



ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ Ι

κ. ΠΕΤΑΛΙΔΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ



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 παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Κεντρικής Μακεδονίας» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- 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.





12η Διάλεξη

Θέματα Σχεδίασης/Σύζευξη



Θέματα σχεδίασης

- Η ποιότητα μιας σχεδίασης κρίνεται τις πιο πολλές φορές από την ανεξαρτησία των κλάσεων που έχουμε δημιουργήσει
- Κλάσεις που είναι ανεξάρτητες μεταξύ τους επαναχρησιμοποιούνται πιο εύκολα, έχουν λιγότερα λάθη και κατανοούνται πιο εύκολα

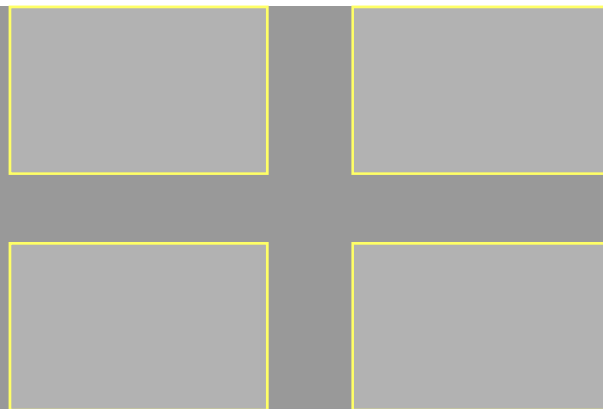


Ανεξαρτησία συστατικών

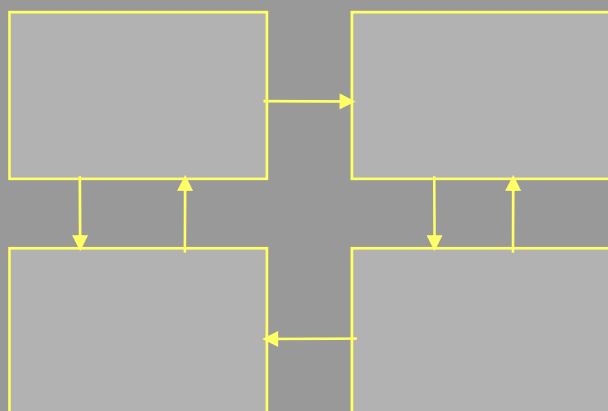
- Για να αναγνωρίσουμε και να μετρήσουμε το βαθμό ανεξαρτησίας μεταξύ των συστατικών χρησιμοποιούμε συχνά δύο έννοιες:
 - Σύζευξη
 - Μετράει την εξάρτηση που υπάρχει μεταξύ δύο συστατικών
 - Συνεκτικότητα
 - Μετράει τη συνοχή ενός συστατικού



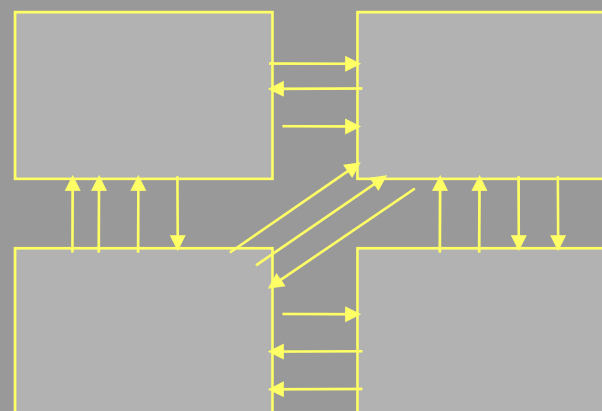
Σύζευξη



Μη συζευγμένα -
καμία εξάρτηση



Χαλαρή σύζευξη
κάποιες εξαρτήσεις



Υψηλή σύζευξη
πολλές εξαρτήσεις

Κατηγορίες σύζευξης

Σύζευξη περιεχομένου

Σύζευξη κοινών δεδομένων

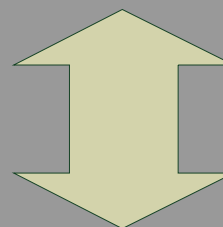
Υψηλή σύζευξη (κακή)

Σύζευξη ελέγχου

Σύζευξη αντιγράφου

Σύζευξη δεδομένων

Μη σύζευξη



Χαμηλή σύζευξη (καλή)



Σύζευξη περιεχομένου

- Όταν το ένα συστατικό α αλλάζει το περιεχόμενο ενός άλλου συστατικού β
 - Παράδειγμα 1
 - Μια συνάρτηση A αλλάζει μια εντολή που βρίσκεται σε μια συνάρτηση B .
 - Συνήθως αυτό είναι δυνατόν μόνο σε assembly
 - Παράδειγμα 2
 - Μια συνάρτηση A περιέχει μια εντολή `goto` σε μια συνάρτηση B

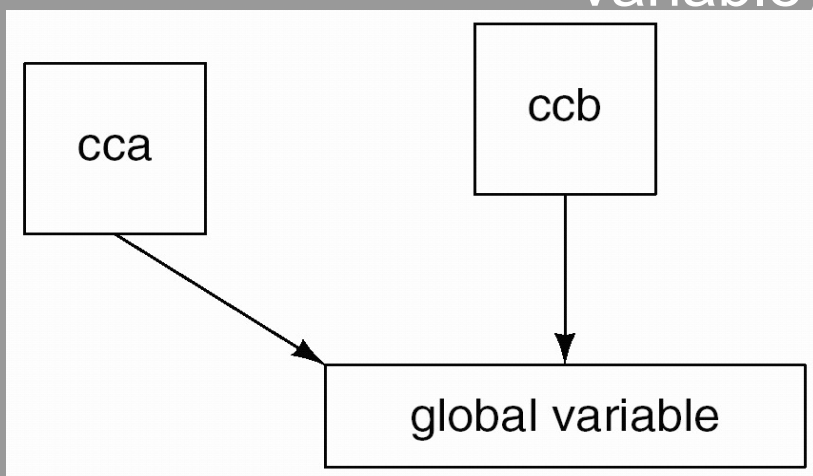
Γιατί είναι κακή;

- Σχεδόν οποιαδήποτε αλλαγή στο B, ακόμα και αλλαγή του compiler που χρησιμοποιείται προϋποθέτει αλλαγή και στο A.
- Αδύνατον να χρησιμοποιηθεί το A χωρίς να χρησιμοποιηθεί και το B



Σύζευξη κοινών δεδομένων

- Όταν δύο συστατικά μοιράζονται κοινά δεδομένα. Πχ
 - Η συνάρτηση cca και η συνάρτηση ccb μπορούν ΚΑΙ να διαβάσουν ΚΑΙ να αλλάξουν μια καθολική μεταβλητή (global variable)



Παράδειγμα

- Παράδειγμα 2
 - Δύο διαφορετικές συναρτήσεις έχουν πρόσβαση στην ίδια βάση δεδομένων και μπορούν να αλλάξουν ΤΗΝ ΙΔΙΑ ΕΓΓΡΑΦΗ
 - Προσοχή ποιοι μπορούν να αλλάζουν στη C++ τα public attributes



Γιατί είναι κακή;

- Γιατί είναι αντίθετη στις αρχές του δομημένου προγραμματισμού
- Ο κώδικας που παράγεται δε διαβάζεται
- Ποιος σταματά τον ακόλουθο βρόγχο;

```
while (global variable == 0)
{
    if (argument xyz > 25)
        module 3 ();
    else
        module 4 ();
}
```



Γιατί είναι κακή; (συνέχεια)

- Τα συστατικά (μια συνάρτηση, μια κλάση) μπορούν να έχουν “παρενέργειες”. Αυτό τα κάνει δυσανάγνωστα αφού πρέπει κανείς να διαβάσει ολόκληρο το συστατικό με προσοχή να δει αν αλλάζει κάποια καθολική μεταβλητή
- Αν αλλάξει η καθολική μεταβλητή πρέπει να αλλάξει και κάθε συστατικό που τη χρησιμοποιεί



Γιατί είναι κακή; (συνέχεια)

- Η επαναχρησιμοποίηση είναι δύσκολη
 - Πρέπει να υπάρχει το ίδιο σύνολο καθολικών μεταβλητών όποτε επαναχρησιμοποιείται το συστατικό
- Κάθε συστατικό έχει πρόσβαση σε περισσότερα δεδομένα από όσα χρειάζεται. Αυτό κάνει ευκολότερη τη δημιουργία λαθών από απροσεξία



Πότε είναι καλή

- Όταν ένα συστατικό έχει ένα τεράστιο αριθμό μεταβλητών και πρέπει να αρχικοποιηθούν, αντί να περαστούν σε παράμετροι όποτε χρησιμοποιείται το συστατικό, ορίζονται σαν καθολικές μεταβλητές και αρχικοποιούνται έτσι πιο γρήγορα



Σύζευξη ελέγχου

- Όταν το ένα συστατικό περνά παραμέτρους στο άλλο που ελέγχουν τη συμπεριφορά του μέσω flags
- Συνήθως εμφανίζεται σε συστατικά που παρουσιάζουν λογική συνεκτικότητα (δείτε τις παρακάτω διαφάνειες)



Παράδειγμα

- Η συνάρτηση A καλεί τη συνάρτηση B.
- Αν περαστεί ως παράμετρος ο αριθμός 1 η συνάρτηση B κάνει ταξινόμηση, αν περαστεί ως παράμετρος ο αριθμός 2 η συνάρτηση B διαβάζει από το δίσκο, αν περαστεί ο αριθμός 3 η B παίζει μουσική κτλ



Γιατί δεν είναι καλή

- Τα συστατικά δεν είναι ανεξάρτητα.
- Το συστατικό που καλείται πρέπει να ξέρει την εσωτερική δομή του συστατικού που το καλεί
 - Επηρεάζει την επαναχρησιμοποίηση

Σύζευξη αντιγράφου

- Πολλές γλώσσες επιτρέπουν μόνο απλές μεταβλητές σαν παραμέτρους σε συναρτήσεις ή μεθόδους
 - Το όνομα ενός υπαλλήλου
 - Ο κωδικός ενός προϊόντος
- Αλλά πολλές γλώσσες επιτρέπουν ολόκληρες δομές
 - Το struct υπάλληλος
 - Την κλάση προϊόν



Σύζευξη αντιγράφου

- Σύζευξη αντιγράφου έχουμε όταν ένα συστατικό A καλεί ένα συστατικό B και περνά μια ολόκληρη δομή σε αυτό σαν παράμετρο και επιπλέον το B χρειάζεται να ξέρει μόνο συγκεκριμένα πεδία από αυτή τη δομή



Παράδειγμα

```
function calculateWithHolding(employeeRecord R);
```

- Δε μπορείτε να πείτε χωρίς να δείτε τον κώδικα ποια πεδία του R αλλάζουν ή χρησιμοποιούνται
- Θα έπρεπε να περνάν σαν παράμετροι ΜΟΝΟ συγκεκριμένα πεδία



Γιατί είναι κακή;

- Δεν είναι εύκολο να ξεκαθαρίσεις τι κάνει το κάθε συστατικό
- Η επαναχρησιμοποίηση δεν είναι εύκολη
- Τα συστατικά αποκτούν πρόσβαση σε δεδομένα που δε θα έπρεπε



Προσοχή

- Δεν είναι κακό να περνάτε ολόκληρες δομές αν αυτό είναι απαραίτητο:

```
invertMatrix (originalMatrix, invertedMatrix);  
printRecord(warehouseRecord);
```




Σύζευξη δεδομένων

- Όταν το ένα συστατικό περνά δεδομένα στο άλλο
- Είναι η καλύτερη σύζευξη που μπορείτε να πετύχετε
 - `displayTimeOfArrival (flightNumber);`
 - `computeProduct (firstNumber, secondNumber);`
 - `getJobWithHighestPriority (jobQueue);`

Γιατί είναι καλή;

- Δεν υπάρχουν τα προβλήματα με τις συζεύξεις αντιγράφου, ελέγχου, κοινών δεδομένων και περιεχομένου
- Είναι πιο εύκολο να συντηρηθούν τα συστατικά
 - Μια αλλαγή σε ένα είναι δύσκολο να προκαλέσει προβλήματα στο άλλο



Συζητήσεις

- Ένα από τα πλεονεκτήματα των αντικειμενοστραφών συστημάτων είναι ότι παρουσιάζουν μικρή σύζευξη καθώς κάθε κλάση μπορεί και περιορίζει τις πράξεις που γίνονται στα δεδομένα της αλλά και περιέχει τους ορισμούς των πράξεων αυτών