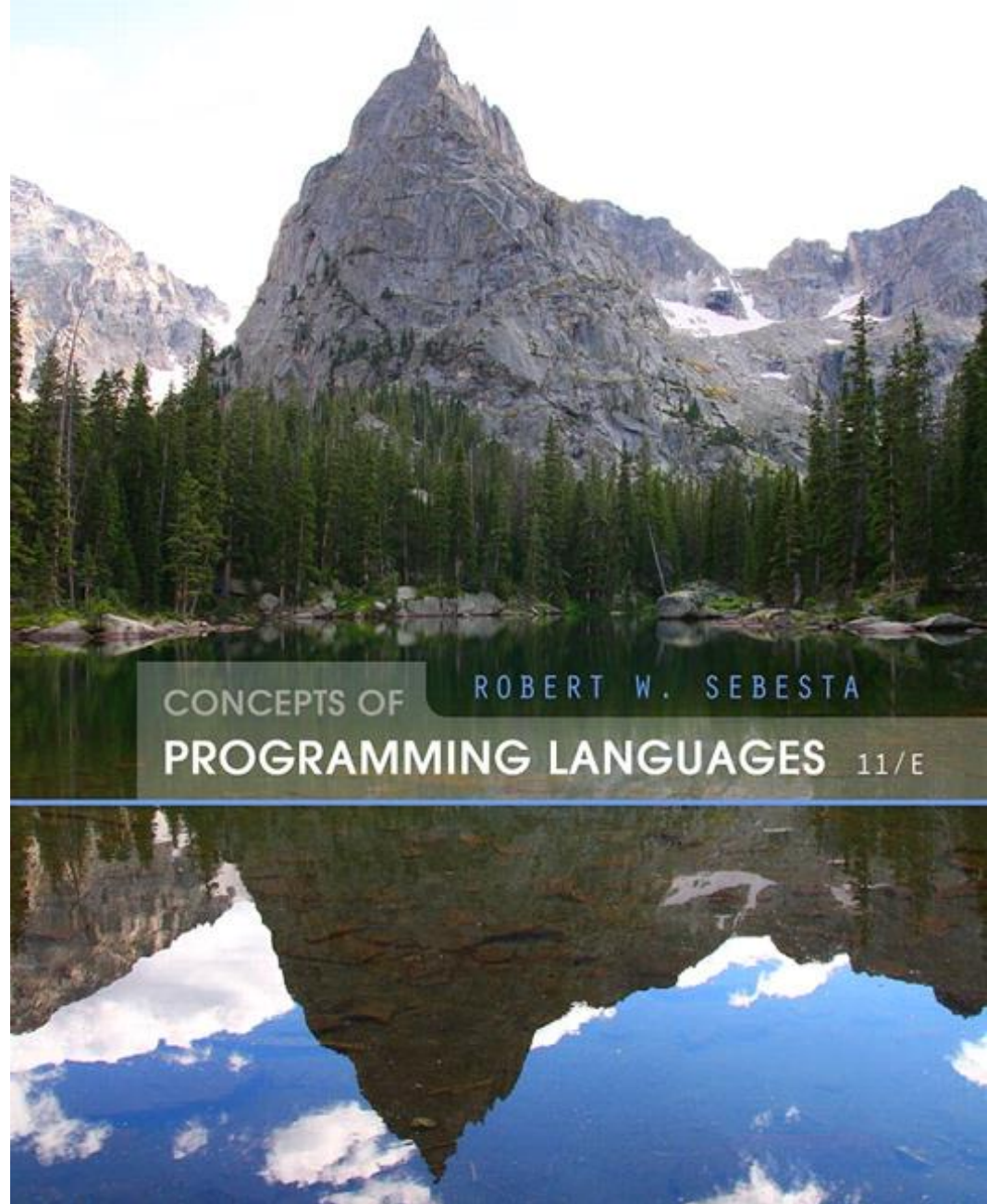


# Κεφάλαιο 5

## Ονόματα, Προσδέσεις, και Εμβέλεις

Γκόγκος Χρήστος  
Τμήμα Πληροφορικής και Τηλεπικοινωνιών (Αρτα)  
Πανεπιστήμιο Ιωαννίνων



# Θέματα κεφαλαίου 5

---

- Εισαγωγή
- Ονόματα
- Μεταβλητές
- Η έννοια της πρόσδεσης
- Εμβέλεια
- Εμβέλεια και διάρκεια ζωής
- Περιβάλλοντα αναφοράς
- Επώνυμες σταθερές

# Εισαγωγή

---

- Οι προστακτικές γλώσσες αποτελούν αφαιρέσεις της αρχιτεκτονικής von Neumann
  - Μνήμη
  - Επεξεργαστής
- Οι μεταβλητές χαρακτηρίζονται από ιδιότητες
  - Η σχεδίαση ενός τύπου, πρέπει να εξετάζει θέματα εμβέλειας, διάρκεια ζωής, ελέγχου τύπων, αρχικοποίησης και συμβατότητας τύπων

# Ονόματα

---

- Θέματα σχεδίασης ονομάτων:
  - Υπάρχει διάκριση πεζών κεφαλαίων;
  - Οι ειδικές λέξεις είναι δεσμευμένες λέξεις ή μόνο λέξεις κλειδιά;

# Ονόματα (συνέχεια)

---

- Μήκος

- Αν είναι πολύ μικρό, μπορεί να μην είναι σε θέση να περιγράψει καλά την επιδιωκόμενη έννοια
- Παραδείγματα γλωσσών:
  - C99: δεν υπάρχει όριο μήκους, αλλά μόνο οι πρώτοι 63 χαρακτήρες είναι σημαντικοί, επίσης, τα εξωτερικά ονόματα έχουν μέγιστο μήκος 31 χαρακτήρων
  - C# και Java: δεν υπάρχει όριο, και όλοι οι χαρακτήρες είναι σημαντικοί
  - C++: χωρίς όριο, αλλά οι διάφορες υλοποιήσεις συχνά επιβάλλουν κάποιο όριο

# Ονόματα (συνέχεια)

---

- Ειδικοί χαρακτήρες
  - PHP: όλα τα ονόματα μεταβλητών πρέπει να ξεκινούν με το σύμβολο του δολαρίου
  - Perl: Όλες τα ονόματα μεταβλητών ξεκινούν με ειδικούς χαρακτήρες που καθορίζουν τον τύπο της μεταβλητής
  - Ruby: Τα ονόματα μεταβλητών που ξεκινούν με @ είναι μεταβλητές στιγμιοτύπων, τα ονόματα μεταβλητών που ξεκινούν με @@ είναι μεταβλητές κλάσεων

# Ονόματα (συνέχεια)

---

- Διάκριση πεζών κεφαλαίων
  - Μειονέκτημα: αναγνωσιμότητα (ονόματα που μοιάζουν είναι διαφορετικά)
    - Τα ονόματα στις C-based γλώσσες έχουν διάκριση πεζών κεφαλαίων (case sensitive)
    - Σε άλλες γλώσσες μπορεί να μην υπάρχει διάκριση πεζών κεφαλαίων
    - Σε γλώσσες όπως η C++, Java, και η C# το πρόβλημα της μειωμένης αναγνωσιμότητας μπορεί να είναι χειρότερο καθώς προκαθορισμένα ονόματα είναι mixed case (π.χ. `IndexOutOfBoundsException`)

# Ονόματα (συνέχεια)

---

- Ειδικές λέξεις
  - Ενισχύουν την αναγνωσιμότητα, χρησιμοποιούνται για το διαχωρισμό των συντακτικών τμημάτων των προτάσεων
  - Μια λέξη κλειδί *keyword* είναι μια λέξη που είναι ειδική μόνο σε συγκεκριμένα contexts
  - Μια δεσμευμένη λέξη (*reserved word*) είναι μια ειδική λέξη που δεν μπορεί να χρησιμοποιηθεί ως όνομα ορισμένο από το χρήστη
  - Πιθανό πρόβλημα με δεσμευμένες λέξεις: Αν είναι πάρα πολλές, συμβαίνουν πολλές συγκρούσεις (π.χ., η COBOL διαθέτει 300 δεσμευμένες λέξεις!)



# Μεταβλητές

---

- Μια μεταβλητή (variable) αποτελεί αφαίρεση ενός κελιού μνήμης
- Οι μεταβλητές μπορούν να περιγραφούν ως μια εξάδα χαρακτηριστικών:
  - Όνομα
  - Διεύθυνση
  - Τιμή
  - Τύπος
  - Διάρκεια ζωής
  - Εμβέλεια

# Χαρακτηριστικά Μεταβλητών

---

- Όνομα – δεν έχουν όλες οι μεταβλητές
- Διεύθυνση – Η διεύθυνση μνήμης με την οποία συσχετίζεται
  - Μια μεταβλητή μπορεί να έχει διαφορετικές διευθύνσεις σε διαφορετικά χρονικά σημεία εκτέλεσης του προγράμματος
  - Αν δύο ονόματα μεταβλητών μπορούν να χρησιμοποιηθούν για να προσπελάσουν την ίδια θέση μνήμης, τότε ονομάζονται ψευδώνυμα (aliases)
  - Τα ψευδώνυμα δημιουργούνται μέσω δεικτών, μεταβλητών αναφοράς καθώς και με unions στη C και C++
  - Τα ψευδώνυμα μπορούν να είναι έχουν αρνητική επίδραση στην αναγνωσιμότητα (οι προγραμματιστές πρέπει να τα θυμούνται όλα)

# Χαρακτηριστικά Μεταβλητών (συνέχεια)

---

- *Τύπος* – καθορίζει το εύρος τιμών των μεταβλητών και το σύνολο των λειτουργιών που ορίζονται για το συγκεκριμένο τύπο, στην περίπτωση της κινητής υποδιαστολής, ο τύπος καθορίζει επίσης και την ακρίβεια
- *Τιμή* – τα περιεχόμενα της θέσης μνήμης με την αντιστοιχείται η μεταβλητή
  - Η I-τιμή μιας μεταβλητής είναι η διεύθυνση της
  - Η r-τιμή μιας μεταβλητής είναι η τιμή της
- *Αφηρημένο κελί μνήμης*– το φυσικό κελί μνήμης ή η συλλογή κελιών μνήμης που αντιστοιχούνται με μια μεταβλητή

# Η έννοια της σύνδεσης

---

Πρόσδεση (*binding*) είναι η αντιστοίχιση ανάμεσα σε μια οντότητα και σε ένα χαρακτηριστικό, όπως η αντιστοίχιση μεταξύ μια μεταβλητής και του τύπου της ή της τιμής της, ή ακόμα και μεταξύ μιας λειτουργίας και ενός συμβόλου

- *Χρόνος πρόσδεσης (binding time)* είναι ο χρόνος κατά τον οποίο συμβαίνει η πρόσδεση

# Πιθανοί χρόνοι πρόσδεσης

---

- Κατά τη σχεδίαση της γλώσσας -- πρόσδεση τελεστών σε λειτουργίες
- Κατά την υλοποίηση της γλώσσας – πρόσδεση του τύπου κινητής υποδιαστολής σε μια αναπαράσταση
- Κατά το χρόνο μεταγλώττισης – πρόσδεση μιας μεταβλητής σε ένα τύπο στη C ή στη Java
- Κατά το χρόνο φόρτωσης – πρόσδεση μιας C ή C++ `static` μεταβλητής σε ένα κελί μνήμης
- Κατά το χρόνο εκτέλεσης – πρόσδεση μιας μη-στατικής, τοπικής μεταβλητής σε ένα κελί μνήμης

# Στατική και Δυναμική Πρόσδεση

---

- Η πρόσδεση είναι στατική (*static*) εάν συμβαίνει πριν την εκτέλεση και παραμένει αναλλοίωτη κατά τη διάρκεια της εκτέλεσης του προγράμματος.
- Η πρόσδεση είναι δυναμική (*dynamic*) αν συμβαίνει κατά την εκτέλεση ή αν μπορεί να αλλάζει κατά την εκτέλεση του προγράμματος

# Πρόσδεση Τύπων

---

- Πως ορίζεται ένας τύπος;
- Πότε πραγματοποιείται η πρόσδεση;
- Αν είναι στατική, ο τύπος μπορεί να καθορίζεται είτε μέσω άμεσης (explicit) δήλωσης είτε μέσω έμμεσης (implicit) δήλωσης

# Άμεση / Έμμεση Δήλωση

---

- Μια άμεση δήλωση *explicit declaration* είναι μια εντολή προγράμματος που χρησιμοποιείται για να δηλώσει τον τύπο μεταβλητών
- Μια έμμεση δήλωση *implicit declaration* είναι ένας μηχανισμός για προκαθορισμένη συμπεριφορά για καθορισμό τύπων μεταβλητών μέσω προκαθορισμένων μετατροπών αντί για εντολές δήλωσης
- Οι Basic, Perl, Ruby, JavaScript, PHP παρέχουν έμμεσες δηλώσεις
  - Πλεονέκτημα: Ευκολία συγγραφής (μια μικρή ευκολία)
  - Μειονέκτημα: Αξιοπιστία (αποτελεί λιγότερο πρόβλημα στην Perl)



# Άμεση/Έμμεση Δήλωση (συνέχεια)

---

- Ορισμένες γλώσσες χρησιμοποιούν type inferencing για να καθορίσουν τον τύπο των μεταβλητών
  - C# – μια μεταβλητή μπορεί να δηλωθεί με το `var` και με μια αρχική τιμή. Η αρχική τιμή θέτει τον τύπο
  - Οι Visual Basic 9.0+, ML, Haskell, και F# χρησιμοποιούν type inferencing. Το πλαίσιο της εμφάνισης μιας μεταβλητής καθορίζει και τον τύπο της

# Δυναμική πρόσδεση τύπων

---

- Δυναμική πρόσδεση τύπων (JavaScript, Python, Ruby, PHP, C# (περιορισμένο))
- Καθορισμός μέσω μιας εντολής ανάθεσης π.χ., JavaScript

```
list = [2, 4.33, 6, 8];
```

```
list = 17.3;
```

- Πλεονέκτημα: ευελιξία (generic προγραμματιστικές μονάδες)
- Μειονεκτήματα:
  - Υψηλό κόστος (έλεγχος δυναμικών τύπων και ερμηνεία)
  - Η ανίχνευση λαθών σε τύπους από τον μεταγλωττιστή γίνεται δύσκολη

# Χαρακτηριστικά μεταβλητών (συνέχεια)

---

- Προσδέσεις αποθήκευσης & διάρκεια ζωής
  - Κατανομή (allocation) – λήψη ενός κελιού από μια δεξαμενή διαθέσιμων κελιών
  - Κατάργηση κατανομής (deallocation) – τοποθέτηση ενός κελιού πίσω στη δεξαμενή διαθέσιμων κελιών
- Η διάρκεια ζωής (lifetime) μιας μεταβλητής είναι ο χρόνος κατά τον οποίο η μεταβλητή είναι προσδεμένη σε ένα συγκεκριμένο κελί δεδομένων

# Κατηγορίες μεταβλητών με βάση το χρόνο ζωής

---

- Στατικές–Προσδένονται σε κελιά μνήμης πριν ξεκινήσει η εκτέλεση και παραμένουν προσδεμένες στα ίδια κελιά μνήμης κατά τη διάρκεια της εκτέλεσης, π.χ., C και C++ `static` μεταβλητές συναρτήσεων
  - Πλεονεκτήματα: αποδοτικότητα (απευθείας διευθυνσιοδότηση), δυνατότητα συγγραφής υποπρογραμμάτων που είναι ευαίσθητα στο ιστορικό κλήσεών τους
  - Μειονέκτημα: έλλειψη ευελιξίας (δεν μπορεί να χρησιμοποιηθεί με αναδρομή)

# Κατηγορίες μεταβλητών με βάση το χρόνο ζωής

---

- Δυναμική δέσμευση στοίβας – Οι προσδέσεις αποθήκευσης δημιουργούνται για τις μεταβλητές όταν οι εντολές δήλωσής τους επεξεργαστούν.  
(Μια δήλωση επεξεργάζεται όταν ο εκτελέσιμος κώδικας που σχετίζεται με αυτή εκτελείται)
- Αν πρόκειται για βαθμωτή τιμή, όλα τα χαρακτηριστικά εκτός από τις διευθύνσεις είναι στατικά προσδεμένα
  - Οι τοπικές μεταβλητές στα C υποπρογράμματα (που δεν έχουν δηλωθεί `static`) και οι Java μέθοδοι
- Πλεονέκτημα: επιτρέπουν την αναδρομή, εξοικονομούν αποθηκευτικό χώρο
- Μειονεκτήματα:
  - Επιβάρυνση δέσμευσης και αποδέσμευσης
  - Τα υποπρογράμματα μπορεί να είναι ευαίσθητα στο ιστορικό κλήσεων τους
  - Λιγότερο αποδοτικές αναφορές στη μνήμη λόγω της έμμεσης διευθυνσιοδότησης που χρησιμοποιείται

# Κατηγορίες μεταβλητών με βάση το χρόνο ζωής

---

- *Δυναμική άμεση δέσμευση σωρού* – Η δέσμευση και η αποδέσμευση γίνεται με άμεσες οδηγίες, που ορίζονται από τον προγραμματιστή, και συμβαίνουν κατά την εκτέλεση
- Μπορούν να προσπελαστούν μόνο μέσω δεικτών ή αναφορών, π.χ. δυναμικά αντικείμενα στη C++ (μέσω `new` και `delete`), όλα τα αντικείμενα στη Java
- Πλεονέκτημα: παρέχει δυναμική διαχείριση της μνήμης
- Μειονέκτημα: Μη αποδοτικό και αναξιόπιστο

# Κατηγορίες μεταβλητών με βάση το χρόνο ζωής

---

- *Δυναμική έμμεση δέσμευση-σωρού* — Η δέσμευση και η αποδέσμευση προκαλείται από εντολές ανάθεσης
  - Όλες οι μεταβλητές στην APL, όλα τα λεκτικά και οι διατάξεις στην Perl, JavaScript, και PHP
- Πλεονέκτημα: ευελιξία (γενικός κώδικας)
- Μειονεκτήματα:
  - Μη-αποδοτικό, καθώς όλες οι μεταβλητές είναι δυναμικές
  - Απώλεια ανίχνευσης λαθών

# Χαρακτηριστικά μεταβλητών: Εμβέλεια

---

- Η εμβέλεια (scope) μιας μεταβλητής είναι το εύρος των εντολών στις οποίες η μεταβλητή είναι ορατή
- Οι τοπικές μεταβλητές (local variables) μιας ενότητας προγράμματος είναι εκείνες οι μεταβλητές που δηλώνονται μέσα στη μονάδα
- Οι μη-τοπικές (nonlocal variables) μεταβλητές μιας ενότητας προγράμματος είναι εκείνες που είναι ορατές στη μονάδα αλλά δεν έχουν δηλωθεί σε αυτή
- Οι καθολικές μεταβλητές (global variables) είναι μια ειδική κατηγορία μη-τοπικών μεταβλητών
- Οι κανόνες εμβέλειας μιας γλώσσας καθορίζουν το πως οι αναφορές σε ονόματα σχετίζονται με τις μεταβλητές



# Στατική εμβέλεια

---

- Βασίζεται στον κώδικα του προγράμματος ως κείμενο
- Για να συνδεθεί μια αναφορά ονόματος σε μια μεταβλητή, θα πρέπει ο προγραμματιστής (ή ο μεταγλωττιστής) να εντοπίσει τη δήλωση
- *Διαδικασία αναζήτησης*: αναζήτηση στις δηλώσεις, πρώτα τοπικά, στη συνέχεια σε αυξανόμενα μεγαλύτερες περικλείουσες εμβέλειες, μέχρι να βρεθεί ένα για το δεδομένο όνομα
- Περικλείουσες στατικές εμβέλειες (σε μια συγκεκριμένη εμβέλεια) είναι οι *στατικοί πρόγονοι του*, ο δε πλησιέστερος στατικός πρόγονος ονομάζεται *στατικός γονέας*
- Ορισμένες γλώσσες επιτρέπουν ένθετες δηλώσεις υποπρογραμμάτων, που δημιουργούν φωλιασμένες στατικές εμβέλειες (π.χ., Ada, JavaScript, Common Lisp, Scheme, Fortran 2003+, F#, και Python)

# Εμβέλεια (συνέχεια)

---

- Οι μεταβλητές μπορεί να κρύβονται από μια ενότητα έχοντας μια «πλησιέστερη» μεταβλητή με το ίδιο όνομα

# Μπλοκς

---

- Μια μέθοδος για τη δημιουργία στατικών εμβελειών μέσα σε προγραμματιστικές μονάδες – από την ALGOL 60
- Παράδειγμα σε C:

```
void sub() {  
    int count;  
    while (...) {  
        int count;  
        count++;  
        ...  
    }  
    ...  
}
```

- Σημείωση: έγκυρο σε C και C++, αλλά όχι σε Java και C# – πολύ επιρρεπές σε λάθη

# Η δομή LET

---

- Οι περισσότερες συναρτησιακές γλώσσες περιέχουν κάποια μορφή της δομής `let`
- Η δομή `let` έχει δύο μέρη
  - Το πρώτο μέρος προσδένει ονόματα σε τιμές
  - Το δεύτερο μέρος χρησιμοποιεί τα ονόματα που έχουν οριστεί στο πρώτο μέρος

- Στη Scheme:

```
(LET (  
  (name1 expression1)  
  ...  
  (namen expressionn)  
  expression  
)
```

# Η δομή **LET** (συνέχεια)

---

- Στην ML:

```
let  
  val name1 = expression1  
  ...  
  val namen = expressionn  
in  
  expression  
end;
```

- Στην F#:

- First part: `let left_side = expression`
- (left\_side is either a name or a tuple pattern)
- All that follows is the second part

# Σειρά Δήλωσης

---

- Η C99, C++, Java, και η C# επιτρέπουν την εμφάνιση δηλώσεων μεταβλητών να εμφανίζονται σε οποιοδήποτε θέση μπορεί να εμφανίζεται κάποια εντολή
  - Στην C99, C++, και Java, η εμφάνιση όλων των τοπικών μεταβλητών είναι από τη δήλωση και μέχρι το τέλος του μπλοκ
  - Στην C#, η εμφάνιση οποιασδήποτε μεταβλητής σε ένα μπλοκ είναι όλο το μπλοκ, άσχετα με τη θέση της δήλωσης μέσα στο μπλοκ
    - Ωστόσο, μια μεταβλητή εξακολουθεί να πρέπει να έχει δηλωθεί πριν χρησιμοποιηθεί

# Σειρά Δήλωσης (συνέχεια)

---

- Στις C++, Java, και C#, οι μεταβλητές μπορούν να δηλωθούν μέσα σε `for` εντολές
  - Η εμβέλεια αυτών των εντολών είναι περιορισμένη στη δομή `for`

# Καθολική Εμβέλεια

---

- Οι C, C++, PHP, και Python επιτρέπουν τη συγγραφή προγραμμάτων που αποτελούνται από μια σειρά ορισμών συναρτήσεων
  - Οι γλώσσες αυτές επιτρέπουν την εμφάνιση δηλώσεων μεταβλητών έξω από τις δηλώσεις των συναρτήσεων
- Η C και η C++ διαθέτουν δηλώσεις και ορισμούς
  - Μια δήλωση έξω από τον ορισμό μιας συνάρτησης καθορίζει ότι ο ορισμός γίνεται σε άλλο αρχείο



# Καθολική Εμβέλεια (συνέχεια)

---

- PHP

- Τα προγράμματα είναι ενσωματωμένα σε έγγραφα HTML markup, σε οποιοδήποτε αριθμό τμημάτων, με κάποια να είναι εντολές και κάποια να είναι ορισμοί συναρτήσεων
- Η εμβέλεια των μεταβλητών που δηλώνονται σε μια συνάρτηση είναι (υπονοούμενα) τοπική
- Η εμβέλεια των μεταβλητών που δηλώνονται εκτός συναρτήσεων είναι από τη δήλωση τους μέχρι το τέλος του προγράμματος, παρακάμπτοντας όμως τις συναρτήσεις που παρεμβάλλονται
  - Οι καθολικές μεταβλητές μπορούν να προσπελαστούν σε μια συνάρτηση μέσω του πίνακα `$GLOBALS` ή με δήλωσή τους ως `global`

# Καθολική Εμβέλεια (συνέχεια)

---

- Python
  - Μπορεί να γίνει αναφορά σε μια καθολική μεταβλητή από μια συνάρτηση, αλλά δεν μπορεί να της ανατεθεί τιμή παρά μόνο αν δηλωθεί ως `global` στην ίδια συνάρτηση

# Αποτίμηση Στατικής Εμβέλειας

---

- Λειτουργεί καλά σε πολλές περιπτώσεις
- Προβλήματα:
  - Στις περισσότερες περιπτώσεις, δίνεται υπερβολική πρόσβαση
  - Καθώς ένα πρόγραμμα αλλάζει, η αρχική δομή καταστρέφεται και οι τοπικές μεταβλητές γίνονται καθολικές, τα υποπρογράμματα επίσης τείνουν να γίνονται καθολικά, αντί να είναι φωλιασμένα

# Δυναμική εμβέλεια

---

- Βασίζεται στην ακολουθία κλήσεων των προγραμματιστικών μονάδων, και όχι στη διαμόρφωση του ίδιου του κώδικα (χρονική έναντι χωρική σχέση)
- Οι αναφορές σε μεταβλητές συνδέονται με τις δηλώσεις εξετάζοντας την αλυσίδα κλήσεων υποπρογραμμάτων που οδήγησαν την εκτέλεση σε αυτό το σημείο

# Παράδειγμα εμβέλειας

---

```
function big() {  
    function sub1() {  
        var x = 7;  
        sub2();  
    }  
    function sub2() {  
        var y = x;  
    }  
    var x = 3;  
    sub1();  
}  
big();
```

Η big καλεί  
τη sub1 που καλεί  
τη sub2 που  
Χρησιμοποιεί τη x

## – Στατική εμβέλεια

- Η αναφορά του x στο sub2 είναι στο x του big

## – Δυναμική εμβέλεια

- Η αναφορά του x στο sub2 είναι στο x ΤΟΥ sub1

# Παράδειγμα εμβέλειας

---

- Αποτίμηση της δυναμικής εμβέλειας:
  - Πλεονέκτημα: ευκολία
  - *Μειονεκτήματα:*
    1. Καθώς εκτελείται ένα υποπρόγραμμα, οι μεταβλητές του είναι ορατές σε όλα τα υποπρογράμματα που καλεί
    2. Δεν είναι δυνατός ο στατικός έλεγχος τύπων
    3. Φτωχή αναγνωσιμότητα – δεν είναι δυνατό να προσδιοριστεί στατικά ο τύπος των μεταβλητών

# Εμβέλεια και χρόνος ζωής

---

- Η εμβέλεια και η διάρκεια ζωής μερικές φορές είναι στενά συνδεδεμένες, αλλά είναι διαφορετικές έννοιες
- Θεωρείστε μια **static** μεταβλητή σε μια συνάρτηση της C ή της C++
  - Προσδένεται με στατικό τρόπο με την εμβέλεια της συνάρτησης, αλλά η μεταβλητή υπάρχει για όλη τη διάρκεια εκτέλεσης του προγράμματος.

# Περιβάλλοντα αναφοράς

---

- Το περιβάλλον αναφοράς μιας εντολής είναι η συλλογή όλων των ονομάτων που είναι ορατά σε αυτή την εντολή
- Σε μια γλώσσα στατικής-εμβέλειας, το περιβάλλον αναφοράς είναι οι τοπικές μεταβλητές και όλες οι ορατές μεταβλητές σε όλες τις περικλείουσες εμβέλειες
- Ένα υποπρόγραμμα είναι **ενεργό** αν η εκτέλεσή του έχει ξεκινήσει αλλά δεν έχει ακόμα τερματιστεί
- Σε μια γλώσσα δυναμικής-εμβέλειας, το περιβάλλον αναφοράς είναι οι τοπικές μεταβλητές συν όλες οι ορατές μεταβλητές σε όλα τα ενεργά υποπρογράμματα



# Επώνυμες Σταθερές

---

- Μια επώνυμη σταθερά είναι μια μεταβλητή που προσδένεται με μια τιμή μόνο μια φορά
- Πλεονεκτήματα: αναγνωσιμότητα και ευκολία τροποποίησης
- Χρησιμοποιούνται για την παραμετροποίηση προγραμμάτων
- Η πρόσδεση τιμών σε επώνυμες σταθερές μπορεί να είναι είτε στατική (ονομάζονται *manifest constants*) ή δυναμική
- Γλώσσες:
  - C++ και Java: εκφράσεις οποιουδήποτε τύπου, με δυναμική πρόσδεση
  - Η C# διαθέτει δύο τύπους, `readonly` και `const`
    - οι τιμές των `const` επώνυμων σταθερών προσδένονται κατά το χρόνο μεταγλώττισης
    - οι τιμές των `readonly` επώνυμων σταθερών προσδένονται δυναμικά

# Σύνοψη

---

- Η διάκριση πεζών-κεφαλαίων και η σχέση ονομάτων με ειδικές λέξεις αποτελούν θέματα σχεδίασης μιας γλώσσας προγραμματισμού
- Οι μεταβλητές χαρακτηρίζονται από την εξάδα: όνομα, διεύθυνση, τιμή, τύπος, διάρκεια ζωής, εμβέλεια
- Πρόσδεση είναι η συσχέτιση ιδιοτήτων με οντότητες του προγράμματος
- Οι βαθμωτές μεταβλητές κατηγοριοποιούνται ως: στατικές, δυναμικές στοίβας, άμεσα δυναμικές σωρού, έμμεσα δυναμικές σωρού
- Ισχυρή τυποποίηση (strong typing) σημαίνει ανίχνευση όλων των λαθών σχετικά με τους τύπους