

# Ανάπτυξη & Σχεδίαση Λογισμικού (HY420)

---

## Διάλεξη 9: Μετρικές Ποιότητας Λογισμικού



# Μετρικές Προϊόντος: Γραμμές Κώδικα



2

- Γραμμές κώδικα
  - Απλό;
  - Αποδοτικό;
  - Καλά ορισμένο;
    - Όχι!
    - Καλύτερος ορισμός (π.χ. για C/C++): Ο αριθμός των ;

# Κυκλωματικός Αριθμός (McCabe, 1976) (1/4)



3

- Ποσοτικοποιεί την **πολυπλοκότητα**
  - Παραδοχή: Πολυπλοκότητα **εξαρτάται από** τη «**ροή ελέγχου**» του προγράμματος
  - Εφαρμόζεται πάνω στο διάγραμμα ροής

$$C = e - n + 2p$$

C: Κυκλωματικός αριθμός (πολυπλοκότητα)

e: Αριθμός ακμών

n: Αριθμός κόμβων στο γράφο

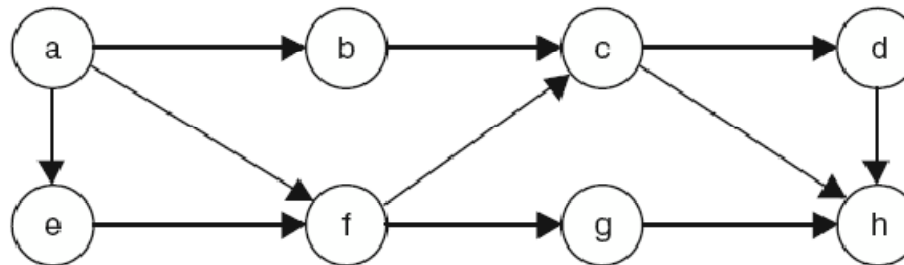
p: Αριθμός ισχυρά συνδεδεμένων τμημάτων του γράφου (κατά κανόνα 1)

- Καλός αριθμός: Έως **10!**

# Κυκλωματικός Αριθμός (McCabe, 1976) (2/4)



4



- Αριθμός κόμβων: 8
- Αριθμός ακμών: 11

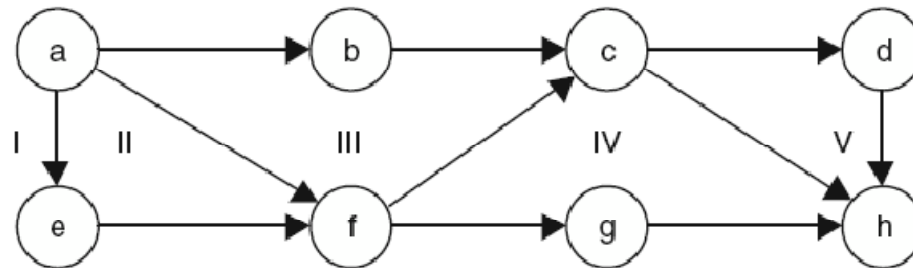
$$C = 11 - 8 + 2 = 5$$

# Κυκλωματικός Αριθμός (McCabe, 1976) (3/4)



5

- Εναλλακτικά:
  - Για επίπεδους γράφους, κυκλωματικός αριθμός = **αριθμός περιοχών** στις οποίες χωρίζει ο γράφος το επίπεδο
  - Τάδε έφη Euler...



# Κυκλωματικός Αριθμός (McCabe, 1976) (4/4)

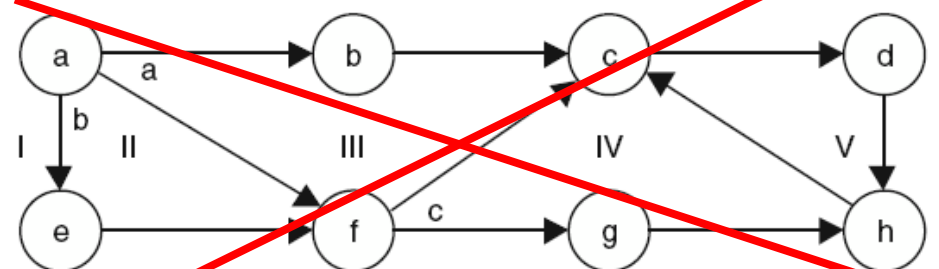
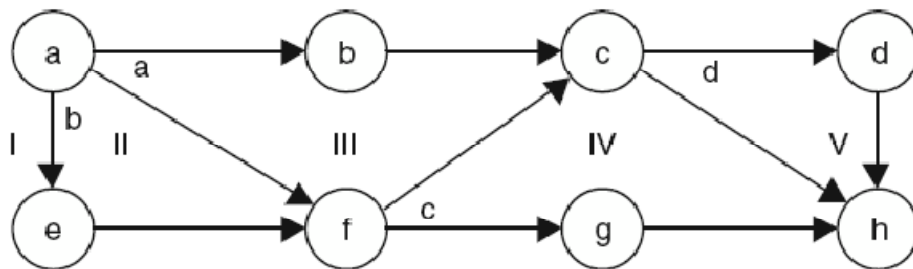


6

- Πρόβλημα:
  - Χρονοβόρο να φτιαχτεί ο γράφος
  - Χρονοβόρο να υπολογιστεί ο κυκλωματικός αριθμός από το γράφο
  - Λύση: McCabe (και πάλι)
    - Εφόσον υπάρχει τελικός κόμβος...

$$C = \pi + 1$$

$\pi$ : Ο αριθμός των εντολών απόφασης στο πρόγραμμα



# Η Επιστήμη Λογισμικού (Halstead, 1970s)



7

- Πρόγραμμα: Σύνολο από **tokens**
  - Έντελα
    - Μεταβλητές, σταθερές
    - $\eta_2$ : Αριθμός διαφορετικών εντέλων στο πρόγραμμα
  - Τελεστές
    - Όλα τα υπόλοιπα
    - $\eta_1$ : Αριθμός διαφορετικών τελεστών στο πρόγραμμα
- Tokens που εμφανίζονται πάντα σε 2αδες, 3αδες κοκ μετρώνται σαν 1
- Δε μετράμε δηλώσεις, εντολές I/O, σχόλια

$$\eta = \eta_1 + \eta_2$$

$\eta$ : Αριθμός διαφορετικών tokens (**μέτρο μεγέθους** προγράμματος)

# Η Επιστήμη Λογισμικού. Παράδειγμα



8

- Έντελα:
  - X Y Z 0 1
  - Σύνολο 5
- Τελεστές:
  - = ; while/end-while > + - ()
  - Σύνολο 7

```
Z = 0 ;  
while X > 0  
    Z = Z + Y ;  
    X = X-1 ;  
end-while ;  
print (Z) ;
```



# Η Επιστήμη Λογισμικού: Δυνατά (εφικτά) έντελα



9

- $\eta_2^*$ : Ο ελάχιστος δυνατός αριθμός εντέλων για την υλοποίηση ενός δεδομένου αλγορίθμου

# Η Επιστήμη Λογισμικού: Μέγεθος Προγράμματος



10

$$N = N_1 + N_2$$

N: Μέγεθος προγράμματος

N<sub>1</sub>: Αριθμός τελεστών

N<sub>2</sub>: Αριθμός εντέλων

Παράδειγμα:

- Τελεστές:  
= 3 ; 5 while-endwhile 1 > 1  
+ 1 - 1 () 1
- Έντελα:  
Z 4 0 2 X 3 Y 1 1 1

$$N = 13 + 11 = 24$$

```
Z = 0 ;  
while X > 0  
    Z = Z + Y ;  
    X = X - 1 ;  
end-while ;  
print (Z) ;
```

# Η Επιστήμη Λογισμικού: Εκτίμηση Μεγέθους Προγράμματος



11

$$\text{est } N = \eta_1 \log_2 \eta_1 + \eta_2 \log_2 \eta_2$$

Παράδειγμα:

$$\eta_1 = 7$$

$$\eta_2 = 5$$

$$\text{est } N = 7 \log_2 7 + 5 \log_2 5 = 24 + 11.6 = 31.26$$

- Εμπειρική παρατήρηση:
  - Αν σχετικό σφάλμα μεταξύ  $N$  και  $\text{est } N > 30\%$ , η μέθοδος επιστήμης λογισμικού μη εφαρμόσιμη

# Η Επιστήμη Λογισμικού: Όγκος Προγράμματος



12

$$V = N \log_2(\eta_1 + \eta_2)$$

- Φυσική σημασία;
  - Ο αριθμός των bits που απαιτούνται για την κωδικοποίηση του προγράμματος
- Δυνατός (εφικτός) όγκος προγράμματος  $V^*$ 
  - 2 τελεστές
    - Όνομα συνάρτησης
    - Τελεστής ομαδοποίησης

$$V^* = (2 + \eta_2^*) \log_2(2 + \eta_2^*)$$

# Η Επιστήμη Λογισμικού: Άλλες Μετρικές



13

- $L = V^* / V$ 
  - Μετράει την «**ποιότητα**» της υλοποίησης σε σχέση με την ελάχιστη δυνατή
- Μη χρησιμοποιούμενες μετρικές
  - Προσπάθεια ανάπτυξης (E)
    - $E = V / L$
    - Μονάδα: emd (Elementary Mental Discriminations)
  - Χρόνος ανάπτυξης (T)
    - $T = E / S$ 
      - S: Σταθερά Stroud, ίση με 18 emd/sec

# Ροή Πληροφορίας (Henry – Kafura)



14

- Ποσοτικοποιεί την **πολυπλοκότητα διασύνδεσης** μεταξύ διαφορετικών μονάδων λογισμικού
  - Ροή πληροφορίας ανάμεσα στις μονάδες

$$HK_i = \text{weight}_i * (\text{fanout}_i * \text{fanin}_i)^2$$

$HK_i$ : Ροή πληροφορίας module  $i$

$\text{fanout}_i$ : Αριθμός εξόδων module  $i$

$\text{fanin}_i$ : Αριθμός εισόδων module  $i$

$\text{weight}_i$ : Πολλαπλασιαστικός παράγοντας. Εξαρτάται από γραμμές κώδικα, πολυπλοκότητα module

$$HK = \sum_{i=1}^n HK_i$$



# Παραγωγικότητα

- Γραμμές Κώδικα / Εργατομέρα Προγραμματιστή
  - 60s: Ακόμα και 1 LOC/workday
  - Τυπική τιμή: 2 – 20 LOC/workday
    - Σε μικρά έργα μπορεί να είναι πολύ μεγαλύτερη

# Αντικειμενοστρεφείς Μετρικές Chidamber - Kemerer



16

- Ζυγισμένος αριθμός μεθόδων ανά κλάση (weighted methods per class – WMC)

$$WMC = \frac{\sum_{i=0}^{n-1} c_i * M_i}{n}$$

n: Αριθμός κλάσεων

M<sub>i</sub>: Μέθοδοι κλάσης i

c<sub>i</sub>: Πολυπλοκότητα κλάσης i (άγνωστο πώς υπολογίζεται...)

- Ένδειξη της πολυπλοκότητας του λογισμικού



# Αντικειμενοστρεφείς Μετρικές Chidamber - Kemerer



17

- **Βάθος δέντρου κληρονομικότητας** (Depth of Inheritance Tree – DIT)
  - Κληρονομικότητα: Προσθέτει πολυπλοκότητα στο λογισμικό
  - Μετρική: Μέγιστη **απόσταση** από ένα φύλλο (κλάση) **ως τη ρίζα** του δέντρου κληρονομικότητας
- **Αριθμός παιδιών** (Number Of Children – NOC)
  - Ποσοτικοποιεί το **πλάτος** του δέντρου κληρονομικότητας
  - Αριθμός υποκλάσεων άμεσα υποκείμενων μιας συγκεκριμένης κλάσης στο δέντρο κληρονομικότητας
  - Υπολογίζεται για κάθε κλάση

# Αντικειμενοστρεφείς Μετρικές Chidamber - Kemerer



18

- **Συσχέτιση** μεταξύ κλάσεων (Coupling Between Object classes – CBO)
  - Υπάρχει συσχέτιση εφόσον attributes / μέθοδοι μιας κλάσης χρησιμοποιούνται από μια άλλη
    - Συμμετρική σχέση!
  - Υπολογίζεται για κάθε κλάση
    - Είναι ο αριθμός των κλάσεων με τις οποίες η κλάση συσχετίζεται

# Αντικειμενοστρεφείς Μετρικές

## Chidamber - Kemerer



19

- **Απόκριση κλάσης** (Response set For Class – RFC)
  - **Σύνολο απόκρισης** κλάσης ( {RS} ): Το σύνολο των μεθόδων που μπορεί πιθανά να εκτελεστούν ως απόκριση σε ερέθισμα που θα ληφθεί από την κλάση
    - Ένωση συνόλου μεθόδων στην κλάση και μεθόδων που καλούνται από αυτές
      - Σε ένα επίπεδο μόνο

$$RFC = | RS |$$

- Υπολογίζεται για κάθε κλάση

# Αντικειμενοστρεφείς Μετρικές

## Chidamber - Kemerer



20

- Έλλειψη συνοχής στις μεθόδους (Lack of Cohesion Of Methods – LCOM)
  - Ποσοτικοποιεί τη συνοχή (την έλλειψή της) στις μεθόδους

$$LCOM = \max (|P| - |Q|, 0)$$

$I_i$ : Σύνολο κοινόχρηστων attributes που χρησιμοποιούνται από τη μέθοδο  $i$

$P$ : Το σύνολο των κενών τομών ανά δύο των  $I_i$

$Q$ : Το σύνολο των μη κενών τομών ανά δύο

των  $I_i$

# Αντικειμενοστρεφείς Μετρικές

## Chidamber - Kemerer



21

- LCOM με απλά ελληνικά...
  - Διμερής γράφος
  - Αριστερό σύνολο κόμβων: μεταβλητές
    - Έστω πλήθος  $m$
  - Δεξιό σύνολο κόμβων: μέθοδοι
    - Έστω πλήθος  $n$
  - Ακμές: Χρήση μεταβλητής από μέθοδο (σύνολο  $Q$ )

$$|P| = m * n - |Q| = m * n - \text{ακμές}$$

$$LCOM = \max(|P| - |Q|, 0) = \max(m * n - 2 * \text{ακμές}, 0)$$

# Αντικειμενοστρεφείς Μετρικές

## Chidamber – Kemerer:

### Παράδειγμα



22

```
class point {
    float x;
    float y;
public:
    point(float newx, float newy) {x=newx; y=newy;}
    getx(){return x;}
    gety(){return y;}
};

class rectangle {
    point pt1, pt2, pt3, pt4;
public:
    rectangle(float pt1x, pt1y, pt2x, pt2y, pt3x, pt3y, pt4x, pt4y)
        { pt1 = new point(pt1x, pt1y); pt2 = new point(pt2x, pt2y);
          pt3 = new point(pt3x, pt3y); pt4 = new point(pt4x, pt4y); }
    float length(point r, point s) {return sqrt((r.getx()-s.getx())^2+
        (r.gety()-s.gety())^2); }
    float area(){return length(pt1,pt2) * length(pt1,pt3); }
};
```

# Αντικειμενοστρεφείς Μετρικές

## Chidamber – Kemerer:

### Παράδειγμα



23

```
class linklistnode {
    rectangle* node;
    linklistnode* next;
public:
    linklistnode(rectangle* newRectangle) {node=newRectangle; next=0;}
    linklistnode* getNext() {return next;}
    rectangle* getRectangle() {return node;}
    void setnext(linklistnode* newnext) {next=newnext;}
};

class rectanglelist {
    linklistnode* top;
public:
    rectanglelist() {top = 0;}
    void addRectangle(float x1, y1, x2, y2, x3, y3, x4, y4) {
        linklistnode* tempLinkListNode; rectangle* tempRectangle;
        tempRectangle = new rectangle(x1,y1,x2,y2,x3,y3,x4,y4);
        tempLinkListNode = new linkListNode(tempRectangle);
        tempLinkListNode->setnext(top);
        top=tempLinkListNode; }
    float totalArea(){float sum; sum=0; linklistnode* temp; temp=top;
        while (temp !=0){sum=sum + temp->getRectangle()->area();
            temp=temp->getNext();}
        return sum;}
};
```

# Αντικειμενοστρεφείς Μετρικές

## Chidamber – Kemerer:

### Παράδειγμα



24

- WMC

Κλάση	Μέθοδοι
-------	---------

point	3
-------	---

rectangle	3
-----------	---

linklistnode	4
--------------	---

rectanglelist	3
---------------	---

$$WMC = \frac{1}{4} * (3+3+4+3) = 3.25 \text{ μέθοδοι/κλάση}$$



# Αντικειμενοστρεφείς Μετρικές

## Chidamber – Kemerer:

### Παράδειγμα



25

- DIT:
  - Δεν υπάρχει κληρονομικότητα
- NOC:
  - Δεν υπάρχει κληρονομικότητα

# Αντικειμενοστρεφείς Μετρικές

## Chidamber – Kemerer:

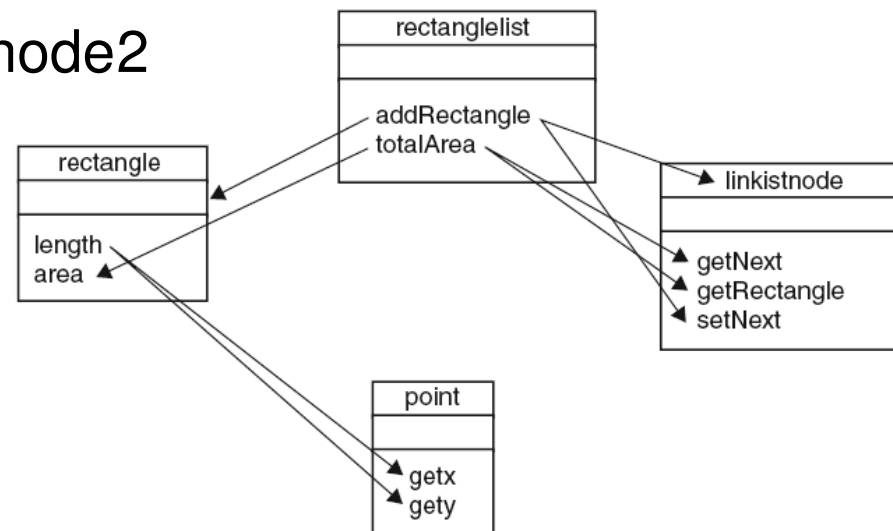
### Παράδειγμα



26

- Συσχέτιση μεταξύ κλάσεων (CBO)

Κλάση	Σχετιζόμενες	CBO
point	rectangle	1
rectangle	point, rectanglelist	2
linklistnode	rectanglelist	1
rectanglelist	rectangle, linklistnode2	



# Αντικειμενοστρεφείς Μετρικές

## Chidamber – Kemerer:

### Παράδειγμα



27

- RFC

Κλάση	Σύνολο απόκρισης	RFC
point	point, getX, getY	3
rectangle	rectangle, point, length getX, getY, area	6
linklistnode	linklistNode, getNext getRectangle, setnext	4
rectanglelist	rectanglelist, addRectangle rectangle, setnext, totalArea getRectangle, area, getNext, linkListNode	9

# Αντικειμενοστρεφείς Μετρικές

## Chidamber – Kemerer:

### Παράδειγμα



28

- LCOM

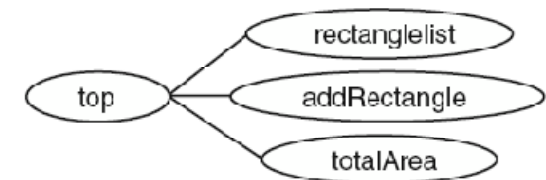
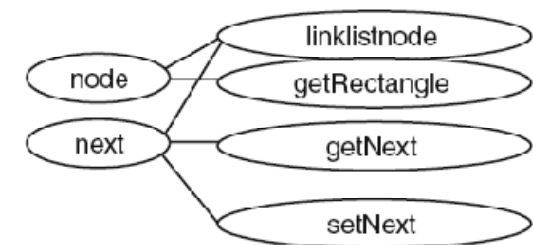
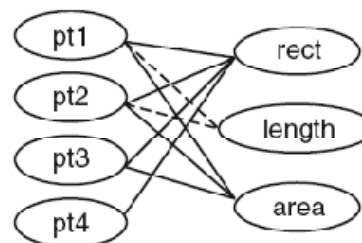
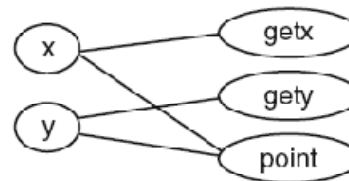
#### Κλάση LCOM

point  $\max((6-4)-4, 0)=0$

rectangle  $\max((12-9)-9, 0)=0$

linklistnode  $\max((8-5)-5, 0)=0$

rectanglelist  $\max((3-3)-3, 0)=0$



# Αντικειμενοστρεφείς Μετρικές MOOD



29

- **Ορατότητα** (visibility)

TC: Αριθμός κλάσεων στο σύστημα

$M_d(C_i)$ : Αριθμός μεθόδων στην κλάση  $C_i$

$M_{m,i}$ : Μέθοδος  $m$  κλάσης  $i$

$Is\_Visible(M_{m,i}, C_j) =$  1 αν  $i \neq j$  και η  $M_{m,i}$  είναι διαθέσιμη  
στη  $C_j$   
0 διαφορετικά

$$V(M_{m,i}) = \frac{\sum_{j=1}^{TC} Is_{Visible}(M_{m,i}, j)}{TC - 1}$$

# Αντικειμενοστρεφείς Μετρικές MOOD: Εμφώλευση



30

- Βαθμός απόκρυψης μεθόδων (MHF)

$$MHF = \frac{\sum_{i=1}^{TC} \sum_{m=1}^{M_d(C_i)} (1 - V(M_{m,i}))}{\sum_{i=1}^{TC} M_d(C_i)}$$

- Βαθμός απόκρυψης χαρακτηριστικών (AHF)

$$AHF = \frac{\sum_{i=1}^{TC} \sum_{m=1}^{A_d(C_i)} (1 - V(A_{m,i}))}{\sum_{i=1}^{TC} A_d(C_i)}$$

# Αντικειμενοστρεφείς Μετρικές

## MOOD: Παράδειγμα



31

```

Class A{
    int a;
public:
    void x();
    void y();
};
Class B {
    int b;
    int bb;
    void w();
public:
    void z();
};
Class C {
    int c;
    void v();
};
    
```

TC = 3

Method	is_vis(A)	is_vis(B)	is_vis(C)	V(M <sub>m,i</sub> )
A::x()	0	1	1	1
A::y()	0	1	1	1
B::w()	0	0	0	0
B::z()	1	0	1	1
C::v()	0	0	0	0

$$\text{MHF} = \{[(1-1)+(1-1)]+[(1-0)+(1-1)]+(1-0)\} / (2+2+1) = 2/5 = 0.4$$

Attribute	A(M <sub>m,i</sub> )			
A.a	0	0	0	0
B.b	0	0	0	0
B.bb	0	0	0	0
C.c	0	0	0	0

$$\text{AHF} = \{(1-0)+[(1-0)+(1-0)]+(1-0)\} / (1+2+1) = 4/4 = 1.0$$

# Αντικειμενοστρεφείς Μετρικές

## MOOD: Κληρονομικότητα



32

- Βαθμός κληρονομικότητας μεθόδων (MIF)  
 $M_d(C_i)$ : Αριθμός μεθόδων που δηλώθηκαν στην κλάση  $i$   
 $M_i(C_i)$ : Αριθμός μεθόδων που κληρονομήθηκαν στην κλάση  $i$   
 $M_a(C_i) = M_d(C_i) + M_i(C_i)$

$$MIF = \frac{\sum_{i=1}^{TC} M_i(C_i)}{\sum_{i=1}^{TC} M_a(C_i)}$$

- Βαθμός κληρονομικότητας χαρακτηριστικών (AIF)
  - Αντίστοιχα

$$AIF = \frac{\sum_{i=1}^{TC} A_i(C_i)}{\sum_{i=1}^{TC} A_a(C_i)}$$



# Αντικειμενοστρεφείς Μετρικές

## MOOD: Παράδειγμα



33

```

Class A{
protected:
    int a;
public:
    void x();
    virtual void y();
};
Class B public A {
    int b;
protected:
    int bb;
public:
    void z();
    void y();
    void w();
};
Class C public B {
    int c;
    void v();
};
    
```

**Κλάση**  $M_d$

$M_i$

$A_d$

$A_i$

A      x(), y()

B      w(), z(), y()

C      v()

A::x()

B::w(),z(),y()

A::x()

a

b, bb

c

A.a

B.bb

$$MIF = (0 + 1 + 4) / [(0 + 1 + 4) + (2 + 3 + 1)] = 5/11$$

$$AIF = (0 + 1 + 1) / [(0 + 1 + 1) + (1 + 2 + 1)] = 2/6 = 1/3$$

# Αντικειμενοστρεφείς Μετρικές

## MOOD: Βαθμός Συσχέτισης



34

- Υπολογίζει τη **συσχέτιση** μεταξύ κλάσεων (CF)

- Εξαιρείται η συσχέτιση λόγω κληρονομικότητας

$is\_client(c_i, c_j) = 1$  αν η κλάση  $i$  έχει σχέση με την  $j$   
0 διαφορετικά

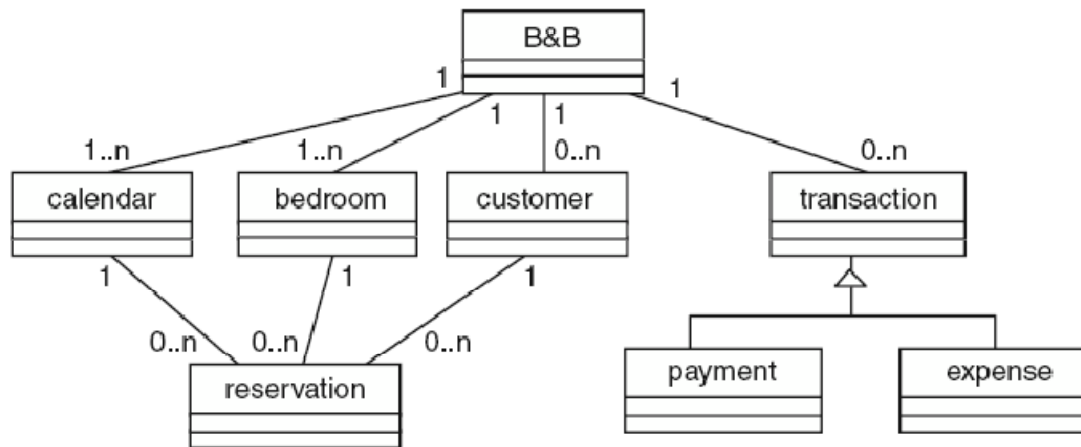
- Σχέση: Κλήση μεθόδου κλάσης  $j$ , αναφορά στην κλάση  $j$ , προσπέλαση χαρακτηριστικού κλάσης  $j$

$$CF = \frac{\sum_{i=1}^{TC} \sum_{j=1}^{TC} is_{client}(c_i, c_j)}{TC^2 - TC}$$

# Αντικειμενοστρεφείς Μετρικές MOOD: Παράδειγμα



35



## Κλάση

B&B  
calendar  
bedroom  
customer  
Transaction  
reservation  
payment  
expense

## Κλάσεις is\_client

calendar, bedroom, customer, transaction  
reservation  
reservation  
reservation

$$CF = (4+1+1+1+0+0+0) / (8^2-8) = 7/56 = 1/8$$

# Αντικειμενοστρεφείς Μετρικές

## MOOD: Βαθμός Πολυμορφισμού



36

- Μέτρο «**δυνατότητας**» **πολυμορφισμού**

$M_o(C_i)$ : Αριθμός overriding μεθόδων κλάσης  $i$

$M_n(C_i)$ : Αριθμός νέων μεθόδων κλάσης  $i$

$DC(C_i)$ : Αριθμός απογόνων κλάσης  $i$

$$PF = \frac{\sum_{i=1}^{TC} M_o(C_i)}{\sum_{i=1}^{TC} (M_n(C_i) DC(C_i))}$$

# Αντικειμενοστρεφείς Μετρικές MOOD: Παράδειγμα



37

```
Class A{
protected:
    int a;
public:
    void x();
    virtual void y();
};
Class B public A {
    int b;
protected:
    int bb;
public:
    void z():
    void y();
    void w();
};
Class C public B {
    int c;
    void v();
};
```

**Κλάση**

**$M_n$**

**$M_o$**

**DC**

A

x(), y()

2

B

w(), z()

y()

1

C

v()

0

$$PF = (0 + 1 + 0) / (2*2 + 2*1 + 0*1) = 1/6$$