ΕΝΟΤΗΤΑ

2

ΠΛΗΡΟΦΟΡΙΚΗ Γ΄ ΛΥΚΕΙΟΥ · ΣΗΜΕΙΩΣΕΙΣ ΜΑΘΗΤΗ

Πρόγραμμα – Τεχνικές Σχεδίασης – Γλώσσες – Προγραμματιστικά Περιβάλλοντα

Παράγραφοι σχολικού βιβλίου: **6.1** · **6.4** · **6.4.1** · **6.4.2** · **6.4.3** · **6.3** · **6.7**

1 Από το πρόβλημα στο πρόγραμμα § 6.1

Για την επίλυση ενός προβλήματος με υπολογιστή διακρίνουμε τρία βασικά στάδια:



Το τρίτο στάδιο αφορά τον προγραμματισμό, δηλαδή τη δημιουργία του προγράμματος.

> ◆ **ΟΡΙΣΜΟΣ – ΠΡΟΓΡΑΜΜΑ**
>
> Πρόγραμμα είναι το **σύνολο των εντολών** που πρέπει να δοθούν στον υπολογιστή, ώστε να υλοποιηθεί ο αλγόριθμος για την επίλυση ενός προβλήματος.

Το πρόγραμμα συνδέεται επίσης με τα δεδομένα και τις δομές δεδομένων πάνω στις οποίες ενεργεί ο αλγόριθμος.

> !! **SOS**
>
> Αλγόριθμος ≠ Πρόγραμμα
>
> Ο αλγόριθμος περιγράφει τη διαδικασία επίλυσης. Το πρόγραμμα διατυπώνει τη λύση σε μορφή που μπορεί να υλοποιηθεί από τον υπολογιστή.

2 Τεχνικές σχεδίασης προγραμμάτων § 6.4

Η ανάπτυξη σύνθετων προγραμμάτων απαιτεί μεθοδικό σχεδιασμό. Στην ενότητα αυτή εξετάζουμε τρεις βασικές έννοιες:



3 Ιεραρχική σχεδίαση § 6.4.1

Η ιεραρχική σχεδίαση ή σχεδίαση από επάνω προς τα κάτω (top-down) βασίζεται στη συνεχή διαίρεση ενός προβλήματος σε μικρότερα και απλούστερα υποπροβλήματα, μέχρι να φτάσουμε σε επίπεδο όπου η υλοποίησή τους είναι απλή.

> **? ΣΚΕΨΟΥ ΤΟ ΕΤΣΙ – Οργάνωση σχολικής εκδρομής**
>
> Μεγάλο πρόβλημα: «Οργάνωση σχολικής εκδρομής».
>
> Το χωρίζουμε σε: Μετακίνηση – Διαμονή – Επισκέψεις – Φαγητό.
>
> Στη συνέχεια, π.χ. η Μετακίνηση αναλύεται σε: προσφορές – επιλογή λεωφορείου – κράτηση.
>
> Ξεκινάμε από το γενικό και προχωρούμε σταδιακά στο ειδικό.

> **!! SOS**
>
> Ιεραρχική σχεδίαση = μεθοδολογία σχεδίασης. Απαντά κυρίως στο ερώτημα: «Πώς θα αναλύσω το μεγάλο πρόβλημα σε απλούστερα υποπροβλήματα;»

4 Τμηματικός προγραμματισμός § 6.4.2

Η ιεραρχική σχεδίαση προγράμματος υλοποιείται με τον τμηματικό προγραμματισμό.

Μετά την ανάλυση του προβλήματος σε υποπροβλήματα, κάθε υποπρόβλημα αποτελεί ανεξάρτητη ενότητα (module), η οποία γράφεται ξεχωριστά από τα υπόλοιπα τμήματα του προγράμματος.

? ΣΚΕΨΟΥ ΤΟ ΕΤΣΙ – Η ομάδα της εκδρομής

Αφού χωρίσουμε την οργάνωση της εκδρομής σε επιμέρους εργασίες, μπορούμε να αναθέσουμε μία ομάδα στη μετακίνηση, μία στη διαμονή και μία στις επισκέψεις.

Κάθε ομάδα αναλαμβάνει ένα συγκεκριμένο τμήμα του συνολικού έργου.

⚠ ΠΡΟΣΟΧΗ

ΠΩΣ ΣΥΝΔΕΟΝΤΑΙ;

Ιεραρχική σχεδίαση: αναλύω το πρόβλημα σε υποπροβλήματα.

Τμηματικός προγραμματισμός: υλοποιώ την παραπάνω ανάλυση, γράφοντας κάθε υποπρόβλημα ως ανεξάρτητη ενότητα (module).

Άρα: η ιεραρχική σχεδίαση υλοποιείται με τον τμηματικό προγραμματισμό.

5 Δομημένος προγραμματισμός § 6.4.3

Ο δομημένος προγραμματισμός είναι μεθοδολογία σύνταξης προγραμμάτων που διευκολύνει την ανάπτυξη σύνθετων προγραμμάτων και τη διαχείρισή τους.

Βασίζεται στη χρήση τριών και μόνο βασικών λογικών δομών:

Ακολουθία → Επιλογή → Επανάληψη

Κάθε πρόγραμμα, όπως και κάθε ενότητά του, έχει μόνο μία είσοδο και μόνο μία έξοδο.

Με τις τρεις αυτές δομές μπορεί να περιγραφεί η λογική οποιουδήποτε αλγορίθμου χωρίς να είναι απαραίτητη η ανεξέλεγκτη μεταφορά της ροής εκτέλεσης.

Η εντολή GOTO

Η GOTO μεταφέρει τη ροή εκτέλεσης σε άλλο σημείο του προγράμματος. Η αλόγιστη χρήση της δημιουργεί πολύπλοκη ροή και δυσκολεύει την κατανόηση, τον έλεγχο και τη συντήρηση του προγράμματος.

!! SOS – Πλεονεκτήματα δομημένου προγραμματισμού

1. Δημιουργία απλούστερων προγραμμάτων.
2. Άμεση μεταφορά των αλγορίθμων σε προγράμματα.
3. Διευκόλυνση ανάλυσης του προγράμματος σε τμήματα.
4. Περιορισμός των λαθών κατά την ανάπτυξη.
5. Διευκόλυνση ανάγνωσης και κατανόησης από τρίτους.
6. Ευκολότερη διόρθωση και συντήρηση.

★ ΠΑΝΕΛΛΑΔΙΚΕΣ – 2016 – Θέμα Α2 (παλαιό σύστημα)

Ζητήθηκαν επιγραμματικά πέντε πλεονεκτήματα του δομημένου προγραμματισμού. Άρα η συγκεκριμένη λίστα είναι θεωρία που χρειάζεται.

6 Φυσικές και τεχνητές γλώσσες § 6.3

Κάθε γλώσσα χαρακτηρίζεται από βασικά στοιχεία που καθορίζουν ποια σύμβολα και λέξεις χρησιμοποιεί και με ποιους κανόνες σχηματίζονται έγκυρες προτάσεις.

Έννοια	Τι περιγράφει
Αλφάβητο	Το σύνολο των στοιχείων που χρησιμοποιούνται από τη γλώσσα.
Λεξιλόγιο	Το σύνολο των αποδεκτών λέξεων της γλώσσας.
Γραμματική	Τους κανόνες σχηματισμού και σύνδεσης των λέξεων.
Σημασιολογία	Το νόημα των λέξεων, εκφράσεων και προτάσεων.

Στη γραμματική διακρίνουμε το τυπικό και το συντακτικό.

- Τυπικό: καθορίζει τις μορφές με τις οποίες μία λέξη είναι αποδεκτή.
- Συντακτικό: καθορίζει τους κανόνες διάταξης και σύνδεσης των λέξεων για τη δημιουργία προτάσεων.

Φυσικές vs τεχνητές γλώσσες

Φυσικές γλώσσες	Τεχνητές γλώσσες
Χρησιμοποιούνται για επικοινωνία μεταξύ ανθρώπων.	Δημιουργούνται συνειδητά για συγκεκριμένο σκοπό.
Εξελίσσονται συνεχώς.	Χαρακτηρίζονται από μεγαλύτερη στασιμότητα.
π.χ. Ελληνικά, Αγγλικά	π.χ. γλώσσες προγραμματισμού

⚠ **ΠΑΓΙΔΑ**

Οι γλώσσες υψηλού επιπέδου ΔΕΝ είναι φυσικές γλώσσες. Οι γλώσσες προγραμματισμού, χαμηλού ή υψηλού επιπέδου, είναι τεχνητές γλώσσες.

Γλώσσα μηχανής

Είναι η γλώσσα που γίνεται άμεσα κατανοητή από τον υπολογιστή και αποτελείται από ακολουθίες δυαδικών ψηφίων (0 και 1).

Γλώσσες υψηλού επιπέδου

Είναι γλώσσες προγραμματισμού περισσότερο προσιτές στον άνθρωπο, οι οποίες πρέπει να μεταφραστούν σε γλώσσα μηχανής για να εκτελεστούν από τον υπολογιστή.

7 Προγραμματιστικά περιβάλλοντα § 6.7

Για τη δημιουργία ενός προγράμματος απαιτούνται εργαλεία που επιτρέπουν τη σύνταξη, τη μετάφραση, τη σύνδεση και τελικά την εκτέλεσή του.

Για τη δημιουργία ενός προγράμματος απαιτούνται τουλάχιστον τρία προγράμματα: ο συντάκτης, ο μεταγλωττιστής και ο συνδέτης.

7.1 Συντάκτης

Ο συντάκτης (editor) είναι το πρόγραμμα που χρησιμοποιείται για τη σύνταξη και τη διόρθωση του πηγαίου προγράμματος.

✎ **ΠΑΡΑΔΕΙΓΜΑ**

Όπως χρησιμοποιούμε έναν επεξεργαστή κειμένου για να γράψουμε και να διορθώσουμε ένα κείμενο, ο προγραμματιστής χρησιμοποιεί τον συντάκτη για να γράψει και να διορθώσει τον κώδικα.

7.2 Πηγαίο και αντικείμενο πρόγραμμα

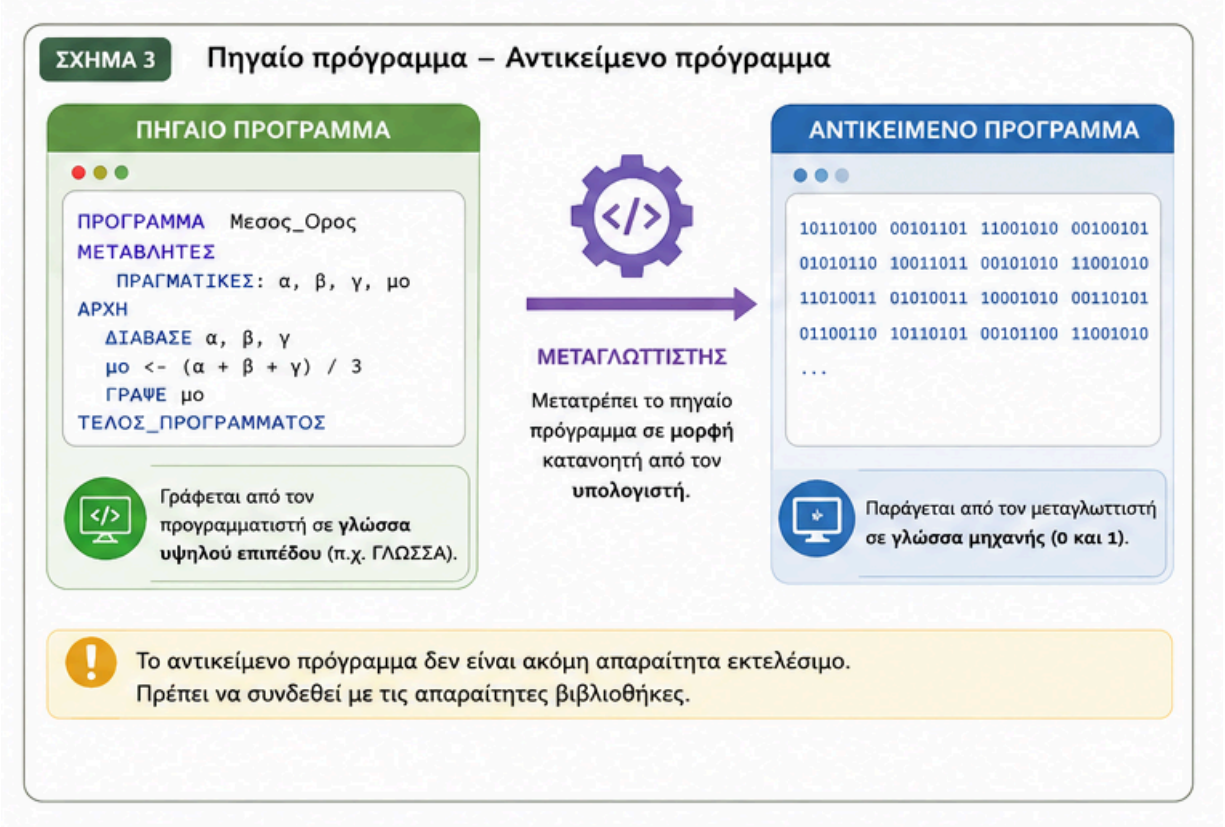
◆ **ΟΡΙΣΜΟΙ**

Πηγαίο πρόγραμμα: το αρχικό πρόγραμμα, όπως γράφεται από τον προγραμματιστή σε γλώσσα υψηλού επιπέδου.

Αντικείμενο πρόγραμμα: το ισοδύναμο πρόγραμμα σε γλώσσα μηχανής που παράγεται από τον μεταγλωττιστή.

⚠ **ΠΡΟΣΟΧΗ**

Το αντικείμενο πρόγραμμα είναι σε μορφή κατανοητή από τον υπολογιστή, αλλά συνήθως δεν είναι ακόμη έτοιμο για εκτέλεση.



★ **ΠΑΝΕΛΛΑΔΙΚΕΣ – 2021 – Θέμα Α2α**

Ζητήθηκε τι ονομάζεται αντικείμενο πρόγραμμα. Η διάκριση πηγαίου – αντικείμενου προγράμματος είναι επομένως SOS.

8 Μεταγλωττιστής και διερμηνευτής

8.1 Μεταγλωττιστής

Ο μεταγλωττιστής (compiler) δέχεται στην είσοδό του ένα πρόγραμμα γραμμένο σε γλώσσα υψηλού επιπέδου και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής.

? ΣΚΕΨΟΥ ΤΟ ΕΤΣΙ – Μετάφραση ολόκληρου βιβλίου

Ένας μεταφραστής παίρνει ολόκληρο ένα ελληνικό βιβλίο και δημιουργεί την αγγλική έκδοσή του. Μετά ο αναγνώστης μπορεί να χρησιμοποιεί τη μεταφρασμένη έκδοση χωρίς να γίνεται νέα μετάφραση κάθε φορά.

Αντίστοιχα, ο μεταγλωττιστής μεταφράζει το πρόγραμμα ως σύνολο και παράγει αντικείμενο πρόγραμμα.

8.2 Διερμηνευτής

Ο διερμηνευτής (interpreter) διαβάζει μία προς μία τις εντολές του αρχικού προγράμματος και για καθεμία εκτελεί αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής.

? ΣΚΕΨΟΥ ΤΟ ΕΤΣΙ – Διερμηνέας σε συζήτηση

Δύο άνθρωποι μιλούν διαφορετικές γλώσσες. Ο πρώτος λέει μία πρόταση, ο διερμηνέας τη μεταφράζει αμέσως και μετά περνούν στην επόμενη.

Έτσι λειτουργεί και ο διερμηνευτής: εντολή → μετάφραση/εκτέλεση → επόμενη εντολή.

	Μεταγλωττιστής	Διερμηνευτής
Τρόπος	Μεταφράζει το πρόγραμμα ως σύνολο.	Διαβάζει και εκτελεί μία εντολή κάθε φορά.
Παραγόμενο πρόγραμμα	Παράγει αντικείμενο πρόγραμμα.	Δεν παράγει αντίστοιχο αντικείμενο πρόγραμμα.
Εκτέλεση	Το μεταφρασμένο πρόγραμμα εκτελείται συνήθως ταχύτερα.	Η εκτέλεση είναι συνήθως πιο αργή.
Εντοπισμός λαθών	Τα συντακτικά λάθη εμφανίζονται κατά τη μεταγλώττιση.	Η διαδικασία σταματά όταν συναντηθεί λάθος.

⚠ ΠΡΟΣΟΧΗ

Διερμηνευτής – πλεονέκτημα: επιτρέπει άμεση εκτέλεση και άμεση διόρθωση των εντολών, γι' αυτό είναι ιδιαίτερα χρήσιμος κατά την ανάπτυξη ενός προγράμματος.

Μεταγλωττιστής – μειονέκτημα: πριν χρησιμοποιηθεί το πρόγραμμα πρέπει να ολοκληρωθεί η διαδικασία μεταγλώττισης και σύνδεσης.

Σύγχρονα προγραμματιστικά περιβάλλοντα: μπορούν να χρησιμοποιούν μεικτές υλοποιήσεις, αξιοποιώντας διερμηνευτή κατά την ανάπτυξη και μεταγλωττιστή για την τελική έκδοση.

✱ ΜΝΗΜΟΝΙΚΟ

Μεταγλωττιστής = «μεταφράζω όλο το έργο».

Διερμηνευτής = «μεταφράζω και εκτελώ βήμα-βήμα».

9 Βιβλιοθήκες και συνδέτης

9.1 Βιβλιοθήκες

Οι βιβλιοθήκες περιέχουν έτοιμα υποπρογράμματα που μπορούν να χρησιμοποιηθούν από άλλα προγράμματα.

? ΣΚΕΨΟΥ ΤΟ ΕΤΣΙ – Εργαλειοθήκη

Όταν συναρμολογείς ένα έπιπλο δεν κατασκευάζεις πρώτα το κατσαβίδι και το σφυρί. Χρησιμοποιείς έτοιμα εργαλεία.

Παρόμοια, ο προγραμματιστής μπορεί να χρησιμοποιεί έτοιμες λειτουργίες από βιβλιοθήκες αντί να τις δημιουργεί κάθε φορά από την αρχή.

✎ ΠΡΑΓΜΑΤΙΚΟ ΠΑΡΑΔΕΙΓΜΑ

Για έναν μαθηματικό υπολογισμό, όπως μια τετραγωνική ρίζα, μια γλώσσα μπορεί να διαθέτει έτοιμη μαθηματική λειτουργία σε βιβλιοθήκη. Ο προγραμματιστής την αξιοποιεί χωρίς να υλοποιήσει ο ίδιος από την αρχή τον αντίστοιχο αλγόριθμο.

9.2 Συνδέτης – φορτωτής

Το αντικείμενο πρόγραμμα συνήθως χρειάζεται να συνδεθεί με άλλα τμήματα και με τις απαραίτητες βιβλιοθήκες. Τη σύνδεση αυτή αναλαμβάνει ο συνδέτης-φορτωτής (linker-loader), ώστε να δημιουργηθεί το εκτελέσιμο πρόγραμμα.

? ΣΚΕΨΟΥ ΤΟ ΕΤΣΙ – Τελική συναρμολόγηση

Μπορεί να έχουμε όλα τα κομμάτια ενός επίπλου και τα απαραίτητα εξαρτήματα, αλλά ακόμη δεν έχουμε έτοιμο έπιπλο. Πρέπει να συναρμολογηθούν σωστά.

Αντίστοιχα, ο συνδέτης συνδυάζει το αντικείμενο πρόγραμμα με τα απαραίτητα τμήματα/βιβλιοθήκες για να προκύψει το εκτελέσιμο πρόγραμμα.

10 Από το πηγαίο στο εκτελέσιμο πρόγραμμα



!! SOS

Αντικείμενο πρόγραμμα ≠ Εκτελέσιμο πρόγραμμα.

Το αντικείμενο πρόγραμμα μπορεί να χρειάζεται σύνδεση με βιβλιοθήκες πριν δημιουργηθεί το εκτελέσιμο.

Η δημιουργία του εκτελέσιμου προγράμματος γίνεται μόνο στην περίπτωση που το αρχικό πρόγραμμα δεν περιέχει συντακτικά λάθη. Τις περισσότερες φορές κάθε πρόγραμμα αρχικά θα έχει λάθη.

11 Λάθη σε ένα πρόγραμμα

	Συντακτικό λάθος	Λογικό λάθος
Τι συμβαίνει	Παραβιάζονται οι κανόνες σύνταξης της γλώσσας.	Η λογική/ο αλγόριθμος δεν οδηγεί στο σωστό αποτέλεσμα.
Εντοπισμός	Εντοπίζεται κατά τη μετάφραση.	Τα λογικά λάθη εμφανίζονται μόνο κατά την εκτέλεση του προγράμματος.
Αποτέλεσμα	Εμποδίζει τη σωστή μετάφραση, άρα δεν δημιουργείται το εκτελέσιμο πρόγραμμα.	Το πρόγραμμα μπορεί να μεταφραστεί και να εκτελεστεί κανονικά, αλλά να δίνει λανθασμένα αποτελέσματα.

✎ **ΠΑΡΑΔΕΙΓΜΑ – Συντακτικό**

Η πρόταση «Θέλω πάω σχολείο» χρησιμοποιεί γνωστές λέξεις, αλλά η σύνταξή της είναι λανθασμένη. Αντίστοιχα, μια εντολή μπορεί να χρησιμοποιεί σωστές λέξεις μιας γλώσσας προγραμματισμού με λάθος σύνταξη.

✎ **ΠΑΡΑΔΕΙΓΜΑ – Λογικό**

Θέλεις να πας από Θεσσαλονίκη στην Αθήνα, αλλά ακολουθείς άψογα οδηγίες που οδηγούν στην Καβάλα. Οι οδηγίες εκτελούνται κανονικά, όμως λύνουν λάθος πρόβλημα. Έτσι και ένα πρόγραμμα μπορεί να εκτελείται χωρίς συντακτικό σφάλμα αλλά να δίνει λανθασμένο αποτέλεσμα.

12 Πανελλαδικές – χαρακτηριστικές παγίδες

Οι παρακάτω τύποι ερωτήσεων έχουν εμφανιστεί σε προηγούμενες εξετάσεις και δείχνουν τι πρέπει να διακρίνεις με ακρίβεια.

Σ/Λ: Ο μεταγλωττιστής δέχεται πρόγραμμα υψηλού επιπέδου και παράγει ισοδύναμο πρόγραμμα σε γλώσσα μηχανής. **ΣΩΣΤΟ**

Σ/Λ: Το πηγαίο πρόγραμμα εκτελείται απευθείας από τον υπολογιστή χωρίς μετάφραση. **ΛΑΘΟΣ**

Σ/Λ: Ο διερμηνευτής διαβάζει μία προς μία τις εντολές και εκτελεί αμέσως την αντίστοιχη ακολουθία εντολών μηχανής. **ΣΩΣΤΟ**

Σ/Λ: Η χρήση διερμηνευτή οδηγεί σε ταχύτερη εκτέλεση από το αντίστοιχο μεταγλωττισμένο εκτελέσιμο πρόγραμμα. **ΛΑΘΟΣ**

Σ/Λ: Τα συντακτικά λάθη εντοπίζονται στη φάση της μεταγλώττισης. **ΣΩΣΤΟ**

Σ/Λ: Τα λογικά λάθη εμφανίζονται υποχρεωτικά στο στάδιο της μεταγλώττισης. **ΛΑΘΟΣ**

13 Τι πρέπει να γνωρίζω πολύ καλά – SOS

- τη σχέση Πρόβλημα → Αλγόριθμος → Πρόγραμμα
- τη διαφορά ιεραρχικής σχεδίασης και τμηματικού προγραμματισμού
- τις τρεις βασικές δομές: ακολουθία – επιλογή – επανάληψη
- τα πλεονεκτήματα του δομημένου προγραμματισμού και γιατί αποφεύγεται η GOTO
- αλφάβητο – λεξιλόγιο – γραμματική – σημασιολογία
- τη διάκριση φυσικών και τεχνητών γλωσσών
- τη διαφορά μεταγλωττιστή και διερμηνευτή
- τη διάκριση πηγαίου – αντικείμενου – εκτελέσιμου προγράμματος
- τον ρόλο βιβλιοθηκών και συνδέτη-φορτωτή
- τη διαδρομή από το πηγαίο στο εκτελέσιμο πρόγραμμα
- τη διαφορά συντακτικών και λογικών λαθών

14 Θέματα Πανελλαδικών – Ενότητα 2

Οι παρακάτω ερωτήσεις έχουν επιλεγεί επειδή αφορούν έννοιες που παραμένουν στη σημερινή ύλη της ενότητας. Οι διατυπώσεις δίνονται συνοπτικά για μελέτη και σύνδεση με τη θεωρία.

Έτος – Θέμα	Τι εξετάστηκε	Σύνδεση με σημειώσεις
2002 – Επαναληπτικές – Θέμα 1.Α4, 1.Β	Λογικά λάθη και Διαφορές μεταξύ μεταγλωττιστή (compiler) και διερμηνευτή (interpreter).	§ 11, 8
2004 – Ημερήσια – Θέμα 1Α	Σ/Λ για μεταγλωττιστή, διερμηνευτή, πηγαίο πρόγραμμα, γλώσσα μηχανής	§§ 7,8
2004 – Επαναληπτικές – Θέμα Γ3-Γ4	Σ/Λ για την GOTO στον δομημένο προγραμματισμό και για τον εντοπισμό συντακτικών λαθών κατά τη μεταγλώττιση.	§§ 5, 11
2007 – Ημερήσια – Θέμα 1Δ	Συμπλήρωση εννοιών συντάκτης – πηγαίο – μεταγλωττιστής – αντικείμενο – συνδέτης/φορτωτής – εκτελέσιμο και σωστή χρονική σειρά των σταδίων.	§§ 7-10
2014 Επαναληπτικές – Θέμα Α1	Σ/Λ σχετικά με τη διάκριση φυσικών και τεχνητών γλωσσών και τις γλώσσες προγραμματισμού.	§ 6
2014 – Θέμα Α1	Σ/Λ: δυνατότητα δημιουργίας εκτελέσιμου προγράμματος όταν το αρχικό πρόγραμμα περιέχει λογικά, αλλά όχι συντακτικά λάθη.	§§ 10, 11
2016 – Θέμα Α2 (Παλαιό σύστημα)	Επιγραμματική αναφορά πέντε πλεονεκτημάτων του δομημένου προγραμματισμού.	§ 5
2021 – Θέμα Α2α	Έννοια του αντικείμενου προγράμματος και σχέση του με τη διαδικασία μετάφρασης.	§§ 7.2, 8