

A. Φυσικές και τεχνητές γλώσσες

Οι γλώσσες προγραμματισμού είναι **τεχνητές** γλώσσες

Μία γλώσσα προσδιορίζεται από το **αλφάβητό της, το λεξιλόγιό της, τη γραμματική της και τέλος τη σημασιολογία της.**

Αλφάβητο μίας γλώσσας καλείται το σύνολο των στοιχείων που χρησιμοποιείται από τη γλώσσα.

Το **λεξιλόγιο** αποτελείται από ένα υποσύνολο όλων των ακολουθιών που δημιουργούνται από τα στοιχεία του αλφαβήτου, τις λέξεις που είναι δεκτές από τη γλώσσα

Η **Γραμματική** αποτελείται από το τυπικό ή τυπολογικό (accidence) και το συντακτικό (syntax).

Τυπικό είναι το σύνολο των κανόνων που ορίζει τις μορφές με τις οποίες μία λέξη είναι **αποδεκτή**. Για **παράδειγμα**, στην ελληνική γλώσσα οι λέξεις γλώσσα, γλώσσας, γλώσσες είναι δεκτές, ενώ η λέξη γλώσσας δεν είναι αποδεκτή.

Συντακτικό είναι το σύνολο των κανόνων που καθορίζει τη νομιμότητα της διάταξης και της σύνδεσης των λέξεων της γλώσσας για τη **δημιουργία προτάσεων**.

Η γνώση του **συντακτικού** επιτρέπει τη δημιουργία σωστών προτάσεων στις φυσικές γλώσσες, ενώ στις γλώσσες προγραμματισμού τη δημιουργία σωστών εντολών.

Οι **φυσικές** γλώσσες **εξελίσσονται** συνεχώς και αυτό γιατί η γλώσσα χρησιμοποιείται για την επικοινωνία μεταξύ ανθρώπων

οι **τεχνητές** γλώσσες χαρακτηρίζονται από στασιμότητα, αφού κατασκευάζονται συνειδητά για ένα συγκεκριμένο σκοπό. Ωστόσο συχνά οι γλώσσες προγραμματισμού **βελτιώνονται** και μεταβάλλονται προς το καλύτερο

B Πρόβλημα και Τεχνικές Σχεδίασης γλωσσών προγραμματισμού για την επίλυση ενός προβλήματος με τον Υπολογιστή

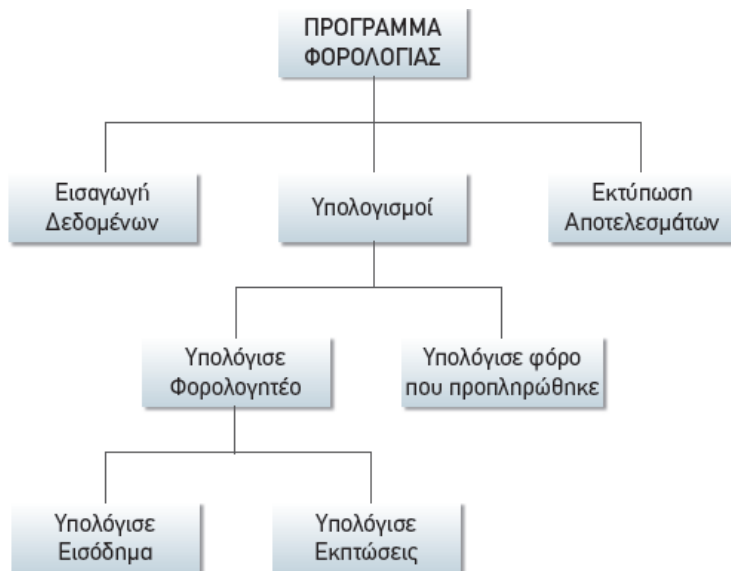
Η επίλυση ενός προβλήματος με τον υπολογιστή περιλαμβάνει :

- Τον ακριβή προσδιορισμό του προβλήματος.
- Την ανάπτυξη του αντίστοιχου αλγορίθμου.
- Τη διατύπωση του αλγορίθμου σε κατανοητή μορφή από τον υπολογιστή δηλ. Την δημιουργία προγραμμάτων και την ανάπτυξη γλωσσών προγραμματισμού

Στις Τεχνικές σχεδίασης προγραμμάτων αναφέρεται η **Ιεραρχική σχεδίαση** προγράμματος

Ο Σκοπός της ιεραρχικής σχεδίασης είναι η διάσπαση του προβλήματος σε μια σειρά από **απλούστερα υποπροβλήματα**, τα οποία να είναι εύκολο να επιλυθούν οδηγώντας στην επίλυση του αρχικού προβλήματος.

Παράδειγμα Ιεραρχικής σχεδίασης



Η ιεραρχική σχεδίαση και υλοποιείται με τον τμηματικό προγραμματισμό.

Κάθε υποπρόβλημα αποτελεί ανεξάρτητη ενότητα (module), που γράφεται ξεχωριστά από τα υπόλοιπα τμήματα προγράμματος.

Επομένως ο τμηματικός προγραμματισμός διευκολύνει τη δημιουργία του προγράμματος, μειώνει τα λάθη και επιτρέπει την ευκολότερη παρακολούθηση, κατανόηση και συντήρηση του προγράμματος

Όλες οι **σύγχρονες γλώσσες προγραμματισμού** υποστηρίζουν το **δομημένο προγραμματισμό** ((structured programming).) την ανάγκη δηλαδή να υπάρχει μία κοινή **μεθοδολογία** στην ανάπτυξη των προγραμμάτων

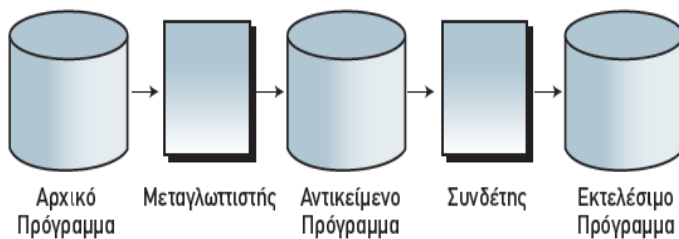
Ο δομημένος προγραμματισμός στηρίζεται σε **λογικές δομές** , όπως τη δομή της ακολουθίας, τη δομή της επιλογής και τη δομή της επανάληψης και έχει στόχο όλα τα προγράμματα να μπορούν να γραφούν χρησιμοποιώντας συνδυασμούς των τριών αυτών λογικών δομών. Να σημειωθεί ότι κάθε πρόγραμμα ή και κάθε ενότητα προγράμματος έχει μία είσοδο και μία έξοδο.

ο όρος δομημένος προγραμματισμός περιέχει τόσο την ιεραρχική σχεδίαση όσο και τον τμηματικό προγραμματισμό.

Πλεονεκτήματα του δομημένου προγραμματισμού είναι :

- Η Δημιουργία απλούστερων προγραμμάτων.
- Η Άμεση μεταφορά των αλγορίθμων σε προγράμματα.
- Η Διευκόλυνση ανάλυσης του προγράμματος σε τμήματα.
- Περιορισμός των λαθών κατά την ανάπτυξη του προγράμματος.
- Διευκόλυνση στην ανάγνωση και κατανόηση του προγράμματος από τρίτους.
- Ευκολότερη διόρθωση και συντήρηση.

Γ Μεταφραστικά Προγράμματα



Μια γλώσσα προγραμματισμού πρέπει να **μετατραπεί** σε μορφή **αναγνωρίσιμη** και **εκτελέσιμη** από τον υπολογιστή, δηλαδή σε **γλώσσα μηχανής**.

Υπάρχουν δύο μεγάλες κατηγορίες μεταφραστικών προγραμμάτων, οι **μεταγλωττιστές (compilers)** και οι **διερμηνευτές (interpreters)**.

Ο **μεταγλωττιστής** δέχεται στην είσοδο ένα πρόγραμμα γραμμένο σε μια γλώσσα υψηλού επιπέδου (**πηγαίο ή αρχικό πρόγραμμα**) και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής (**αντικείμενο**) το οποίο εκτελείται οποτεδήποτε από τον υπολογιστή και είναι τελείως ανεξάρτητο από το αρχικό πρόγραμμα.

Αντίθετα ο **διερμηνευτής** διαβάσει **μία προς μία τις εντολές** του αρχικού προγράμματος και για καθεμία εκτελεί αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής.

Το **αρχικό πρόγραμμα** λέγεται **πηγαίο πρόγραμμα (source)**, ενώ το **πρόγραμμα που παράγεται από το μεταγλωττιστή** λέγεται **αντικείμενο πρόγραμμα (object)**.

Το **αντικείμενο πρόγραμμα** συνήθως χρειάζεται να συμπληρωθεί και να **συνδεθεί** με άλλα **τμήματα προγράμματος** που είτε τα γράφει ο προγραμματιστής είτε βρίσκονται στις **βιβλιοθήκες (libraries)** της γλώσσας. Το **πρόγραμμα που επιτρέπει αυτή τη σύνδεση ονομάζεται συνδέτης – φορτωτής (linker- loader)**.

Το αποτέλεσμα του συνδέτη είναι η **παραγωγή του εκτελέσιμου προγράμματος (executable)**, το οποίο είναι το τελικό πρόγραμμα που εκτελείται από τον

υπολογιστή. Για το λόγο αυτό η συνολική διαδικασία αποκαλείται μεταγλώπηση και σύνδεση

Η δημιουργία του **εκτελέσιμου προγράμματος** γίνεται μόνο στην περίπτωση που το αρχικό πρόγραμμα **δεν περιέχει συντακτικά λάθη**.

Τα λάθη του προγράμματος είναι γενικά δύο ειδών, λογικά και συντακτικά. Τα **λογικά λάθη** εμφανίζονται μόνο στην **εκτέλεση**, ενώ τα **συντακτικά λάθη** στο **στάδιο της μεταγλώπησης**.

Τα **λογικά λάθη** οφείλονται σε σφάλματα κατά την **υλοποίηση του αλγορίθμου**, ενώ τα **συντακτικά** παρουσιάζονται κατά την εγγραφή και σύνταξη των εντολών του αρχικού (πηγαίου) προγράμματος

Ο μεταγλωπστής ή ο διερμηνευτής ανιχνεύει τα **συντακτικά λάθη** και εμφανίζει κατάλληλα διαγνωστικά μηνύματα.

Για την αρχική σύνταξη των προγραμμάτων και τη διόρθωσή τους χρησιμοποιείται ένα ειδικό πρόγραμμα που ονομάζεται **συντάκτης** (editor). Ο συντάκτης είναι ουσιαστικά ένας μικρός επεξεργαστής κειμένου, με δυνατότητες που διευκολύνουν τη γρήγορη γραφή των εντολών των προγραμμάτων.

Για τη δημιουργία, τη μετάφραση και την εκτέλεση ενός προγράμματος απαιτούνται τουλάχιστον τρία προγράμματα: **ο συντάκτης, ο μεταγλωπστής και ο συνδέτης**. Τα σύγχρονα **προγραμματιστικά περιβάλλοντα** παρέχουν αυτά τα προγράμματα με **ενιαίο τρόπο**.

Το κάθε προγραμματιστικό περιβάλλον έχει φυσικά διαφορετικά εργαλεία και ιδιότητες. Για παράδειγμα, ένα περιβάλλον οπτικού (visual) προγραμματισμού πρέπει να περιέχει οπωσδήποτε και ειδικό συντάκτη που να διευκολύνει τη δημιουργία γραφικών αντικειμένων (για παράδειγμα, φόρμες, λίστες, παράθυρα διαλόγου κ.α)

Η χρήση μεταγλωπστή έχει το **μειονέκτημα** ότι, προτού χρησιμοποιηθεί ένα πρόγραμμα, πρέπει να περάσει από τη διαδικασία της μεταγλώπησης και σύνδεσης. Από την άλλη μεριά η χρήση διερμηνευτή έχει το **πλεονέκτημα** της άμεσης εκτέλεσης και συνεπώς και της άμεσης διόρθωσης. Όμως η εκτέλεση του προγράμματος μερικές φορές, καθίσταται πιο αργή από εκείνη του ισοδύναμου εκτελέσιμου προγράμματος που παράγει ο μεταγλωπστής.

Τα σύγχρονα προγραμματιστικά περιβάλλοντα παρουσιάζονται συνήθως **μεικτές υλοποιήσεις μετάφρασης** , όπου χρησιμοποιείται διερμηνευτής κατά τη φάση δημιουργίας του προγράμματος και μεταγλωπστής για την τελική έκδοση και εκμετάλλευση του προγράμματος.

Τα σύγχρονα ολοκληρωμένα προγραμματιστικά περιβάλλοντα περιέχουν όλα τα προγράμματα και τα εργαλεία που απαιτούνται και βοηθούν τη συγγραφή, την εκτέλεση και κύρια τη διόρθωση των προγραμμάτων.

6.3 Φυσικές και τεχνητές γλώσσες

E2014-A1-1

Οι γλώσσες χαμηλού επιπέδου είναι τεχνητές γλώσσες, ενώ οι γλώσσες υψηλού επιπέδου είναι φυσικές γλώσσες. Σ Λ

E2004-Θ1Ε

1. Τι καλείται αλφάβητο μιας γλώσσας;
2. Από τι αποτελείται το λεξιλόγιο μιας γλώσσας;
3. Τι είναι το τυπικό μιας γλώσσας;
4. Τι είναι το συντακτικό μιας γλώσσας;

2017-ΘΑ2β

Να αναφέρετε ονομαστικά τα στοιχεία που προσδιορίζουν μία γλώσσα.

2018-A2β

Τι εννοούμε με τον όρο «τυπικό» και τι με τον όρο «συντακτικό» μιας γλώσσας;

6.4 Τεχνικές σχεδίασης προγραμμάτων

6.4.1 Ιεραρχική σχεδίαση προγράμματος

E2015-ΘΑ1-3

Ο ιεραρχικός προγραμματισμός χρησιμοποιεί τη στρατηγική της συνεχούς διαίρεσης του προβλήματος σε υποπροβλήματα. Σ Λ

2019-A1-4, B2019-A1-4

Σκοπός της ιεραρχικής σχεδίασης είναι η διάσπαση του προβλήματος σε μια σειρά από απλούστερα υποπροβλήματα. Σ Λ

6.4.2 Τμηματικός προγραμματισμός

ΕΠ2020-ΘΑ2γ, ΞΠ2020-ΘΑ2γ

Πώς ορίζεται ο τμηματικός προγραμματισμός;

6.4.3 Δομημένος προγραμματισμός

E2004-Θ1Γ3

Η εντολή GOTO που αλλάζει τη ροή εκτέλεσης ενός προγράμματος είναι απαραίτητη στο δομημένο προγραμματισμό . Σ Λ

B2001-Θ1Δ1

Ποια είναι τα πλεονεκτήματα του δομημένου προγραμματισμού;

E2003-Θ1Δ

Σε ποιες στοιχειώδεις λογικές δομές στηρίζεται ο δομημένος προγραμματισμός;

Να αναφέρετε τέσσερα πλεονεκτήματα του δομημένου προγραμματισμού.

Π2016-A2

Να αναφέρετε επιγραμματικά πέντε από τα πλεονεκτήματα του δομημένου προγραμματισμού.

E2020-ΘΑ2α, ΕΠ2020-ΘΑ2α, Ξ2020-ΘΑ2α, ΞΠ2020-ΘΑ2α

Να αναφέρετε επιγραμματικά τα πλεονεκτήματα του δομημένου προγραμματισμού (μονάδες 6).

6.7 Προγραμματιστικά περιβάλλοντα

2004-Θ1Α

2. Ο μεταγλωττιστής δέχεται στην είσοδό του ένα πρόγραμμα γραμμένο σε μια γλώσσα υψηλού επιπέδου και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής. Σ Λ

3. Το πηγαίο πρόγραμμα εκτελείται από τον υπολογιστή χωρίς μεταγλώττιση. Σ Λ

4. Ο διερμηνευτής διαβάζει μία προς μία τις εντολές του πηγαίου προγράμματος και για κάθε μια εκτελεί αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής. Σ Λ

E2002-Θ1Α4

Τα λογικά λάθη είναι συνήθως λάθη σχεδιασμού και δεν προκαλούν τη διακοπή της εκτέλεσης του προγράμματος. Σ Λ

E2004-Θ1Γ4

Τα συντακτικά λάθη στον πηγαίο κώδικα εμφανίζονται κατά το στάδιο της μεταγλώττισής του. Σ Λ

E2006-Θ1Α3

Το πρόγραμμα που παράγεται από το μεταγλωττιστή λέγεται εκτελέσιμο. Σ Λ

2014-A1-3

Το εκτελέσιμο πρόγραμμα δημιουργείται ακόμα και στην περίπτωση που το αρχικό πρόγραμμα περιέχει λογικά, αλλά όχι συντακτικά λάθη Σ Λ

2016-A1-5

Το πρόγραμμα Συντάκτης εντοπίζει τα συντακτικά λάθη του προγράμματος. Σ Λ

Π2016-A1-4, ΠΒ2016-A1-4

Οι μεταβλητές που χρησιμοποιούνται σ' ένα πρόγραμμα αντιστοιχούνται από το μεταγλωττιστή σε συγκεκριμένες θέσεις μνήμης του υπολογιστή. Σ Λ

Ξ2016-A1-1

Ο μεταγλωττιστής δέχεται στην είσοδο ένα πρόγραμμα σε γλώσσα μηχανής και παράγει στην έξοδο ένα ισοδύναμο πρόγραμμα σε μια γλώσσα υψηλού επιπέδου. Σ Λ

2017-A1-2

Η χρησιμοποίηση του διερμηνευτή για τη μετάφραση ενός προγράμματος έχει ως αποτέλεσμα την ταχύτερη εκτέλεσή του. Σ Λ

E2017-A1-4, Ξ2017-A1-4

Ο μεταγλωττιστής διαβάζει μία προς μία τις εντολές του αρχικού προγράμματος και για κάθε μία εκτελεί αμέσως μία ισοδύναμη ακολουθία εντολών μηχανής. Σ Λ

E2017-A1-5, Ξ2017-A1-5

Τα συντακτικά λάθη εντοπίζονται στη φάση της μεταγλώττισης. Σ Λ

E2018-A1-1, Ξ2018-A1-1

Τα λογικά λάθη εμφανίζονται στο στάδιο της μεταγλώττισης. Σ Λ

E2019-A1-1, Ξ2019-A1-5

Η δημιουργία του εκτελέσιμου προγράμματος γίνεται μόνο στην περίπτωση που το αρχικό πρόγραμμα δεν περιέχει λογικά λάθη. Σ Λ

2021-A1-1

Ο μεταγλωττιστής εντοπίζει τα λογικά λάθη. Σ Λ

Ε2021-A1-4, Ξ2021-A1-4

Τα λογικά λάθη δεν προκαλούν τη διακοπή της εκτέλεσης του προγράμματος. Σ Λ

2022-A1-5

Η χρήση του διερμηνευτή καθιστά την εκτέλεση του προγράμματος πιο αργή. Σ Λ

2014-A4β

Ποιος είναι ο ρόλος του συντάκτη σε ένα προγραμματιστικό περιβάλλον;

2014-A4γ

Ποιος είναι ο ρόλος του συνδέτη-φορτωτή σε ένα προγραμματιστικό περιβάλλον;

2014-A4δ

Ποιος είναι ο ρόλος του μεταγλωττιστή σε ένα προγραμματιστικό περιβάλλον;

2008-Θ1Β2

Ποια η διαφορά μεταξύ διερμηνευτή και μεταγλωττιστή;

2002-Θ1Γ

Να περιγράψετε τη διαδικασία για τη μετατροπή με μεταγλωττιστή ενός πηγαίου προγράμματος σε εκτελέσιμο πρόγραμμα, συμπεριλαμβανομένης της ανίχνευσης και διόρθωσης λαθών.

Ε2002-Θ1Β

Ποιες είναι οι διαφορές μεταξύ μεταγλωττιστή (compiler) και διερμηνευτή (interpreter).

2009-Θ1Γ2α

Πότε εμφανίζονται τα συντακτικά λάθη ενός προγράμματος και πότε τα λογικά;

2021-A2α

Τι ονομάζεται αντικείμενο πρόγραμμα;

2007-Θ1Δ

Δίνονται οι παρακάτω προτάσεις:

Π1. Ο συνδέτης-φορτωτής μετατρέπει το _____ 1 _____ πρόγραμμα σε _____ 2 _____ πρόγραμμα

Π2. Ο συντάκτης χρησιμοποιείται για να δημιουργηθεί το _____ 3 _____ πρόγραμμα

Π3. Ο μεταγλωττιστής μετατρέπει το _____ 4 _____ πρόγραμμα σε _____ 5 _____ πρόγραμμα

και οι παρακάτω λέξεις:

α. αντικείμενο

β. εκτελέσιμο

γ. πηγαίο.

1. Να γράψετε στο τετράδιό σας τους αριθμούς (1–5) των κενών διαστημάτων των προτάσεων και δίπλα το γράμμα της λέξης (α, β, γ) που αντιστοιχεί σωστά.

ΣΗΜΕΙΩΣΗ: Κάποιες από τις λέξεις χρησιμοποιούνται περισσότερες φορές από μία.

2. Κατά την ανάπτυξη ενός προγράμματος σε ένα προγραμματιστικό περιβάλλον, με ποια χρονική σειρά πραγματοποιούνται τα βήματα που περιγράφουν οι παραπάνω προτάσεις; Να απαντήσετε γράφοντας τα Π1, Π2, Π3 με τη σωστή σειρά.

E2015-ΘΑ3

Προκειμένου να επιλύσουμε ένα πρόβλημα με τον υπολογιστή, εκτελούμε κάποια βήματα.

Τα βήματα αυτά δίνονται παρακάτω με τυχαία σειρά:

α. Σύνδεση του προγράμματος

β. Σύνταξη του προγράμματος

γ. Ανάπτυξη του αντίστοιχου αλγορίθμου

δ. Διατύπωση του προβλήματος

ε. Μεταγλώττιση του προγράμματος

στ. Εκτέλεση του προγράμματος

ζ. Κατανόηση του προβλήματος

η. Ανάλυση του προβλήματος

Κάποια από τα βήματα αυτά (α, β,..., η) περιλαμβάνονται στη μεσαία στήλη του Πίνακα Ι.

Απάντηση

Προηγούμενο	Βήμα	Επόμενο
β	ε	α
η	γ	β
δ	ζ	η
ε	α	στ

Προηγούμενο	Βήμα	Επόμενο
	ε	
	γ	
	ζ	
	α	

Πίνακας Ι

Να αντιγράψετε τον Πίνακα Ι στο τετράδιό σας και, για καθένα από τα βήματα που δίνονται στη μεσαία στήλη, να συμπληρώσετε, με βάση τη σωστή σειρά εκτέλεσης των βημάτων:

α. στη στήλη Προηγούμενο, το γράμμα που αντιστοιχεί στο αμέσως προηγούμενο βήμα (μονάδες 4).

β. στη στήλη Επόμενο, το γράμμα που αντιστοιχεί στο αμέσως επόμενο βήμα (μονάδες 4).