

(ΕΛ/ΛΑΚ) Ελεύθερο Λογισμικό / Λογισμικού Ανοικτού Κώδικα

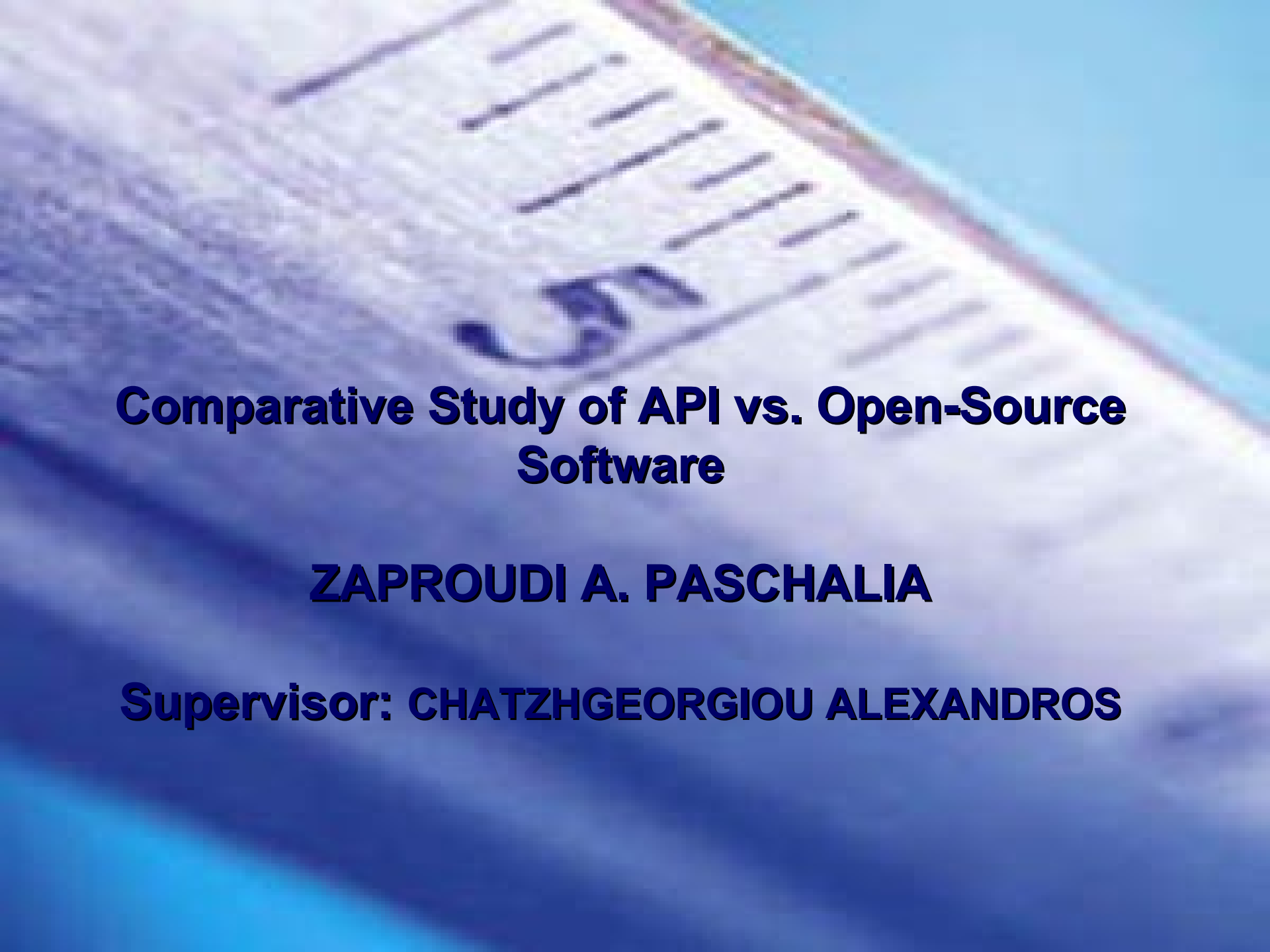


## Συνέδριο ΕΛ/ΛΑΚ 2010

Επιχειρηματικότητα,  
Δημόσιος Τομέας,  
Εκπαίδευση & Έρευνα

Παρασκευή 14 και Σάββατο 15 Μαΐου  
**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ:**  
Πολυτεχνειούπολη Ζωγράφου,  
Αίθουσα Τελετών

εγγραφές: [conf.ellak.gr](http://conf.ellak.gr)



**Comparative Study of API vs. Open-Source  
Software**

**ZAPROUDI A. PASCHALIA**

**Supervisor: CHATZHGEORGIU ALEXANDROS**

# ΕΙΣΑΓΩΓΗ

*«Κάθε στοιχείο σε μία βιβλιοθήκη γράφεται για να διατηρηθεί στον χρόνο» J. Tulach. [1]*

Η διαδικασία της σχεδίασης, πρέπει να είναι πρωταρχικό μέλημα για να καλύψει:

- Τις νέες απαιτήσεις των πελατών.
- Την συμβατότητα με προηγούμενες εκδόσεις.

# Κατηγορίες Λογισμικού

## Βιβλιοθήκες ανοικτού κώδικα:

*«Τα APIs είναι γραμμένα από έμπειρους για μη έμπειρους»  
(APIs are written by experts for non experts).*

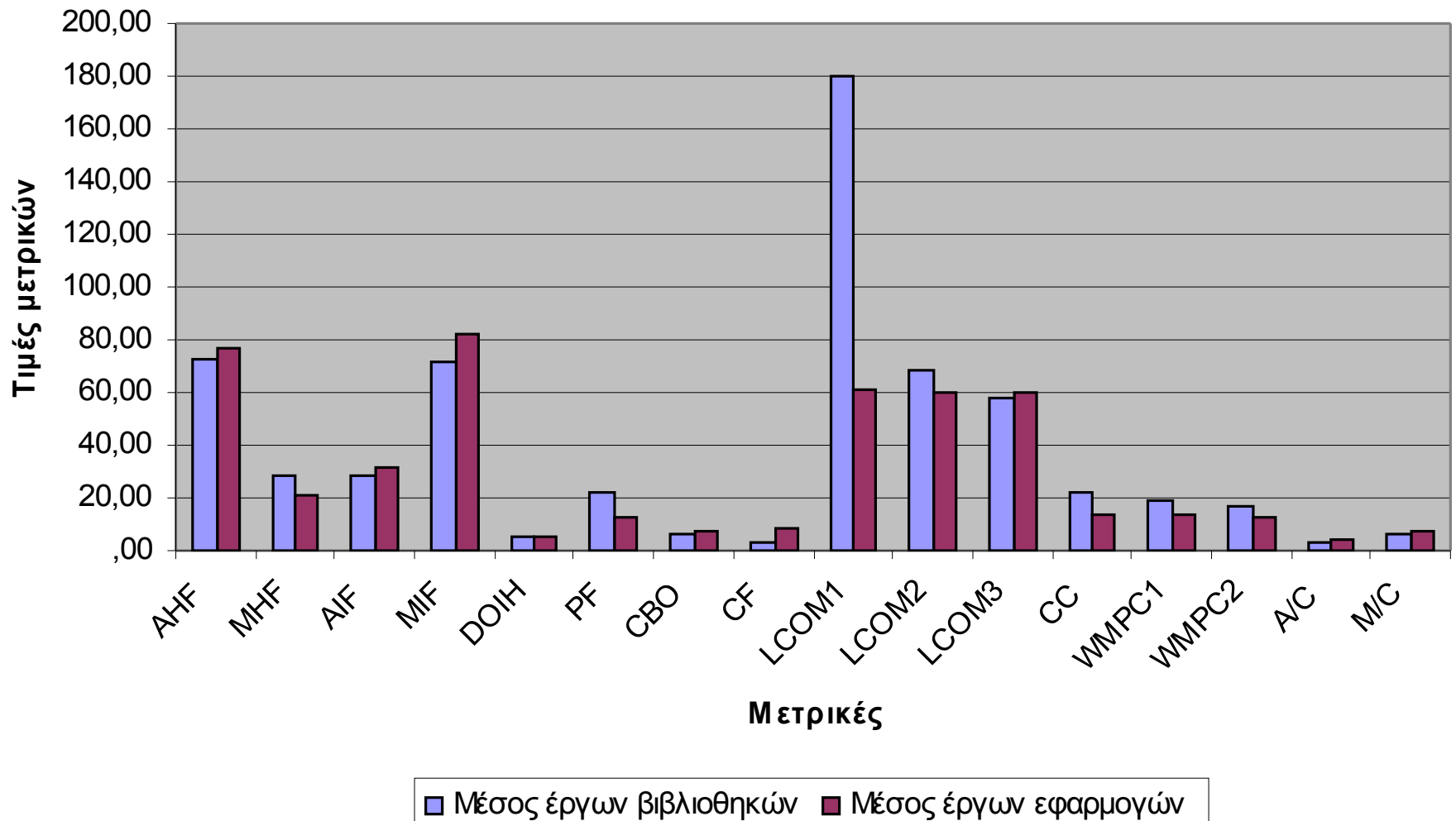
- Πιθανή εσωτερική πολυπλοκότητα, που όμως ποτέ δεν φτάνει στον τελικό χρήστη-προγραμματιστή εξαιτίας της έννοιας της αφαίρεσης (abstraction).
- Παραδείγματα προσεγγμένης σχεδίασης.
- Άριστη τεκμηρίωση (documentation)

## Εφαρμογές ανοικτού κώδικα:

- Μπορούν να χρησιμοποιηθούν από οποιονδήποτε χρήστη.
- Δεν χαρακτηρίζονται από υψηλής ποιότητας αρχιτεκτονική σχεδίαση.

# ΣΥΓΚΡΙΤΙΚΗ ΜΕΛΕΤΗ

Σύγκριση Μέσων



# ΠΕΡΙΓΡΑΦΙΚΑ ΜΕΤΡΑ

## Έργα Βιβλιοθηκών

	Minimum	Maximum	Mean	Std. Deviation
AHF	41	91	73,08	15,171
MHF	3	58	28,24	15,611
AIF	0	88	28,80	22,405
MIF	7	92	71,60	19,166
DOIH	1	7	4,80	1,472
PF	4	100	22,52	22,089
CF	1	11	3,48	2,519
CBO	0	14	6,16	3,105
LCOM1	6	1234	179,64	261,384
LCOM2	50	82	68,04	7,966
LCOM3	35	72	58,16	11,119
CC	5	85	22,24	17,064
WMPC1	5	77	19,24	14,495
WMPC2	3	65	17,00	12,563
A/C	0,7	5,6	2,748	1,188
M/C	2,3	12,0	6,156	2,619

## Έργα Εφαρμογών

	Minimum	Maximum	Mean	Std. Deviation
AHF	56	98	76,72	11,219
MHF	1	44	21,52	11,244
AIF	0	77	31,72	20,070
MIF	57	97	81,84	11,205
DOIH	3	9	5,76	1,877
PF	0	41	12,16	12,555
CF	0	38	8,68	11,679
CBO	4	13	7,68	2,780
LCOM1	19	168	60,84	43,108
LCOM2	23	88	60,36	12,910
LCOM3	29	75	59,48	9,161
CC	7	35	14,20	6,410
WMPC1	7	33	13,64	6,324
WMPC2	7	28	12,44	5,245
Attr/Class	1,3	12,6	4,404	2,583
Meth/class	3,1	16,2	7,38	3,267

# Descriptives

Έργα Βιβλιοθηκών				Έργα Εφαρμογών			
			Statistic				Statistic
<b>MIF</b>	Mean		<b>71,60</b>	<b>MIF</b>	Mean		<b>81,84</b>
	95% Confidence Interval for Mean	Lower Bound	63,69		95% Confidence Interval for Mean	Lower Bound	77,21
		Upper Bound	79,51			95% Confidence Interval for Mean	Upper Bound
	5% Trimmed Mean		<b>73,76</b>		5% Trimmed Mean		<b>82,36</b>
Interquartile Range		22	Interquartile Range		16		
<b>PF</b>	Mean		<b>22,52</b>	<b>PF</b>	Mean		<b>12,16</b>
	95% Confidence Interval for Mean	Lower Bound	13,40		95% Confidence Interval for Mean	Lower Bound	6,98
		Upper Bound	31,64			95% Confidence Interval for Mean	Upper Bound
	5% Trimmed Mean		<b>19,69</b>		5% Trimmed Mean		<b>11,27</b>
Interquartile Range		25	Interquartile Range		21		
<b>LCOM2</b>	Mean		<b>68,04</b>	<b>LCOM2</b>	Mean		<b>60,36</b>
	95% Confidence Interval for Mean	Lower Bound	64,75		95% Confidence Interval for Mean	Lower Bound	55,03
		Upper Bound	71,33			95% Confidence Interval for Mean	Upper Bound
	5% Trimmed Mean		<b>68,23</b>		5% Trimmed Mean		<b>60,84</b>
Interquartile Range		13	Interquartile Range		13		
<b>CC</b>	Mean		<b>22,24</b>	<b>CC</b>	Mean		<b>14,20</b>
	95% Confidence Interval for Mean	Lower Bound	15,20		95% Confidence Interval for Mean	Lower Bound	11,55
		Upper Bound	29,28			95% Confidence Interval for Mean	Upper Bound
	5% Trimmed Mean		<b>20,07</b>		5% Trimmed Mean		<b>13,54</b>
Interquartile Range		15	Interquartile Range		6		

# Συνοπτικά αποτελέσματα του t-test

Independent Samples Test							
	t-test for Equality of Means						
						95% Confidence Interval of the Difference	
	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
MIF	-2,306	48	,025	-10,240	4,440	-19,168	-1,312
DOIH	-2,012	48	,050	-,960	,477	-1,919	-,001
PF	2,039	48	,047	10,360	5,082	,143	20,577
CF	-2,176	26,228	,039	-5,200	2,389	-10,110	-,290
LCOM1	2,242	25,305	,034	118,800	52,983	9,746	227,854
LCOM2	2,531	48	,015	7,680	3,034	1,580	13,780
CC	2,205	30,640	,035	8,040	3,646	,601	15,479
A_C	-2,913	33,713	,006	-1,6560	,5685	-2,8117	-,5003





# ΣΥΜΠΕΡΑΣΜΑΤΑ

## ● Κληρονομικότητα (Inheritance)

*«Μην εκθέτεις μεγάλες σε βάθος ιεραρχίες, είναι απίθανο να βελτιώσεις την χρησιμοποίηση του API» (Don't expose a deep hierarchy of classes, is unlikely to improve an API's usability)*

*J. Tulach.[1]*

Η μετρική MIF(Method Inheritance Factor) για την μέτρηση του βαθμού χρησιμοποίησης της Κληρονομικότητας παρουσίασε σημαντική διαφορά στην σύγκριση των έργων των δύο κατηγοριών λογισμικού.

- **Λογισμικό βιβλιοθηκών:** υπάρχουν σοβαρές ενδείξεις ότι χρησιμοποιούν σε μικρότερο βαθμό την κληρονομικότητα. (mean=71.60)
- **Λογισμικό εφαρμογών:** σοβαρές ενδείξεις για πλεονασματική κληρονομικότητα (mean=81.84), η οποία οδηγεί σε δυσκολία συντήρησης, ελέγχου ατελειών και αποσφαλμάτωσης, καθώς και δυσκολία επέκτασης.

## ● Πολυμορφισμός ( Polymorphism )

*«Αναφέρομαι σε αντικείμενα με τα interfaces τους» (Refer to objects by their interfaces).*

*J. Bloch [2]*

Έργα που ακολουθούν αυτή την αρχή, μπορούν ευκολότερα να επεκταθούν, παρουσιάζουν μεγαλύτερο βαθμό επαναχρησιμοποίησης και οπωσδήποτε καλύτερη συντήρηση του συστήματος.

Νέες κλάσεις μπορούν να προστεθούν με λίγες ή καθόλου τροποποιήσεις. Αυτό δίνει μεγάλη «εκφραστική» δυνατότητα στους προγραμματιστές.

Από τις μετρήσεις που έδωσε η μετρική PF (Polymorphism Factor) παρατηρείται:

- **τα έργα του λογισμικού εφαρμογών** φαίνεται να μην εκμεταλλεύονται πολύ τον μηχανισμό του Πολυμορφισμού. (mean= 12.16)
- αντίθετα με **τα έργα του λογισμικού βιβλιοθηκών** για τα οποία διαπιστώνεται ότι τον χρησιμοποιούν σε πολύ καλό βαθμό. (mean=22.52)

## ● Σύζευξη (coupling) ή Εξάρτηση (dependency)

*«Το API πρέπει να είναι τόσο μικρό, όσο είναι δυνατό αλλά όχι μικρότερο.» (API Should Be As Small As Possible But No Smaller).*

*J. Bloch [2]*

Σ' ένα υψηλής σύζευξης (coupling) σύστημα αν χρειαστεί να τροποποιήσουμε ένα στοιχείο του, τότε ενδεχομένως να «παραβιαστεί» όλο το σύστημα. Οι εξαρτήσεις μεταξύ των μονάδων ενός συστήματος ελαττώνονται, όσο το μέγεθος κάθε τμήματός του μειώνεται.

## **Αυξημένη σύζευξη**, σημαίνει:

- μικρό βαθμό ενθυλάκωσης (encapsulation)
- μείωση του χαρακτηριστικού της ευαναγνωσιμότητας (readability).
- δημιουργεί δυσκολία στην συντήρηση (maintenance) και στον έλεγχο ατελειών, λόγω των πολλών εξαρτήσεων μεταξύ των μονάδων.
- δυσκολία επέκτασης (extensibility)
- δυσκολία επαναχρησιμοποίησης (reusability).

Η μετρική CF (Coupling Factor), για την μέτρηση του βαθμού της Σύζευξης παρουσίασε στατιστικά σημαντική διαφορά ( $p\text{-value}=0.39$  σε επίπεδο στατιστικής σημαντικότητας 5%) στην σύγκριση των έργων των δύο κατηγοριών λογισμικού.

➤ Στα **έργα του λογισμικού εφαρμογών** ( $\text{mean}=8.68$ ) φαίνεται να υπάρχει μεγάλη σύζευξη μεταξύ των μονάδων τους, το οποίο είναι σημάδι «κακής» σχεδίασης.

➤ Στα **έργα του λογισμικού βιβλιοθηκών** ( $\text{mean}=3.48$ ) παρατηρείται μικρότερη σύζευξη.

## ● Συνοχή ή Συνεκτικότητα (Cohesion) – Πολυπλοκότητα

*«Μία καλά σχεδιασμένη βιβλιοθήκη θα πρέπει να αποκρύπτει κάθε είδους περίπλοκων και επικίνδυνων σημείων για τους χρήστες της». [2]*

Έτσι στα έργα του λογισμικού βιβλιοθηκών περιμένουμε πιθανώς σε μεγαλύτερο βαθμό την πολυπλοκότητα και ενδεχομένως σε μικρότερο βαθμό την συνεκτικότητα για κάθε κλάση.



Συνδυάζοντας τα αποτελέσματα που έδωσε η στατιστική ανάλυση σε ότι αφορά την συνεκτικότητα (Μετρική LCOM2 (Lack of Cohesion Of Method) ) και την κυκλωματική πολυπλοκότητα (Μετρική CC (Cyclomatic Complexity)) παρατηρούμε:

➤ Για τα **έργα βιβλιοθηκών** υπάρχουν σοβαρές ενδείξεις ότι τα τμήματά τους, παρουσιάζουν μικρότερη συνεκτικότητα ( $mean_{LCOM2}=68.04$ ) και μεγαλύτερη πολυπλοκότητα ( $mean_{CC}=22.24$ ).

➤ Για τα **έργα εφαρμογών** έχουμε μεγαλύτερη συνεκτικότητα ( $mean_{LCOM2}=60.36$ ) και μικρότερη πολυπλοκότητα ( $mean_{CC}=14.20$ )



Όπως διαπιστώθηκε κατά την εκπόνηση αυτής της εργασίας, είναι εμφανής η έλλειψη της πειραματικής αξιολόγησης σε σημαντικούς τομείς της αντικειμενοστρεφούς σχεδίασης και κατ' επέκταση της αντικειμενοστρεφούς τεχνολογίας.

Σαν συνέχεια αυτής της έρευνας:

- Θα μπορούσε να ερευνηθεί, το αν παραβιάζονται ή κατά πόσο εφαρμόζονται ευρετικοί κανόνες στα έργα καθώς και αν και κατά πόσο αξιοποιούνται τα διάφορα πρότυπα σχεδίασης.
- Η συνέχιση της έρευνας μπορεί να αφορά όχι μόνο επιλεγμένες γενιές των προϊόντων λογισμικού που θα επιλεγούν αλλά και την εξέλιξη που παρουσιάζουν τα έργα εξετάζοντας πολλαπλές γενιές.
- Η μελέτη πολλαπλών γενεών λογισμικού ενδέχεται να αποκαλύψει διαφορές σχετικά με την εξέλιξη της ποιότητας του λογισμικού σε βιβλιοθήκες και έργα εφαρμογών ανοικτού κώδικα.



# ΑΝΑΦΟΡΕΣ

- [ 1 ] Tulach Jaroslav, «Practical API Design, Confessions of a Java Framework Architect», Apress, 233 Spring Street, New York, 2008
  
- [ 2 ] Bloch Joshua, “Effective Java”, The Java series, Addison-Wesley, Upper Saddle River, NJ, 2001
  
- [ 3 ] Victor Basili, Lionel Briand and Walcelio Melo. “A Validation of Object-Oriented Design Metrics as Quality Indicators”, IEEE Transactions on Software Engineering. Vol. 22, No. 10, October 1996