



Εισαγωγή στις Γλώσσες Προγραμματισμού

Αθ. Ανδρούτσος



Πρόγραμμα

- Ένα **πρόγραμμα** είναι μία ακολουθία εντολών, γραμμένων σε μία γλώσσα προγραμματισμού
- Τις εντολές αυτές τις δέχεται ως είσοδο ένα άλλο πρόγραμμα που ονομάζεται **Μεταγλωττιστής**, ο οποίος μεταφράζει (Language Translator) τις εντολές σε γλώσσα μηχανής που στη συνέχεια εκτελούνται



Εκτέλεση Προγράμματος

Γλώσσες Προγραμματισμού

- Για να εκτελεστεί ένα πρόγραμμα θα πρέπει και οι **εντολές και τα δεδομένα** να φορτωθούν στη μνήμη, σύμφωνα με την αρχιτεκτονική Von Neumann
- Επομένως, τα συστατικά στοιχεία ενός προγράμματος δεν είναι μόνο οι **εντολές** - οι οποίες παρέχουν λειτουργικότητα (π.χ. μία πρόσθεση)- αλλά και τα **δεδομένα** επί των οποίων εφαρμόζεται η λειτουργικότητα



Δεδομένα (1)

- Λέγοντας δεδομένα εννοούμε κάθε πληροφορία του πραγματικού κόσμου που μπορεί να αναπαρασταθεί στη μνήμη ενός συστήματος υπολογιστή
- Κατηγορίες (Τύποι) τέτοιων δεδομένων μπορεί να είναι οι αριθμοί, χαρακτήρες, αλλά και πιο σύνθετες δομές, όπως ακολουθίες αριθμών ή χαρακτήρων



Δεδομένα (2)

- Θα δούμε ότι το σύστημα τύπων δεδομένων (**Type System**) που μας παρέχει μία γλώσσα προγραμματισμού για να μπορούμε να ορίζουμε δεδομένα, απλά ή σύνθετα, είναι από τα πιο θεμελιώδη και σημαντικά υποσυστήματα μίας γλώσσας



Αλγόριθμοι

Γλώσσες Προγραμματισμού

- Η έννοια του αλγορίθμου -και συνακόλουθα της αλγοριθμικής σκέψης- είναι μία τεχνική έννοια που σημαίνει μία ακολουθία βημάτων για την επίλυση ενός προβλήματος
- Υπάρχουν διάφορα προβλήματα που συνδέονται με αλγοριθμικές εφαρμογές και αφορούν κυρίως θεμελιώδη τεχνικά προβλήματα σε επιμέρους τομείς



Αλγοριθμικές εφαρμογές

Γλώσσες Προγραμματισμού

- Τέτοια προβλήματα που συνήθως συνδέονται με την έννοια του αλγορίθμου και της αλγοριθμικής σκέψης είναι η εύρεση του ελάχιστου ή μέγιστου στοιχείου μία ακολουθίας αριθμών ή η ταξινόμηση μίας ακολουθίας στοιχείων, η εύρεση της βέλτιστης διαδρομής –για παράδειγμα σε όρους χρόνου- μεταξύ εναλλακτικών διαδρομών, κλπ.



Αλγοριθμική σκέψη

Γλώσσες Προγραμματισμού

- Η καλλιέργεια της αλγοριθμικής σκέψης είναι σημαντικό στοιχείο από μόνη της στην εξέλιξη ενός προγραμματιστή
- Αποτελεί επίσης προαπαιτούμενο για την περαιτέρω προγραμματιστική εξέλιξη στην επίλυση ανώτερου επιπέδου προβλημάτων που σχετίζονται με εμπορικές εφαρμογές



Εμπορικές Εφαρμογές (1)

Γλώσσες Προγραμματισμού

- Για την ανάπτυξη εφαρμογών που απευθύνονται στο εμπορικό κύκλωμα απαιτούνται δεξιότητες ευρύτερες από την αλγοριθμική σκέψη, όπως μεθοδολογίες και αρχιτεκτονικές ανάπτυξης λογισμικού (software architectures), καλές πρακτικές προγραμματισμού (design patterns), γνώση εργαλείων (programming tools), κλπ.



Εμπορικές Εφαρμογές (2)

Γλώσσες Προγραμματισμού

- Επίσης απαιτούνται και γνώσεις άλλων συστατικών στοιχείων, όπως
 - **Βάσεις Δεδομένων**, που χρησιμοποιούνται για την εισαγωγή, διαγραφή, ενημέρωση και κυρίως αναζήτηση δεδομένων που έχουν αποθηκευτεί μόνιμα σε εξωτερικές οργανωμένες συλλογές δεδομένων
 - **Σχεδιασμός και ανάπτυξη διεπαφών με τον χρήστη (Οθονών)** ώστε η αλληλεπίδραση του χρήστη με την εφαρμογή να είναι φιλική προς τον χρήστη (user friendly)



Προγραμματιστικές Δεξιότητες

Γλώσσες Προγραμματισμού

- Η γνωσιακή διαδρομή ενός προγραμματιστή θα πρέπει να είναι να καλλιεργήσει πρώτα την αλγοριθμική σκέψη, να εξελίξει επίσης τις γνώσεις του στο χώρο των Βάσεων Δεδομένων και στον Σχεδιασμό Διεπαφών και στη συνέχεια να συνθέσει τις γνώσεις αυτές στο ανώτερο επίπεδο του σχεδιασμού και υλοποίησης εφαρμογών



Γραμματική Γλωσσών

Γλώσσες Προγραμματισμού

- Οι γλώσσες προγραμματισμού μας παρέχουν ένα σετ ειδικών λέξεων που αποτελούν μέρος της Γραμματικής της γλώσσας για να μπορούμε να περιγράψουμε αλγόριθμους ή εφαρμογές
- Το σετ αυτό των ειδικών λέξεων ονομάζεται keywords και μαζί με τους συντακτικούς και εννοιολογικούς κανόνες της γλώσσας αποτελούν τη Γραμματική της γλώσσας



Ανάπτυξη Προγραμμάτων

Γλώσσες Προγραμματισμού

- Για να γράψουμε ένα πρόγραμμα θα πρέπει να γνωρίζουμε τα keywords της γλώσσας, τους συντακτικούς και εννοιολογικούς κανόνες
- Περαιτέρω θα πρέπει να γνωρίζουμε και τα έτοιμα προγράμματα που παρέχει μία γλώσσα προγραμματισμού –και που είναι οργανωμένα σε βιβλιοθήκες προγραμμάτων, ώστε τελικά να συνθέσουμε ένα πρόγραμμα



Μεταγλωττιστής (1)

- Ο Μεταγλωττιστής (compiler) είναι ένα πρόγραμμα που δέχεται ως είσοδο ένα άλλο πρόγραμμα γραμμένο σε μία γλώσσα προγραμματισμού, που ονομάζεται **γλώσσα πηγής (source language)** και παράγει ένα ισοδύναμο πρόγραμμα σε μία **άλλη γλώσσα (target language)** που μπορεί και εκτελείται από την μηχανή



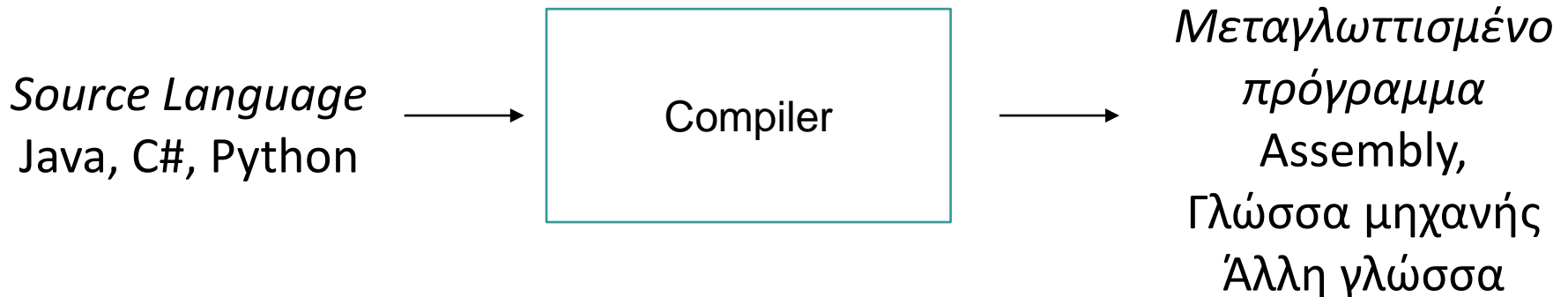
Μεταγλωττιστής (2)

- Οι μεταγλωττιστές αποτελούνται από τρία τουλάχιστον σημαντικά υποσυστήματα:
 - *Λεκτικός Αναλυτής*, που μετατρέπει τον source code σε tokens
 - *Συντακτικός Αναλυτής*, που ελέγχει αν η σύνταξη του προγράμματος αν είναι ορθή με βάση μια Γραμματική της γλώσσας
 - *Εννοιολογικός Αναλυτής*, που ελέγχει αν το πρόγραμμα συμμορφώνεται με το Type System της γλώσσας



Μεταγλωττιστής (3)

- Η έννοια του μεταγλωττιστή είναι γενικότερη και περιλαμβάνει τη μετάφραση σε τελική γλώσσα, την μετάφραση σε άλλη γλώσσα (transpilers), την μετάφραση σε ενδιάμεση γλώσσα ή και την εκτέλεση ενός προγράμματος





Μεταγλωττιστής (4)

- Υπάρχουν τρεις βασικοί τύποι μεταφραστών γλωσσών:
 - Οι μεταγλωττιστές μίας φάσης, που σε ένα βήμα μεταγλωττίζουν μία γλώσσα πηγής σε εκτελέσιμο πρόγραμμα, όπως συμβαίνει με τις γλώσσες C, C++
 - Οι μεταγλωττιστές δύο φάσεων που μεταγλωττίζουν ένα πρόγραμμα πρώτα σε ενδιάμεση γλώσσα (intermediate language) και μετά το εκτελούν όπως οι Java, C#, JavaScript, Python
 - Οι διερμηνευτές (interpreters)



Διερμηνευτές

Γλώσσες Προγραμματισμού

- Οι διερμηνευτές (interpreters) και η διαδικασία της διερμηνεύσης ενός προγράμματος πηγής είναι η διαδικασία μετάφρασης και ταυτόχρονα εκτέλεσης του προγράμματος
- Διερμηνευόμενες γλώσσες θεωρούνται οι Python, JavaScript αλλά και οι μεταγλώττιση δύο φάσεων στη δεύτερη φάση χρησιμοποιεί διερμηνεύση



Μεταγλώττιση δύο φάσεων (1)

Γλώσσες Προγραμματισμού

- Για την ακρίβεια όλες οι σύγχρονες γλώσσες, όπως Java, C#, Python, JavaScript και άλλες χρησιμοποιούν μεταγλώττιση δύο φάσεων, όπου στην 1^η φάση γίνεται μεταγλώττιση σε ενδιάμεσο κώδικα και στη συνέχεια διερμηνευση



Μεταγλώττιση δύο φάσεων (2)

Γλώσσες Προγραμματισμού

- Σε κάποιες γλώσσες όπως Java, C# η μεταγλώττιση δύο φάσεων είναι φανερή γιατί δημιουργούνται ενδιάμεσα αρχεία και σε άλλες γλώσσες, όπως Python, JavaScript δεν είναι φανερή γιατί ο ενδιάμεσος κώδικας δεν αποθηκεύεται σε αρχεία αλλά διατηρείται στη μνήμη



Μεταγλώττιση δύο φάσεων (3)

Γλώσσες Προγραμματισμού

- Σε κάθε περίπτωση, σε όλες τις σύγχρονες γλώσσες στη 2^η φάση γίνεται διερμήνευση του ενδιάμεσου κώδικα σε τελικό κώδικα γλώσσας μηχανής και το πρόγραμμα εκτελείται άμεσα χωρίς να δημιουργείται κάποιο τελικό αρχείο (π.χ. .exe όπως συμβαίνει στη μεταγλώττιση μίας φάσης)
- Ο διερμηνευτής επομένως είναι το περιβάλλον εκτέλεσης του προγράμματος (runtime)



Write once, run anywhere

Γλώσσες Προγραμματισμού

- Η παραγωγή ενδιάμεσου κώδικα γίνεται για να προσδώσουμε ευελιξία στη γλώσσα και τα προγράμματα να μπορούν να εκτελούνται σε οποιαδήποτε μηχανή και οποιοδήποτε Λειτουργικό Σύστημα έχει τον διερμηνευτή χωρίς να χρειάζεται τα προγράμματα να ξαναμεταγλωττίζονται (write once, run anywhere)



Jitter

- Επίσης, σε όλες αυτές τις γλώσσες ο διερμηνευτής συνήθως συνοδεύεται από ένα σύστημα JIT (Just In Time compilation, Jitter) που βελτιστοποιεί τη διαδικασία διερμήνευσης προσδίδοντάς της μεγαλύτερη ταχύτητα, μιας και από μόνοι τους οι διερμηνευτές είναι πιο αργοί από το να εκτελέσουμε ένα τελικό (π.χ. .exe) πρόγραμμα



Γλώσσες Προγραμματισμού (1)

Γλώσσες Προγραμματισμού

- Ιστορικά, οι πρώτες γλώσσες προγραμματισμού όπως η Fortran, Pascal, C, Cobol ήταν Συναρτησιακές/Διαδικασιακές (functional/procedural) γιατί η βασική δομή οργάνωσης του κώδικα ήταν οι συναρτήσεις/διαδικασίες, ώστε ο κώδικας να είναι επαναχρησιμοποιήσιμος
- Θα δούμε ότι η **επαναχρησιμοποίηση του κώδικα** είναι διαρκής στόχος μέχρι και σήμερα στην τεχνολογία λογισμικού



Γλώσσες Προγραμματισμού (2)

- Οι σύγχρονες γλώσσες όπως C++, Java, C# είναι Αντικειμενοστραφείς, μιας και η βασική δομή οργάνωσης του κώδικα είναι η *κλάση* (πρόκειται για μία δομή δεδομένων) που επίσης στοχεύει στην επαναχρησιμοποίηση του κώδικα
- Άλλες γλώσσες προγραμματισμού είναι ταυτόχρονα και Αντικειμενοστραφείς και Συναρτησιακές, όπως οι Python, JavaScript