

ΕΝΟΤΗΤΕΣ 6.4 (Τεχνικές Σχεδίασης Προγραμμάτων)

Ερωτήσεις Σωστό / Λάθος

1. Η ιεραρχική σχεδίαση διασπά ένα πρόβλημα σε υποπροβλήματα και τα επιλύει, λύνοντας το αρχικό πρόβλημα
2. Ο τμηματικός προγραμματισμός υλοποιεί τη φιλοσοφία της ιεραρχικής σχεδίασης
3. Ο τμηματικός προγραμματισμός εξασφαλίζει ότι δεν υπάρχει λάθος στο πρόγραμμα
4. Χάρη στο δομημένο προγραμματισμό δημιουργούνται προγράμματα απλούστερα, κατανοητά και εύκολα στη διόρθωση
5. Ο δομημένος προγραμματισμός εμπεριέχει τις έννοιες του ιεραρχικού και του τμηματικού προγραμματισμού
6. Ο δομημένος προγραμματισμός στηρίζεται στις δομές ακολουθίας, επιλογής και επανάληψης
7. Η εντολή GOTO που αλλάζει τη ροή ενός προγράμματος είναι απαραίτητη στον προγραμματισμό **(ΕΞΕΤΑΣΕΙΣ 2004)**
8. Οι σύγχρονες γλώσσες προγραμματισμού δεν υποστηρίζουν την GOTO
9. Ο δομημένος προγραμματισμός βοηθά στην ανάπτυξη απλούστερων προγραμμάτων
10. Το μόνο μειονέκτημα του δομημένου προγραμματισμού είναι ότι η διόρθωση των λαθών γίνεται δυσκολότερα
11. Η δημιουργία με γραφικό τρόπο του περιβάλλοντος της εφαρμογής αποτελεί τον δομημένο προγραμματισμό

Ερωτήσεις Ανάπτυξης

1. Ποιος είναι ο σκοπός της «από πάνω προς τα κάτω» σχεδίασης προγραμμάτων; Πώς γίνεται η παράσταση της ιεραρχικής σχεδίασης προγράμματος; (Συνεχής διαίρεση ενός προβλήματος σε υποπροβλήματα ώστε, όταν αυτά επιλυθούν, να έχει επιλυθεί το αρχικό πρόβλημα. Για την απεικόνισή τους χρησιμοποιούνται διαγραμματικές τεχνικές)
2. Τι υλοποιεί ο τμηματικός προγραμματισμός; (Υλοποιεί την ιεραρχική σχεδίαση, ώστε κάθε υποπρόβλημα να αποτελεί μια ξεχωριστή ενότητα. Διευκολύνει τη δημιουργία του προγράμματος, μειώνει τα λάθη, κάνει ευκολότερη την παρακολούθηση, την κατανόηση και τη συντήρηση του προγράμματος)
3. Ποιες είναι οι αρχές του δομημένου προγραμματισμού; (Αρχικός στόχος ήταν η κατάργηση της εντολής GOTO, αλλά κατέληξε να είναι βασική μεθοδολογία προγραμματισμού. Όλα τα προγράμματα μπορούν να γραφούν με τις δομές ακολουθίας, επιλογής και επανάληψης. Επίσης κάθε ενότητα προγράμματος πρέπει να έχει μόνο μία είσοδο και μία έξοδο. Ο δομημένος προγραμματισμός εμπεριέχει τόσο την ιεραρχική σχεδίαση όσο και τον τμηματικό προγραμματισμό)
4. Ποια είναι τα πλεονεκτήματα του δομημένου προγραμματισμού; (Δημιουργία απλούστερων προγραμμάτων, άμεση μεταφορά αλγορίθμων σε πρόγραμμα, ανάλυση προγραμμάτων σε τμήματα, περιορισμός των λαθών, διευκόλυνση στην ανάγνωση και κατανόηση, ευκολότερη διόρθωση και συντήρηση)
5. Σε ποιες στοιχειώδεις δομές στηρίζεται ο δομημένος προγραμματισμός; (Δομή ακολουθίας, δομή επιλογής και δομή επανάληψης. Όλα τα προβλήματα μπορούν να επιλυθούν με τη χρήση αυτών των δομών)
6. Γιατί η εντολή GOTO κρίνεται ακατάλληλη; (Γιατί αλλάζει τη ροή του προγράμματος και δυσκολεύει την παρακολούθησή, την κατανόηση και τη συντήρηση)

Ασκήσεις

1. Δίνεται το παρακάτω τμήμα αλγορίθμου σε φυσική γλώσσα κατά βήματα:
Βήμα 1: Αν $A > 0$ τότε πήγαινε στο Βήμα 5
Βήμα 2: Αν $A = 0$ τότε πήγαινε στο Βήμα 7
Βήμα 3: Τύπωσε «Αρνητικός»
Βήμα 4: Πήγαινε στο Βήμα 8
Βήμα 5: Τύπωσε «Θετικός»
Βήμα 6: Πήγαινε στο Βήμα 8

Βήμα 7: Τύπωσε «Μηδέν»**Βήμα 8: Τύπωσε «Τέλος»**

- α. Να σχεδιάσετε το ισοδύναμο διάγραμμα ροής
β. Να κωδικοποιήσετε το τμήμα αλγορίθμου σε ψευδογλώσσα σύμφωνα με τις αρχές του δομημένου προγραμματισμού

(ΕΞΕΤΑΣΕΙΣ 2007)

2. Δραστηριότητες ΔΤ1 και ΔΣ2 από το 6^ο Κεφάλαιο του Τετραδίου Μαθητή
3. Σε καθεμία από τις παρακάτω περιπτώσεις να μετατρέψετε τον αδόμητο αλγόριθμο σε αλγόριθμο που ακολουθεί τις αρχές του δομημένου προγραμματισμού
Υπόδειξη: Αρχικά να σχηματίσετε το διάγραμμα ροής και στη συνέχεια με τη βοήθεια του, να σχηματίσετε το δομημένο αλγόριθμο

I.Σημείο A: **Διάβασε** x**Αν** $x \bmod 2 = 1$ **τότε** $\mu_1 \leftarrow \mu_1 + 1$ **αλλιώς** $\mu_2 \leftarrow \mu_2 + 1$ **Τέλος_αν****Αν** $x = 99$ **τότε** **Πήγαινε** στο Σημείο B**αλλιώς** **Πήγαινε** στο Σημείο A**Τέλος_αν**Σημείο B: **Εμφάνισε** μ_1, μ_2 **II.** $\Sigma \leftarrow 0$ $x \leftarrow 0$ Σημείο A: **Αν** $x \leq 20$ **τότε** $\Sigma \leftarrow \Sigma + x$ $x \leftarrow x + 1$ **Πήγαινε** στο Σημείο A**αλλιώς** **Εμφάνισε** Σ **Τέλος_αν****III.**

Εντολή 1

Σημείο A:

Αν Συνθήκη 1 **τότε**

Εντολή 2

Τέλος_αν**Αν** Συνθήκη 2 **τότε** **Πήγαινε** στο Σημείο A**Τέλος_αν**

Εντολή 3