



Ελληνική Δημοκρατία  
Τεχνολογικό Εκπαιδευτικό  
Ίδρυμα Ηπείρου

# Προγραμματισμός I

Ενότητα 1 : Εισαγωγικά Στοιχεία της Pascal

Αλέξανδρος Τζάλλας



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Τμήμα Μηχανικών Πληροφορικής Τ.Ε

## Προγραμματισμός Ι

Ενότητα 1 : Εισαγωγικά Στοιχεία της Pascal

Αλέξανδρος Τζάλλας

Λέκτορας

Άρτα, 2015





# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.





# Χρηματοδότηση

- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «**Εκπαίδευση και Δια Βίου Μάθηση**» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Ηπείρου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



# Σκοποί ενότητας

- Να γίνει μια εισαγωγή στις έννοιες του δομημένου προγραμματισμού
- Να αναλυθούν οι βασικές δομικές μονάδες της γλώσσας Pascal



# Περιεχόμενα ενότητας

- Αρχές Ανάπτυξης Προγραμμάτων
- Είδη Προγραμματισμού
- Είδη Προγραμματισμού
- Διαγράμματα ροής
- Σύνολο χαρακτήρων της Pascal
- Λεξιλόγιο-Τελεστές
- Δηλώσεις Μεταβλητών
- Εκφράσεις/Συναρτήσεις στην Pascal



# Πως να μάθουμε να προγραμματίζουμε;

- Δυστυχώς ο προγραμματισμός δεν διδάσκεται
- Δεν μπορούμε να μάθουμε να προγραμματίζουμε διαβάζοντας βιβλία για προγραμματισμό
- Δεν μπορούμε να μάθουμε να προγραμματίζουμε γράφοντας προγράμματα στο χαρτί
- Ο μόνος τρόπος να μάθουμε να προγραμματίζουμε είναι να γράφουμε προγράμματα στον υπολογιστή, να τα εκτελούμε, να βρίσκουμε τα λάθη που σίγουρα έχουμε κάνει, να τα διορθώνουμε, να εκτελούμε πάλι τα προγράμματα, να ξαναβρίσκουμε λάθη, να τα διορθώνουμε πάλι, μέχρι να καταφέρουμε να κάνουμε τα προγράμματά μας να επιλύουν τα προβλήματά μας όπως πρέπει
- Επικουρικά, η ανάγνωση και κατανόηση καλογραμμένων, τεκμηριωμένων και ορθών προγραμμάτων είναι πάντοτε χρήσιμη, αλλά από μόνη της δεν αρκεί



# Προγραμματισμός

- Ο Η/Υ χρησιμοποιείται για την επεξεργασία δεδομένων
- Για να γίνει η επεξεργασία αυτή πρέπει να του δοθούν συγκεκριμένες οδηγίες από τον άνθρωπο
- Το σύνολο των οδηγιών προς τον υπολογιστή για την επίλυση ενός προβλήματος αποτελεί το πρόγραμμα





# Αλγόριθμος

- Πριν δημιουργήσουμε ένα πρόγραμμα για να λύσουμε ένα πρόβλημα πρέπει να σχεδιάσουμε έναν αλγόριθμο για αυτό το πρόβλημα
- Μια σαφώς καθορισμένη ακολουθία εντολών ή κανόνων που οδηγεί σε ένα προκαθορισμένο αποτέλεσμα (λύση ενός προβλήματος) μεσώ ενός πεπερασμένου πλήθους βημάτων
- Έάν πρόγραμμα είναι η διατύπωση ενός αλγορίθμου σε μια συγκεκριμένη γλώσσα προγραμματισμού
- Το πρόγραμμα είναι η υλοποίηση του αλγορίθμου



# Παράδειγμα

## Μέγιστος Κοινός Διαιρέτης

Να βρεθεί ο **ΜΚΔ** δύο αριθμών  
(Αλγόριθμος Ευκλείδη)

**Βήμα 1** : Δώσε δύο αριθμούς A & B

**Βήμα 2**: Βρες το μεγαλύτερο & αντικατέστησε τον με το υπόλοιπο της διαίρεσης αυτού με τον άλλο αριθμό

**Βήμα 3**: Αν το υπόλοιπο είναι μηδέν τότε πήγαινε στο **Βήμα 4**, αλλιώς στο **Βήμα 2**

**Βήμα 4**: Τύπωσε **ΜΚΔ**

**Βήμα 5**: Τέλος

Αν  $A=25$  &  $B=20$  έχουμε:

1. Αν  $A=25$ ,  $B=20$
2.  $A=5$ ,  $B=20$  (Αντικατάσταση A με το υπόλοιπο της διαίρεσης  $25:20$ )
3.  $A=5$ ,  $B=0$  (Αντικατάσταση του B με το υπόλοιπο της διαίρεσης  $20:5$ )
4. **ΜΚΔ=5**
5. Τέλος



# Αρχές Ανάπτυξης Προγραμμάτων<sub>1/2</sub>

- **Αλγόριθμος**
  - Ο προγραμματιστής πρέπει να διατυπώσει τη λύση του προβλήματος σε γενικές γραμμές
- **Ψευδοκώδικας / Διάγραμμα ροής**
  - Η περιγραφή γίνεται χρησιμοποιώντας ψευδοκώδικα ή διαγράμματα ροής
  - Έλεγχος της λογικής της προτεινομένης λύσης χρησιμοποιώντας καταλληλά δεδομένα
- **Κωδικοποίηση σε γλώσσα υψηλού επιπέδου**
  - Η ακολουθία των λειτουργιών του ψευδοκώδικα ή του διαγράμματος ροής κωδικοποιείται σε γλώσσα προγραμματισμού



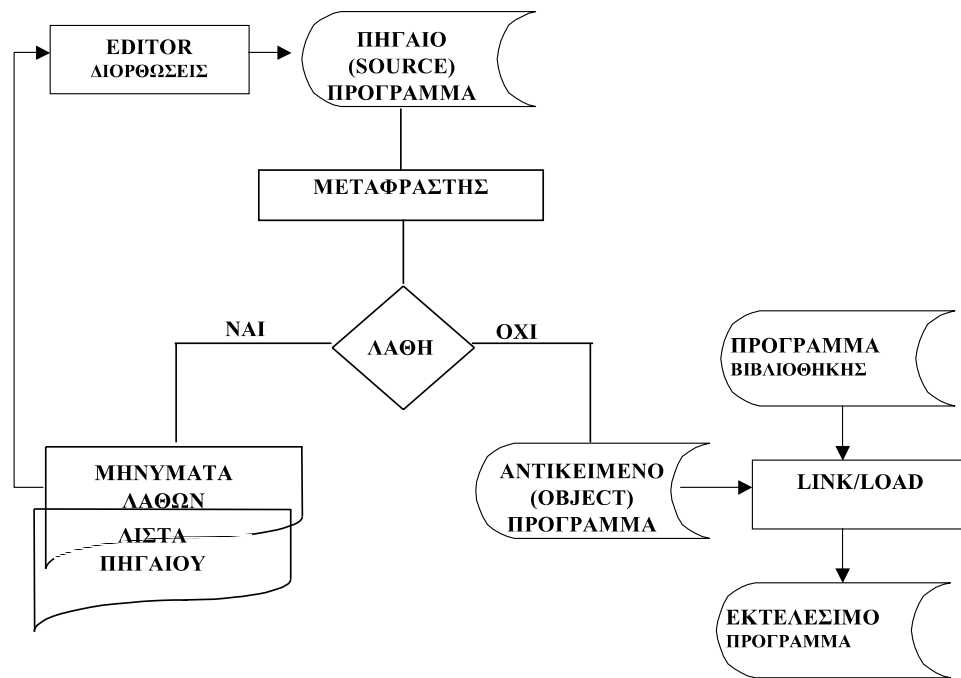
# Αρχές Ανάπτυξης Προγραμμάτων<sub>2/2</sub>

- **Μεταγλώττιση - Σύνδεση**
  - ✓ Ο μεταφραστής ελέγχει το πηγαίο πρόγραμμα για γραμματικά λάθη (ορθογραφία εντολών) και συντακτικά λάθη (εντολές που λείπουν)
  - ✓ Το πηγαίο πρόγραμμα μετατρέπεται σε γλώσσα μηχανής
- **Διορθώσεις**
  - ✓ Τα γραμματικά και συντακτικά λάθη διορθώνονται από τον προγραμματιστή και επαναλαμβάνεται η μεταγλώττιση
  - ✓ Διαδοχική απομάκρυνση των σφαλμάτων (debugging)
- **Δοκιμή**
  - ✓ Λάθη λογικής που παραμένουν στο πρόγραμμα μπορούν να βρεθούν εξετάζοντας την έξοδο
  - ✓ Εκτός από συνηθισμένα δεδομένα, δίνονται σκόπιμα λάθη για να διαπιστωθεί αν θα εντοπιστούν



# Υλοποίηση Προγραμμάτων

Ολοκληρωμένο  
(γραφικό)  
περιβάλλον για την  
γραφή –  
μεταγλώττιση –  
σύνδεση και  
εκτέλεση ενός  
προγράμματος





# Περιγραφή Αλγορίθμων

- Ως εργαλείο επικοινωνίας, ένας αλγόριθμος πρέπει:
  - να έχει συγκεκριμένη δομή
  - να εκφράζεται σε μορφή κατανοητή από όλους
  - να εκφράζεται με καθορισμένη σύνταξη ώστε να μην είναι διφορούμενη η ερμηνεία του
- Διακρίνουμε δυο μεγάλες κατηγορίες εργαλείων περιγραφής αλγορίθμων:
  - εργαλεία που βασίζονται στη γλώσσα (ψευδοκώδικας)
  - εργαλεία που βασίζονται σε σχήματα (διαγράμματα ροής)



# Ψευδοκώδικας

- Ο ψευδοκώδικας είναι κάτι που μοιάζει με κώδικα προγραμματισμού, αλλά δεν είναι.
- Ο ψευδοκώδικας είναι μια δομημένη γλώσσα που χρησιμοποιεί:
  - στοιχεία (συντακτικές δομές) από τις γλώσσες προγραμματισμού
  - συμβολισμούς (σημασιολογικά στοιχεία) από τα Μαθηματικά και τη Μαθηματική Λογική
  - στοιχεία (λεκτικές περιγραφές) από τη φυσική γλώσσα



# Χαρακτηριστικά Ψευδοκώδικα

- Ο ψευδοκώδικας χρησιμοποιείται για την ανάπτυξη ενός εργαλείου περιγραφής αλγορίθμων.
- Να συνδυάζει ταυτόχρονα την αυστηρότητα και την ακρίβεια που έχουν οι γλώσσες προγραμματισμού με την εκφραστική δύναμη της φυσικής γλώσσα.
- Ο ψευδοκώδικας είναι ανεξάρτητος από οποιαδήποτε γλώσσα προγραμματισμού.
- Ένας αλγόριθμος που περιγράφεται με ψευδοκώδικας μπορεί άμεσα να «μεταφραστεί» σε ένα πρόγραμμα.
- Γίνεται κατανοητός τόσο από προγραμματιστές όσο και από μη προγραμματιστές.





# Δομές Ελέγχου

- Μια απλή δομημένη γλώσσα μπορεί να περιέχει τρεις δομές ελέγχου:
  - ακολουθία
  - απόφαση
  - επανάληψη



# Γλώσσα Ψευδοκώδικα<sub>1/3</sub>

Είδος Εντολής	Συνοπτική Έκφραση	Λειτουργία Εντολής
<b>Εισόδου</b>	<b>Διάβασε</b>	Εισαγωγή δεδομένων από το πληκτρολόγιο ή μαγνητική μονάδα εισόδου
<b>Εξόδου</b>	<b>Εμφάνισε</b>	Εμφάνιση δεδομένων ή αποτελεσμάτων επεξεργασίας στην οθόνη
	<b>Τύπωσε</b>	Εκτύπωση δεδομένων ή αποτελεσμάτων επεξεργασίας στον εκτυπωτή
	<b>Γράψε</b>	Εγγραφή δεδομένων ή αποτελεσμάτων επεξεργασίας σε μαγνητική μονάδα εξόδου
<b>Αντικατάστασης</b>	<b>Αντικατέστησε ή βάλε</b>	Το περιεχόμενο μιας περιοχής μνήμης αντικαθίσταται με το περιεχόμενο μιας άλλης περιοχής
<b>Αριθμητικών πράξεων</b>	<b>Πρόσθεσε, αφάιρεσε, πολλαπλασίασε, διαίρεσε, υπολόγισε</b>	Εκτελούνται αριθμητικές πράξεις ή παραστάσεις



# Γλώσσα Ψευδοκώδικα<sub>2/3</sub>

Είδος Εντολής	Συνοπτική Έκφραση	Λειτουργία Εντολής
<b>Ελέγχου και επιλογής</b>	<b>Αν</b> η συνθήκη αληθεύει <b>Τότε</b> εκτέλεσε την εντολή 1,2,... <b>Αλλιώς</b> εκτέλεσε την εντολή 10, 20, ...	Γίνεται έλεγχος μιας συνθήκης με διπλή επιλογή ενέργειας ανάλογα με το αν η συνθήκη αληθεύει ή όχι
<b>Δηλώσεις ορίων</b>	<b>Αρχή, Τέλος</b>	Λέξεις που δηλώνουν την αρχή και το τέλος μιας ομάδας εντολών
<b>Εντολές επανάληψης</b>	<b>Επανάλαβε εφόσον</b> η συνθήκη είναι αληθής αρχή εντολή 1 ... εντολή n τέλος	Αρχίζει και συνεχίζει την εκτέλεση των εντολών που επαναλαμβάνονται μεταξύ <b>αρχή</b> και <b>τέλος</b> εφόσον η συνθήκη είναι αληθής. Διαφορετικά δεν αρχίζει καθόλου ή σταματά η εκτέλεση των εντολών επανάληψης όταν η συνθήκη γίνει ψευδής
	<b>Επανάλαβε</b> εντολή 1 εντολή 2 .... εντολή n <b>μέχρι</b> η συνθήκη να γίνει αληθής	Οι εντολές που περιλαμβάνονται μεταξύ των <b>επανάλαβε-μέχρι</b> εκτελούνται όσο η συνθήκη είναι ψευδής. Όταν η συνθήκη γίνει αληθής σταματά η εκτέλεση των εντολών



# Γλώσσα Ψευδοκώδικα<sub>3/3</sub>

Είδος Εντολής	Συνοπτική Έκφραση	Λειτουργία Εντολής
<b>Εντολές επανάληψης</b>	<b>Για n φορές επανέλαβε</b> <b>αρχή</b> εντολή 1 εντολή 2 ... εντολή n <b>τέλος</b>	Εκτελεί τις εντολές που περιλαμβάνονται μεταξύ των <b>αρχή ... τέλος</b> για προκαθορισμένο αριθμό επαναλήψεων
<b>Κλήση υποπρογράμματος</b>	<b>Κάλεσε όνομα υποπρογράμματος</b>	Εκτελεί τις εντολές που περιέχονται στο υποπρόγραμμα που έχει περιγραφεί αναλυτικά σε άλλο σημείο του προγράμματός μας ή της βιβλιοθήκης υποπρογραμμάτων της γλώσσας προγραμματισμού που χρησιμοποιούμε



# Παράδειγμα 1

---

## ΑΛΓΟΡΙΘΜΟΣ MAX-XY

---

### ΔΕΔΟΜΕΝΑ

X,Y,MAX: REAL;

### ΑΡΧΗ

ΔΙΑΒΑΣΕ(X,Y);

ΕΑΝ (X > Y) ΤΟΤΕ

MAX:=X

### ΑΛΛΙΩΣ

MAX:=Y

ΕΑΝ-ΤΕΛΟΣ;

ΤΥΠΩΣΕ(MAX)

### ΤΕΛΟΣ

---



# Παράδειγμα 2

---

ΑΛΓΟΡΙΘΜΟΣ MAX-XYZ

---

ΔΕΔΟΜΕΝΑ

X,Y,Z: REAL;

ΑΡΧΗ

ΔΙΑΒΑΣΕ(X,Y,Z);

ΕΑΝ (X > Y) ΤΟΤΕ

ΕΑΝ (X > Z) ΤΟΤΕ

ΤΥΠΩΣΕ(X)

ΑΛΛΙΩΣ

ΤΥΠΩΣΕ(Z)

ΕΑΝ-ΤΕΛΟΣ

ΑΛΛΙΩΣ

ΕΑΝ (Y > Z) ΤΟΤΕ

ΤΥΠΩΣΕ(Y)

ΑΛΛΙΩΣ

ΤΥΠΩΣΕ(Z)

ΕΑΝ-ΤΕΛΟΣ

ΕΑΝ-ΤΕΛΟΣ

ΤΕΛΟΣ

---



# Παράδειγμα 3

---

## ΑΛΓΟΡΙΘΜΟΣ ΜΚΔ

---

### ΔΕΔΟΜΕΝΑ

X, Y, Z, MKD: INTEGER;

### ΑΡΧΗ

ΔΙΑΒΑΣΕ (X,Y) ;

Z := X MOD Y ;

ΕΑΝ (Z <> 0) ΤΟΤΕ

    X := Y ;

    Y := Z

### ΑΛΛΙΩΣ

    MKD := Y

ΕΑΝ-ΤΕΛΟΣ

### ΤΕΛΟΣ

---



# Διαγράμματα Ροής (Law Chat)<sub>1/2</sub>

- Επιδεικνύουν την ακολουθία των βημάτων που περιγράφει ο αλγόριθμος με ένα διαγραμματικό τρόπο
- Δείχνει την γενική ακολουθία των γεγονότων με την οποία μπορεί να λυθεί το πρόβλημα
- Δίνουν μια πολύ καλή προσέγγιση για το σχεδιασμό προγραμμάτων
  - Απευθύνονται στους προγραμματιστές
- Δεν θεωρούνται ο καλύτερος τρόπος για την τεκμηρίωση των προγραμμάτων
  - Δεν απευθύνονται στους χρήστες





# Διαγράμματα Ροής (Law Chat)<sub>2/2</sub>

- Χρησιμοποιούνται διάφορα γεωμετρικά σχήματα με ορισμένη σημασία
- Τα σχήματα συνδέονται με συνεχείς γραμμές και χρησιμοποιούνται τόξα για να δείξουν την ροή των γεγονότων
- Ένα διάγραμμα ροής δεν είναι ένα συμπληρωμένο πρόγραμμα, αλλά αποτελεί μια βοήθεια για τον προγραμματιστή

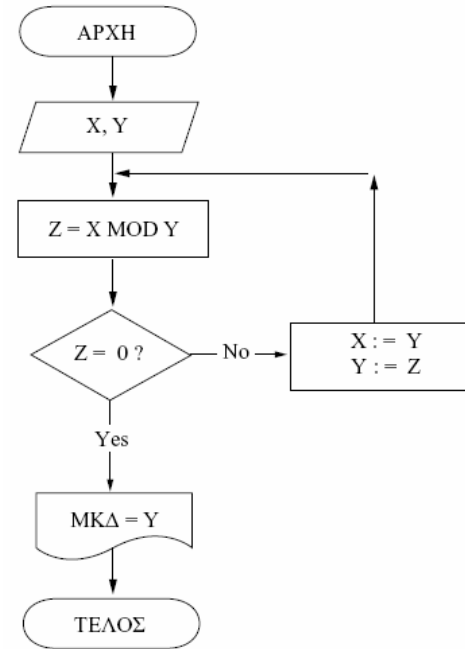
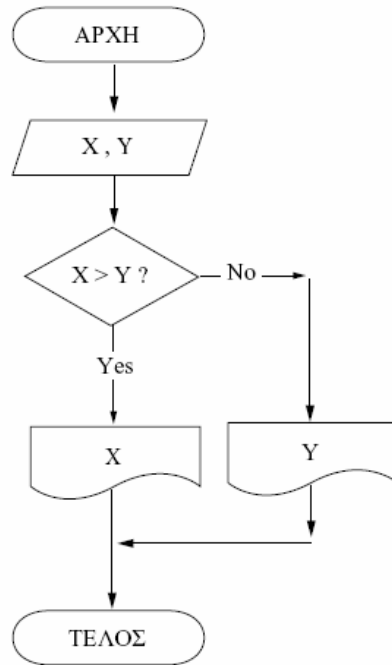


# Σύμβολα Διαγραμμάτων Ροής

ΣΥΜΒΟΛΟ	ΕΝΝΟΙΑ
	Παριστάνει την αρχή ή το τέλος μιας διαδικασίας
	Παριστάνει τον υπολογισμό ή μια επεξεργασία για την οποία υπάρχει μια είσοδος και μια έξοδος
	Παριστάνει την εκτέλεση ομάδας εντολών. Η ανάλυση αυτής της ομάδας εντολών γίνεται ξεχωριστά
	Μέσο εισαγωγής ή παρουσίασης στοιχείων
	Απόφαση για μια συνθήκη. Υπάρχουν 2 έξοδοι ανάλογα με το αν ικανοποιείται ή όχι η συνθήκη
	Απόφαση με πολλαπλές εξόδους
	Σημείο διακλάδωσης
	Εκτύπωση
	Ροή εκτέλεσης



# Παράδειγμα





# Γλώσσες Προγραμματισμού

- Γλώσσες μηχανής
  - Άμεση επικοινωνία με το υλικό του υπολογιστή
- Συμβολικές γλώσσες
  - Χρήση συμβολικών ονομάτων για τις βασικές εντολές
- Υψηλού επιπέδου
  - Προσανατολισμένες στο πρόβλημα και ουχί στον υπολογιστή



# Γλώσσες Μηχανής (Machine Language)

- Η μόνη γλώσσα που «καταλαβαίνει» άμεσα ο υπολογιστής
- Κάθε υπολογιστής έχει τη δική του διαφορετική γλώσσα μηχανής
- Κάθε κωδικός της γλώσσας μηχανής συμβολίζει μία συγκεκριμένη λειτουργία, π.χ. Πολλαπλασιασμός = 1001
- Ο προγραμματισμός σε γλώσσα μηχανής είναι εξαιρετικά επίπονη διαδικασία
- Ξένη προς τον τρόπο που σκέφτεται ο άνθρωπος
- Μεγάλη ταχύτητα εκτέλεσης των προγραμμάτων



# Συμβολικές Γλώσσες (Assembly)

- Για την καλύτερη απομνημόνευση των κωδίκων μηχανής χρησιμοποιούνται μνημονικοί κωδικοί (συντομεύσεις των Αγγλικών)
  - Πολλαπλασιασμός = MULT = 1001
  - ADD32 6,1,2
- Η μετάφραση από τα συμβολικά ονόματα στους αριθμητικούς κωδικούς της γλώσσα μηχανής γίνεται μέσω του συμβολομεταφραστή (assembler)



# Γλώσσες Υψηλού Επιπέδου

- Δεν εξαρτώνται από την μηχανή αλλά από το πρόβλημα
- Επιτρέπουν στον προγραμματιστή να γράψει εντολές χρησιμοποιώντας κώδικες λέξεις και συμβατικούς μαθηματικούς συμβολισμούς
- Οι εντολές μοιάζουν σχεδόν όπως τα Αγγλικά και περιέχουν τα συνηθισμένα μαθηματικά σύμβολα
  - If A then B
  - $D=A*B+C$



# Μεταγλώττιση (Compilation)

- Ο υπολογιστής δεν «καταλαβαίνει» άμεσα μια γλώσσα υψηλού επιπέδου
- Η μετάφραση από μια γλώσσα υψηλού επιπέδου στη γλώσσα μηχανής γίνεται μέσω των μεταγλωττιστών (compilers)
  - ✓ Οδηγίες του προγραμματιστή: αποθηκευμένες στο πηγαίο πρόγραμμα (source program)
  - ✓ Αντίστοιχες οδηγίες επιπέδου μηχανής: αποθηκευμένες στο αντικειμενικό ή τελικό πρόγραμμα (object program)
  - ✓ Σε ορισμένα μικρά υπολογιστικά συστήματα (π.χ. PC) το αντικειμενικό πρόγραμμα πρέπει να προετοιμαστεί για εκτέλεση από άλλο ειδικό πρόγραμμα, τον συνετή (linker)





# Μεταγλωττιστές

- Μεταφράζουν μια συγκεκριμένη γλώσσα υψηλού επιπέδου στη γλώσσα μηχανής ενός συγκεκριμένου είδους
- Μεγάλα προγράμματα που τοποθετούνται μονιμά στη δευτερεύουσα μνήμη και όταν χρειαστεί μεταφέρονται στην κυρία μνήμη
- Όταν τελειώσει η διαδικασία μεταγλώττισης είναι διαθέσιμο το εκτελέσιμο πρόγραμμα για τον χειρισμό των δεδομένων που δίνει τα αποτελέσματα στην έξοδο



# Είδη Προγραμματισμού

- Διαδικαστικός προγραμματισμός
  - **Pascal**, Basic, C
- Δηλωτικός προγραμματισμός
  - Prolog
- Αντικειμενοστρεφής προγραμματισμός
  - C++, Java
- Οπτικός προγραμματισμός
  - Visual Basic, Visual C++, Visual Prolog



# Διαδικαστικός Προγραμματισμός

## Procedural Programming

- Κάθε πρόγραμμα είναι μία διαδικασία ή ένα σύνολο εργασιών που δίνονται με προκαθορισμένο τρόπο και σειρά στο πρόγραμμα
- Οι περισσότερες γλώσσες προγραμματισμού είναι διαδικαστικές
- Ευνοείται άμεσα η δημιουργία και εκτεταμένη χρήση βιβλιοθηκών προγραμμάτων που περιέχουν υπορουτίνες σχετικές με κοινά θέματα (π.χ. βιβλιοθήκη γραφικών)



# Δηλωτικός Προγραμματισμός

## Declarative Programming

- Ένα πρόβλημα αποτελείται από σχέσεις που περιγράφουν τα συστατικά του προβλήματος και τους συσχετισμούς μεταξύ τους
- Η λύση του προβλήματος προκύπτει από την εφαρμογή του μηχανισμού συμπερασμού της γλώσσας
- Το πρόγραμμα τερματίζεται όταν βρεθούν όλες οι δυνατές λύσεις του προβλήματος
- **Παραλληλισμός:** χάρτης  $\leftrightarrow$  σαφείς οδηγίες για την εύρεση της πορείας



# Αντικειμενοστρεφής Προγραμματισμός

## Object Oriented Programming

- Βασίζεται σε αντικείμενα (objects) διαφόρων κατηγοριών (classes), τα οποία αλληλεπιδρούν μεταξύ τους
- Πρώτα ορίζονται τα κατάλληλα για το πρόβλημα αντικείμενα και στη συνέχεια χρησιμοποιούνται για να περιγραφεί βήμα προς βήμα η εξεύρεση της λύσης



# Οπτικός Προγραμματισμός

## Visual Programming

- Παρέχεται ευκολία ανάπτυξης ολοκληρωμένων εφαρμογών
- Έτοιμες βιβλιοθήκες γραφικών που μπορούν να χρησιμοποιηθούν σε διαφορετικές εφαρμογές
- Περισσότερο οικείος στους προγραμματιστές



# Εξέλιξη Γλωσσών Προγραμματισμού<sub>1/4</sub>

- Ο ψευδοκώδικας είναι κάτι που μοιάζει με κώδικα προγραμματισμού, αλλά δεν είναι.
- Ο ψευδοκώδικας είναι μια δομημένη γλώσσα που χρησιμοποιεί:
  - στοιχεία (συντακτικές δομές) από τις γλώσσες προγραμματισμού
  - συμβολισμούς (σημασιολογικά στοιχεία) από τα Μαθηματικά και τη Μαθηματική Λογική
  - στοιχεία (λεκτικές περιγραφές) από τη φυσική γλώσσα



# Εξέλιξη Γλωσσών Προγραμματισμού<sub>2/4</sub>

- **BASIC (Beginners All-purpose Symbolic Instruction Code)**
  - J. Kemmeny, T. Kurtz, 1961
  - Ιδιαίτερα διαδεδομένη σε αρχάριους και απλούς χρήστες υπολογιστών
  - Quick Basic, Visual Basic: οι πιο πετυχημένες εκδόσεις
- **PASCAL**
  - K. Jensen, N. Wirth, 1971
  - Γενικής χρήσης
  - Ακολουθεί τις αρχές του δομημένου προγραμματισμού





# Εξέλιξη Γλωσσών Προγραμματισμού<sub>3/4</sub>

- **PROLOG (PROgramming in LOGic)**
  - 1970
  - Λογικός προγραμματισμός, χειρισμός συμβόλων
  - Βασίζεται στην μαθηματική λογική (κατηγορηματικός λογισμός πρώτης τάξης)
  - Χρησιμοποιείται σε εφαρμογές τεχνητής νοημοσύνης

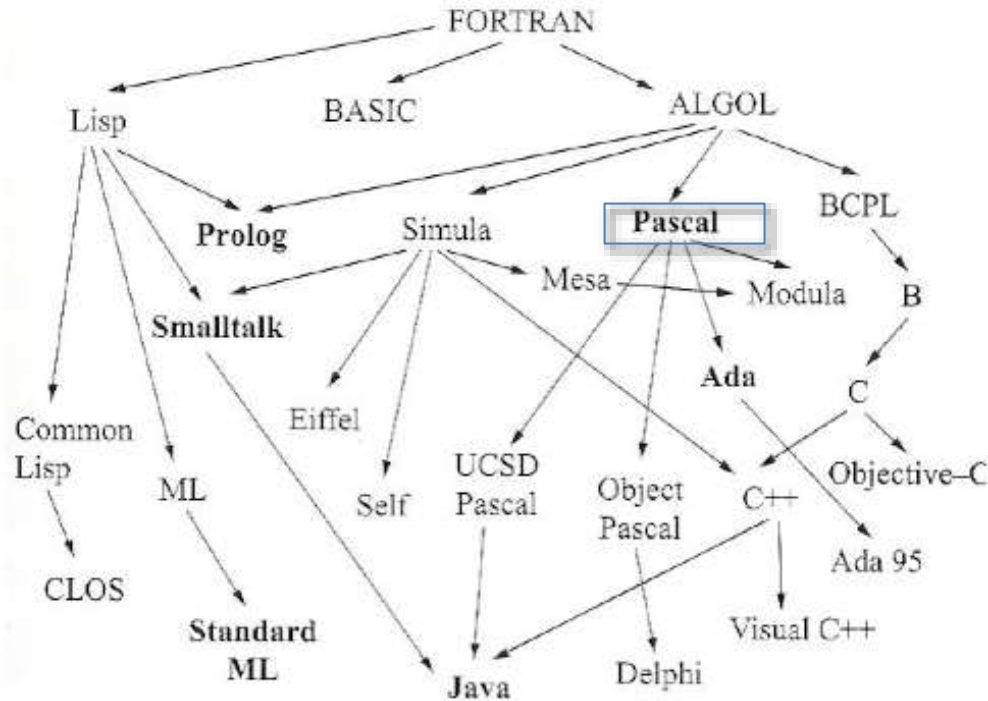


# Εξέλιξη Γλωσσών Προγραμματισμού<sub>4/4</sub>

- C
  - D. Ritchie, 1972
  - Υποστήριξη πολύ εξειδικευμένων τελεστών
  - Προσεγγίζει την ταχύτητα της γλώσσας Assembly
  - Χρησιμοποιήθηκε για την κατασκευή του λειτουργικού συστήματος Unix
  - C++: προγραμματισμός προσανατολισμένος σε αντικείμενα



# Γενεαλογικό Δένδρο Γλωσσών Προγραμματισμού





# Γιατί Pascal?

- Απλή και εύκολη στην εκμάθηση
- Περιλαμβάνει ικανοποιητικό σύνολο εντολών ελέγχου
- Διαχειρίζεται πολλούς τύπους δεδομένων
- Είναι γλώσσα γενικού σκοπού
- Υπάρχουν μεταφραστές της Pascal για τα περισσότερα υπολογιστικά συστήματα
- Ιδανική για διδασκαλία βασικών αρχών προγραμματισμού



# Βασικά στοιχεία ενός προγράμματος Pascal

- Ένα πρόγραμμα για Η/Υ απαρτίζεται από μια ακολουθία συμβόλων ή χαρακτήρων που σχηματίζουν το κείμενο του προγράμματος
- Οι κανόνες της γλώσσας αποτελούνται από 2 τμήματα που ονομάζονται:
  - **Σύνταξη:** ορίζουν τον τρόπο με τον οποίο το σύνολο των λέξεων μιας γλώσσας (λεξιλόγιο) μπορεί να συνδυαστεί ώστε να σχηματιστούν προτάσεις
  - **Σημασιολογία:** δίνουν έννοια σ' αυτούς τους συνδυασμούς των λέξεων



# Σύνολο χαρακτήρων της Pascal<sub>1/2</sub>

- Για να κωδικοποιήσουμε στον ΗΥ ένα πρόγραμμα χρησιμοποιούμε:
  - το αλφάβητο της γλώσσας
  - τους συντακτικούς της κανόνες
- Το σύνολο χαρακτήρων καθορίζει το ποιους χαρακτήρες μπορούμε να χρησιμοποιήσουμε όταν γράφουμε τα προγράμματά μας
- Η Pascal χρησιμοποιεί:
  - Τα γράμματα του λατινικού αλφάβητου, κεφαλαία: **A B C D E F  
G H I J K L M N O P Q R S T U V W X Y Z**
  - και μικρά: **a b c d e f g h i j k l m n o p q r s t u v w x y z**



# Σύνολο χαρακτήρων της Pascal<sub>1/2</sub>

- Τα ψηφία του δεκαδικού συστήματος: **0 1 2 3 4 5 6 7 8 9**
- Το κενό (**blank**)
- ορισμένους ειδικούς χαρακτήρες: **+ - \* / = < > [ ] . , : ; ^ ( )**



# Λεξιλόγιο

- Το λεξιλόγιο της Pascal περιλαμβάνει 8 κατηγορίες:
  - Ονόματα
  - Αριθμούς
  - Χαρακτήρες
  - Αλυσίδες χαρακτήρων
  - Τελεστές
  - Διαχωριστές
  - Λέξεις κλειδιά
  - Σχόλια





# Λεξιλόγιο-Ονόματα<sub>1/2</sub>

- Ονόματα ορίζονται από τον προγραμματιστή και χρησιμοποιούνται για τον προσδιορισμό των διαφόρων ποσοτήτων μέσα σε ένα πρόγραμμα
- Για κάθε όνομα ισχύουν τα εξής:
  - Αποτελούνται από ένα γράμμα που ακολουθείται από έναν οποιαδήποτε αριθμό γραμμάτων και ψηφίων, π.χ. Value1,result, mkd, κ.τ.λ.
  - Επιτρέπεται η χρήση του χαρακτήρα “\_”, π.χ. Value\_1, value\_of\_x, κλπ.
  - Θεωρητικά το μήκος τους είναι απεριόριστο (εξαρτάται από τον μεταφραστή), π.χ. Θεωρούνται ίδια τα ονόματα Logariasmos\_No1 Logariasmos\_No2 LOGARIASMOS\_No3
  - Δεν επιτρέπεται η χρήση ονομάτων που έχουν ειδική σημασία για την Pascal



# Λεξιλόγιο-Ονόματα<sub>2/2</sub>

Επιτρεπτά ονόματα	Μη επιτρεπτά Ονόματα	Αιτιολογία
Name	Name's	Περιέχει το χαρακτήρα "' "
Elem_number	Elem-Number	Περιέχει το χαρακτήρα "-"
K1	Elem Number	Περιέχει κενό χαρακτήρα
ForNumbers	For	Ειδική σημασία στην Pascal
F10KL2	2Def	Αρχίζει από ψηφίο



# Λεξιλόγιο-Αριθμούς<sub>1/4</sub>

- **Ακέραιοι (integers):** Ακολουθίες ψηφίων που μπορεί να προηγείται το πρόσημο + (προαιρετικό) ή -
- Οι τιμές έχουν εύρος:
  - [ -MAXINT,..-1,0,+1,..,+MAXINT ], όπου  
MAXINT=32767



# Λεξιλόγιο-Αριθμούς<sub>2/4</sub>

Επιτρεπτοί ακέραιοι	Μη επιτρεπτοί ακέραιοι	Αιτιολογία
45 0 -8 32000	43,600  2.5786	Περιέχει χαρακτήρα μη-ψηφίο ", "  Περιέχει χαρακτήρα μη-ψηφίο "."



# Λεξιλόγιο-Αριθμούς<sub>3/4</sub>

- **Πραγματικοί (reals):** Αριθμοί αρνητικοί ή θετικοί που μπορεί να περιέχουν και δεκαδικά ψηφία
- Περιέχουν υποδιαστολή ή είναι
- εκφρασμένοι σε εκθετική μορφή:
  - $38E27 \rightarrow 8 \times 10^{27}$
  - $-0,092E-19 \rightarrow -0,092 \times 10^{-19}$



# Λεξιλόγιο-Αριθμούς<sub>4/4</sub>

Επιτρεπτοί πραγματικοί	Μη επιτρεπτοί Πραγματικοί	Αιτιολογία
23.56	23.	Δεν έχει δεκαδικά ψηφία
-49.375	45,89	Περιέχει κόμμα
9674785.0	1.345.78	Περιέχει 2 δεκαδικά ψηφία
34E21	36E2*6	Περιέχει το χαρακτήρα *
-14.487E6	-8.2E4.1	Ο εκθέτης δεν είναι ακέραιος



# Λεξιλόγιο-Χαρακτήρες

- Είναι χαρακτήρες από το σύνολο των χαρακτήρων που υποστηρίζει το υπολογιστικό σύστημα μέσα σε αποστρόφους. Π.χ. ‘α’, ‘Α’, ‘&’, ‘4’
  - ‘5’ διαφέρει από το 5
  - ‘Α’ διαφέρει από το ‘α’
  - ‘ ’ ο κενός χαρακτήρας



# Λεξιλόγιο-Αλυσίδες Χαρακτήρων

- Αλυσίδα χαρακτήρων (string) είναι μια ακολουθία χαρακτήρων μεταξύ αποστρόφων.
  - 'algorithm'
  - 'Pascal Vers'
  - '5'
  - '236'
  - '5+2'





# Λεξιλόγιο-Τελεστές<sub>1/4</sub>

- Είναι σύμβολα που δηλώνουν μια αριθμητική ή λογική πράξη μεταξύ αριθμητικών ή αλφαριθμητικών δεδομένων:
  - Αριθμητικοί
  - Σύγκρισης
  - Λογικοί



# Λεξιλόγιο-Τελεστές<sub>2/4</sub>

Αριθμητικοί Τελεστές	Λειτουργία	Σύνταξη
-	Αρνηση	-X
*	Πολ/σμός αριθμών ή τομή συνόλων	X*Y
/	Διαίρεση αριθμών	X/Y
DIV	Ακέραιο πηλίκο	X DIV Y
MOD	Ακέραιο υπόλοιπο	X MOD Y
+	Πρόσθεση αριθμών ή ένωση συνόλων	X+Y
-	Αφαίρεση αριθμών ή αφαίρεση συνόλων	X-Y



# Λεξιλόγιο-Τελεστές<sub>3/4</sub>

Τελεστές σύγκρισης	Λειτουργία	Σύνταξη
=	Ισότητα	$X=Y$
<> ή ><	Διάφορο	$X<>Y$ $X><Y$
<	Μικρότερο	$X<Y$
>	Μεγαλύτερο	$X>Y$
<= ή =<	Μικρότερο ή ίσο	$X<=Y$ ή $X=<Y$
>= ή =>	Μεγαλύτερο ή ίσο	$X>=Y$ ή $X=>Y$



# Λεξιλόγιο-Τελεστές<sub>4/4</sub>

+ Λογικοί τελεστές

X	Y	NOT X	X AND Y	X OR Y
A	A	Ψ	A	A
A	Ψ	Ψ	Ψ	A
Ψ	A	A	Ψ	A
Ψ	Ψ	A	Ψ	Ψ



# Λεξιλόγιο-Διαχωριστές

- Είναι χαρακτήρες με συγκεκριμένη λειτουργία
  - := καταχώρηση
  - ( ) παρενθέσεις
  - [ ] αγκύλες για τους πίνακες και τα σύνολα
  - , . ; : σημεία στίξης



# Λεξιλόγιο-Λέξεις Κλειδιά

- ☉ And
- ☉ Array
- ☉ Begin for
- ☉ case
- ☉ const goto
- ☉ div
- ☉ Do
- ☉ Downto
- ☉ else
- end
- file
- function
- if
- in
- label
- mod
- ☉ Nil
- ☉ not
- ☉ of
- ☉ or
- ☉ packed
- ☉ procedure
- ☉ program
- ☉ record
- ☉ repeat
- set
- then
- to
- type
- until
- var
- while
- with



# Λεξιλόγιο-Σχόλια

- Σχόλιο είναι ένα κείμενο μέσα σε αγκύλες, το οποίο μπορεί να τοποθετηθεί οπουδήποτε μέσα στο πρόγραμμα για επεξήγηση τμημάτων του προγράμματος
- Αντί για αγκύλες μπορούμε να χρησιμοποιούμε και τους χαρακτήρες (\* και \*).
  - Για παράδειγμα: {This is a remark} {Κείμενο και στα Ελληνικά}  
(\* Σχόλιο \*)



# Δεδομένα

- Τα δεδομένα παριστάνουν πληροφορίες που πρέπει να δοθούν σε ένα πρόγραμμα. Το πρόγραμμα επεξεργάζεται τα δεδομένα για να παράγει αποτελέσματα
- Τα δεδομένα προέρχονται από το εξωτερικό περιβάλλον (συσκευές εισόδου) ή τη βοηθητική μνήμη του υπολογιστή (συσκευές μαγνητικών δίσκων)
- Αντιπροσωπεύουν γεγονότα, παρατηρήσεις, μετρήσεις, ιδέες για τον κόσμο κ.α.





# Δεδομένα

- Τα δεδομένα που χρειάζεται ένα πρόγραμμα είναι δύο ειδών:
  - Σταθερές (constants)
  - Μεταβλητές (variables)



# Δεδομένα-Σταθερές

- Μερικά από τα δεδομένα που χρησιμοποιούνται σε ένα πρόγραμμα δεν αλλάζουν ποτέ
- Τα δεδομένα αυτά μπορούν να παρασταθούν με σταθερές (constants)
- Οι ορισμοί των σταθερών ενός προγράμματος ακολουθούν την επικεφαλίδα του και αρχίζουν με την ειδική λέξη `const`

```
const meresergasias = 5;
```

```
const asteriskos = '*';
```

```
const nai = true, pi = 3.14159265368;
```



# Δεδομένα-Μεταβλητές

- **Μεταβλητές (variables):** δεδομένα τα οποία υπόκεινται σε αλλαγή
  - τρέχον άθροισμα σ' ένα υπολογιστή τσέπης
- Οι μεταβλητές παριστάνουν θέσεις της κύριας μνήμης του υπολογιστή
- Η «τιμή» μιας μεταβλητής αναφέρεται στο περιεχόμενο της θέσης μνήμης που αντιστοιχεί στη μεταβλητή αυτή
- Μπορούμε να χρησιμοποιήσουμε όσες μεταβλητές χρειαζόμαστε και να τις ονομάσουμε όπως μας βολεύει



# Δηλώσεις Μεταβλητών<sub>1/3</sub>

- Οι δηλώσεις των μεταβλητών ακολουθούν τους ορισμούς των σταθερών (αν αυτοί υπάρχουν)

`var merokamato: integer;`

`aktina, periferia, emvadon: real;`

- Η δήλωση μιας μεταβλητής αντιστοιχίζει απλά το αναγνωριστικό της με τη διεύθυνση μιας θέσης της μνήμης του υπολογιστή και περιγράφει τον τύπο των τιμών που μπορούν να αποθηκευτούν στη θέση αυτή
- Κατά τη δημιουργία της, η τιμή μιας μεταβλητής είναι αόριστη



# Δηλώσεις Μεταβλητών<sub>2/3</sub>

- Τα δεδομένα έχουν κάποιο τύπο (type) που ορίζει τη σημασία τους, το σύνολο των τιμών (values) που μπορούν να πάρουν και περιορίζει τι μπορεί να γίνει με αυτές.
- Η Pascal υποστηρίζει 5 βασικούς **απλούς τύπους (simple types)** δεδομένων:
  - integer : τα δεδομένα παίρνουν ακέραιες τιμές
  - real : τα δεδομένα παίρνουν πραγματικές τιμές
  - boolean : παίρνουν μόνο από τις δύο λογικές τιμές TRUE ή FALSE
  - char : δέχονται σαν τιμές ένα χαρακτήρα, π.χ. 'a' , 'G' , '4' , '&'
  - string : αλυσίδα χαρακτήρων, π.χ. 'Monday', 'Στέφανος'



# Δηλώσεις Μεταβλητών<sub>3/3</sub>

- Η Pascal εφαρμόζει αυστηρό έλεγχο τύπων, δεν επιτρέπει την ανάμειξη δεδομένων διαφόρων τύπων
  - π.χ. δεν μπορεί να γίνει πολλαπλασιασμός ενός δεδομένου τύπου integer με ένα δεδομένου τύπου char



# Εκφράσεις στην Pascal

## Σειρά προτεραιότητας

- NOT
- \* / DIV MOD AND
- + - OR
- = <> > < >= <=



# Αριθμητικές Συναρτήσεις

- $\text{ABS}(\text{num}) \rightarrow \text{abs}(-2.54)=2.54$
- $\text{COS}(\text{num})$
- $\text{EXP}(\text{num})$
- $\text{FRAC}(\text{num}) \rightarrow \text{frac}(3.78)=0.78$
- $\text{INT}(\text{num}) \rightarrow \text{int}(3.78)=3.0$
- $\text{SQR}(\text{num}) \rightarrow \text{sqr}(2)=4$
- $\text{SQRT}(\text{num}) \rightarrow \text{sqrt}(9)=3$





# Βαθμωτές Συναρτήσεις

- $PRED(num) \rightarrow Pred(10)=9$
- $ODD(num) - \text{περιττός} \rightarrow Odd(7)=true$
- $SUCC(num) \rightarrow Succ(10)=11$



# Συναρτήσεις Μετατροπής

- $ORD(num) \rightarrow (A,B,C,D,E,F), Ord(C)=2$
- $ROUND(num) \rightarrow Round(3.78)=4$
- $TRUNC(num) \rightarrow Trunc(3.78)=3$



# Παράδειγμα 1

Άλγεβρα	Pascal
$(-a + b)c$	$(-a+b) *c$
$\frac{ab}{c}$	$a*b \text{ div } c$
$a + \frac{b}{c}$	$a+b \text{ div } c$
$x(y+w(3-v))$	$x*(y+w*(3-v))$
$\frac{y+2}{x+4}$	$(y+2) \text{ div } (x+4)$



# Παράδειγμα 2

Άλγεβρα

$$y^4$$

$$|y - 7|$$

$$|x^2 - x + 3|$$

Pascal

$$\text{sqr}(\text{sqr}(y)) \text{ ή } \text{sqr}(y*y) \text{ ή } y*y*y*y$$

$$\text{abs}(y - 7)$$

$$\text{abs}(\text{sqr}(x) - x + 3)$$



# Βιβλιογραφία

Βλαχάβας Ι. (1994). Η γλώσσα προγραμματισμού Pascal. Εκδόσεις Γαρταγάνης Διονύσιος.

Κάβουρας Ι.Κ. (1999). Δομημένος Προγραμματισμός με Pascal. Εκδόσεις Κλειδάριθμος.

Αλεβίζου Θ., & Καμπουρέλης Α. (1995). Μαθήματα Προγραμματισμού: Εισαγωγή με τη Γλώσσα Pascal. Εκδόσεις Παπασωτηρίου.

Cooper D. (1993). Oh! Pascal!, An Introduction to Computing, του. Εκδόσεις Norton.

Larry R.N. (1998). Advanced Programming in Pascal with Data Structures. Εκδόσεις Macmillan USA.

Τσελίκης Γ.Σ., Τσελίκας Ν.Δ. (2012). C: από τη Θεωρία στην Εφαρμογή (Β' Έκδοση). Εκδόσεις Παπασωτηρίου.

Aho A.V., Hopcroft J.E., & Ullman J.D. (1974). The design and analysis of computer algorithms. Εκδόσεις Addison Wesley.

Abelson H., Sussman G.J., Sussman J. (1985). Structure and Interpretation of Computer Programs, MIT Press, McGraw Hill Book Company.



# Σημείωμα Αναφοράς

Copyright Τεχνολογικό Ίδρυμα Ηπείρου. Αλέξανδρος Τζάλλας.  
Προγραμματισμός Ι.

Έκδοση: 1.0 Άρτα, 2015. Διαθέσιμο από τη δικτυακή  
διεύθυνση:

<http://eclass.teiep.gr/OpenClass/courses/COMP111/>



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού-Μη Εμπορική Χρήση-Όχι Παράγωγα Έργα 4.0 Διεθνές [1] ή μεταγενέστερη. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, Διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/deed.el>



# Τέλος Ενότητας

Επεξεργασία: Ευάγγελος Καρβούνης  
Άρτα, 2015



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ





# Τέλος Ενότητας

## Εισαγωγικά Στοιχεία της Pascal



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ