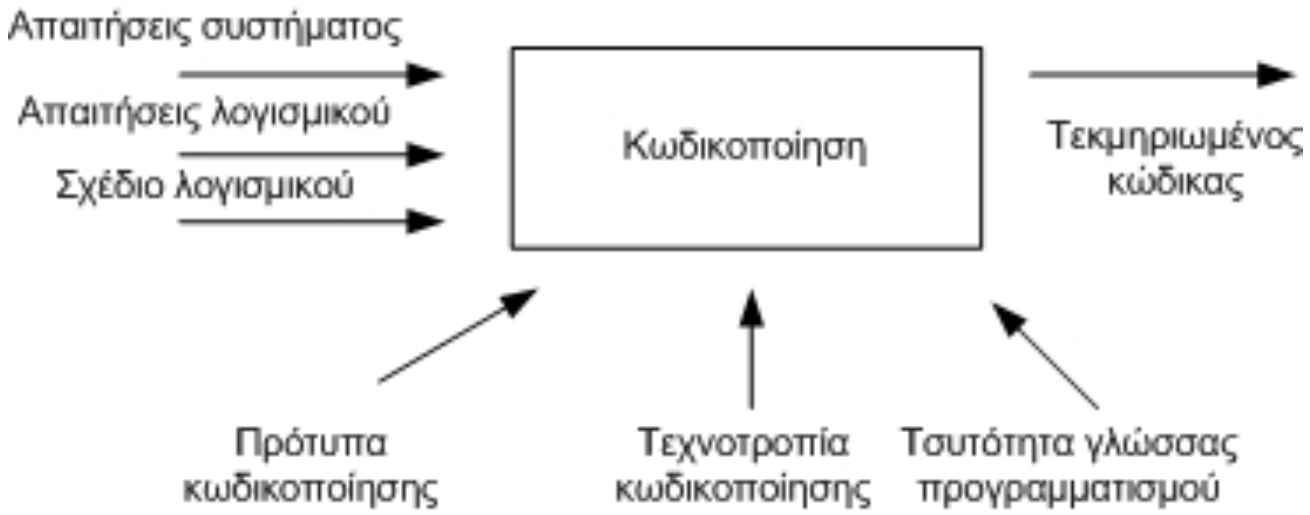
# Κωδικοποίηση και Έλεγχος Ορθότητας

# περιεχόμενα περουσίασης

- Κωδικοποίηση
- Πρότυπα και διαδικασίες κωδικοποίησης
- Τεκμηρίωση
- Διαχείριση εκδόσεων
- Έλεγχος ορθότητας λογισμικού

# κωδικοποίηση

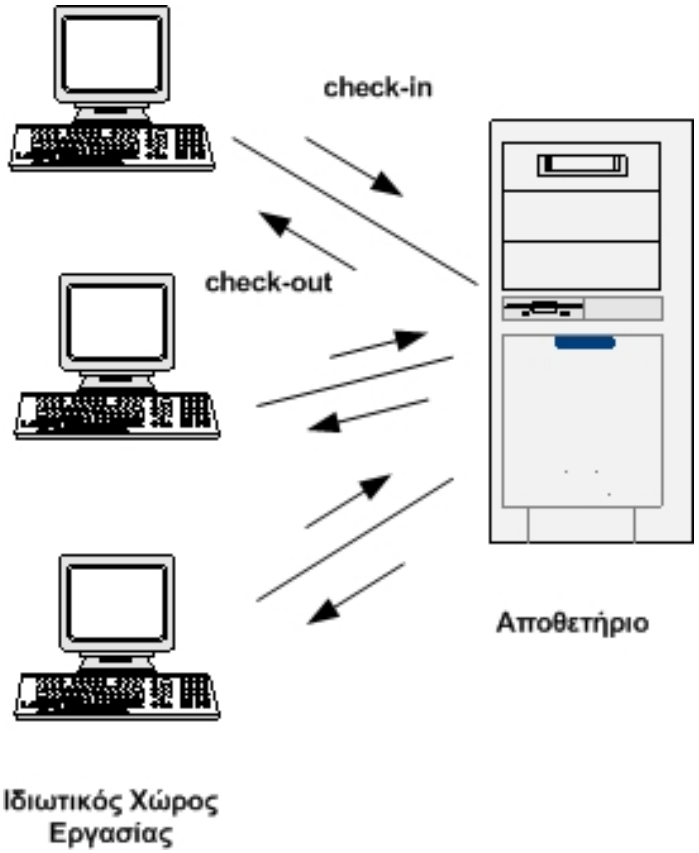


# διαχείριση εκδόσεων

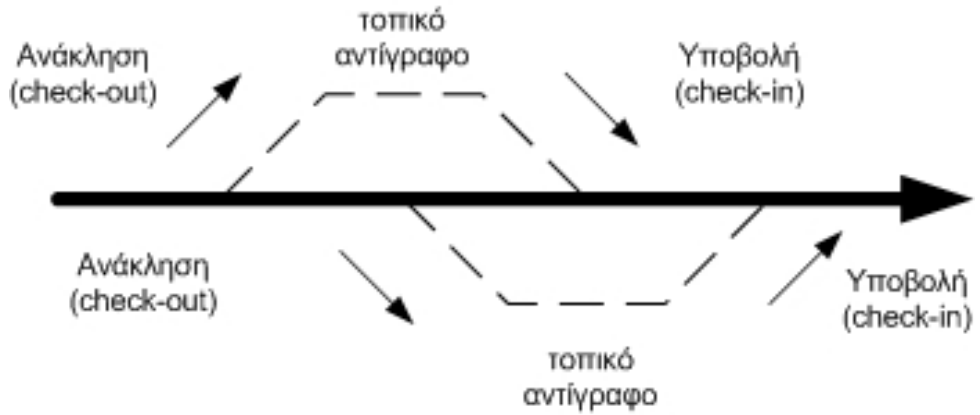
Ένα σύστημα διαχείρισης εκδόσεων προσφέρει:

- Την παροχή της αναίρεσης στην ομάδα ανάπτυξης.
- Διατήρηση διαφορετικών εκδόσεων ενός αρχείου του πηγαίου κώδικα.
- Ελεγχόμενη πρόσβαση στον πηγαίο κώδικα του λογισμικού.
- Παράλληλη εργασία των προγραμματιστών στην ίδια βάση κώδικα.
- Εύκολος εντοπισμός για το ποιος έχει κάνει αλλαγές στον κώδικα και γιατί.
- Την παραγωγή πολλών κυκλοφοριών (releases) του λογισμικού.
- Μία χρονομηχανή που είναι σε θέση να μας δώσει την ακριβή εικόνα του έργου σε μία συγκεκριμένη ημερομηνία.

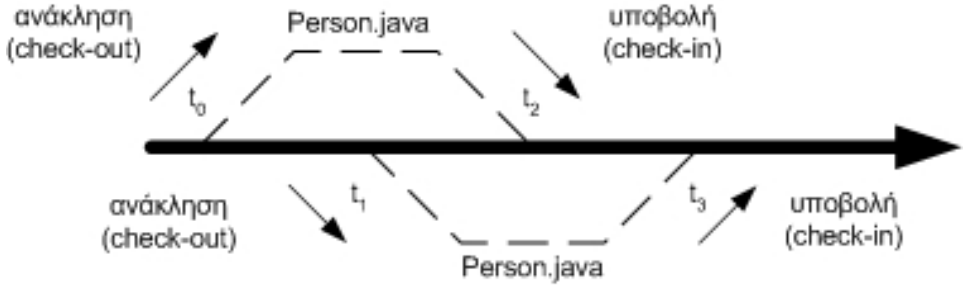
# αποθετήριο



# βασική γραμμή (κορμός)

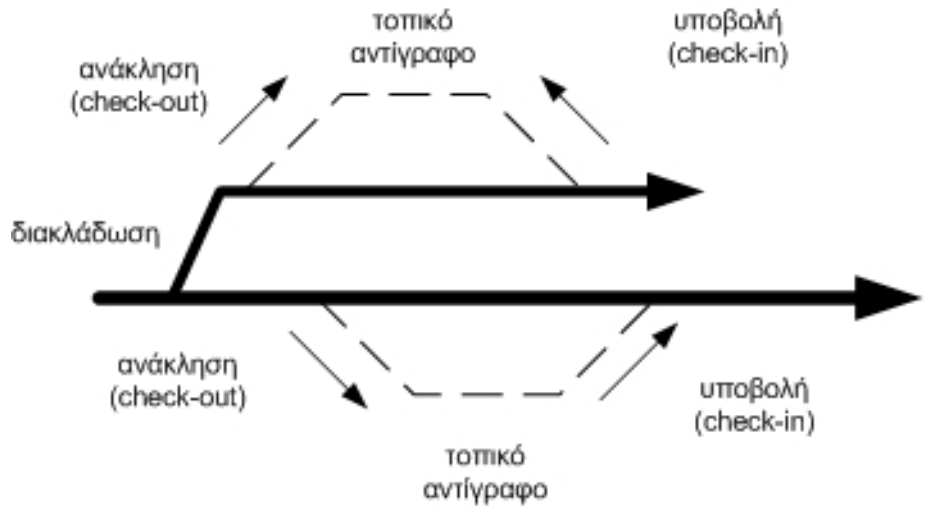



- Βασική γραμμή κώδικα

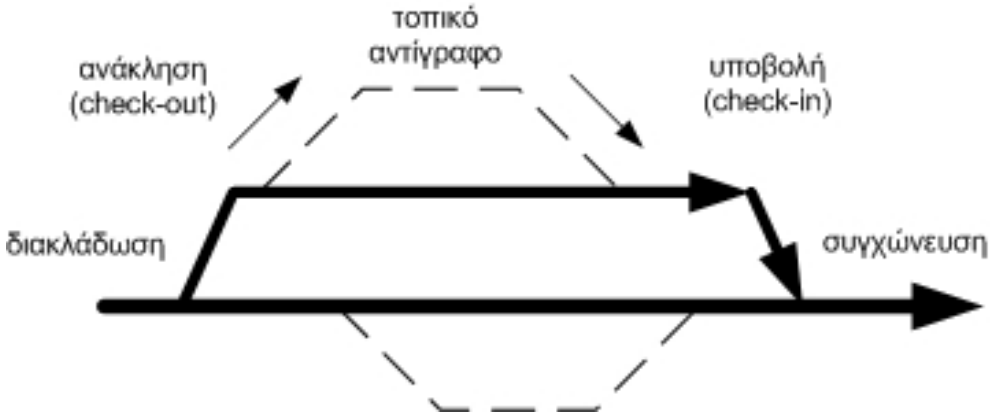



- Συγκρούσεις

# διακλάδωση - συγχώνευση




- Διακλάδωση



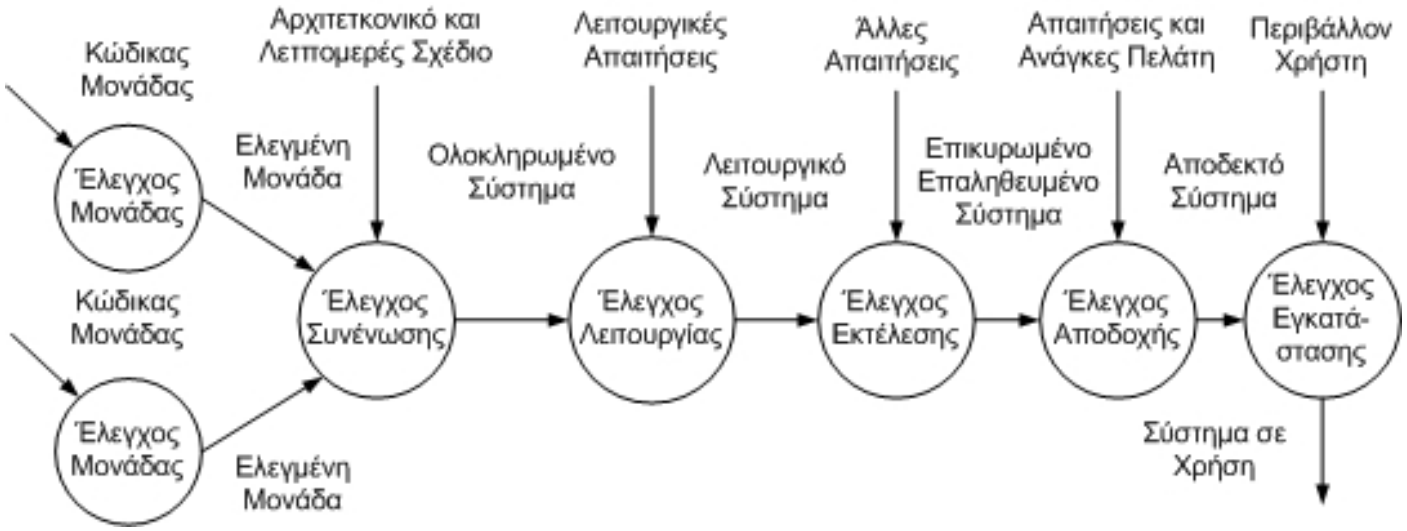

- Συγχώνευση

# έλεγχος ορθότητας

Κατηγορίες σφαλμάτων

- αλγοριθμικά σφάλματα
- υπολογιστικά λάθη και λάθη ακρίβειας
- σφάλματα υπερφόρτωσης
- σφάλματα ορίων ή χωρητικότητας
- σφάλματα χρονισμού ή σφάλματα συντονισμού
- σφάλματα απόδοσης
- σφάλματα ανάκαμψης
- σφάλματα υλικού και λογισμικού συστήματος
- σφάλματα τεκμηρίωσης

# στάδια ελέγχου



# έλεγχος κλειστού κουτιού

- Με τον έλεγχο κλειστού κουτιού δεν έχουμε γνώση της εσωτερικής δομής και λογικής του αντικειμένου ελέγχου.
- Θα πρέπει όμως με κάποιο τρόπο να επιλέξουμε εκείνες τις δοκιμασίες ελέγχου οι οποίες έχουν τη μεγαλύτερη πιθανότητα ανακάλυψης κάποιου σφάλματος.
- Θα εξετάσουμε δύο τεχνικές οι οποίες, αν και δεν εγγυώνται την ανακάλυψη όλων των σφαλμάτων, αυξάνουν την πιθανότητα εντοπισμού τους.

# παράδειγμα: έλεγχος κλειστού κουτιού

- Έστω η μέθοδος calculateSalary(int month), της κλάσης Employee. Η μέθοδος υπολογίζει τη μισθοδοσία ενός υπαλλήλου για κάποιο μήνα του χρόνου.
- Η είσοδος σε αυτή τη μέθοδος είναι ένας ακέραιος ο οποίος αναπαριστά τον μήνα για τον οποίο υπολογίζεται η μισθοδοσία.
- Είναι προφανές ότι δεν μπορούμε να ελέγξουμε τη μέθοδο με όλους τους δυνατούς ακέραιους, αλλά δε θα πρέπει και πάλι να ελέγχουμε «στην τύχη».

# διαμερισμός ισοδυναμίας

- Μία τεχνική η οποία χρησιμοποιείται πολύ συχνά για τον έλεγχο κλειστού κουτιού είναι ο διαμερισμός ισοδυναμίας (equivalence partitioning).
- Σύμφωνα με αυτή την τεχνική θα πρέπει να χωρίσουμε το χώρο των δυνατών εισόδων σε κλάσεις ισοδυναμίας (equivalence classes), έτσι ώστε η επιλογή μίας δοκιμασίας ελέγχου από κάθε διαφορετική κλάση να θεωρείται ως αντιπροσωπευτική για όλη την κλάση.
- Με αυτό τον τρόπο περιορίζουμε τον αριθμό των δοκιμασιών ελέγχου, έχοντας ταυτόχρονα αυξήσει την πιθανότητα να ανακαλύψουμε κάποιο σφάλμα.

# παράδειγμα: διαμερισμός ισοδυναμίας

- Στο παράδειγμα της μεθόδου calculateSalary μπορούμε να χωρίσουμε το χώρο της εισόδου της μεθόδους σε τρεις κλάσεις.
- Η πρώτη κλάση περιλαμβάνει μία δοκιμασία ελέγχου για month μικρότερο του 1,
- η δεύτερη κλάση περιλαμβάνει μία δοκιμασία ελέγχου για month μεταξύ 1 και 12 και
- η τρίτη κλάση για month μεγαλύτερο του 12.

# διαμερισμός ισοδυναμίας

Οι κλάσεις ισοδυναμίας θα πρέπει να ικανοποιούν τα ακόλουθα κριτήρια:

- Κάθε πιθανή είσοδος ανήκει σε μια από τις κλάσεις
- Κανένα σύνολο δεδομένων εισόδου δεν ανήκει σε περισσότερες από μια κλάσεις
- Αν η εκτέλεση του προγράμματος δείχνει την ύπαρξη ενός σφάλματος, όταν ένα συγκεκριμένο μέλος μιας κλάσης χρησιμοποιείται ως είσοδος, τότε το ίδιο σφάλμα μπορεί να προκύψει και με τη χρησιμοποίηση κάθε άλλου μέλους της κλάσης ως εισόδου

# ανάλυση συνοριακών τιμών

- Μία δεύτερη τεχνική είναι η ανάλυση των συνοριακών τιμών (boundary value analysis).
- Η εμπειρία στον έλεγχο του λογισμικού έχει δείξει ότι η πιθανότητα ανακάλυψης σφαλμάτων αυξάνεται, όταν οι δοκιμασίες ελέγχου σχετίζονται με τις συνοριακές συνθήκες (boundary conditions).
- Αν επανέλθουμε στο παράδειγμα της μεθόδου calculateSalary, οι συνοριακές τιμές οι οποίες θα πρέπει να συμπεριληφθούν στις δοκιμασίες ελέγχου είναι οι τιμές 0, 1, 12, 13.

# ανάλυση συνοριακών τιμών

- Βλέπουμε ότι οι συνοριακές τιμές που επιλέξαμε είναι οι συνοριακές τιμές των κλάσεων ισοδυναμίας στις οποίες χωρίσαμε το χώρο των εισόδων της calculateSalary.
- Η ανάλυση συνοριακών τιμών συμπληρώνει το διαμερισμό ισοδυναμίας, αναλύοντας τις συνοριακές συνθήκες όχι μόνο στην είσοδο αλλά και στην έξοδο.

# αξιολόγηση ελέγχου κλειστού κουτιού

Υπάρχουν πλεονεκτήματα και μειονεκτήματα στο έλεγχο κλειστού κουτιού.

- Ένα προφανές πλεονέκτημα είναι ότι ο έλεγχος ενός κλειστού κουτιού είναι ελεύθερος περιορισμών που επιβάλλει η εσωτερική δομή και λογική του αντικειμένου που ελέγχεται.
- Το μειονέκτημα του ελέγχου κλειστού κουτιού είναι ότι για ορισμένα αντικείμενα ελέγχου η παραγωγή του συνόλου των αντιπροσωπευτικών δοκιμασιών ελέγχου που να επιδεικνύει τη σωστή λειτουργικότητα σε όλες τις περιπτώσεις είναι αδύνατη.

# αξιολόγηση ελέγχου κλειστού κουτιού

- Ας θεωρήσουμε για παράδειγμα ότι μία μονάδα λογισμικού δέχεται ως είσοδο το ακαθάριστο εισόδημα και τις περιοχές των συντελεστών φορολογίας και παράγει στην έξοδο το ποσό του φόρου εισοδήματος.
- Θα μπορούσαμε να έχουμε έναν πίνακα φόρων που θα δείχνει τις αναμενόμενες εξόδους για συγκεκριμένα δεδομένα εισόδου, αλλά ίσως να μην είμαστε σε θέση να γνωρίζουμε γενικά πώς υπολογίζεται ο φόρος.
- Ο αλγόριθμος για τον υπολογισμό του φόρου εξαρτάται από τις περιοχές του εισοδήματος, αλλά τόσο τα όρια των περιοχών όσο και τα σχετιζόμενα με αυτά ποσοστά είναι μέρος της εσωτερικής λογικής της μονάδας προγράμματος.
- Εξετάζοντας αυτή τη μονάδα σαν κλειστό κουτί, δεν είναι δυνατόν να επιλέγουμε αντιπροσωπευτικές δοκιμασίες ελέγχου, επειδή δε θα γνωρίζουμε αρκετά για τις περιοχές του εισοδήματος

# έλεγχος ανοικτού κουτιού

- Με τον έλεγχο ανοιχτού κουτιού είμαστε ενήμεροι για την εσωτερική δομή και λογική του αντικειμένου ελέγχου.
- Η επιδίωξη του ελέγχου ανοιχτού κουτιού είναι να επιλέξουμε δοκιμασίες ελέγχου οι οποίες εκτελούν όλες τις εντολές ή όλες τις πιθανές διαδρομές της ροής ελέγχου μέσα στη μονάδα προγράμματος ή στις μονάδες προγράμματος.
- Για να επιλέξουμε δοκιμασίες ελέγχου βασιζόμαστε στην κάλυψη του κώδικα που παρέχει ο έλεγχος.

# καλύψεις στον έλεγχο ανοικτού κουτιού

- Υπάρχουν τρεις διαφορετικές κατηγορίες καλύψεων, έτσι ώστε να αξιολογήσουμε την κάλυψη του κώδικα για τις δοκιμασίες ελέγχου που επιλέγουμε.
  - Κάλυψη εντολών (statement coverage): Κάθε εντολή της μονάδας εκτελείται τουλάχιστον μία φορά.
  - Κάλυψη διακλαδώσεων (branch coverage): Για κάθε σημείο απόφασης στη μονάδα επιλέγεται κάθε διακλάδωση τουλάχιστον μία φορά.
  - Κάλυψη διαδρομών (path coverage): Κάθε διακριτή διαδρομή του προγράμματος εκτελείται τουλάχιστον μια φορά.

# καλύψεις στον έλεγχο ανοικτού κουτιού

- Σε γενικές γραμμές η πλήρης κάλυψη εντολών απαιτεί λιγότερες δοκιμασίες ελέγχου από την κάλυψη διακλαδώσεων, που με τη σειρά της απαιτεί λιγότερες σε σύγκριση με την κάλυψη διαδρομών.
- Όσο πιο πολύπλοκη είναι μία μονάδα, τόσο περισσότερες δοκιμασίες ελέγχου διαδρομών απαιτεί.
- Συνήθως ο έλεγχος διαδρομών είναι περισσότερο επιθυμητός, αφού εξετάζει το πρόγραμμα πιο ολοκληρωμένα, από τους άλλους δυο τύπους ελέγχου
- Παρ' όλα αυτά, για πολύπλοκα προγράμματα ο έλεγχος διαδρομών μπορεί να αποβεί υπερβολικά χρονοβόρος.

# συμπεράσματα

- Τόσο οι έλεγχοι κλειστού κουτιού όσο και οι έλεγχοι ανοιχτού κουτιού δεν μπορεί να είναι εξαντλητικοί.
- Στους ελέγχους κλειστού κουτιού δεν μπορούμε πάντα να εξαντλήσουμε το χώρο δεδομένων εισόδου.
- Από την άλλη μεριά στον έλεγχο ανοιχτού κουτιού δεν μπορούμε να εξαντλήσουμε όλες τις πιθανές διαδρομές εκτέλεσης της λογικής του λογισμικού.
- Ο έλεγχος κλειστού κουτιού πάσχει από αβεβαιότητα για το αν οι δοκιμασίες ελέγχου που επιλέχθηκαν θα ανακαλύψουν ένα συγκεκριμένο σφάλμα.
- Ο έλεγχος ανοιχτού κουτιού είναι ανοιχτός στον κίνδυνο να δοθεί περισσότερη προσοχή από όση θα έπρεπε στην εσωτερική δομή του κώδικα.

# συμπεράσματα

- Υπάρχει επίσης ο κίνδυνος να καταλήξουμε σε έναν έλεγχο του τι κάνει το πρόγραμμα και όχι του τι θα έπρεπε να κάνει.
- Όταν ελέγχουμε το λογισμικό, δε χρειάζεται να επιλέξουμε αποκλειστικά μεταξύ του ανοικτού και του κλειστού κουτιού. Μπορούμε να θεωρήσουμε αυτούς τους δύο τρόπους ελέγχου ως τα δύο άκρα μιας νοητής συνεχούς γραμμής ελέγχου. Κάθε προσέγγιση ελέγχου μπορεί να βρίσκεται κάπου ενδιάμεσα. Το ποια προσέγγιση θα διαλέξουμε εξαρτάται από ένα σύνολο παραγόντων στους οποίους συμπεριλαμβάνονται οι ακόλουθοι:
  - ο αριθμός των πιθανών λογικών διαδρομών,
  - η φύση των δεδομένων εισόδου,
  - η έκταση και η ποσότητα των υπολογισμών που απαιτούνται και
  - η πολυπλοκότητα των αλγορίθμων