



# Τεχνολογία Λογισμικού

Ενότητα #5 : Έλεγχος ορθής λειτουργίας λογισμικού

Σταμέλος Ιωάννης  
Τμήμα Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



# 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 Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο την 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.





ΑΡΙΣΤΟΤΕΛΕΙΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΧΤΑ  
ΑΚΑΔΗΜΑΪΚΑ  
ΜΑΘΗΜΑΤΑ



# ΕΛΕΓΧΟΣ ΟΡΘΗΣ ΛΕΙΤΟΥΡΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ (Software Testing)

# Περιεχόμενα ενότητας

1. Έλεγχος Ορθής Λειτουργίας Λογισμικού (Software Testing).
2. Test Cases.
3. Βασικές Αρχές Δυναμικού Ελέγχου.
4. Είδη Δυναμικού Ελέγχου.
5. Στρατηγικές Δυναμικού Ελέγχου.





ΑΡΙΣΤΟΤΕΛΕΙΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΟΝΙΚΗΣ

---

# ΕΛΕΓΧΟΣ ΟΡΘΗΣ ΛΕΙΤΟΥΡΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ (Software Testing)

# ΕΛΕΓΧΟΣ ΟΡΘΗΣ ΛΕΙΤΟΥΡΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ (Software Testing)

- ΣΤΑΤΙΚΟΣ ΕΛΕΓΧΟΣ: Ο κώδικας **εξετάζεται** για τον εντοπισμό λαθών ή αδύναμων σημείων.
- ΔΥΝΑΜΙΚΟΣ ΕΛΕΓΧΟΣ: Ένα συστατικό κώδικα **εκτελείται** με ένα σύνολο από δεδομένα για τα οποία τα αποτελέσματα είναι γνωστά (δοκιμή, test): αποκάλυψη ύπαρξης λαθών και όχι απόδειξη απουσίας λαθών.
- Δοκιμαστικά δεδομένα (test, data), περιπτώσεις ελέγχου (test cases).



# ΠΡΟΒΛΗΜΑ ΕΛΕΓΧΟΥ

- Έστω ένα πρόγραμμα που δέχεται στην είσοδό του 3 ακέραιους αριθμούς που αντιπροσωπεύουν μήκη γεωμ. τμημάτων και αποφασίζει αν μπορούν αυτά να είναι οι πλευρές ενός τριγώνου, και τι είδους είναι το τρίγωνο (ισοσκελές, ορθογώνιο, κλπ).
- Να δοθεί ένα σύνολο εισόδων που θα εγγυάται την ορθή λειτουργία αυτού του προγράμματος για οποιονδήποτε συνδυασμό 3 τμημάτων.





# Test Cases -1-

- 1. Valid scalene (5, 3, 4) => orthogonal.
- 2. Valid isosceles (3, 3, 4) => isosceles.
- 3. Valid equilateral (3, 3, 3,) => equilateral.
- 4. First permutation of 2 sides (50, 50, 25) => isosceles.
- 5. Second perm of 2 sides (25, 50, 50) => isosceles.
- 6. Third perm of 2 sides (50, 25, 50) => isosceles.
- 7. One side zero (1000, 1000, 0) => invalid.



# Test Cases -2-

- 8. One side has negative length (3, 3, -4) => invalid.
- 9. first perm of two equal sides (5, 5, 10) => invalid.
- 10. Second perm of 2 equal sides (10, 5, 5) => invalid.
- 11. Third perm of 2 equal sides (5, 10, 5) => invalid.
- 12. Three sides >0, sum of 2 smallest < largest (8,2,5) => invalid.
- 13. Perm 2 of line lengths in test 12 (2, 5, 8) => invalid.
- 14. Perm 3 of line lengths in test 12 (2, 8, 5) => invalid.



# Test Cases -3-

- 15. Perm 4 of line lengths in test 12 (8, 5, 2) => inv.
- 16. Perm 5 of line lengths in test 12 (5, 8, 2) => inv.
- 17. Perm 6 of line lengths in test 12 (5, 2, 8) => inv.
- 18. All sides zero (0, 0, 0) => inv.
- 19. Non-integer input, side a (@, 4, 5) => inv.
- 20. Non-integer input, side b (3, \$, 5) => inv.
- 21. Non-integer input, side c (3, 4, %) => inv.



# Test Cases -4-

- 22. Missing input a (, 4, 5) => invalid.
- 23. Missing input b (3, , 5) => invalid.
- 24. Missing input c (3, 4, ) => invalid.
- 25. Three sides > 0, one side equals the sum of the other two (12, 5, 7) => inv.
- 26. Perm 2 of line lengths in test 25 (12, 7, 5) => inv.
- 27. Perm 3 of line lengths in test 25 (7, 5, 12) => inv.
- 28. Perm 4 of line lengths in test 25 (7, 12, 5) => inv.
- 29. Perm 5 of line lengths in test 25 (5, 12, 7) => inv.
- 30. Perm 6 of line lengths in test 25 (5, 7, 12) => inv.



# Test Cases -5-

- 31. Three sides at max values (32767, 32767, 323767) => inv.
- 32. Two sides at max values (32767, 32767, 1) => inv.
- 33. One side at max values (32767, 1, 1) => inv.



# ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΔΥΝΑΜΙΚΟΥ ΕΛΕΓΧΟΥ (1)

**Γενικά ισχύουν τα ακόλουθα:**

- Ο κώδικας ενός προγραμματιστή πρέπει να ελέγχεται από κάποιον άλλο προγραμματιστή.
- Ο κώδικας μιας εταιρείας πρέπει να ελέγχεται από άλλη εταιρεία.
- Ένα συστατικό κώδικα πρέπει να δοκιμασθεί ώστε να διαπιστωθεί ότι δεν κάνει αυτό που δεν πρέπει να κάνει.



# ΒΑΣΙΚΕΣ ΑΡΧΕΣ

## ΔΥΝΑΜΙΚΟΥ ΕΛΕΓΧΟΥ (2)

- Οι περιπτώσεις ελέγχου απομνημονεύονται για μελλοντική χρήση.
- Μια περίπτωση ελέγχου είναι καλή όταν έχει μεγάλη πιθανότητα ανακάλυψης λαθών.
- Μια περίπτωση ελέγχου είναι επιτυχής όταν αποκαλύψει λάθη άγνωστα μέχρι τώρα.
- Ο δυναμικός έλεγχος είναι μια εξαιρετικά δημιουργική και διανοητικά προκλητική και δύσκολη εργασία.



# ΕΙΔΗ ΔΥΝΑΜΙΚΟΥ ΕΛΕΓΧΟΥ

Επίπεδα δυναμικού ελέγχου και σειρά εκτέλεσης ελέγχων:

- Έλεγχος μονάδας και συνένωσης (φάση κωδικοποίησης).
- Έλεγχος συστήματος και αποδοχής (φάση ελέγχου).





# ΣΤΡΑΤΗΓΙΚΕΣ ΔΥΝΑΜΙΚΟΥ ΕΛΕΓΧΟΥ

Δυο στρατηγικές για την επιλογή περιπτώσεων ελέγχου:

- Στρατηγική μαύρου κουτιού (black box testing): είναι γνωστές οι προδιαγραφές του συστατικού αλλά άγνωστος ο τρόπος κατασκευής.
- Στρατηγική άσπρου κουτιού (white box testing): είναι γνωστός και ο τρόπος κατασκευής του συστατικού.



# ΣΤΡΑΤΗΓΙΚΗ ΜΑΥΡΟΥ ΚΟΥΤΙΟΥ

Λέγεται και Λειτουργικός έλεγχος (Functional Testing).

- Θεωρητικά πρέπει να δοκιμασθούν όλα τα δυνατά δεδομένα εισόδου (έγκυρα και άκυρα) και, αν η μονάδα περιέχει μνήμη, όλες οι δυνατές ακολουθίες δεδομένων εισόδου.
- Αυτό είναι ανέφικτο οπότε χρησιμοποιούνται μερικές μέθοδοι απλοποίησης.



# ΙΣΟΔΥΝΑΜΗ ΔΙΑΜΕΡΙΣΗ (Equivalence Partitioning) -1-

- Όλοι οι δυνατοί συνδυασμοί των δεδομένων εισόδου χωρίζονται σε ένα πεπερασμένο σύνολο κλάσεων ισοδύναμων τιμών.
- Οι τιμές μιας κλάσης είναι ισοδύναμες με την έννοια ότι ανακαλύπτουν το ίδιο σύνολο λαθών.
- Το πλήθος των περιπτώσεων ελέγχου είναι το πολύ ίσο με το πλήθος των κλάσεων.



# ΙΣΟΔΥΝΑΜΗ ΔΙΑΜΕΡΙΣΗ (Equivalence Partitioning) -2-

- Για κάθε περίπτωση ελέγχου απαιτείται μια μόνο τιμή από κάθε κλάση.
- Οι κλάσεις προσδιορίζονται από τις συνθήκες εισόδου (περιορισμοί των δεδομένων εισόδου).



# ΠΑΡΑΔΕΙΓΜΑΤΑ ΙΣΟΔΥΝΑΜΗΣ ΔΙΑΜΕΡΙΣΗΣ (1)

- Αν μια συνθήκη προδιαγράφει ένα διάστημα τιμών υπάρχει μια έγκυρη κλάση και δυο άκυρες (π.χ.  $1.0 \leq x \leq 3.5$  και  $x < 1.0$ ,  $x > 3.5$ ).
- Αν μια συνθήκη προδιαγράφει ένα πλήθος τιμών με μια μέγιστη τιμή υπάρχει μια έγκυρη κλάση και δυο άκυρες (π.χ. Αριθμός παιδιών μέχρι 5:  $x$  ανήκει στο  $[1, 2, 3, 4, 5]$  και  $x = 0$ ,  $x > 5$ ).



# ΠΑΡΑΔΕΙΓΜΑΤΑ ΙΣΟΔΥΝΑΜΗΣ ΔΙΑΜΕΡΙΣΗΣ (2)

- Κάθε σύνολο τιμών, για τις οποίες υπάρχει υπόνοια ότι το συστατικό τις μεταχειρίζεται διαφορετικά, δημιουργεί μια έγκυρη κλάση και μια άκυρη (με μια μόνο τιμή διαφορετική από τις τιμές του συνόλου αυτού).
- Κατά τον ορισμό των περιπτώσεων ελέγχου πρέπει να επιλέγονται εκείνες οι περιπτώσεις που καλύπτουν πολλές ταυτόχρονα κλάσεις: συμπύκνωση δοκιμών (test compaction).



# ΑΝΑΛΥΣΗ ΣΥΝΟΡΙΑΚΩΝ ΤΙΜΩΝ (Boundary value analysis)

- Συνοριακές τιμές σε μια κλάση ισοδύναμων τιμών είναι οι τιμές των άκρων της.
- Έχει παρατηρηθεί ότι πάνω στις συνοριακές τιμές (τόσο εισόδου όσο και εξόδου) καθώς και στις τιμές αμέσως πριν και μετά γίνονται λάθη.



# ΔΟΚΙΜΕΣ ΣΥΝΟΡΙΑΚΩΝ ΤΙΜΩΝ (1)

- Αν μια συνθήκη εισόδου προδιαγράφει ένα διάστημα τιμών απαιτούνται 'έγκυρες' περιπτώσεις για τα άκρα του διαστήματος και 'άκυρες' για τις τιμές έξω ακριβώς από τα άκρα.
- Αν μια συνθήκη εισόδου προδιαγράφει ένα πλήθος τιμών απαιτούνται 'έγκυρες' περιπτώσεις με το μικρότερο και μεγαλύτερο πλήθος καθώς και 'άκυρες' με ένα μικρότερο και ένα μεγαλύτερο από αυτό το πλήθος (π.χ. Για αρχείο που περιέχει 1 με 200 εγγραφές, απαιτούνται περιπτώσεις ελέγχου με 0,1,200,201 εγγραφές).





# ΔΟΚΙΜΕΣ ΣΥΝΟΡΙΑΚΩΝ ΤΙΜΩΝ (2)

- Αν η είσοδος ή η έξοδος είναι ένα διατεταγμένο σύνολο (π.χ ένα ακολουθιακό αρχείο, μια γραμμική λίστα, ένας πίνακας) απαιτούνται περιπτώσεις ελέγχου με το πρώτο και το τελευταίο στοιχείο ελέγχου.



# ΠΡΟΣΕΓΓΙΣΗ ΑΙΤΙΟΥ-ΑΠΟΤΕΛΕΣΜΑΤΟΣ (cause-effect analysis)

- Με βάση τις προδιαγραφές κατασκευάζεται ένας γράφος που στους αρχικούς κόμβους έχει αίτια, στους τελικούς κόμβους αποτελέσματα και στους ενδιάμεσους κόμβους περιορισμούς.
- Στη συνέχεια κατασκευάζεται ένας πίνακας απόφασης από τον οποίο μπορούν να παραχθούν περιπτώσεις ελέγχου.
- Η εφαρμογή της μεθόδου υποστηρίζεται από εργαλεία (π.χ. SoftTest).



# Προδιαγραφή Απαίτησης

- Αν ο πελάτης έχει ηλικία μικρότερη των 18 ετών και παίζει τένις, να του αποστέλλεται επιστολή σχετική με προϊόντα τένις. Αν ο πελάτης έχει ηλικία 18 ή μεγαλύτερη ή έχει άδεια οδήγησης αυτοκινήτου, να του αποστέλλεται επιστολή σχετικά με προϊόντα αυτοκινήτου. Αν συμβαίνουν και τα δυο, να προστεθεί το όνομα του στην λίστα Α.



# Προδιαγραφή Απαίτησης: Είσοδοι

TITLE 'Implied (small "t" or "f") node states'.

## NODES

Under18 = 'The person is under 18' | /b.

Plays\_tennis = 'The person plays tennis' |  
'The person does not play tennis'.

Over18 = 'The person is 18 or older' | /b.

Has\_license = 'The person has a motorcycle license' |  
'The person does not have a motorcycle license'.



# Έξοδοι

T\_brochure = 'Send a tennis brochure' |

'Do not send a tennis brochure' obs.

M\_brochure = 'Send a motorcycle brochure'

| 'Do not send a motorcycle brochure' obs.

A\_list = 'Put them on the "A" mailing list' |

'Do not put them on the "A" mailing

list'.



# Περιορισμοί – Λογικές Σχέσεις

## CONSTRAINTS

$\text{excl}(\text{Under18}, \text{Over18}).$

$\text{req}(\text{Has\_license}, \text{Over18}).$

## RELATIONS

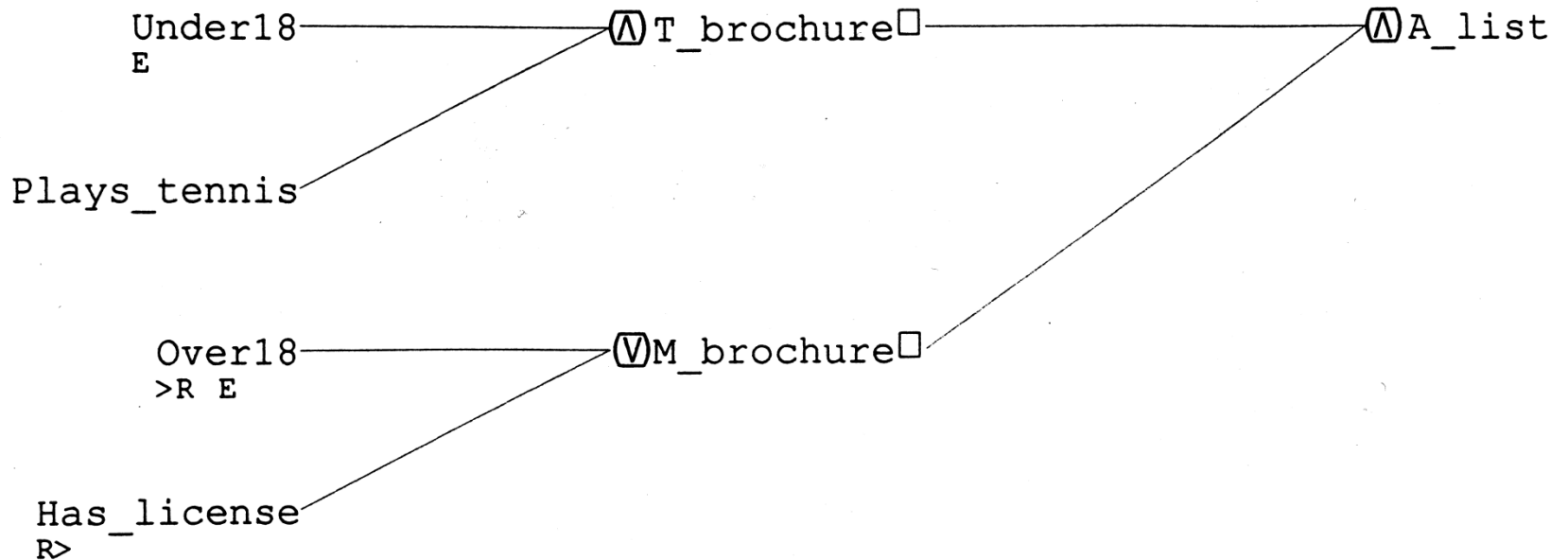
$\text{T\_brochure} \text{ :- Under18 and Plays\_tennis.}$

$\text{M\_brochure} \text{ :- Over18 or Has\_license.}$

$\text{A\_list} \text{ :- M\_brochure and T\_brochure.}$



# Γράφος Αιτίου- Αποτελέσματος



# Εντοπισμός Λογικού Λάθους

Functional Variations for:

A\_list:- T\_brochure and M\_brochure.

<INFEASIBLE> T01—Due to constraint(s)

ACROSS relationships (or faulty logic)

7. If T\_brochure and M\_brochure then A\_list

T11—Possible graph logic error. TRUE state of

A\_list always Infeasible

T12—Note: TRUE state of A\_list not covered in

any test case





# Δοκιμές

Αρκούν 3 Δοκιμές (από 2 στην τετάρτη = 16 συνολικά) για 100% κάλυψη λογικών λαθών.

- TEST#1 = Under18, Plays\_tennis, not Over18, not Has\_license. – Send a tennis brochure, Do not send a motorcycle license, Do not put them on the “A” mailing list.
- TEST#2 not Under18, Plays\_tennis, Over18, not Has\_license – Do not send a tennis brochure, Send a motorcycle license, Do not put them on the “A” mailing list.
- TEST#3 Under18, does not Plays\_tennis, not Over18, not Has\_license – DO not send a tennis brochure, Do not send a motorcycle license, Do not put them on the “A” mailing list.



# ΠΡΟΣΕΓΓΙΣΗ ΑΙΤΙΟΥ-ΑΠΟΤΕΛΕΣΜΑΤΟΣ (cause-effect-analysis)

- Στην πράξη η κατασκευή του γράφου δεν είναι εύκολη και ο αριθμός περιπτώσεων ελέγχου είναι πολύ μεγάλος.
- Με τη μέθοδο όμως αυτή είναι δυνατό να ελεγχθούν συνδυασμοί συνθηκών εισόδου.
- Ο ελεγκτής πρέπει να χρησιμοποιεί τη διαίσθηση του για να διαισθανθεί και να ανακαλύψει σχεδιαστικά λάθη.



# ΑΣΚΗΣΕΙΣ -1-

Να σχεδιασθούν περιπτώσεις ελέγχου για τα ακόλουθα προγράμματα:

- Πρόγραμμα που αθροίζει τις αγορές ενός πελάτη και υπολογίζει έκπτωση 5% αν το συνολικό ποσό είναι πάνω από 60€ και 20% αν είναι πάνω από 300€.
- Πρόγραμμα που εξετάζει τρεις αριθμούς και αποφασίζει αν μπορούν να είναι μήκη πλευρών ενός τριγώνου και τι είδους τρίγωνο είναι αυτό.



# ΑΣΚΗΣΕΙΣ -2-

- Πρόγραμμα που διαβάζει μια γραμμή κειμένου και μετράει τον αριθμό των χαρακτήρων που δεν είναι κενοί (blank).
- Πρόγραμμα που διαβάζει μια γραμμή κειμένου και αντικαθιστά τις ακολουθίες κενών χαρακτήρων με ένα κενό χαρακτήρα.
- Αφηρημένος τύπος δεδομένων 'χορδή' (string) ο οποίος προσφέρει τις λειτουργίες της σύνδεσης δύο χορδών, μήκους χορδής και επιλογής υποχορδής (με βάση δυο ακεραίους) πάνω σε μια σειρά αλφαριθμητικών χαρακτήρων.



# ΣΤΡΑΤΗΓΙΚΗ ΤΟΥ ΔΙΑΦΑΝΟΥΣ ΚΟΥΤΙΟΥ

- Λέγεται και δομικός έλεγχος (Structural Testing).
- Ο καθορισμός των περιπτώσεων ελέγχου γίνεται με βάση το σχέδιο ή/και τον κώδικα.
- Συνήθως επιδιώκεται να καλυφθούν όλα τα δυνατά μονοπάτια εκτέλεσης στο γράφο ροής του προγράμματος.



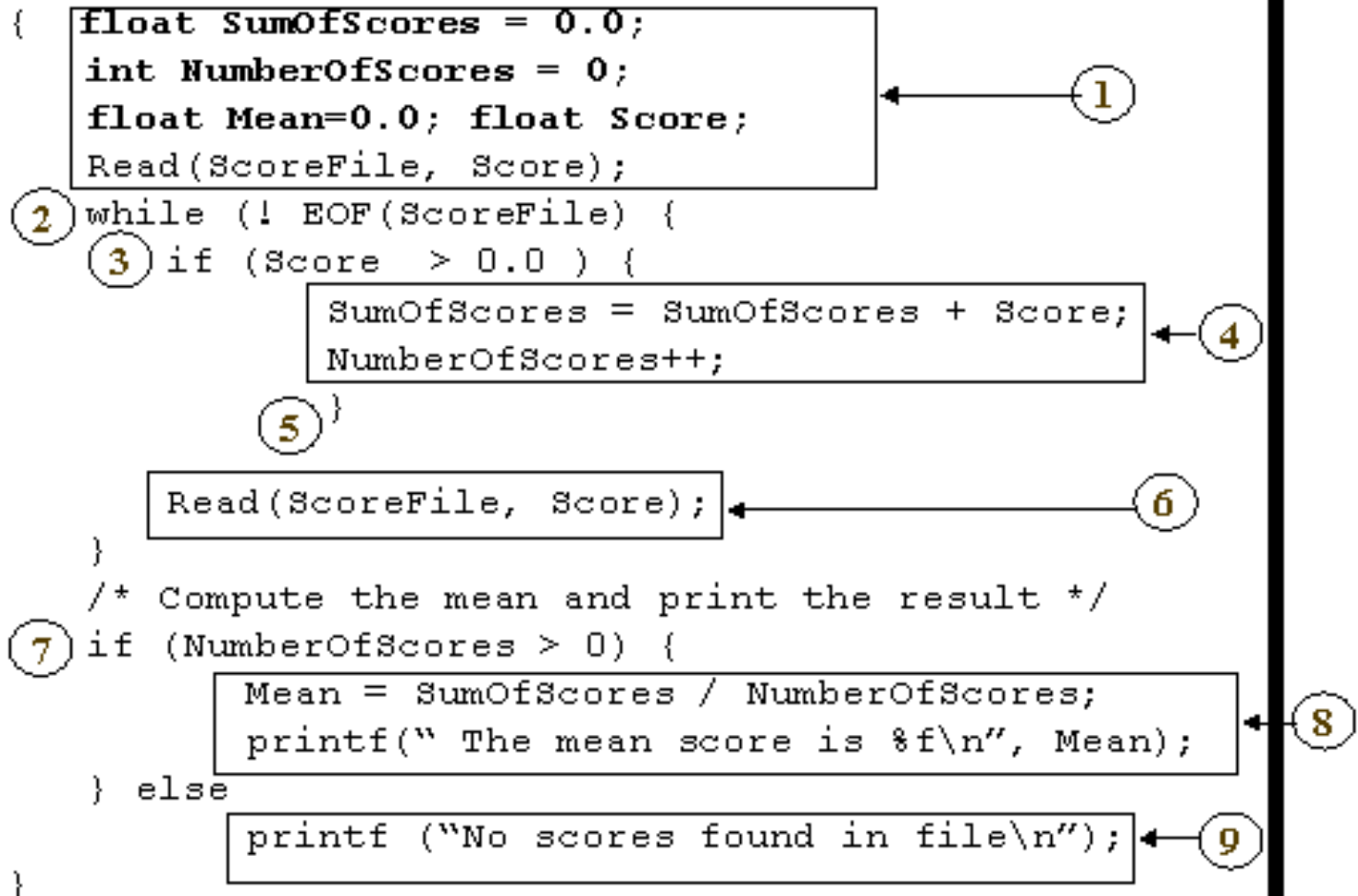
# Έλεγχος με βάση τα μονοπάτια

```
FindMean(float Mean, FILE ScoreFile)
{ SumOfScores = 0.0; NumberOfScores = 0; Mean = 0;
  Read(ScoreFile, Score); /*Read in and sum the scores*/
  while (! EOF(ScoreFile) ) {
      if ( Score > 0.0 ) {
          SumOfScores = SumOfScores + Score;
          NumberOfScores++;
      }
      Read(ScoreFile, Score);
  }
  /* Compute the mean and print the result */
  if (NumberOfScores > 0 ) {
      Mean = SumOfScores/NumberOfScores;
      printf("The mean score is %f \n", Mean);
  } else
      printf("No scores found in file\n");
}
```

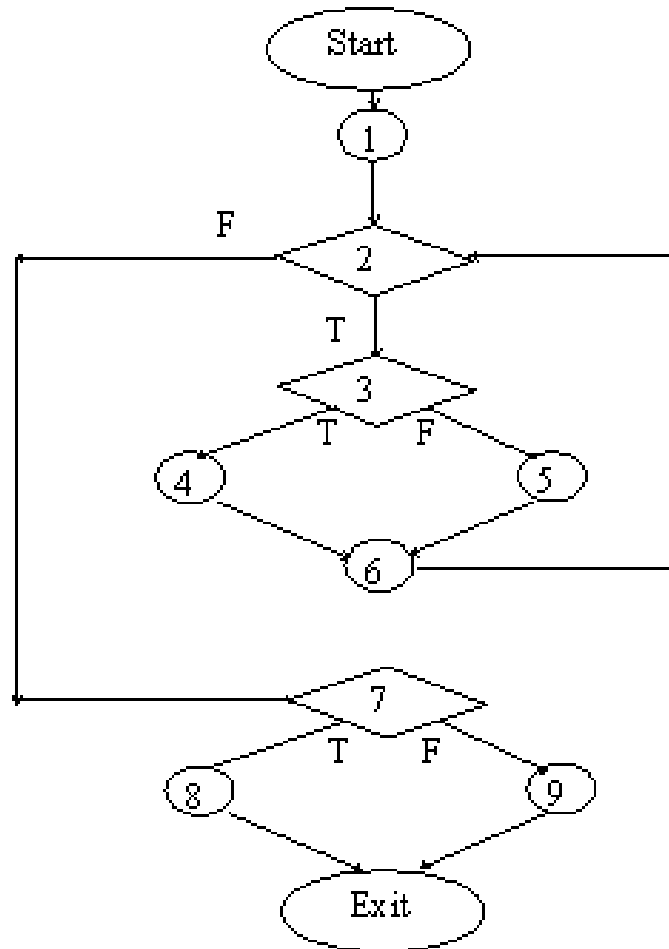


# Κόμβοι Γράφου Ροής

```
FindMean (FILE ScoreFile)
```

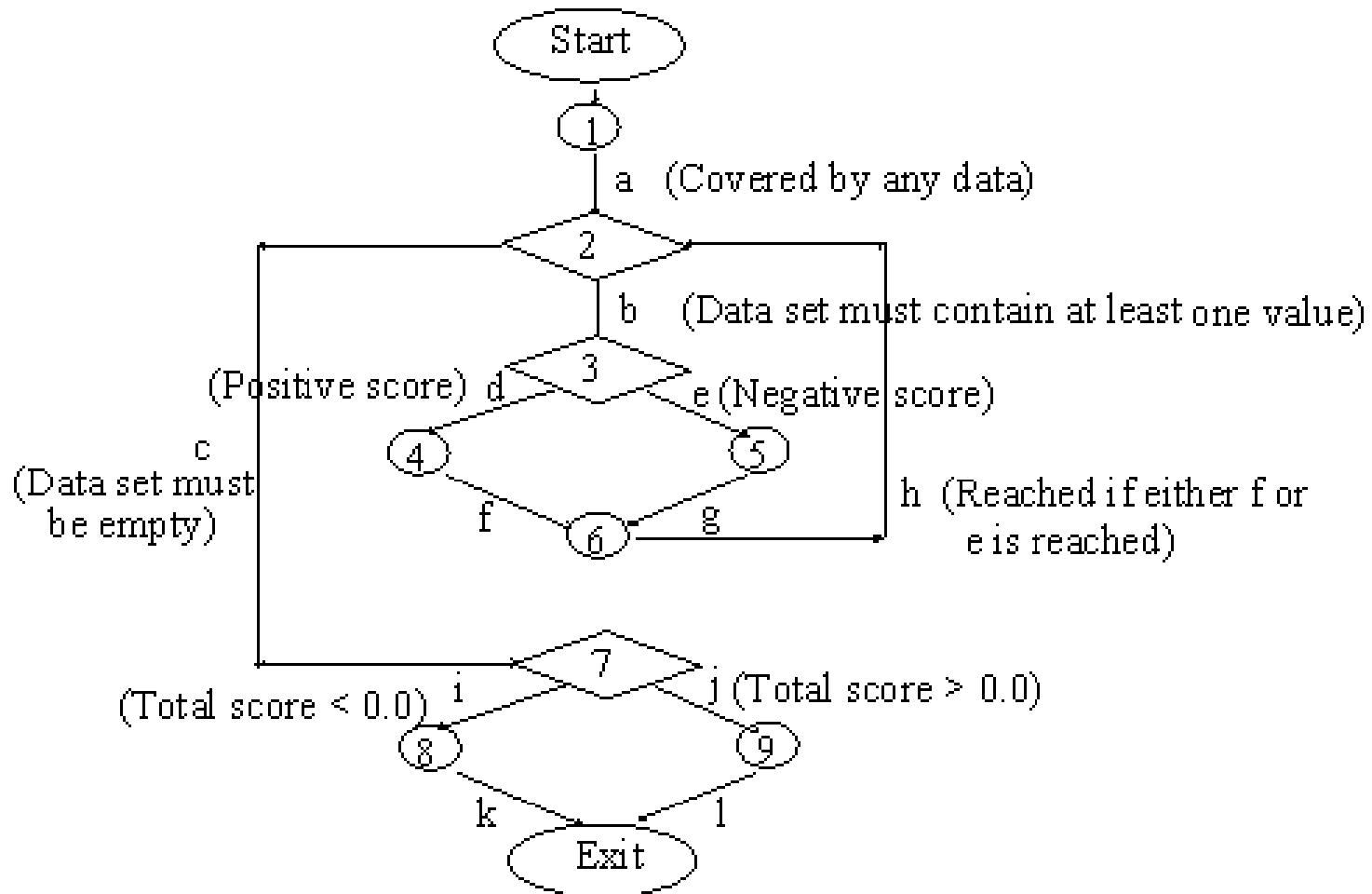


# Γράφος Ροής





# Περιπτώσεις Δοκιμής



# Κυκλωματική Πολυπλοκότητα

- Έχει προταθεί ως μέτρο της πολυπλοκότητας του κώδικα (McCabe, 1976) και μετρά τον αριθμό των γραμμικά ανεξάρτητων μονοπατιών μέσα σε ένα συστατικό.
- $V(G)=E-N+2$   
όπου  $V(G)$  η κυκλωματική πολυπλοκότητα,  
 $E$  ο αριθμός των ακμών,  
 $N$  ο αριθμός των κόμβων
- Στο προηγούμενο παράδειγμα  $V(G)=12-10+2 =4$ .



# ΠΡΟΒΛΗΜΑΤΑ

- Ο αριθμός των μονοπατιών είναι συνήθως πολύ μεγάλος.
- Δεν καλύπτονται λάθη μη-συμμόρφωσης προς τις προδιαγραφές.
- Δεν ανακαλύπτονται μονοπάτια που λείπουν.
- Δεν ανακαλύπτονται λάθη που αφορούν ειδικές τιμές δεδομένων.



# ΚΡΙΤΗΡΙΑ ΚΑΛΥΨΗΣ ΤΟΥ ΔΟΜΙΚΟΥ ΕΛΕΓΧΟΥ

- Στη στρατηγική άσπρου κουτιού χρησιμοποιούνται τα λεγόμενα κριτήρια κάλυψης. Ακολουθεί λίστα από τέτοια κριτήρια από το ασθενέστερο προς το ισχυρότερο.



# ΚΡΙΤΗΡΙΑ ΚΑΛΥΨΗΣ

- Κάλυψη εντολών.
- Κάλυψη απόφασης ή κάλυψη διακλάδωσης.
- Κάλυψη συνθήκης (συνδυασμοί ελέγχου που επιτρέπουν μια απόφαση).
- Κάλυψη απόφασης/συνθήκης.
- Κάλυψη πολλαπλής συνθήκης (πολλαπλές διαδοχικές συνθήκες).



# ΤΕΛΙΚΟ ΣΥΜΠΕΡΑΣΜΑ

- Στην πράξη απαιτείται συνδυασμός των δύο στρατηγικών (διαφανούς/αδιαφανούς κουτιού).
- Είναι δυνατό να χαραχθεί μια γενική στρατηγική που συνδυάζει τον λειτουργικό και δομικό έλεγχο (όπου αυτό είναι δυνατό).



# ΕΦΑΡΜΟΓΗ ΔΥΝΑΜΙΚΟΥ ΕΛΕΓΧΟΥ -1-

Απαιτεί την κατασκευή ορισμένων προϊόντων:

- Ειδικός κώδικας (π.χ. οδηγοί-test drivers).
- Περιπτώσεις ελέγχου.
- Ειδικά έγγραφα τεκμηρίωσης (π.χ. ημερολόγιο, εκθέσεις).



# ΕΦΑΡΜΟΓΗ ΔΥΝΑΜΙΚΟΥ ΕΛΕΓΧΟΥ -2-

## Ανθρώπινοι ρόλοι:

- Διοικητής.
- Αναλυτής.
- Σχεδιαστής.
- Προγραμματιστής.
- Ελεγκτής.
- Αξιολογητής.





# ΕΦΑΡΜΟΓΗ ΔΥΝΑΜΙΚΟΥ ΕΛΕΓΧΟΥ -1-

## Έλεγχος μονάδας (Unit Testing)

- Αποσκοπεί στο να δειχθεί ότι κάθε μονάδα προγράμματος χωριστά δεν πληροί τις προδιαγραφές της.
- Ετοιμάζονται περιπτώσεις ελέγχου ξεκινώντας με τη στρατηγική του άσπρου κουτιού που συμπληρώνονται με περιπτώσεις από τη στρατηγική του μαύρου κουτιού.



# ΕΦΑΡΜΟΓΗ ΔΥΝΑΜΙΚΟΥ ΕΛΕΓΧΟΥ -2-

## Δυο είδη ελέγχου μονάδας

- Μη-Αυξητικός (κάθε μονάδα ξεχωριστά και στη συνέχεια συνένωση)-απαιτούνται οδηγοί και στελέχη.
- Αυξητικός (κάθε μονάδα ελέγχεται σε συνδυασμό με άλλες μονάδες που έχουν ήδη ελεγχθεί) - απαιτούνται είτε οδηγοί είτε στελέχη ανάλογα με τον τρόπο εφαρμογής του ελέγχου.



# ΑΥΞΗΤΙΚΟΣ ΕΛΕΓΧΟΣ

- Η επιλογή των μονάδων μπορεί να γίνει με διάφορα κριτήρια (π.χ. προηγείται ο έλεγχος των μονάδων που κάνουν είσοδο - έξοδο).
- Δεν απαιτείται έλεγχος ενοποίησης.



# ΕΛΕΓΧΟΣ ΕΝΟΠΟΙΗΣΗΣ (1)

## Λέγεται και Έλεγχος Ολοκλήρωσης (Integration Testing)

- Αποσκοπεί στο ναδειχθεί ότι διαδοχικές ενώσεις μονάδων του προγράμματος σε όλο και μεγαλύτερα συστατικά δεν πληρούν τις προδιαγραφές.
- Συχνά συνενώνονται μεγάλες μονάδες λογισμικού που έχουν ελεγχθεί εσωτερικά με αυξητικό ή μη-αυξητικό τρόπο.



# ΕΛΕΓΧΟΣ ΕΝΟΠΟΙΗΣΗΣ (2)

- Απαιτείται ένα πλάνο ελέγχου συνένωσης που μπορεί να επιτρέψει την παράλληλη διεξαγωγή της κωδικοποίησης, ελέγχου μονάδας και ελέγχου ενοποίησης διαφόρων τμημάτων του συστήματος.
- Ο λόγος είναι ότι η υλοποίηση των διαφόρων μονάδων συνήθως ολοκληρώνεται σε διαφορετικές χρονικές στιγμές, σύμφωνα με το πλάνο του έργου.



# ΕΛΕΓΧΟΣ ΣΥΣΤΗΜΑΤΟΣ

## System Testing

- Αποσκοπεί στο να δειχθεί ότι το σύστημα δεν πληροί τις απαιτήσεις από το σύστημα και απαιτήσεις από το λογισμικό.
- Χρησιμοποιείται η στρατηγική του μαύρου κουτιού και ελέγχονται οι λειτουργικές απαιτήσεις, οι απαιτήσεις διασύνδεσης και οι μη λειτουργικές απαιτήσεις (επίδοσης, περιορισμών στη σχεδίαση, ασφάλειας, αξιοπιστίας, ξεκινήματος ύστερα από αιφνίδιο σταμάτημα, διαθεσιμότητας κτλ).
- Επίσης διενεργούνται έλεγχοι πίεσης (stress tests) και έλεγχοι όγκου δεδομένων (volume tests).



# ΕΛΕΓΧΟΣ ΑΠΟΔΟΧΗΣ

## Acceptance Testing

- Αποσκοπεί στο να πεισθεί ο πελάτης ότι το σύστημα που κατασκευάστηκε για λογαριασμό του, πληροί τις προδιαγραφές του.



# ΤΕΚΜΗΡΙΩΣΗ ΔΥΝΑΜΙΚΟΥ ΕΛΕΓΧΟΥ

- Τυπικά έγγραφα τεκμηρίωσης ελέγχου
  - Προαπαιτούμενα έγγραφα
    - Έγγραφα έργου
    - Τεκμηρίωση Μονάδων / Συστατικών προς έλεγχο.
  - Προετοιμασία εκτέλεσης του ελέγχου
    - Πλάνο Ελέγχου
    - Προδιαγραφές σχεδίου ελέγχου
    - Προδιαγραφές περιπτώσεων ελέγχου
    - Προδιαγραφές διαδικασίας διεκπεραίωσης ελέγχου.
  - Μετά την εκτέλεση του ελέγχου
    - Ημερολόγιο ελέγχου
    - Έκθεση απροόπτων
    - Περιληπτική έκθεση ελέγχου.
- Πρότυπο ελέγχου ΙΕΕΕ: 8 έγγραφα τεκμηρίωσης.





# ΔΥΝΑΜΙΚΟΣ ΕΛΕΓΧΟΣ ΚΑΙ ΠΑΡΑΛΑΒΗ -1-

## Επαλήθευση

- Επαληθευμένο λογισμικό: έχει γίνει έλεγχος συστήματος από την ομάδα που κατασκεύασε το λογισμικό ( $\alpha$ -test).

## Επικύρωση

- Επικυρωμένο λογισμικό: έχει γίνει έλεγχος συστήματος από ομάδα ανεξάρτητη από εκείνη που το κατασκεύασε ( $\beta$ -test).



# ΔΥΝΑΜΙΚΟΣ ΕΛΕΓΧΟΣ ΚΑΙ ΠΑΡΑΛΑΒΗ -2-

## Πιστοποίηση λογισμικού

- Πιστοποιημένο λογισμικό: το λογισμικό έχει περάσει τους προηγούμενους ελέγχους και έχει παραμείνει για ένα διάστημα σε δοκιμαστική λειτουργία στο πραγματικό περιβάλλον λειτουργίας κάτω από την εποπτεία του πελάτη.



# ΕΡΓΑΛΕΙΑ ΥΠΟΣΤΗΡΙΞΗΣ

- Υπάρχουν ειδικά εργαλεία για την υποστήριξη της εφαρμογής του δυναμικού ελέγχου:
  - Αναλυτές (coverage analyzers).
  - Συγκριτές (comparators).
  - Γεννήτριες δοκιμαστικών δεδομένων (test data generators).
  - ...



# ΑΠΟΣΦΑΛΜΑΤΩΣΗ

## Εντοπισμός και εκκαθάριση λαθών

- Διάφορες στρατηγικές (Επαγωγή, Αναδρομή, Οπισθοδρόμηση).
- Συμβατικές τεχνικές (διαγνωστικά μηνύματα, στιγμιαία αντίγραφα της μνήμης, επιλεκτική λήψη ιχνών δεδομένων και ροής ελέγχου, σημεία διακοπής).



# ΚΡΙΤΗΡΙΑ ΤΕΡΜΑΤΙΣΜΟΥ -1-

**Πότε πρέπει να τερματίζει ο έλεγχος;**

Κριτήριο τερματισμού για τον έλεγχο μονάδας  
μπορεί να είναι:

- Όλες οι περιπτώσεις ελέγχου να είναι ανεπιτυχείς και οι περιπτώσεις αυτές να έχουν παραχθεί με βάση το κριτήριο κάλυψης πολλαπλής συνθήκης και να έχει γίνει ανάλυση συνοριακών τιμών στη διασύνδεση της μονάδας.



# ΚΡΙΤΗΡΙΑ ΤΕΡΜΑΤΙΣΜΟΥ -2-

Κριτήριο τερματισμού για τον έλεγχο συστήματος μπορεί να είναι

- Όλες οι περιπτώσεις ελέγχου να είναι ανεπιτυχείς και οι περιπτώσεις αυτές να έχουν παραχθεί με βάση την προσέγγιση αιτίου-αποτελέσματος και να έχει γίνει ανάλυση συνοριακών τιμών με ταυτόχρονη χρήση της διαίσθησης των ελεγκτών.



# ΚΡΙΤΗΡΙΑ ΤΕΡΜΑΤΙΣΜΟΥ -3-

## Άλλα κριτήρια μπορεί να είναι

- Αποκάλυψη ενός ορισμένου αριθμού λαθών ή εκπνοή μιας ορισμένης χρονικής περιόδου του ελέγχου, όποιο συμβεί τελευταίο.
- Ο ρυθμός ανακάλυψης λαθών να μην είναι ανοδικός.
- Συνδυασμός των παραπάνω.



# Σημείωμα Χρήσης Έργων Τρίτων

Όλα τα σχήματα/διαγράμματα έχουν συμπεριληφθεί μετά από κατάλληλη τροποποίηση, από το σύγγραμμα «S. L. Pfleeger (Γ. Σταμέλος), «Τεχνολογία Λογισμικού, Θεωρία και Πράξη», Εκδ. ΚΛΕΙΔΑΡΙΘΜΟΣ, 2012.»





# Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Σταμέλος Ιωάννης.  
«Τεχνολογία Λογισμικού. Έλεγχος ορθής λειτουργίας λογισμικού». Έκδοση:  
1.0. Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση:  
<http://eclass.auth.gr/courses/OCRS221/>



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Μη Εμπορική Χρήση - Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>





# Τέλος ενότητας

Επεξεργασία: <Τέγος Στέργιος >  
Θεσσαλονίκη, <Χειμερινό Εξάμηνο 2013-2014>





ΑΡΙΣΤΟΤΕΛΕΙΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΟΝΙΚΗΣ

---

# Σημειώματα

# Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

