μαθητών > 0 τότε
! Εδώ θα μπορούσαμε να έχουμε πραγματικό υπολογισμό
Εμφάνισε "Μέσος όρος: 15.5 (παράδειγμα)"

αλλιώς

Εμφάνισε "Δεν υπάρχουν μαθητές στο σύστημα."
Τέλος_αν

αλλιώς_αν επιλογή = 3 τότε

Εμφάνισε "--- ΣΤΑΤΙΣΤΙΚΑ ---"
Εμφάνισε "Συνολικός αριθμός μαθητών: ", πλήθος_μαθητών
Εμφάνισε "Ποσοστό επιτυχίας: 85% (παράδειγμα)"

αλλιώς_αν επιλογή = 4 τότε
Εμφάνισε "--- ΕΞΑΓΩΓΗ ΑΝΑΦΟΡΑΣ ---"
Εμφάνισε "Η αναφορά δημιουργήθηκε στο αρχείο 'report.txt'"

αλλιώς ! επιλογή = 5
Εμφάνισε "--- ΕΞΟΔΟΣ ---"
Εμφάνισε "Συνολικοί μαθητές: ", πλήθος_μαθητών
Εμφάνισε "Το σύστημα τερματίζεται..."
Τέλος_αν

! Μικρή παύση μετά από κάθε επιλογή (εκτός από έξοδο)
Αν επιλογή ≠ 5 τότε
Εμφάνισε "Πατήστε Enter για συνέχεια..."
Διάβασε ! Αναμονή για Enter
Τέλος_αν
Τέλος_αν

Μέχρις_ότου επιλογή = 5
Εμφάνισε "Αντίο!"
Τέλος Σύστημα_Διαχείρισης_Μαθητών

7. Τμηματικός Προγραμματισμός: Συστηματική Προσέγγιση

Θεωρητική Ανάλυση:

Ο τμηματικός προγραμματισμός σπάει ένα μεγάλο πρόβλημα σε **μικρότερα, διαχειρίσιμα τμήματα**. Κάθε τμήμα εκτελεί μια συγκεκριμένη λειτουργία και μπορεί να επαναχρησιμοποιηθεί.

Πλεονεκτήματα:

1. **Μειωμένη πολυπλοκότητα** - Κάθε τμήμα είναι πιο απλό
2. **Ευκολότερος εντοπισμός σφαλμάτων** - Δοκιμάζουμε κάθε τμήμα ξεχωριστά
3. **Επαναχρησιμοποίηση κώδικα** - Τα ίδια τμήματα χρησιμοποιούνται πολλές φορές
4. **Καλύτερη συνεργασία** - Πολλοί προγραμματιστές μπορούν να δουλέψουν παράλληλα
5. **Ευκολότερη συντήρηση** - Αλλαγές γίνονται σε συγκεκριμένα τμήματα

Λεπτομερές Παράδειγμα 7.1: Σύστημα υπολογισμού μισθοδοσίας

! =====
! ΚΥΡΙΟΣ ΑΛΓΟΡΙΘΜΟΣ
! =====

Αλγόριθμος Κύριος_Μισθοδοσίας

Πραγματικός: ωρομίσθιο, ώρες_εργασίας, καθарός_μισθός, ασφαλιστικές, φόροι,
συνολικό_κόστος

Κάλεσε Εισαγωγή_Δεδομένων(ωρομίσθιο, ώρες_εργασίας)
Κάλεσε Υπολογισμός_Μισθού(ωρομίσθιο, ώρες_εργασίας, καθарός_μισθός)
Κάλεσε Υπολογισμός_Κρατήσεων(καθарός_μισθός, ασφαλιστικές, φόροι)
Κάλεσε Υπολογισμός_Συνολικού_Κόστους(καθарός_μισθός, ασφαλιστικές,
συνολικό_κόστος)

Κάλεσε Εκτύπωση_Αποτελεσμάτων(ωρομίσθιο, ώρες_εργασίας, καθαρός_μισθός,
ασφαλιστικές, φόροι, συνολικό_κόστος)
Τέλος Κύριος_Μισθοδοσίας

! =====
! ΥΠΟΑΛΓΟΡΙΘΜΟΙ
! =====

Αλγόριθμος Εισαγωγή_Δεδομένων(ωρομίσθιο, ώρες_εργασίας)
! Εισαγωγή και έλεγχος δεδομένων
Εμφάνισε "ΣΥΣΤΗΜΑ ΜΙΣΘΟΔΟΣΙΑΣ"
Εμφάνισε "====="

Επανάλαβε
Εμφάνισε "Δώσε το ωρομίσθιο (π.χ. 8.50): "
Διάβασε ωρομίσθιο
Μέχρις_ότου ωρομίσθιο > 0

Επανάλαβε
Εμφάνισε "Δώσε τις ώρες εργασίας (π.χ. 160): "
Διάβασε ώρες_εργασίας
Μέχρις_ότου ώρες_εργασίας > 0 και ώρες_εργασίας <= 300
Τέλος Εισαγωγή_Δεδομένων

Αλγόριθμος Υπολογισμός_Μισθού(ωρομίσθιο, ώρες_εργασίας, καθαρός_μισθός)
Πραγματικός: υπερωρίες, επιπλέον

! Υπολογισμός βασικού μισθού
Αν ώρες_εργασίας <= 40 τότε
καθαρός_μισθός ← ώρες_εργασίας * ωρομίσθιο
αλλιώς
! Υπερωρίες με 1.5 φορές το ωρομίσθιο
υπερωρίες ← ώρες_εργασίας - 40
επιπλέον ← υπερωρίες * (ωρομίσθιο * 1.5)
καθαρός_μισθός ← (40 * ωρομίσθιο) + επιπλέον
Τέλος_αν
Τέλος Υπολογισμός_Μισθού

Αλγόριθμος Υπολογισμός_Κρατήσεων(καθαρός_μισθός, ασφαλιστικές, φόροι)
! Υπολογισμός ασφαλιστικών εισφορών (15%)
ασφαλιστικές ← καθαρός_μισθός * 0.15

! Υπολογισμός φόρου με κλιμάκια
Αν καθαρός_μισθός <= 1000 τότε
φόροι ← καθαρός_μισθός * 0.10
αλλιώς_αν καθαρός_μισθός <= 2000 τότε
φόροι ← (1000 * 0.10) + ((καθαρός_μισθός - 1000) * 0.20)
αλλιώς

φόροι ← $(1000 * 0.10) + (1000 * 0.20) + ((\text{καθαρός_μισθός} - 2000) * 0.30)$
Τέλος_αν
Τέλος Υπολογισμός_Κρατήσεων

Αλγόριθμος Υπολογισμός_Συνολικού_Κόστους(καθαρός_μισθός, ασφαλιστικές,
συνολικό_κόστος)
! Συνολικό κόστος για τον εργοδότη
συνολικό_κόστος ← καθαρός_μισθός + ασφαλιστικές
Τέλος Υπολογισμός_Συνολικού_Κόστους

Αλγόριθμος Εκτύπωση_Αποτελεσμάτων(ωρομίσθιο, ώρες_εργασίας, καθαρός_μισθός,
ασφαλιστικές, φόροι, συνολικό_κόστος)
Πραγματικός: καθαρά_διαθέσιμα

! Εκτύπωση αναλυτικής απόδειξης
Εμφάνισε ""
Εμφάνισε "ΑΠΟΔΕΙΞΗ ΜΙΣΘΟΔΟΣΙΑΣ"
Εμφάνισε "=====
Εμφάνισε "Ωρομίσθιο: ", ωρομίσθιο:8:2, " €"
Εμφάνισε "Ωρες εργασίας: ", ώρες_εργασίας:8
Εμφάνισε "Βασικός μισθός: ", καθαρός_μισθός:8:2, " €"
Εμφάνισε "Ασφαλιστικές εισφορές: ", ασφαλιστικές:8:2, " €"
Εμφάνισε "Φόροι: ", φόροι:8:2, " €"
Εμφάνισε "-----"

καθαρά_διαθέσιμα ← καθαρός_μισθός - ασφαλιστικές - φόροι
Εμφάνισε "Καθαρά διαθέσιμα: ", καθαρά_διαθέσιμα:8:2, " €"
Εμφάνισε "Συνολικό κόστος εργοδότη: ", συνολικό_κόστος:8:2, " €"
Τέλος Εκτύπωση_Αποτελεσμάτων

8. Πρακτικές Ασκήσεις με Αναλυτικές Λύσεις

Άσκηση 1: Υπολογισμός τριγωνομετρικών συναρτήσεων

! Σκοπός: Εμφάνιση τιμών ημιτόνου για γωνίες 0-360 μοιρών

Αλγόριθμος Τριγωνομετρικός_Πίνακας

Πραγματικός: γωνία_μοίρες, γωνία_ακτίνια, ημίτονο, συνημίτονο

Εμφάνισε "Γωνία(°) Ημίτονο Συνημίτονο"

Εμφάνισε "=====
====="

Για γωνία_μοίρες από 0 μέχρι 360 με_βήμα 15

! Μετατροπή από μοίρες σε ακτίνια

γωνία_ακτίνια ← γωνία_μοίρες * $\pi / 180$

! Υπολογισμός τριγωνομετρικών συναρτήσεων

ημίτονο ← $\text{HM}(\text{γωνία_ακτίνια})$

συνημίτονο ← $\text{ΣΥΝ}(\text{γωνία_ακτίνια})$

! Εμφάνιση αποτελεσμάτων με σωστή μορφοποίηση

Εμφάνισε γωνία_μοίρες:8, " ", ημίτονο:8:4, " ", συνημίτονο:8:4
Τέλος_επανάληψης
Τέλος Τριγωνομετρικός_Πίνακας

Άσκηση 2: Ανάλυση αριθμού σε πρώτους παράγοντες

Αλγόριθμος Πρώτοι_Παράγοντες
Ακέραιος: αριθμός, διαιρέτης, πλήθος_παραγόντων
Εμφάνισε "Δώσε έναν ακέραιο αριθμό (>1): "
Διάβασε αριθμός

Αν αριθμός ≤ 1 τότε

Εμφάνισε "Ο αριθμός πρέπει να είναι μεγαλύτερος του 1"
αλλιώς

Εμφάνισε "Οι πρώτοι παράγοντες του ", αριθμός, " είναι:"

πλήθος_παραγόντων $\leftarrow 0$

διαιρέτης $\leftarrow 2$

Όσο αριθμός > 1 επανάλαβε

! Έλεγχος αν ο διαιρέτης διαιρεί τον αριθμό

Αν αριθμός mod διαιρέτης = 0 τότε

πλήθος_παραγόντων \leftarrow πλήθος_παραγόντων + 1

! Εμφάνιση του παράγοντα

Αν πλήθος_παραγόντων = 1 τότε

Εμφάνισε διαιρέτης

αλλιώς

Εμφάνισε " × ", διαιρέτης

Τέλος_αν

! Διαίρεση του αριθμού με τον παράγοντα

αριθμός \leftarrow αριθμός div διαιρέτης

αλλιώς

! Προχωράμε στον επόμενο πιθανό διαιρέτη

διαιρέτης \leftarrow διαιρέτης + 1

Τέλος_αν

Τέλος_επανάληψης

Εμφάνισε ""

Εμφάνισε "Συνολικοί πρώτοι παράγοντες: ", πλήθος_παραγόντων

Τέλος_αν

Τέλος Πρώτοι_Παράγοντες

9. Συμβουλές

Κατανόηση της ιεραρχίας πράξεων:

1. Παρενθέσεις
2. Συναρτήσεις και ύψωση σε δύναμη (^)

3. Πολλαπλασιασμός (*), Διαίρεση (/), DIV, MOD

4. Πρόσθεση (+), Αφαίρεση (-)

1. **Σωστή χρήση τελεστών σύγκρισης:**

- Χρησιμοποιείτε πάντα = για ισότητα, όχι ==
- Προτιμήστε <= και >= αντί για συνδυασμούς

2. **Αποφυγή ατέρμονων βρόχων:**

- Βεβαιωθείτε ότι η συνθήκη τερματισμού θα γίνει κάποτε αληθής
- Μην ξεχάσετε να αυξήσετε τους μετρητές

3. **Χρήση εσοχών (indentation):**

- Κάθε εμφωλευμένη δομή πρέπει να έχει επιπλέον εσοχή
- Αυτό βοηθά στην οπτική κατανόηση της δομής

4. **Δοκιμή με ακραίες περιπτώσεις:**

- Τέστ με μηδενικές τιμές
- Τέστ με πολύ μεγάλους ή μικρούς αριθμούς
- Τέστ με μη έγκυρες εισόδους

10. Σύνοψη Κλειδίων Σημείων

Έννοια	Ορισμός	Παράδειγμα
Ακολουθία	Εκτέλεση εντολών με σειρά	Διάβασε α; $\Sigma \leftarrow \alpha + \beta$; Εμφάνισε Σ
Επιλογή	Απόφαση βάσει συνθήκης	Αν $\alpha > 0$ τότε ... αλλιώς ...
Επανάληψη	Εκτέλεση πολλών φορών	Όσο ... επανάλαβε ...
Μετρητής	Μεταβλητή που μετρά επαναλήψεις	i σε Για i από 1 μέχρι 10
Άθροιστής	Μεταβλητή για άθροισμα	$\Sigma \leftarrow \Sigma + \alpha$
Έγκυρη είσοδος	Επαλήθευση δεδομένων	Μέχρις_ότου $\alpha > 0$ και $\alpha < 100$
Εμφωλευμένος βρόχος	Βρόχος μέσα σε βρόχο	Για i ... Για j ...
Τμηματικός προγρ.	Σπάσιμο σε υποαλγόριθμους	Κάλεσε Υπολογισμός()