

Γ' Λυκείου

# Πληροφορική

## Περιεχόμενα

### Κεφάλαιο 13: Εκσφαλμάτωση Προγράμματος (Βιβλίο I)

- 13.1 Κατηγορίες λαθών

### Κεφάλαιο 5 : Εκσφαλμάτωση Προγράμματος (Βιβλίο II)

- 5.1 Κατηγορίες λαθών
  - 5.1.1 Συντακτικά λάθη
  - 5.1.2 Λάθη που οδηγούν σε αντικανονικό τερματισμό του προγράμματος
  - 5.1.3 Λογικά λάθη.
- 5.2 Εκσφαλμάτωση
  - 5.2.1 Εκσφαλμάτωση λογικών λαθών στις δομές επιλογής
  - 5.2.2 Εκσφαλμάτωση λογικών λαθών στις δομές επανάληψης

### Κεφάλαιο 2: Τεχνικές Σχεδίασης Αλγορίθμων (Βιβλίο II)

- 2.1 Μέθοδος Διαίρει και Βασίλευε

Ένας προγραμματιστής, ανεξάρτητα από πόσο ικανός είναι, όταν δημιουργεί ένα πρόγραμμα, είναι φυσικό να κάνει κάποιο λάθος. Ειδικά σε μεγάλες εφαρμογές που κατασκευάζει πολύπλοκα υπολογιστικά υποπρογράμματα ή χρησιμοποιεί αρκετές συσκευές υλικού, είναι αδύνατο να αποφύγει τέτοιες ανεπιθύμητες καταστάσεις.

Μπορούμε να διακρίνουμε τις εξής κατηγορίες λαθών:

1. Συντακτικά λάθη (λάθη κατά την υλοποίηση του προγράμματος)
2. Λάθη που οδηγούν σε αντικανονικό τερματισμό του προγράμματος (λάθη κατά την εκτέλεση του προγράμματος)
3. Λογικά λάθη που παράγουν λανθασμένα αποτελέσματα

## 1. Συντακτικά λάθη (λάθη κατά την υλοποίηση)

### Οφείλονται:

- σε αναγραμματισμούς ονομάτων εντολών  
(π.χ. δεν γράψαμε σωστά μια δεσμευμένη λέξη)
- λάθος σύνταξης εντολών  
(π.χ. δεν τερματίσαμε μια δομή επιλογής ή μια δομή επανάληψης)
- μη δήλωση δεδομένων  
(π.χ. δεν δηλώσαμε μια μεταβλητή που χρησιμοποιούμε στο πρόγραμμά μας).

### Εμφανίζονται:

κατά τη διαδικασία της μεταγλώττισης ή διερμηνεύσης. Ο μεταγλωττιστής ή ο διερμηνευτής παρουσιάζει προειδοποιητικά μηνύματα, με τη βοήθεια των οποίων εύκολα και γρήγορα εντοπίζουμε και διορθώνουμε τα λάθη.

Τα σύγχρονα προγραμματιστικά περιβάλλοντα μας προφυλάσσουν αυτόματα από τα λάθη κατά την υλοποίηση, αφού παρέχουν εργαλεία αυτόματου ελέγχου σύνταξης των εντολών και παρακολουθούν τον προγραμματιστή κατά τη συγγραφή του προγράμματος.

### Αντιμετώπιση:

Είναι τα πλέον εύκολα στην αντιμετώπιση τους. Ο προγραμματιστής επιστρέφει στο πηγαίο πρόγραμμα, διορθώνει τα λάθη και ξαναυποβάλλει το πηγαίο για μεταγλώττιση.

### Παραδείγματα:

- αντί για το σωστό ΔΙΑΒΑΣΕ X να γράψουμε ΔΙΑΒΣΕ X
- Να ξεχάσουμε το ΤΕΛΟΣ\_ΑΝ σε μια δομή επιλογής
- Να μην γράψουμε το ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ σε μια ΟΣΟ ή ΜΙΑ ΓΙΑ
- Να χρησιμοποιούμε την μεταβλητή ΠΛΗΘΟΣ χωρίς να την δηλώσουμε στην παράγραφο ΜΕΤΑΒΛΗΤΕΣ.

## 2. Λάθη που οδηγούν σε αντικανονικό τερματισμό του προγράμματος (λάθη κατά την εκτέλεση του προγράμματος)

Η πρόληψη τέτοιων λαθών είναι αρκετά δύσκολη, αφού συνήθως οφείλονται σε καταστάσεις που δεν είναι εύκολο να ελεγχθούν από τον προγραμματιστή, ενώ πολλές φορές εμφανίζονται μετά από μεγάλο χρονικό διάστημα. Τα λάθη αυτά συνήθως οδηγούν σε αντικανονικό τερματισμό του προγράμματος.

### Οφείλονται:

- στην κλήση μιας διαδικασίας με δεδομένα που δεν μπορεί να χειριστεί όπως
  - αναζήτηση διαγραμμένων αρχείων
  - η προσπάθεια διαίρεσης ενός αριθμού με το μηδέν
  - προσπάθεια υπολογισμού τετραγωνικής ρίζας αρνητικού αριθμού
  - η υπερχείλιση μιας αριθμητικής μεταβλητής
- από δυσλειτουργία του υλικού μέρους του υπολογιστή, όπως
  - η καταστροφή του σκληρού δίσκου του συστήματος,
  - ο τερματισμός μιας σύνδεσης δικτύου και η
  - αποσύνδεση του εκτυπωτή.

### Εμφανίζονται:

Κατά την εκτέλεση του προγράμματος (συνεπώς δεν ανιχνεύονται κατά την μεταγλώττιση του) Τα λάθη που προκαλούνται κατά το χρόνο εκτέλεσης του προγράμματος είναι πιο επώδυνα γιατί συνήθως εμφανίζονται σε πραγματικό περιβάλλον εκτέλεσης και τις περισσότερες φορές προκαλούν τον αντικανονικό τερματισμό της εφαρμογής και το κρέμασμα (crash) του συστήματος.

### Αντιμετώπιση:

Όταν ένα λάθος προκληθεί κατά την εκτέλεση της εφαρμογής, είναι δυνατό να αντιμετωπισθεί μόνο με τη χρήση εντολών προγράμματος που το παγιδεύουν και εκτελούν τις κατάλληλες διαδικασίες χειρισμού του.

## 3. Λογικά λάθη

### Οφείλονται:

σε σφάλματα στη λογική επίλυσης του προβλήματος ή σε λανθασμένη διατύπωση του αλγόριθμου, είναι δηλαδή συνήθως λάθη σχεδιασμού.

Τα λογικά λάθη δεν προκαλούν την διακοπή της εκτέλεσης του προγράμματος.

### Εμφανίζονται:

κατά την εκτέλεση του προγράμματος. Τα αντιλαμβανόμαστε αφού ολοκληρωθεί η εκτέλεση του προγράμματος, από τα λαθεμένα αποτελέσματα που παράγει το πρόγραμμα

Τα λογικά λάθη δεν ανιχνεύονται στο στάδιο της μεταγλώττισης του προγράμματος.

### Αντιμετώπιση:

είναι τα πλέον σοβαρά και δύσκολα στη ανίχνευση τους. Διαπιστώνονται με τη διαδικασία ελέγχου και την ανάλυση των αποτελεσμάτων του προγράμματος

### Παράδειγμα:

- αντί του σωστού  $A \leftarrow B + \Gamma$  γράφω  $A \leftarrow B * \Gamma$
- Για τον υπολογισμό του μέσου όρου 3 αριθμών πρέπει να γράψω  $MO \leftarrow (A+B+\Gamma)/3$ . Αν ξεχάσω τις παρενθέσεις και γράψω  $MO \leftarrow A+B+\Gamma/3$  τότε θα υπολογιστεί λάθος ο μέσος όρος (λογικό λάθος).

## 5.2 Εκσφαλμάτωση προγράμματος

Η διαδικασία ελέγχου, εντοπισμού και διόρθωσης των σφαλμάτων ενός προγράμματος καλείται **εκσφαλμάτωση** (debugging). Στόχος της διαδικασίας εκσφαλμάτωσης είναι ο εντοπισμός των σημείων του προγράμματος που προκαλούν προβλήματα στη λειτουργία του.

Για τον εντοπισμό ενός λάθους δεν υπάρχουν ιδιαίτερα μυστικά και τρικ. Η εκσφαλμάτωση είναι ένα πρόβλημα λογικής και όσο πιο καλά αντιλαμβάνεται ο προγραμματιστής τον τρόπο που εργάζεται το πρόγραμμα, τόσο πιο εύκολα και σύντομα θα εντοπίσει λάθη που προκαλούν δυσλειτουργίες.

Για τα **συντακτικά λάθη** που παρουσιάζονται κατά το χρόνο σχεδιασμού, δεν χρειάζεται ιδιαίτερη μνεία. Όπως αναφέρθηκε, τις περισσότερες φορές το περιβάλλον προγραμματισμού τα ανιχνεύει αυτόματα και προτείνει τη διόρθωσή τους. Ακόμη και αν το περιβάλλον δεν προτείνει τη διόρθωση, ο μεταγλωττιστής (και ο διερμηνευτής) συλλαμβάνει και περιγράφει το λάθος και στη συνέχεια ο προγραμματιστής μπορεί πολύ εύκολα να το διορθώσει.

Τα λάθη που κυρίως μας απασχολούν στη φάση της εκσφαλμάτωσης είναι τα **λογικά λάθη** και τα **λάθη που παρουσιάζονται κατά το χρόνο εκτέλεσης του προγράμματος**.

### 5.2.1 Εκσφαλμάτωση λογικών λαθών στις δομές επιλογής

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

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

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.

### 5.2.2 Εκσφαλμάτωση λογικών 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.

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

- 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

**ΑΡΑ:** Μελετώντας τα παραδείγματα της παραγράφου (Βιβλίο II σελ. 120), παρατηρούμε ότι η ανίχνευση των λογικών λαθών γίνεται με την εξής μέθοδο: δίνουμε στην είσοδο κάποιες συγκριμένες τιμές και συγκρίνουμε το αποτέλεσμα του προγράμματος με το αναμενόμενο αποτέλεσμα

## 2.1 Μέθοδος 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).

Χαρακτηριστικό παράδειγμα της μεθόδου «διαίρει και βασίλευε» είναι η δυαδική αναζήτηση.

Τα παραδείγματα του βιβλίου (Βιβλίο II σελ. 68)

Κατηφόρης Παναγιώτης