

ΠΛΗΡΟΦΟΡΙΚΗ Ι (Python)

Εισαγωγή

Περιεχόμενο μαθήματος:

- Αλγοριθμική επίλυση προβλημάτων
- Προγραμματισμός με Python
- Εφαρμογές σε μαθηματικά και μη προβλήματα

Ανάπτυξη προγραμμάτων: IDLE

Βιβλίο:



Αλγόριθμος: Βήμα προς βήμα διαδικασία για την επίλυση κάποιου προβλήματος. Το πλήθος των βημάτων πρέπει να είναι πεπερασμένο.

- Είσοδος (≥ 0 δεδομένα)
- Έξοδος (≥ 1 αποτελέσματα)
- Ορισμένος (περιέχει σαφείς οδηγίες)
- Κάθε οδηγία, μεμονωμένα: εξαιρετικά απλή
- Καλύπτει όλες τις δυνατές καταστάσεις
- Εξασφαλίζει τον τερματισμό (πεπερασμένος αριθμός βημάτων ή χρόνος)

Πρόγραμμα: Ακριβής διατύπωση ενός αλγορίθμου σε μια γλώσσα προγραμματισμού.

Τρόποι αναπαράστασης/περιγραφής αλγορίθμων:

- Φυσική γλώσσα
- Διάγραμμα ροής (flowchart)
- Ψευδοκώδικας (pseudocode)
- Κώδικας

Υπολογιστική επίλυση προβλήματος:

- 1) Ανάλυση δεδομένων του προβλήματος
- 2) Μαθηματική διατύπωση του προβλήματος
- 3) Ανάπτυξη του **αλγόριθμου** (σχεδιασμός ή επιλογή κατάλληλου αλγορίθμου (συνήθως: ένα πρόβλημα – πολλοί αλγόριθμοι)
- 4) Διατύπωση αλγόριθμου σε γλώσσα προγραμματισμού: **πρόγραμμα**
- 5) Εκτέλεση προγράμματος για συγκεκριμένα δεδομένα
- 6) Ερμηνεία αποτελεσμάτων

Βασικές αλγοριθμικές ενέργειες

- Είσοδος / Έξοδος
- Πράξεις ή αναθέσεις τιμών σε μεταβλητές
- Έλεγχος ποσοτήτων – επιλογή δράσης
- Επαναληπτική εκτέλεση
- Τερματισμός

Παράδειγμα αλγόριθμου – Ταξινόμηση φουσαλίδας (bubble sort)

Πρόβλημα: Να ταξινομηθεί κατά αύξουσα σειρά μία λίστα αριθμών.

Ο αλγόριθμος της φυσαλίδας σε φυσική γλώσσα:

- 1) Σύγκρινε κάθε ζευγάρι στοιχείων (1ο με 2ο, 2ο με 3ο, ..., (n-1)ο με n-οστό) και εάν τα στοιχεία κάποιου ζευγαριού είναι λάθος ταξινομημένα, εναλλάξέ τα (swap)
- 2) Εάν έγινε έστω και μία εναλλαγή στοιχείων στο προηγούμενο βήμα, τότε επανέλαβε τη διαδικασία σύγκρισης των ζευγαριών (βήμα 1), διαφορετικά τερμάτισε τον αλγόριθμο. (σε κάθε νέα επανάληψη του 1ου βήματος, το τελευταίο ζευγάρι της προηγούμενης επανάληψης της διαδικασίας δε χρειάζεται να συγκριθεί, άρα συγκρίνεται πάντα ένα ζευγάρι λιγότερο)

Π.χ.,

Αρχική λίστα:	5	3	8	1	4	5	7	6
1η σύγκριση:	3	5	8	1	4	5	7	6
2η σύγκριση:	3	5	8	1	4	5	7	6
...								
(n-1)η σύγκριση:	3	5	1	4	5	7	6	8
Μετά το 1ο πέρασμα:	3	5	1	4	5	7	6	8
...								
Μετά το 2ο πέρασμα:	3	1	4	5	5	6	7	8
...								

Γλώσσες προγραμματισμού

- i. Γλώσσες υψηλού επιπέδου (Γ.Υ.Ε.) (Java, C, C++, Python, Fortran, Pascal, κ.α.)
- ii. Γλώσσες χαμηλού επιπέδου (γλώσσα μηχανής (Γ.Μ.), assembly)

Οι Η/Υ εκτελούν γλώσσα μηχανής

Γ.Μ.:

- διαφορετική για κάθε τύπο επεξεργαστή
- εντολές = αλληλουχίες από bits (0 και 1)
- οι εντολές εκτελούνται άμεσα από τη CPU

Assembly (λίγο ανώτερο επίπεδο από Γ.Μ.):

- διαφορετική για κάθε τύπο επεξεργαστή
- μνημονικά ονόματα αντί για bits (ίδιο πλήθος εντολών)
- μετατροπή σε Γ.Μ. από κατάλληλο πρόγραμμα (assembler)

Γ.Υ.Ε.:

- Ανεξάρτητες από τον τύπο επεξεργαστή (υψηλή φορητότητα)
- Υψηλή αφαίρεση / ευκολότερος προγραμματισμός
- Μετατροπή σε Γ.Μ. από κατάλληλα προγράμματα:
 - i) μεταγλωττιστές (compilers) ή
 - ii) διερμηνείς (interpreters)

Μεταγλωττιστές (Compilers)

- Προγράμματα που μετατρέπουν ένα πηγαίο πρόγραμμα (Γ.Υ.Ε.) σε εκτελέσιμο (Γ.Μ.).
- Το παραγόμενο εκτελέσιμο πρόγραμμα “τρέχει” αυτόνομα σε κάθε υπολογιστή που “καταλαβαίνει” τη συγκεκριμένη Γ.Μ. (δεν απαιτείται πλέον ο μεταγλωττιστής).

Διερμηνείς (Interpreters)

- Προγράμματα που εκτελούν άμεσα ένα πηγαίο πρόγραμμα (Γ.Υ.Ε.).
- Δεν παράγουν ολοκληρωμένο εκτελέσιμο κώδικα, αλλά ανά εντολή.
- Για την εκτέλεση του πηγαίου προγράμματος σε κάποιον υπολογιστή απαιτείται και η ύπαρξη του διερμηνέα.
- Τα ερμηνευόμενα προγράμματα είναι πιο αργά από τα μεταγλωττισμένα.

Σφάλματα προγραμματισμού

- α) Συντακτικά σφάλματα (syntax errors)
 - Ανιχνεύονται από τον compiler / interpreter
 - β) Σφάλματα κατά την εκτέλεση (run-time error)
 - Ανιχνεύονται από τον interpreter (σε γλώσσες που διαθέτουν)
 - γ) Λογικά σφάλματα (logic errors / bugs)
 - Ανιχνεύονται από τον προγραμματιστή (το πρόγραμμα εκτελείται αλλά το αποτέλεσμα είναι λάθος)
- Δυσκολία εντοπισμού σφαλμάτων (συνήθως): $\gamma > \beta > \alpha$
 - Διαδικασία εντοπισμού και επίλυσης σφαλμάτων: **debugging**



Βασικά χαρακτηριστικά της Python

- Γλώσσα προγραμματισμού υψηλού επιπέδου
- Υποστηρίζει τόσο Διαδικαστικό όσο και Αντικειμενοστραφή Προγραμματισμό
- Χρησιμοποιεί διερμηνέα και είναι scripting language
- Εκτελείται:
 - είτε ανά εντολή (εντολή = μία ή πολλές γραμμές κώδικα) στο Python shell / κέλυφος (Διαδραστική λειτουργία)
 - είτε ως πλήρες αρχείο προγράμματος με κατάληξη .py (Λειτουργία script)
- Δυναμική γλώσσα στον ορισμό τύπου μεταβλητών
- Εύκολη στη σύνταξη και στην εκμάθηση, κατανόηση και ανάγνωση
- Υποχρεωτική στοίχιση (indentation), όχι μόνο για μορφοποίηση
- Δημιουργία μικρότερων σε μέγεθος προγραμμάτων σε σχέση με άλλες γλώσσες προγρ/σμού
- Μεγάλη πρότυπη βιβλιοθήκη (standard library) έτοιμων συναρτήσεων
- Πληθώρα υλοποιήσεων: CPython, Cython, Jython, PyPy, ...
- Πληθώρα IDEs: IDLE, Ipython, PythonAnywhere (on-line), ...

Ιστορικά στοιχεία

- Αναπτύχθηκε από τον Ολλανδό Guido van Rossum στις αρχές της δεκαετίας του '90
- Πρώτη έκδοση: Python 0.9.0 – Φεβρουάριος 1991
- Python 1.0 – Ιανουάριος 1994
- Python 2.0 – Οκτώβριος 2000
- Python 3.0 (backwards-incompatible) – Δεκέμβριος 2008
- Τρέχουσα έκδοση: Python 3.13.0 (7 Οκτωβρίου 2024)