



A. EVANS, K. MARTIN, M. A. POATSY

Εισαγωγή στην
πληροφορική
Θεωρία και πράξη
3η έκδοση

Κεφάλαιο 10

Στο παρασκήνιο: Προγραμματισμός λογισμικού

Στόχοι (1 από 3)

- 10.1 Η σημασία του προγραμματισμού για τους προγραμματιστές και τους χρήστες λογισμικού.
- 10.2 Τα στάδια του κύκλου ζωής ανάπτυξης συστήματος (SDLC).
- 10.3 Προγραμματισμός και τα βήματα του κύκλου ζωής ανάπτυξης προγράμματος (PDLC).
- 10.4 Παραγωγή ολοκληρωμένης διατύπωσης προβλήματος από την περιγραφή μιας εργασίας.
- 10.5 Πώς χρησιμοποιούν οι προγραμματιστές τις μεθοδολογίες ελέγχου ροής και σχεδίασης κατά την ανάπτυξη αλγόριθμων.

Στόχοι (2 από 3)

- 10.6 Οι κατηγορίες των γλωσσών προγραμματισμού και οι ρόλοι του μεταγλωττιστή και του ολοκληρωμένου περιβάλλοντος προγραμματισμού (IDE) στην κωδικοποίηση.
- 10.7 Ο ρόλος της αποσφαλμάτωσης στην ανάπτυξη προγραμμάτων.
- 10.8 Η σημασία της δοκιμής και της τεκμηρίωσης στην ανάπτυξη προγραμμάτων.
- 10.9 Οι κινητήριες δυνάμεις πίσω από τη δημοτικότητα διάφορων γλωσσών προγραμματισμού.

Στόχοι (3 από 3)

- 10.10 Σύνοψη των προβλημάτων επιλογής της κατάλληλης γλώσσας προγραμματισμού για συγκεκριμένη εργασία.
- 10.11 Σύγκριση και εντοπισμός διαφορών μεταξύ των σύγχρονων γλωσσών προγραμματισμού.
- 10.12 Οι βασικές αρχές της ανάπτυξης των γλωσσών προγραμματισμού του μέλλοντος.

Τι είναι ο προγραμματισμός

- Κάποιες εργασίες είναι περίπλοκες
 - Απαιτούν δημιουργική σκέψη
 - Απαιτούν ανθρώπινη παρέμβαση
- Μερικές εργασίες είναι ιδανικές για αυτοματισμό
 - Επαναλαμβανόμενες
 - Βασίζονται σε ψηφιακές πληροφορίες
 - Ακολουθούν μια ξεκάθαρη σειρά βημάτων

Τα στάδια του κύκλου ζωής ανάπτυξης συστήματος - Η σημασία του προγραμματισμού

(Στόχος 10.1)

- Μια καριέρα στον προγραμματισμό προσφέρει:
 - Άφθονες εργασίες
 - Υψηλοί μισθοί
 - Τηλεργασία
- Ο προγραμματισμός είναι η μόνη λύση, για τους χρήστες που δεν μπορούν να βρουν υπάρχον προϊόν λογισμικού για μια εργασία.
- Οι μακροεντολές εκτελούν περίπλοκες ακολουθίες βημάτων με μία μόνο εντολή.



Κύκλος ζωής ανάπτυξης συστήματος - Τα στάδια του κύκλου ζωής ανάπτυξης συστήματος (1 από 3)

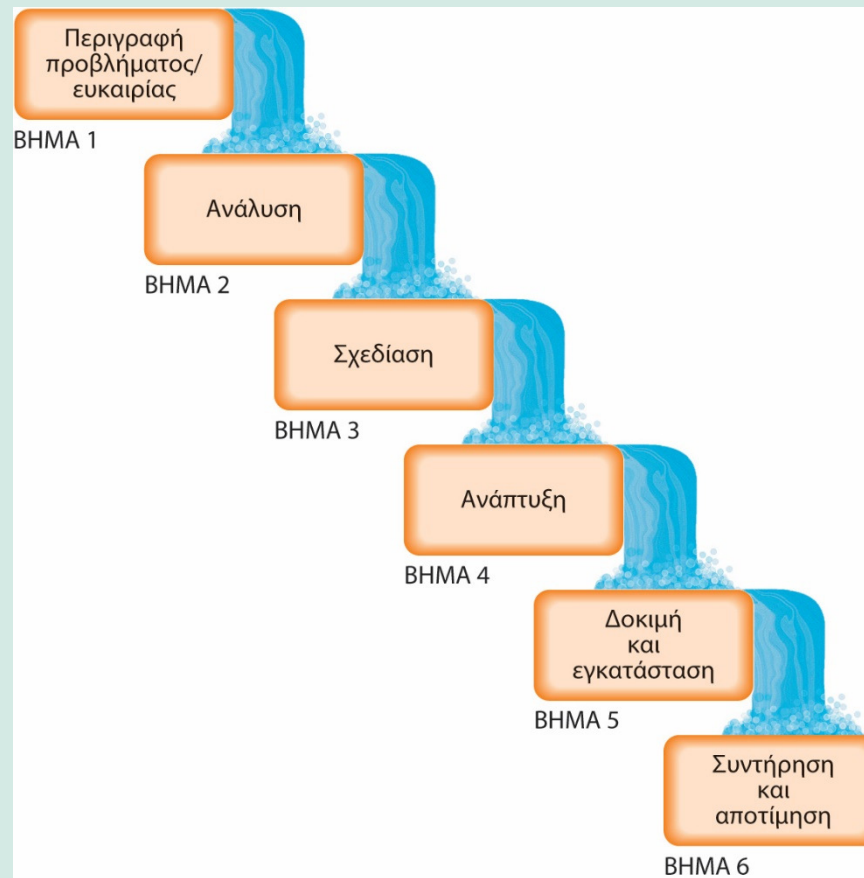
(Στόχος 10.2)

- Πληροφοριακό σύστημα: συλλογή μερών που συνεργάζονται μεταξύ τους προκειμένου να πετύχουν έναν κοινό στόχο
- Ένα πληροφοριακό σύστημα περιλαμβάνει:
 - Δεδομένα
 - Ανθρώπους
 - Διαδικασίες
 - Υλικό
 - Λογισμικό

Κύκλος ζωής ανάπτυξης συστήματος - Τα στάδια του κύκλου ζωής ανάπτυξης συστήματος (2 από 3)

(Στόχος 10.2)

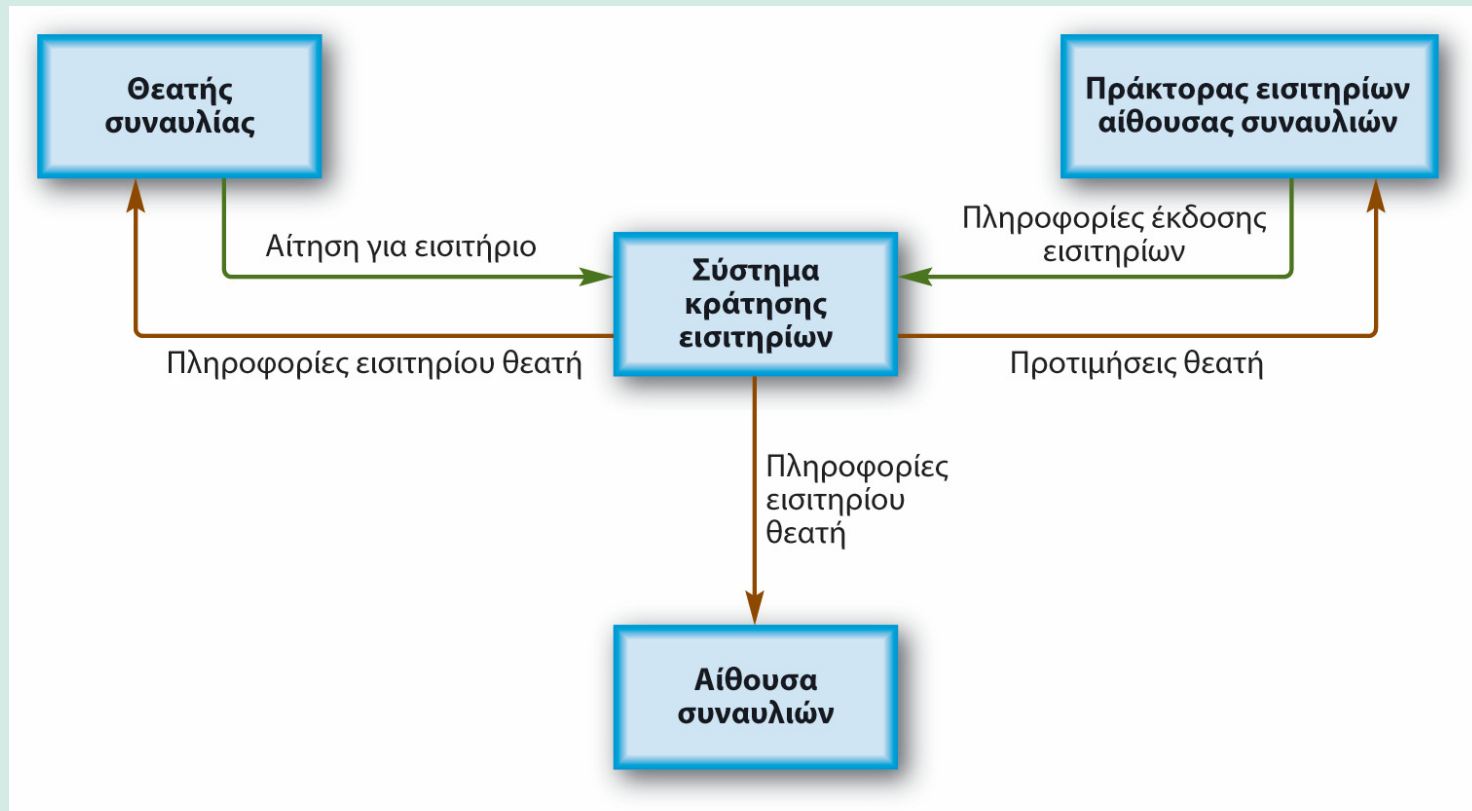
- Τα βήματα του SDLC



Κύκλος ζωής ανάπτυξης συστήματος - Τα στάδια του κύκλου ζωής ανάπτυξης συστήματος (3 από 3)

(Στόχος 10.2)

- Διάγραμμα ροής δεδομένων

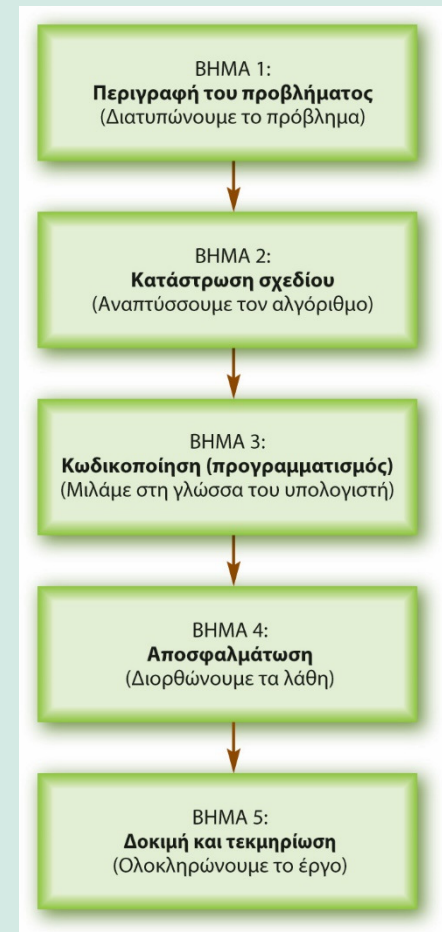


Κύκλος ζωής προγράμματος

Ο κύκλος ζωής ανάπτυξης προγράμματος

(Στόχος 10.3)

- Προγραμματισμός: διεργασία μετάφρασης μιας εργασίας σε μια σειρά από εντολές που θα χρησιμοποιήσει ο υπολογιστής για να εκτελέσει τη συγκεκριμένη εργασία
- Κύκλος ζωής ανάπτυξης προγράμματος (PDLC): τα στάδια που περιλαμβάνει η διεργασία του προγραμματισμού



Κύκλος ζωής προγράμματος

Η διατύπωση του προβλήματος (1 από 2)

(Στόχος 10.4)

- Διατύπωση του προβλήματος
 - Γίνονται κατανοητοί οι στόχοι κάθε προσπάθειας προγραμματισμού
- Αλληλεπίδραση με τους χρήστες
 - Τα δεδομένα είναι η ακατέργαστη πρώτη ύλη
 - Οι πληροφορίες είναι το αποτέλεσμα
 - Η μέθοδος είναι η διαδικασία της μετατροπής των εισόδων στις σωστές εξόδους

Κύκλος ζωής προγράμματος

Η διατύπωση του προβλήματος (2 από 2)

(Στόχος 10.4)

- Χειρισμός σφαλμάτων: ό,τι κάνει ένα πρόγραμμα όταν συμβαίνει κάποιο σφάλμα
- Σχέδιο δοκιμών
 - Προδιαγράφει συγκεκριμένες τιμές των εισόδων που οι προγραμματιστές περιμένουν από τους χρήστες
 - Παραθέτει τις ακριβείς τιμές εξόδων που θα επέστρεφε ένα τέλειο πρόγραμμα για τις αντίστοιχες τιμές των εισόδων

Ανάπτυξη αλγόριθμου

Ανάπτυξη αλγόριθμου (1 από 5)

(Στόχος 10.5)

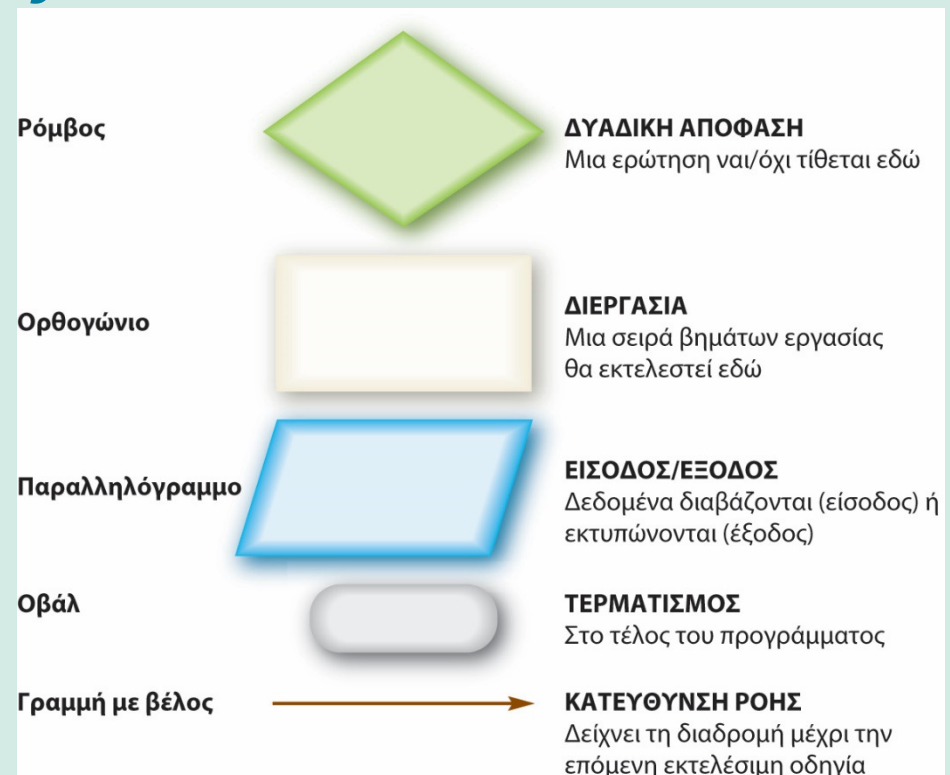
- Αλγόριθμος: σύνολο συγκεκριμένων διαδοχικών βημάτων που περιγράφουν ακριβώς τι πρέπει να κάνει το πρόγραμμα υπολογιστή για να ολοκληρώσει το έργο του
- Οι αλγόριθμοι αναπαριστώνται με διαγράμματα ροής
 - Παρέχουν οπτική αναπαράσταση μοτίβων
- Ο ψευδοκώδικας είναι μια προσέγγιση για την τεκμηρίωση ενός αλγόριθμου που βασίζεται σε κείμενο

Ανάπτυξη αλγόριθμου

Ανάπτυξη αλγόριθμου (2 από 5)

(Στόχος 10.5)

- Τα διαγράμματα ροής παρέχουν οπτική αναπαράσταση των μοτίβων από τα οποία αποτελείται ο αλγόριθμος



Κύκλος ζωής προγράμματος

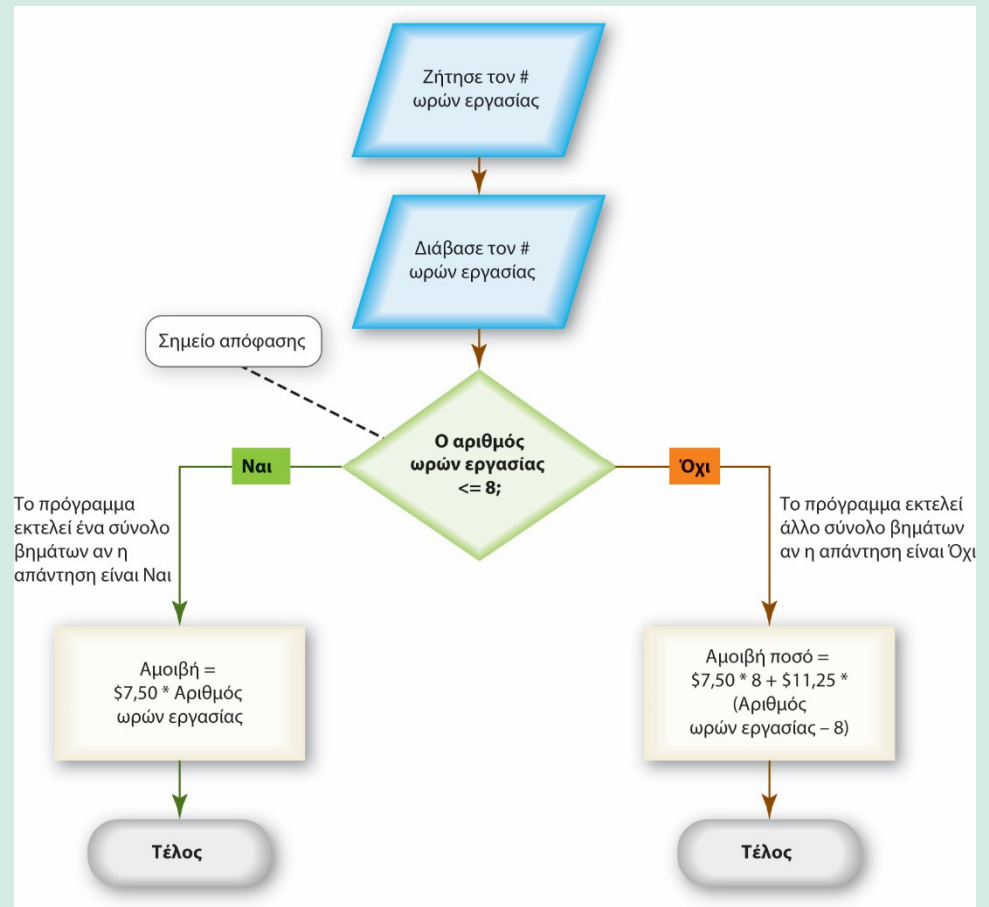
Ανάπτυξη αλγόριθμου (3 από 5)

(Στόχος 10.5)

- Ροή ελέγχου

- Σημεία αποφάσεων

- Δυαδικές αποφάσεις
 - Βρόχοι

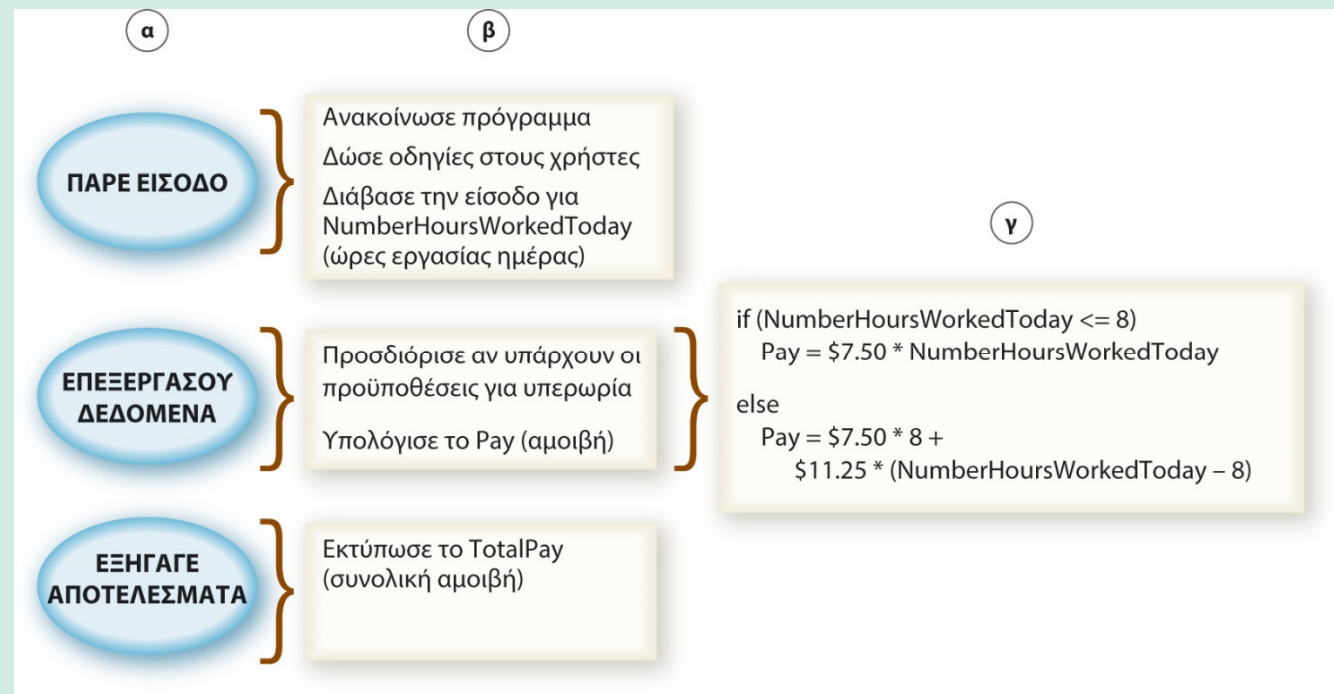


Κύκλος ζωής προγράμματος

Ανάπτυξη αλγόριθμου (4 από 5)

(Στόχος 10.5)

- Σχεδίαση από πάνω προς τα κάτω: προσέγγιση κατά την οποία ένα πρόβλημα αναλύεται σε μια σειρά από εργασίες υψηλού επιπέδου

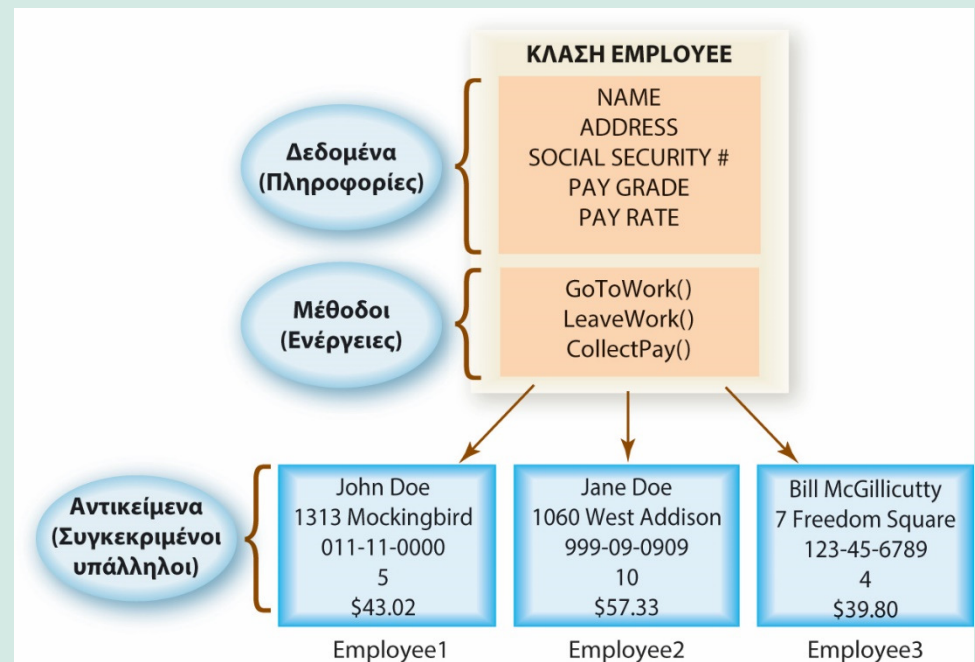


Κύκλος ζωής προγράμματος

Ανάπτυξη αλγόριθμου (5 από 5)

(Στόχος 10.5)

- Αντικειμενοστρεφής ανάλυση: οι προγραμματιστές αναγνωρίζουν πρώτα όλες τις κατηγορίες εισόδων για να λύσουν ένα πρόβλημα
- Επαναχρησιμοποίηση
- Κληρονομικότητα
- Κλάσεις
 - Βασική κλάση
 - Παραγόμενη κλάση



Κύκλος ζωής προγράμματος

Κωδικοποίηση (1 από 7)

(Στόχος 10.6)

- Κωδικοποίηση: μετάφραση ενός αλγόριθμου σε γλώσσα προγραμματισμού
- Οι προγραμματιστές μεταφέρονται από τον αλγόριθμο στον κώδικα:
 - Αναγνωρίζοντας τις απαραίτητες πληροφορίες
 - Αναγνωρίζοντας τη ροή από κάθε βήμα
 - Μετατρέποντας τον αλγόριθμο σε μια συγκεκριμένη γλώσσα προγραμματισμού

Κύκλος ζωής προγράμματος

Κωδικοποίηση (2 από 7)

(Στόχος 10.6)

- Γλώσσα προγραμματισμού
 - Ένα είδος «κώδικα» για το σύνολο των εντολών που η CPU γνωρίζει πώς να εκτελεί
 - Οι γλώσσες προγραμματισμού των υπολογιστών χρησιμοποιούν ειδικές λέξεις και αυστηρούς κανόνες
 - Επιτρέπουν τον έλεγχο της CPU χωρίς να χρειάζεται να γνωρίζουν όλες τις λεπτομέρειες της κατασκευής της

Κύκλος ζωής προγράμματος

Κωδικοποίηση (3 από 7)

(Στόχος 10.6)

Δείγματα κώδικα για διαφορετικές γενιές γλωσσών

Γενιά	Παράδειγμα	Δείγμα κώδικα
1GL	Μηχανής	Τα bit περιγράφουν τις εντολές προς τη CPU. 1110 0101 1001 1111 0000 1011 1110 0110
2GL	Συμβολοσειράς	Οι λέξεις περιγράφουν τις εντολές προς τη CPU. ADD Register 3, Register 4, Register 5
3GL	FORTRAN, BASIC, C/C++, Java	Τα σύμβολα περιγράφουν τις εντολές προς τη CPU. TotalPay = Pay + OvertimePay
4GL	SQL	Πιο ισχυρές εντολές επιτρέπουν την εκτέλεση περίπλοκων εργασιών σε μία μόνο πρόταση. SELECT isbn, title, price, price*0.06 AS sales_tax FROM books WHERE price>100.00 ORDER BY title;
5GL	Prolog	Οι προγραμματιστές μπορούν να κατασκευάζουν εφαρμογές χωρίς να δίνουν αλγόριθμο . Find all the people who are Mike's cousins as: ?-cousin (Mike, family)

Κύκλος ζωής προγράμματος

Κωδικοποίηση (4 από 7)

(Στόχος 10.6)

- Μεταφερισιμότητα: δυνατότητα μεταφοράς μιας ολοκληρωμένης λύσης εύκολα από έναν τύπο υπολογιστή σε άλλο
- Μεταβλητές: κάθε είσοδος και κάθε έξοδος που χειρίζεται το πρόγραμμα
 - Μια δήλωση μεταβλητής λέει στο λειτουργικό σύστημα ότι το πρόγραμμα πρέπει να δεσμεύσει χώρο αποθήκευσης στη RAM
- Σχόλια εισάγονται προκειμένου να εξηγηθεί ο σκοπός ενός αποσπάσματος κώδικα

Κύκλος ζωής προγράμματος Κωδικοποίηση (5 από 7)

(Στόχος 10.6)

The diagram illustrates the mapping of C++ code to the program life cycle steps. The code is as follows:

```
#include <iostream>
using namespace std;

void
main()
{
    //Begin by asking for some variables to be stored in RAM

    float NumberHoursWorkedToday = 0.0;
    float Pay=0.0, TotalPay=0.0;
    int Day;

    //Set up a loop to ask the user how many hours they worked each day
    for( Day = 1; Day <= 7; Day++)
    {
        //Read the input data from the screen
        cout << " Enter the number of hours you worked on day " << Day << " : ";
        cin >> NumberHoursWorkedToday;

        //Check the input makes sense
        if ( NumberHoursWorkedToday < 0 )
        {
            //Wait a minute! We need to handle possible errors here.
            //Print a warning message to the user if they enter a negative value.

            cout << "You can't work negative hours! Try again : ";
            cin >> NumberHoursWorkedToday;
        }

        if ( NumberHoursWorkedToday <= 8 )
        {
            //Compute pay earned today at normal base rate
            Pay = NumberHoursWorkedToday * 7.50;
        }
        else
        {
            //Compute the pay earned using the overtime rule
            Pay = ( 8 * 7.50 ) + ( 11.25 * ( NumberHoursWorkedToday - 8 ) );
        }

        // Update the total pay you have earned so far this week
        TotalPay = TotalPay + Pay;
    } // when we hit this brace, we bounce back up to the beginning of the loop

    //Now we're free from the loop! Let's print then go spend our paycheck!
    cout << " Your totalpay for this week comes to : " << TotalPay;

    // The program is done . |
}
```

The steps and their corresponding code sections are:

- ΒΗΜΑ 1:** Δήλωση μεταβλητών (Declaration of variables) - `float NumberHoursWorkedToday = 0.0; float Pay=0.0, TotalPay=0.0; int Day;`
- ΒΗΜΑ 2:** Δημιουργία βρόχου (Loop creation) - `for(Day = 1; Day <= 7; Day++)`
- ΒΗΜΑ 3:** Συλλογή δεδομένων εισόδου (Input data collection) - `cout << " Enter the number of hours you worked on day " << Day << " : "; cin >> NumberHoursWorkedToday;`
- ΒΗΜΑ 4:** Έλεγχος για σφάλματα (Error checking) - `if (NumberHoursWorkedToday < 0)`
- ΒΗΜΑ 5:** Επεξεργασία δεδομένων (Data processing) - `if (NumberHoursWorkedToday <= 8)`
- ΒΗΜΑ 6:** Ενημέρωση μεταβλητών (Variable update) - `TotalPay = TotalPay + Pay;`

Additional annotations:

- Σχόλια (πράσινο):** Comments in the code.
- Λέξεις-κλειδιά (μπλε):** Keywords in the code.
- Εντολές (μαύρο):** Statements in the code.
- Βρόχος:** The loop structure.

Κύκλος ζωής προγράμματος

Κωδικοποίηση (6 από 7)

(Στόχος 10.6)

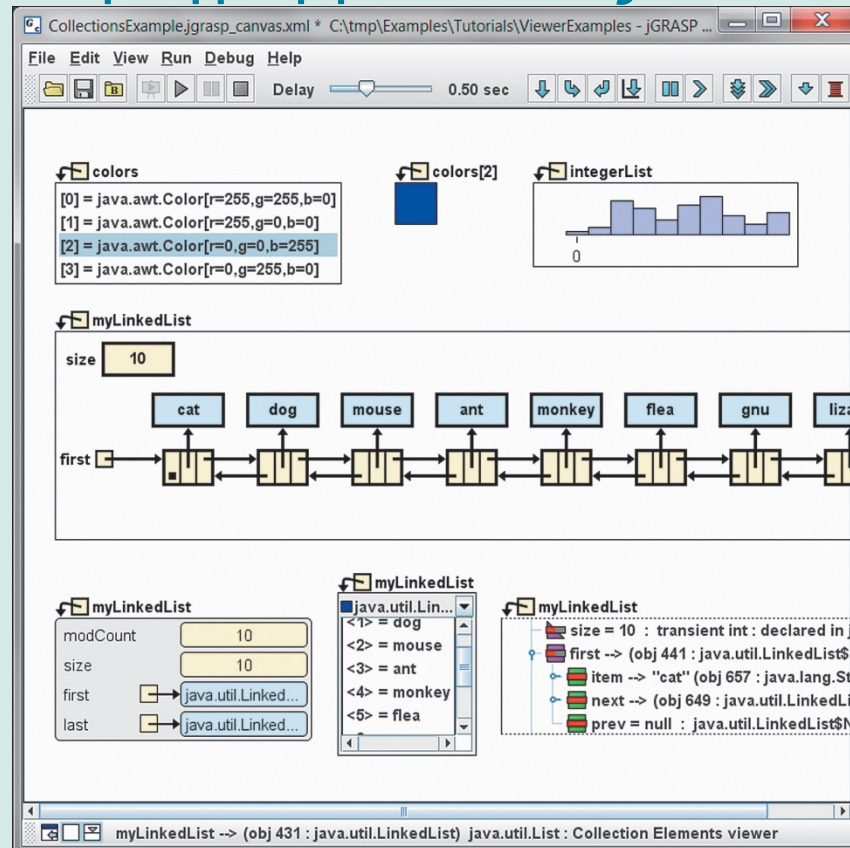
- Μεταγλώττιση: διαδικασία κατά την οποία ο κώδικας μετατρέπεται σε γλώσσα μηχανής
- Ο μεταγλωττιστής καταλαβαίνει τη σύνταξη της γλώσσας προγραμματισμού και τη δομή της CPU
 - Διαβάζει τον πηγαίο κώδικα
 - Δημιουργεί ένα εκτελέσιμο πρόγραμμα
- Κάποιες γλώσσες προγραμματισμού χρησιμοποιούν διερμηνευτή
 - Μεταφράζει τον πηγαίο κώδικα σε μια ενδιάμεση μορφή

Κύκλος ζωής προγράμματος

Κωδικοποίηση (7 από 7)

(Στόχος 10.6)

- IDE: βοηθά τους προγραμματιστές να γράφουν και να δοκιμάζουν τα προγράμματά τους



Κύκλος ζωής προγράμματος

Αποσφαλμάτωση

(Στόχος 10.7)

- Αποσφαλμάτωση: διεργασία διόρθωσης σφαλμάτων σε ένα πρόγραμμα
- Με το πλάνο δοκιμών, γνωρίζουν οι προγραμματιστές ότι το πρόγραμμα έλυσε το πρόβλημα
- Λογικά σφάλματα
 - Εντοπίζονται μόνο όταν το πρόγραμμα εκτελείται
- Σφάλματα εκτέλεσης
 - Διαίρεση με το μηδέν
- Πρόγραμμα αποσφαλμάτωσης

Κύκλος ζωής προγράμματος

Δοκιμή και τεκμηρίωση

(Στόχος 10.8)

- Εσωτερική δοκιμή
- Εξωτερική δοκιμή
- Διορθώσεις μετά από δοκιμές σε έκδοση beta
- Ολοκλήρωση έργου
- Έκδοση για κατασκευαστές(RTM)
- Τεκμηρίωση
- Γενική διαθεσιμότητα (GA)

Πολλές γλώσσες προγραμματισμού

Ανάγκη για διαφορετικές γλώσσες

(Στόχος 10.9)

- Εξετάστε τις αγγελίες προσφοράς εργασίας για προγραμματιστές
- Σε συγκεκριμένους κλάδους, τείνουν να κυριαρχούν συγκεκριμένες γλώσσες
- Ένα μάθημα εισαγωγής στον προγραμματισμό περιλαμβάνει
 - Σχεδίαση
 - Ανάπτυξη αλγόριθμων
 - Τεχνικές αποσφαλμάτωσης
 - Διαχείριση έργου



Πολλές γλώσσες προγραμματισμού

Επιλογή της κατάλληλης γλώσσας

(Στόχος 10.10)

- Πρέπει να λαμβάνονται υπόψη:
 - Διαθέσιμος χώρος
 - Απαιτούμενη ταχύτητα
 - Διαθέσιμοι οργανωτικοί πόροι
 - Τύπος εφαρμογής
- Γλώσσες οπτικού προγραμματισμού
 - Scratch
 - App Inventor

Εξερεύνηση γλωσσών προγραμματισμού

Μια ξενάγηση στις σύγχρονες γλώσσες (1 από 4)

(Στόχος 10.11)

- Visual Basic

- Πρωτότυπο
- Ταχεία ανάπτυξη εφαρμογών
- Χρησιμοποιήθηκε για την κατασκευή πολλών εφαρμογών
- .Net Framework
- Διεπαφή προγραμματισμού εφαρμογών

- C and C++

- Εκτέλεση δύσκολων και επαναλαμβανόμενων αριθμητικών πράξεων

Εξερεύνηση γλωσσών προγραμματισμού

Μια ξενάγηση στις σύγχρονες γλώσσες (2 από 4)

(Στόχος 10.11)

- Java and C#

- Συλλέγουν πληροφορίες από δικτυωμένους υπολογιστές
- Αρχιτεκτονικά ουδέτερες
- Η C# ανταγωνίζεται την Java

- Objective C και Swift

- Χρησιμοποιείται για τον προγραμματισμό εφαρμογών για το macOS
- Αντικειμενοστρεφείς
- Η Apple παρουσίασε τη Swift ως γλώσσα προγραμματισμού στα iOS και macOS

Εξερεύνηση γλωσσών προγραμματισμού

Μια ξενάγηση στις σύγχρονες γλώσσες (3 από 4)

(Στόχος 10.11)

- Γλώσσα σημείωσης υπερκειμένου (HTML)
 - Η βασικότερη γλώσσα μορφοποίησης για εφαρμογές web
- Γλώσσες σεναρίου
 - Περιορίζονται στην εκτέλεση εξειδικευμένων εργασιών
 - Παραδείγματα: JavaScript and PHP
- ASP, JSP και PHP
 - Χρησιμοποιούνται στην κατασκευή διαδικτυακών τόπων με διαδραστικές δυνατότητες

Εξερεύνηση γλωσσών προγραμματισμού

Μια ξενάγηση στις σύγχρονες γλώσσες (4 από 4)

(Στόχος 10.11)

- AJAX, XML και JSON
 - Δημιουργούν εφαρμογές web που ενημερώνονται από μόνες τους και επικοινωνούν με άλλους υπολογιστές
- Ανάπτυξη εφαρμογών για κινητά
 - Ειδικές γλώσσες και εργαλεία υποστήριξης

Εξερεύνηση γλωσσών προγραμματισμού

Το μέλλον των γλωσσών προγραμματισμού

(Στόχος 10.12)

- Δεν είναι εύκολο να το προβλέψει κανείς
- Γλώσσες οπτικού προγραμματισμού
- Πάντα θα χρειάζεται λίγη ποικιλία



Απαγορεύεται η αναδημοσίευση ή αναπαραγωγή του παρόντος έργου με οποιονδήποτε τρόπο χωρίς γραπτή άδεια του εκδότη, σύμφωνα με το Ν. 2121/1993 και τη Διεθνή Σύμβαση της Βέρνης (που έχει κυρωθεί με τον Ν. 100/1975)