

Άσκηση 1 (προθεσμία 4/11/2015)

Γράψτε κώδικα-σκελετό που να δείχνει δύο από τα παρακάτω πρότυπα σχεδίασης (*design patterns*):

- Bridge
- Observer
- Visitor

Δεν είναι ανάγκη τα πρότυπα να συνδυαστούν - μπορούν να υλοποιηθούν από διαφορετικό σύνολο κλάσεων.

Ο κώδικας δεν χρειάζεται να παράγει αποτελέσματα (π.χ. οι μέθοδοι μπορεί απλά να τυπώνουν «κλήθηκε η μέθοδος ΧΨ»), αλλά πρέπει να έχει τέτοια δομή που το πρότυπο σχεδίασης να έχει νόημα. Για παράδειγμα για το πρότυπο Visitor μπορείτε να φανταστείτε ένα μεταγλωττιστή και να ονομάσετε τις κλάσεις σας αντίστοιχα (π.χ. AST, Stmt, Expr, PrintVisitor) ακόμα και αν οι κλάσεις δεν έχουν καθόλου κώδικα εκτός από το σκελετό που υλοποιεί το πρότυπο.

Χρησιμοποιείτε C++ ή Java (ή, ισοδύναμα, C#). Διαλέξτε τη γλώσσα που ξέρετε λιγότερο (δηλαδή αν ξέρετε καλά Java, διαλέξτε τη C++ και αντιστρόφως). Ο κώδικας πρέπει να μπορεί να περάσει από τον compiler και να τρέχει χωρίς δυναμικά σφάλματα (ακόμα κι αν δεν παράγει τίποτα).

Γράψτε μια σύντομη αναφορά (το πολύ 1 σελίδα) που να απαντάει τουλάχιστον στις εξής ερωτήσεις:

- Γιατί ταίριαζε για τη χρήση αυτή το πρότυπο που διαλέξατε;
- Ποια είναι τα πλεονεκτήματα/μειονεκτήματα της υλοποίησής σας;
- Πέρα από τις πολύ βασικές αντικειμενοστρεφείς λειτουργίες (υποσκελισμό μεθόδων, δυναμική κλήση) υπάρχουν στοιχεία στη γλώσσα που διαλέξατε που σας βοήθησαν να υλοποιήσετε το πρότυπο πιο εύκολα; (Τέτοια στοιχεία είναι π.χ. τα templates στη C++, οι τοπικές/εσωτερικές κλάσεις στη Java, η υπερφόρτωση/overloading και στις δύο γλώσσες.) Υπάρχουν στοιχεία σε άλλες γλώσσες που θα βοηθούσαν;

Η άσκηση παραδίδεται με email, έως τα μεσάνυχτα (11:59μμ) της προθεσμίας.