

Αντικειμενοστρεφής Προγραμματισμός–Python

Κ.Π. Γιαλούρης

Στόχοι του μαθήματος

- Απόκτηση αναλυτικής και αλγοριθμικής σκέψης
- Απόκτηση δεξιοτήτων επίλυσης προβλημάτων
- Γνώση προγραμματισμού σε Python

Στόχοι του σημερινού μαθήματος

- ❑ Κατανόηση της έννοιας **Αλγόριθμος**
- ❑ Εισαγωγή στον Προγραμματισμό
- ❑ Κατανόηση βασικών εννοιών
 - Τύπος δεδομένων
 - Μεταβλητή
 - Εντολή εισόδου
 - Εντολή εξόδου
- ❑ Εξοικείωση με τα περιβάλλοντα της Python

Αλγόριθμος

- Αλγόριθμος είναι η πεπερασμένη ακολουθία βημάτων ή ενεργειών που πρέπει να εκτελεστούν για την επίλυση ενός προβλήματος.

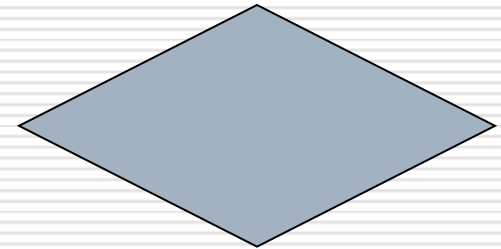
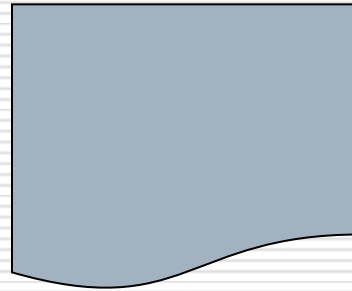
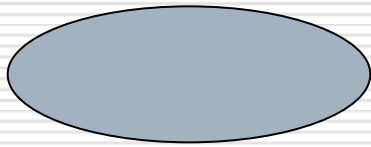
Ένας καλός αλγόριθμος πρέπει:

- ❑ Να περιγράφει με ακρίβεια τα δεδομένα που θα χρησιμοποιήσει.
- ❑ Κάθε βήμα του να είναι τελείως καθορισμένο και να φανερώνει τι ακριβώς κάνει.
- ❑ Να δίνει με ακρίβεια τα αποτελέσματα που παράγονται σε κάθε βήμα του.
- ❑ Να ολοκληρώνεται πάντα ύστερα από ένα πεπερασμένο αριθμό βημάτων που καθένα να εκτελείται σε πεπερασμένο χρόνο.

Μέθοδοι παράστασης αλγορίθμου

- Φραστική
- Διάγραμμα ροής
- Ψευδοκώδικας
- Κωδικοποίηση σε γλώσσα προγραμματισμού

Μέθοδοι παράστασης αλγορίθμου



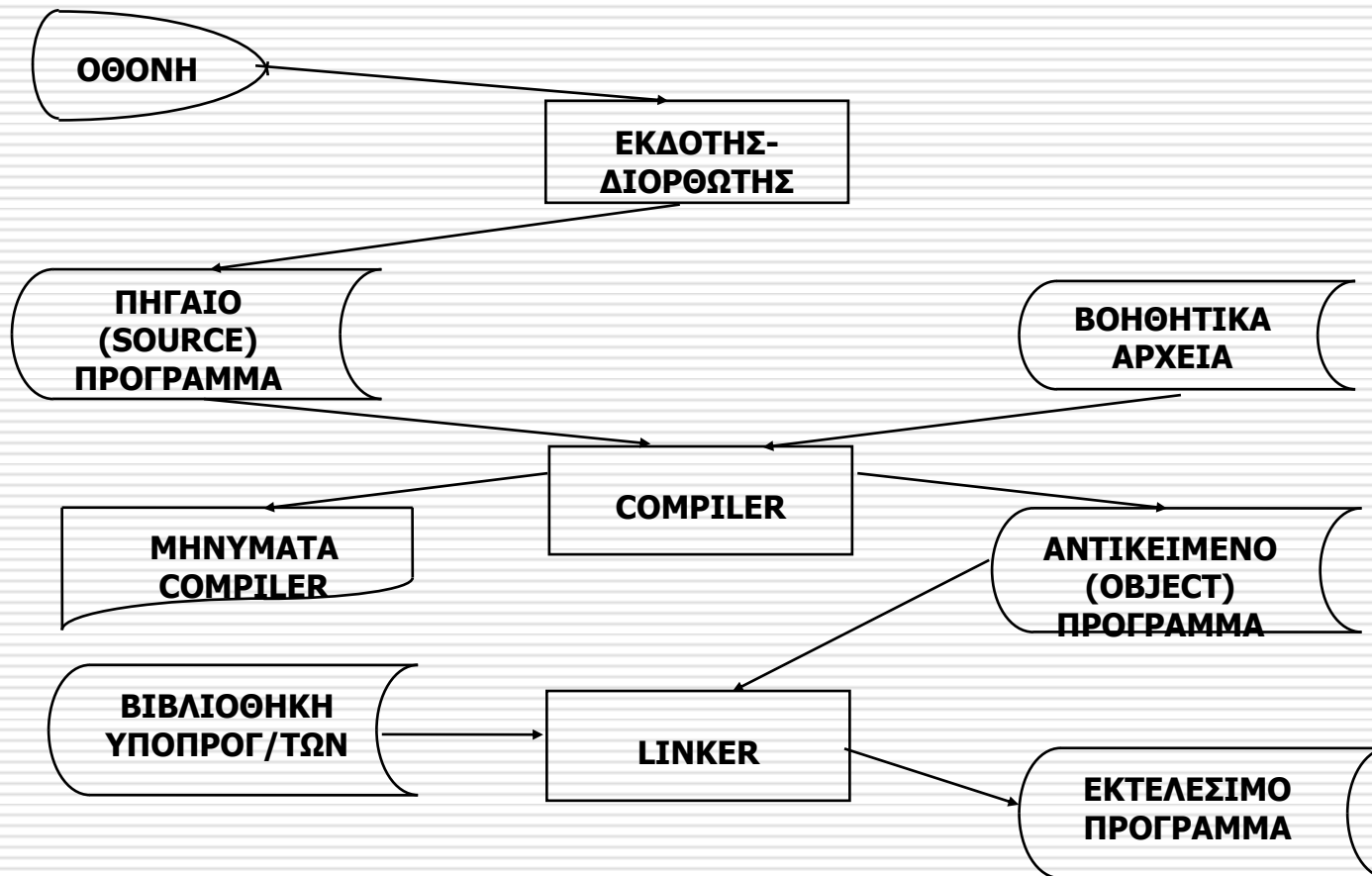
Γλώσσες Προγραμματισμού

- Γλώσσες μηχανής
- Συμβολικές Γλώσσες
- Γλώσσες Υψηλού Επιπέδου
 - FORTRAN, COBOL, ALGOL, BASIC, RPG, ADA,, PASCAL, C, PROLOG, LISP, LOGO, JAVA, PERL κ.α.
- Γλώσσες προγραμματισμού 4ης Γενιάς
- Φυσικές γλώσσες

Μεταφραστικά προγράμματα

- ❑ Συμβολομεταφραστής (assembler)
- ❑ Μεταγλωτιστής (compiler)
- ❑ Διερμηνέας (Interpreter)

Διαδικασία γραφής εκτέλεσης προγράμματος



Μορφές προγραμματισμού

- ❑ Δομημένος (structural)
- ❑ Διαδικασιακός (Procedural)
- ❑ Συναρτησιακός (functional)
- ❑ Λογικός (Logic oriented)
- ❑ Αντικειμενοστρεφής (Object oriented)

Δομημένος Προγραμματισμός

- Στόχος του δομημένου προγραμματισμού είναι να αλλάξει τη διαδικασία καταγραφής του αλγόριθμου από μια επίπονη λειτουργία (*δοκιμής και λάθους*) σε μια ποιοτική και ελεγχόμενη λειτουργία.

Δομημένος Προγραμματισμός

Πλεονεκτήματα

- ❑ Διευκόλυνση στην ανάπτυξη του αλγόριθμου κατά τμήματα.
- ❑ Ευκολία και ταχύτητα στην κωδικοποίηση.
- ❑ Καλύτερη ποιότητα προγραμμάτων.
- ❑ Ευκολία στις διορθώσεις και τη συντήρηση.
- ❑ Τεκμηρίωση που περιέχεται σχεδόν εξ' ολοκλήρου στο ίδιο το πρόγραμμα

Δομημένος Προγραμματισμός

□ Βασικές Δομές

- Ακολουθία

- Επιλογή

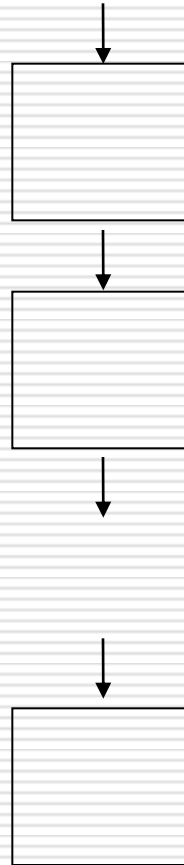
- Επανάληψη

□ Βασικές Τεχνικές

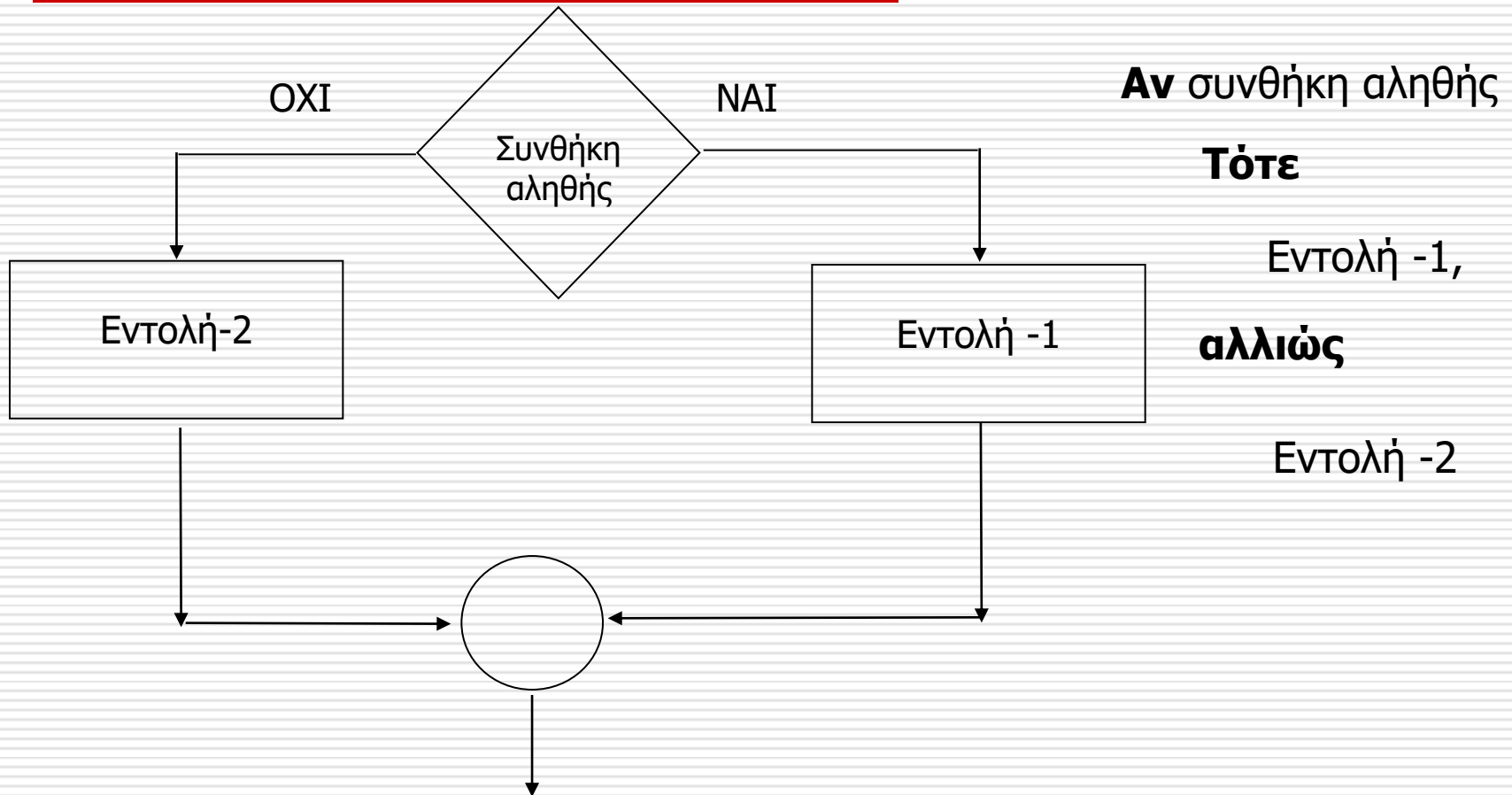
- Ιεραρχικός προγραμματισμός

- Τμηματικός προγραμματισμός

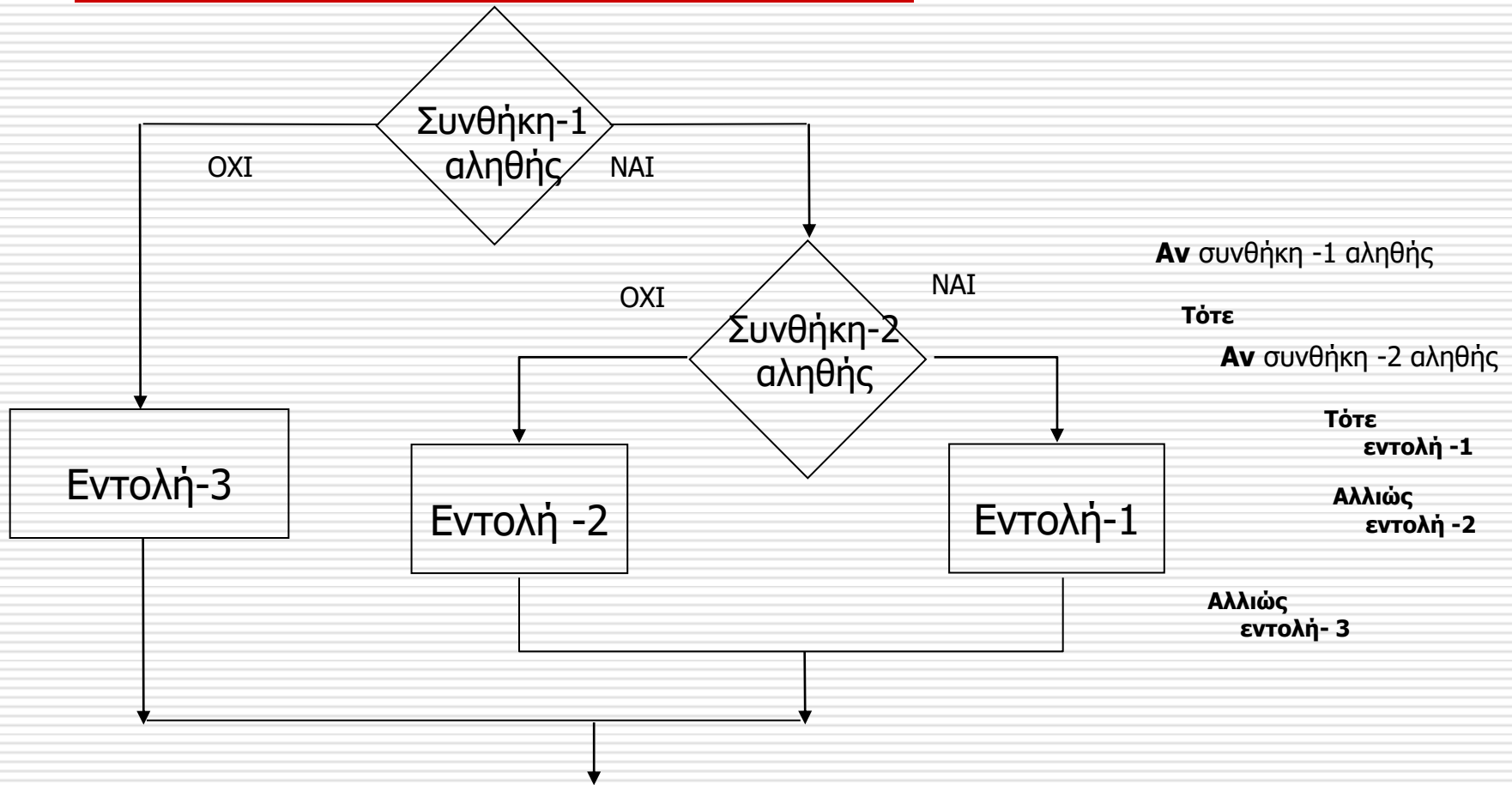
Δομημένος Προγραμματισμός ακολουθία



Δομημένος προγραμματισμός επιλογή



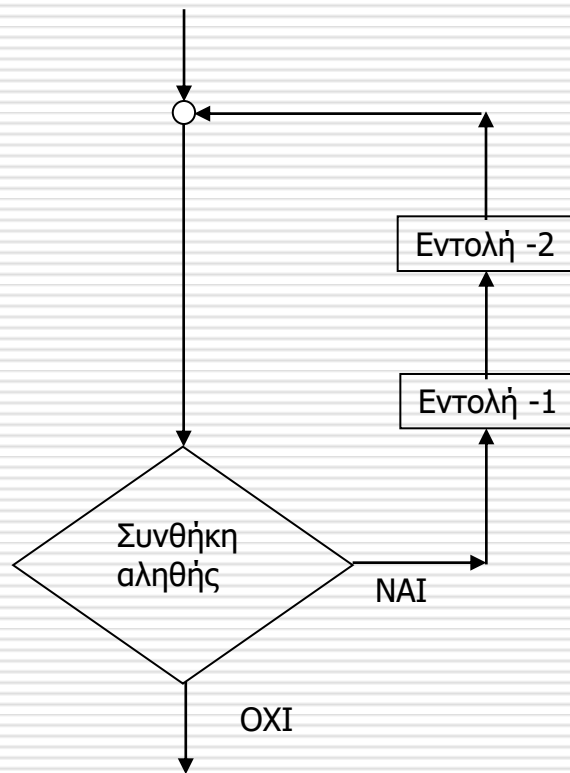
Δομημένος προγραμματισμός επιλογή



Δομημένος προγραμματισμός

Επανάληψη

Εφόσον (συνθήκη αληθής) **Επανάλαβε**



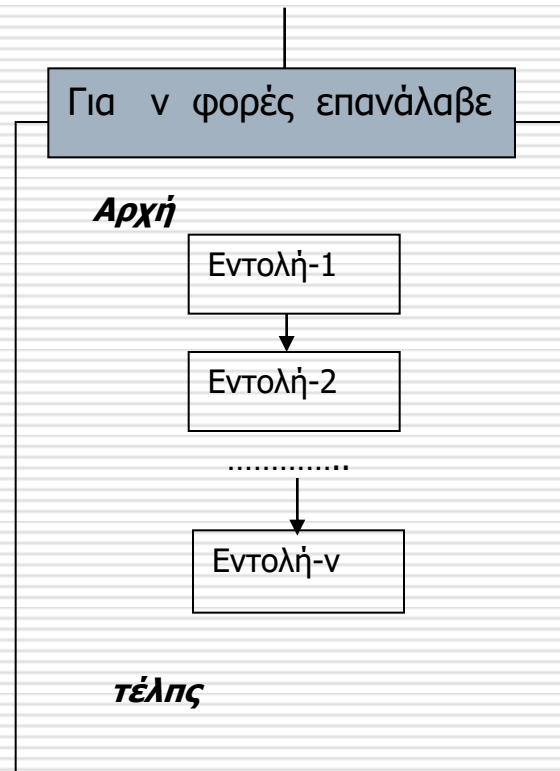
Αρχή επανάληψης

Εντολή -1

Εντολή -2

Τέλος επανάληψης

Δομημένος προγραμματισμός Επανάληψη



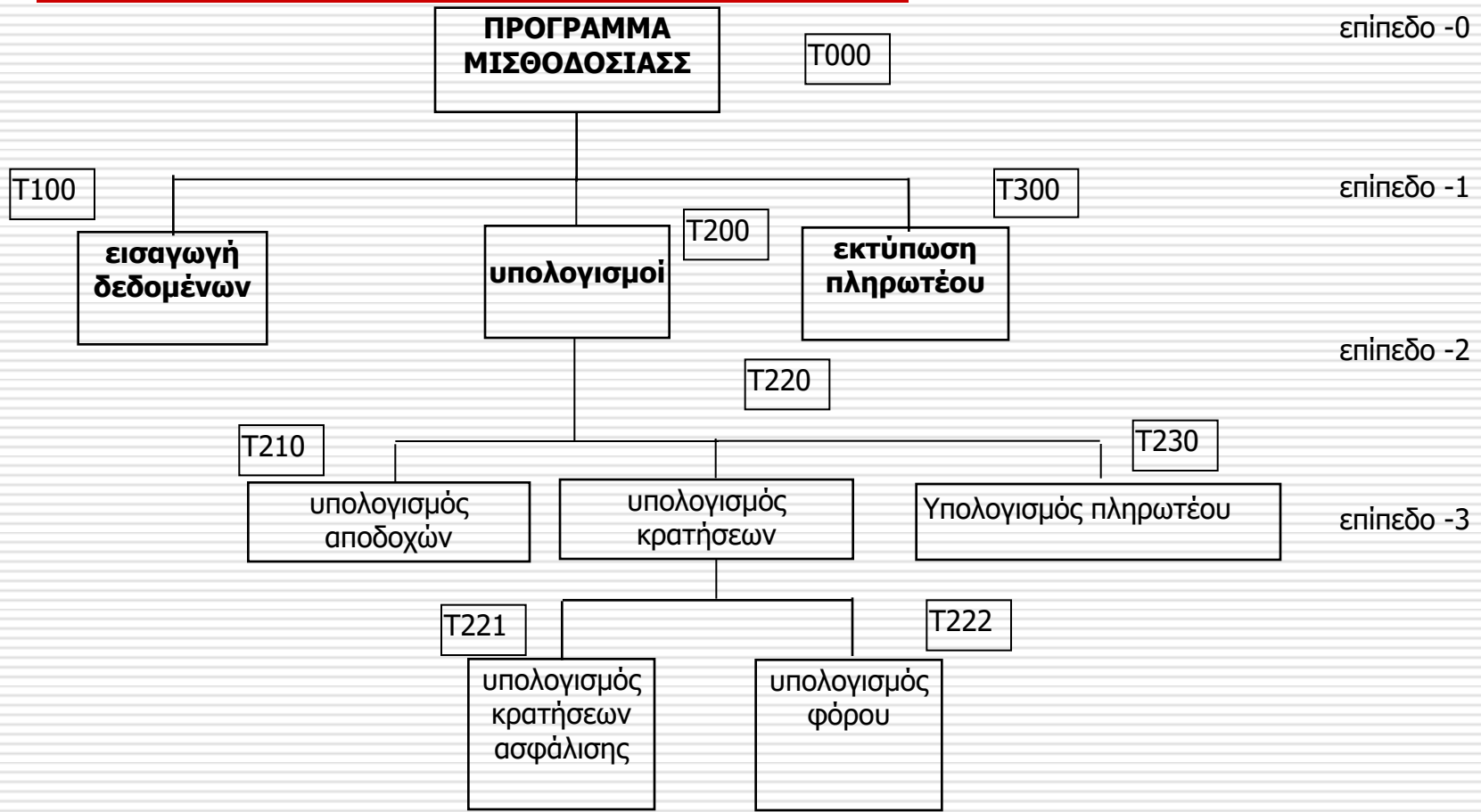
Δομημένος Προγραμματισμός

- Βασικές Τεχνικές
 - Ιεραρχικός προγραμματισμός
 - Τμηματικός προγραμματισμός

Πρόβλημα

- *Να υπολογισθεί το πληρωτέο ποσό ενός ωρομίσθιου εργαζόμενου, όταν είναι γνωστές οι ώρες εργασίας, η ωριαία αποζημίωση, το ποσοστό κρατήσεων για ασφάλιση και το ποσοστό κράτησης φόρου, εφόσον οι καθαρές αποδοχές του υπερβαίνουν ένα ελάχιστο όριο***

Δομημένος Προγραμματισμός HIPO



Δεδομένα

- Ώρες Εργασίας.*
- Ωριαία Αποζημίωση.*
- Ποσοστό Ασφάλισης.*
- Ποσοστό Φόρου.*
- Ελάχιστο Όριο αποδοχών που φορολογείται.*

Ζητούμενα

- *Πληρωτέο Ποσό*

Συνθήκη:

- *Ο φόρος υπολογίζεται εφόσον οι καθαρές αποδοχές υπερβαίνουν ένα δεδομένο όριο.*

Σχέσεις που συνδέουν Δεδομένα - Ζητούμενα

Μισθός =

*(Ώρες Εργασίας) * (Ωριαία Αποζημίωση)*

Κρατήσεις Ασφάλισης =

*(Μισθός) * (Ποσοστό Ασφάλισης)*

Φορολογητέο ποσό =

(Μισθός) - (Κρατήσεις Ασφάλισης)

Φόρος =

*(Φορολογητέο Ποσό) * (Ποσοστό Φόρου)*

Πληρωτέο Ποσό =

(Φορολογητέο Ποσό) - (Φόρος)

