

## Άσκηση 1 (προθεσμία 30/11/2019)

Στην άσκηση αυτή θα χρησιμοποιήσετε δύο διαφορετικούς μηχανισμούς ευέλικτης εξέλιξης προγράμματος και θα τους συγκρίνετε. Η άσκηση έχει 2 προγραμματιστικά μέρη και ένα 3ο για τη σύγκριση.

1) Πάνω σε βιβλιοθήκη που χρησιμοποιεί το το πρότυπο σχεδίασης (design pattern) Visitor υλοποιήστε έναν συγκεκριμένο, γενικό PrintVisitor χρησιμοποιώντας την τεχνολογία dynamic proxy της Java.

Συγκεκριμένα, ξεκινήστε γράφοντας μια πολύ απλή ιεραρχία κλάσεων (π.χ. 5 κλάσεις + 1 interface Visitor) που υλοποιεί το Visitor pattern: οι κλάσεις έχουν κατάλληλες μεθόδους accept, κτλ. Πλέον αυτές τις κλάσεις θα τις θεωρήσετε αμετάβλητες. Στη συνέχεια, αντί να γράψετε με το χέρι έναν PrintVisitor που θα έχει ξεχωριστές μεθόδους visit για τις 5 κλάσεις, θα χρησιμοποιήσετε dynamic proxy που θα σας δημιουργήσει την ώρα της εκτέλεσης μια νέα κλάση που θα υλοποιεί το interface Visitor. Η λογική και των 5 μεθόδων αυτού του visitor θα υπάρχει (χρησιμοποιώντας reflection) μέσα στη δική σας μέθοδο invoke του invocation handler που χρησιμοποιείτε στον dynamic proxy.

2) Για αυτό το μέρος της άσκησης χρειάζεστε τη γλώσσα AspectJ. Κατεβάστε τη από το <http://www.eclipse.org/aspectj/> (ο σύνδεσμος υπάρχει και στη σελίδα του μαθήματος).

Υλοποιήστε μια απλή δομή αναζήτησης με δυαδικό δέντρο (χωρίς υποχρέωση να είναι ισορροπημένο), με τουλάχιστον τις συνηθισμένες λειτουργίες insert, remove και lookup. Στη συνέχεια βελτιώστε τη δομή σας ώστε να λειτουργεί σωστά σε πολυνηματικό περιβάλλον απλά με την προσθήκη ενός ξεχωριστού aspect (δηλαδή χωρίς να αγγίζετε τον αρχικό κώδικα της δομής). Μη συγχρονίσετε παραπάνω απ'όσο είναι απαραίτητο: λειτουργίες που μόνο διαβάζουν δεδομένα πρέπει να μπορούν να εκτελεστούν ταυτόχρονα (π.χ. πολλαπλές λειτουργίες lookup μπορεί να εκτελούνται την ίδια στιγμή αλλά ένα lookup δεν μπορεί να εκτελείται ταυτόχρονα με ένα insert ή ένα remove).

3) Συγκρίνετε αυτό που κάνατε στα δύο παραπάνω βήματα σε μια σύντομη αναφορά (ενδεικτικά: 1 σελίδα). Δεν θα μπορούσατε να κάνετε το βήμα 1 με AspectJ; Το βήμα 2 με dynamic proxy; Ποια είναι τα πλεονεκτήματα και μειονεκτήματα της κάθε προσέγγισης; Σκεφτείτε για παράδειγμα τις εξής διαστάσεις:

- πότε η ευελιξία είναι στατική και πότε δυναμική
- ασφάλεια και ταχύτητα εκτέλεσης για τον κάθε μηχανισμό
- compile έναντι run-time dependencies σε κάθε περίπτωση (δηλ. ποιος μηχανισμός επιτρέπει να μην υπάρχει κάποιος κώδικας στο περιβάλλον compilation αλλά μόνο στο περιβάλλον τελικής εκτέλεσης).

Δώστε αυτόματο τρόπο μεταγλώττισης και εκτέλεσης για διευκόλυνση του βαθμολογητή.

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