

Κεφάλαιο 2ο

Από τον Αλγόριθμο στην ανάπτυξη προγράμματος

2.1 Κύκλος ανάπτυξης προγράμματος/λογισμικού

Η διαδικασία ανάπτυξης λογισμικού αποτελεί μια εργασία που εξελίσσεται σε διακριτές φάσεις ή στάδια και θεωρείται υποσύνολο του κύκλου ζωής ενός συστήματος λογισμικού που ξεκινά από την ανάλυση απαιτήσεων και τελειώνει με την παύση λειτουργίας του. Μεταξύ των βασικών μεθοδολογιών (μοντέλων) -που έχουν προταθεί και ακολουθούνται- είναι το μοντέλο του Καταρράκτη (Waterfall model) και αυτό της Σπειροειδούς προσέγγισης (Spiral model).

2.1.1 Μοντέλο του καταρράκτη

Πρόκειται για το μοντέλο που υποδιαιρεί τη διαδικασία ανάπτυξης ενός συστήματος λογισμικού στις ακόλουθες φάσεις:

- Ανάλυση απαιτήσεων.
- Σχεδίαση.
- Υλοποίηση.
- Ολοκλήρωση.
- Λειτουργία και συντήρηση.

2.1.2 Μοντέλο σπείρας

Στο μοντέλο της σπείρας, η ανάπτυξη ακολουθεί μια εξελικτική διαδικασία με την επαναληπτική εκτέλεση ενός κύκλου φάσεων. Σε καθεμία φάση δημιουργείται μια ενδιάμεση έκδοση του τελικού προϊόντος, η οποία βελτιώνεται κατά τον επόμενο κύκλο κ.ο.κ. Η διαδικασία αυτή συνεχίζεται μέχρι να παραχθεί μια έκδοση που να ικανοποιεί τις απαιτήσεις των χρηστών.

Παρόμοια με τη διαδικασία ανάπτυξης ενός λογισμικού, εργαζόμαστε και κατά την ανάπτυξη ενός προγράμματος, ακολουθώντας πάλι μια μεθοδολογία υλοποίησης του *κύκλου ανάπτυξης προγράμματος* (program development life cycle-PDLC).

Μια από τις μεθοδολογίες ανάπτυξης προγράμματος αποτελείται από φάσεις, όπως:

- Ανάλυση του προβλήματος.
- Σχεδίαση.
- Συγγραφή κώδικα.
- Έλεγχος και εκσφαλμάτωση.
- Τεκμηρίωση.

2.2 Η λογική συγγραφής προγράμματος ανάλογα με το είδος προγραμματισμού

Από τη δεκαετία του 1960 μέχρι σήμερα, έχουν αναπτυχθεί διάφορα είδη προγραμματισμού που τα υποστήριξαν πολλές γλώσσες και θα μπορούσαμε να τα κατηγοριοποιήσουμε σε μεγάλες κατευθύνσεις, τα λεγόμενα **Προγραμματιστικά Υποδείγματα** (programming paradigms).

Τα βασικά προγραμματιστικά υποδείγματα είναι τα ακόλουθα:

- Ο Προστακτικός προγραμματισμός (imperative programming).
- Ο Δηλωτικός προγραμματισμός (declarative programming).

2.2.1 Προστακτικός προγραμματισμός

Ο **Προστακτικός προγραμματισμός** βασίζεται σε εντολές που υλοποιούν τα βήματα ενός αλγόριθμου, ενεργώντας σε μεταβλητές και αλλάζοντας την κατάσταση τους.

Βρίσκεται πιο κοντά στη λογική λειτουργίας του υπολογιστή. Γλώσσες που ακολούθησαν το είδος αυτό είναι οι κλασικές γλώσσες προγραμματισμού, όπως Cobol, Fortran, Pascal, C κ.ά.

2.2.1.1 Δομημένος και μη προγραμματισμός

Κατά την αρχική περίοδο του προγραμματισμού, η διακλάδωση της ροής γίνονταν με την εντολή goto, κατάσταση που οδηγούσε σε μη δομημένα προγράμματα (**μη δομημένος προγραμματισμός** - unstructured programming). Στη συνέχεια δημιουργήθηκε το πρότυπο του **Δομημένου προγραμματισμού** (structured programming), με τις εντολές σε ομάδες (blocks) να ακολουθούν την **Ιεραρχική** λογική ροής και τη δυνατότητα χρήσης υπορουτινών και διάφορων άλλων δομών, όπως η if-then-else.

Ο **Διαδικαστικός** προγραμματισμός (procedural programming) αποτελεί μια υποκατηγορία του Δομημένου προγραμματισμού, με το πρόγραμμα να αποτελείται από αυτοτελείς ομάδες εντολών, τις **διαδικασίες** (procedures). Η γλώσσα προγραμματισμού που αξιοποίησε αρχικά το είδος αυτό είναι η Pascal, από το 1970.

Ο **Αντικειμενοστραφής** προγραμματισμός (object-oriented programming) βασίζεται, σε αντίθεση με το Διαδικαστικό προγραμματισμό, σε **αντικείμενα** που αλληλεπιδρούν μεταξύ τους, αποτελώντας πρότυπο που ταιριάζει περισσότερο στη λογική οργάνωσης και λειτουργίας του πραγματικού κόσμου.

2.2.2 Δηλωτικός προγραμματισμός

Ο **Δηλωτικός** προγραμματισμός (declarative programming paradigm) βασίζεται στην περιγραφή του σκοπού, τον οποίο ζητείται από το πρόγραμμα να επιτύχει. Στο γενικό αυτό υπόδειγμα ανήκουν διάφορες υποκατηγορίες προγραμματισμού, όπως είναι ο Συναρτησιακός και ο Λογικός.

Ο **Συναρτησιακός** προγραμματισμός (functional programming) βασίζεται σε μαθηματικές συναρτήσεις, με γλώσσες όπως Lisp, Logo κ.ά.

Στο **Λογικό** προγραμματισμό (logic programming), ένα πρόγραμμα είναι ένα σύνολο από αξιώματα ή κανόνες οι οποίοι καθορίζουν σχέσεις ανάμεσα σε αντικείμενα. Υπολογισμός ενός λογικού προγράμματος είναι ένα συμπέρασμα που συνάγεται από τα αποτελέσματά του.

Στο προγραμματιστικό πρότυπο του Δηλωτικού προγραμματισμού μπορούμε να θεωρήσουμε ότι ανήκουν και άλλες γλώσσες, που δεν υπάγονται στις δύο προηγούμενες κατηγορίες.

Χαρακτηριστικές είναι η HTML (HyperText Markup Language), γλώσσα σήμανσης-χαρακτηρισμού υπερκειμένου και εν μέρει η SQL (Structured Query Language) γλώσσα για τη διαχείριση δεδομένων, σε ένα Σύστημα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (RDBMS-Relational Database Management System).

2.2.3 Λοιπά πρότυπα και τεχνικές προγραμματισμού

Εκτός από τα παραπάνω υποδείγματα προγραμματισμού υπάρχουν και άλλα, τα οποία είτε δεν μπορούν να χαρακτηριστούν πλήρως ως προγραμματιστικά υποδείγματα είτε αποτελούν τεχνικές και μεθοδολογίες προγραμματισμού τις οποίες θα αναφέρουμε στη συνέχεια.

Παράλληλος προγραμματισμός (parallel programming). Επιτρέπει ταυτόχρονη εκτέλεση διαδικασιών από διαφορετικούς επεξεργαστές.

Προγραμματισμός οδηγούμενος από γεγονότα (event-driven programming). Αποτελεί περισσότερο τεχνική αρχιτεκτονικής ενός προγράμματος σχετικά με τη ροή του, παρά προγραμματιστικό υπόδειγμα. Η ροή του προγράμματος εξαρτάται από την ύπαρξη **Γεγονότων** (events), όπως είναι για παράδειγμα ένα μήνυμα ενός αισθητήρα ή μια ενέργεια του χρήστη με το πάτημα του ποντικιού ή ενός πλήκτρου. Παράδειγμα αποτελεί η Microsoft Visual-Basic.

Οπτικός προγραμματισμός (visual programming). Δεν αποτελεί υπόδειγμα, αλλά εκφράζει τη δυνατότητα γλωσσών ή περιβαλλόντων προγραμματισμού να παρέχουν τη δυνατότητα

δημιουργίας του προγράμματος μέσω γραφικών αντικειμένων, αντί της πληκτρολόγησης του κειμένου που αντιστοιχεί σε εντολές. Οι γλώσσες οπτικού προγραμματισμού βασίζονται, άλλες σε γραφικά με τη μορφή εικονιδίων (icon-based languages), άλλες σε διαγράμματα (diagram languages) και τέλος, άλλες σε φόρμες (form based languages).

Στην κατηγορία αυτή ανήκουν περιβάλλοντα όπως το Authorware της Adobe, περιβάλλοντα δημιουργίας σεναρίων όπως το Kodu της Microsoft και το Alice, το MIT Scratch, το Greenfoot.

Προγραμματισμός δέσμης ενεργειών (script programming) είναι τύπος - και όχι υπόδειγμα- προγραμματισμού δημιουργίας μικρών τμημάτων κώδικα και όχι ολοκληρωμένων προγραμμάτων. Είναι υψηλού επιπέδου προγραμματισμός που διερμηνεύεται κατά την εκτέλεση από ένα άλλο πρόγραμμα, όπως ένας φυλλομετρητής.

Αρθρωτός ή Τμηματικός Προγραμματισμός (modular programming). Σχετίζεται περισσότερο με τεχνική σχεδίασης λογισμικού παρά με πρότυπο. Χαρακτηρίζεται από τη διαίρεση του προβλήματος σε απλούστερα τμήματα, αυτά με τη σειρά τους σε επί μέρους μικρότερα κ.ο.κ. Παρέχει απλούστευση της επίλυσης ενός προβλήματος, ευκολία κωδικοποίησης και συντήρησης. Γενικά ως τμήμα (module) θεωρούμε ένα σύνολο ενεργειών το οποίο εκτελεί μια καθορισμένη λειτουργία ενός προγράμματος και είναι κατά το δυνατόν ανεξάρτητο από τα άλλα τμήματα.

Ιεραρχικός σχεδιασμός

Η μέθοδος ανάλυσης ενός προβλήματος σε μικρότερα είναι εκείνη με την οποία αντιμετωπίζουμε το πρόβλημα ως μια πολυεπίπεδη δομή. Έτσι, για τη σχεδίασή του, ξεκινάμε από το υψηλότερο επίπεδο και στη συνέχεια το αναλύουμε σε όλο και χαμηλότερα, έως ότου φθάσουμε στο κατώτερο επίπεδο ανάλυσης. Η τεχνική αυτή ονομάζεται **ιεραρχικός σχεδιασμός** (top down design).

2.3 Αντικειμενοστρεφής προγραμματισμός

Στην παράγραφο αυτή θα αναφερθούμε συνοπτικά στον Αντικειμενοστρεφή Προγραμματισμό (Object-oriented programming - OOP). Πρόκειται για είδος προγραμματισμού που περιστρέφεται γύρω από την έννοια της **Κλάσης** (Class), η οποία περιγράφει **Αντικείμενα** (Objects), τα οποία περιέχουν δεδομένα στη μορφή **Ιδιοτήτων** (Properties) και κώδικα στη μορφή **Μεθόδων** (Methods).

Παράδειγμα 1

Κλάση: Όχημα

Ιδιότητες: Χρώμα, Τιμή, Αριθμός Τροχών, Ταχύτητα

Μέθοδοι: Επιτάχυνε, Φρέναρε, Στρίψε

Ένα Αντικείμενο έχει συγκεκριμένες Ιδιότητες, όπως για παράδειγμα το Αντικείμενο “Αυτοκίνητο 1” με Ιδιότητες: Χρώμα: ‘Μαύρο’, Τιμή: 10000, Αριθμός τροχών: 4, Ταχύτητα: 120 km/h, ή το Αντικείμενο “Μοτοσικλέτα 2” με Ιδιότητες: Χρώμα: ‘Κόκκινο’, Τιμή: 4000, Αριθμός τροχών: 2, Ταχύτητα: 100 km/h. Έτσι, μια κλάση αποτελεί ένα αφηρημένο σχήμα το οποίο αποκτά συγκεκριμένη υπόσταση όταν δημιουργούνται αντικείμενα όπως το “Αυτοκίνητο 1” και η “Μοτοσικλέτα 2”, καθένα με τα δικά του χαρακτηριστικά. Οι μέθοδοι που περιγράφονται στην κλάση εμπεριέχουν κώδικα. Για παράδειγμα, οι μέθοδοι Επιτάχυνε και Φρέναρε, αυξάνουν και μειώνουν την ταχύτητα του αντικειμένου αντίστοιχα.

Παράδειγμα 2

Κλάση: Ζώο

Ιδιότητες: Ηλικία, Ύψος

Μέθοδοι: Μεγάλωσε, Ψήλωσε

Η μεταβλητή “Καμηλοπάρδαλη 1” είναι ένα αντικείμενο της Κλάσης Ζώο. Όταν το αντικείμενο δημιουργείται, οι ιδιότητες Ηλικία και Ύψος αρχικοποιούνται. Οι μέθοδοι της κλάσης ορίζουν τη συμπεριφορά των αντικειμένων και τον τρόπο με τον οποίο μεταβάλλεται η κατάστασή τους. Για παράδειγμα, οι μέθοδοι Μεγάλωσε και Ψήλωσε, αυξάνουν τις τιμές Ηλικία και Ύψος ενός αντικειμένου αντίστοιχα.