

Προγραμματιστικές Συμβάσεις

- Γενικά χρησιμοποιούμε camelCase.
- Τα ονόματα κλάσεων ξεκινάνε με κεφαλαίο. π.χ. class MatrixFactorization
- Τα ονόματα μεταβλητών ξεκινάνε με πεζό. π.χ. int totalAuctionsCounter
- Περιορίζουμε ονόματα σε 24 χαρακτήρες. π.χ. totalLiveAuctionsIntegerCounter → totalLiveAuctions.
- Όλα τα πεδία και οι συναρτήσεις είναι static, εκτός αν δεν γίνεται να είναι (δηλαδή αν χρειάζεται να χρησιμοποιηθούν από άλλο αρχείο)
- Χρησιμοποιούμε ελάχιστα σχόλια. Σε περίπτωση που χρειάζεται εξήγηση η υλοποίηση, ‘σπάμε’ τον κώδικα ή εισάγουμε ‘ενδιάμεσα σημεία’ κώδικα που κάνουν πιο κατανοητή την λειτουργία. Δίνουμε προτεραιότητα στην αναγνωσιμότητα έναντι της λακωνικότητας και σε εύλογες περιπτώσεις έναντι επίδοσης.
- Πρώτα γράφουμε documentation και μετά κώδικα (ψευδο – test driven development).
- Στο documentation γράφουμε για κάθε μη στατική συνάρτηση: Είσοδο, έξοδο και σύντομη περιγραφή της λειτουργίας και για κάθε μη στατική κλάση μία σύντομη περιγραφή της έννοιας/οντότητας του ‘κόσμου’ που αναπαριστά (όπου έχει νόημα προφανώς)
- Βάζουμε #indef ... #endif σε κάθε .h αρχείο.
- Σε κάθε .cpp αρχείο η κάθε συνάρτηση είναι πλησιέστερα (από πάνω, αφού από κάτω θα έχει error) στην πρώτη άλλη συνάρτηση που την χρησιμοποιεί (αν δεν χρησιμοποιείται από κάποια, τότε πλησιέστερα στο τέλος του αρχείου) και μεταξύ τους είναι ταξινομημένες όπως και στο αντίστοιχο .h αρχείο
- Χρησιμοποιούμε κοινές εκδόσεις στα εργαλεία (compiler, libraries, ...) και τις συγκεκριμένες που αναφέρονται στις διαλέξεις (όπου αναφέρονται)
- Ενημερώνουμε κάθε εξέλιξη στην ανάπτυξη του project (ανάθεση module, ολοκλήρωση module, bug reports) μέσω κάποιου εργαλείου (μάλλον JIRA)
-