

Εργασία 3 (Java Threads).

1. Μελετήστε και εκτελέστε τον κώδικα των παραδειγμάτων 6, 7 και 8 στο σύνδεσμο

<http://www.it.uom.gr/teaching/ParallelDistributedJava/threadsoftware.html>

1.1. Στα παραδείγματα 6, 7

Δημιουργείτε νήματα `AccountHolder` τα οποία να εκτελούν αρκετές διαδοχικές συναλλαγές διακοπτόμενες από τυχαίες καθυστερήσεις σε διάφορα σημεία. Προσπαθείστε να πετύχετε συνθήκες ανταγωνισμού (λανθασμένο αποτέλεσμα). Διορθώστε το αποτέλεσμα με τη βοήθεια αμοιβαίου αποκλεισμού, όπως στο παράδειγμα 7.

1.2. Στο παράδειγμα 8

Υλοποιείτε την εξής αλλαγή. Η καθολική μεταβλητή `sum` τοποθετείται σε μια κλάση με μοναδική `synchronized` μέθοδο την `addsum`. Δηλαδή κατασκευάζουμε ένα `monitor`. Τα νήματα αντί για `synchronized(this)` καλούν αυτή τη μέθοδο.

2. Υπολογισμός του π με αριθμητική ολοκλήρωση (επισυνάπτεται κώδικας σε C).

Συγκρίνετε τους χρόνους εκτέλεσης για 1εκ, 100 εκ, 1δισ, 10δισ και 1, 2, 4, 8, 16 νήματα στοιχεί των παρακάτω εκδοχών

2.1. Λύση με ακολουθιακή άθροιση (<https://software.intel.com/en-us/articles/writing-parallel-programs-a-multi-language-tutorial-introduction>)

2.2. Λύση με `synchronized sum` αλλά χωρίς `partial sums` ανά thread (πολύ αργή)

2.3. Λύση με `partial sums` ανά thread και `synchronized (this)` στο `sum` – όπως το αρχικό παράδειγμα 8 (άσκηση 1.2).

2.4. Λύση με δημιουργία κλάσης `sum` και `synchronized` μεθόδου – όπως το τροποποιημένο παράδειγμα 8 (άσκηση 1.2).

3. Μέτρηση συχνότητας εμφάνισης χαρακτήρων στο κείμενο (επισυνάπτεται κώδικας σε C, επίσης μπορείτε να βρείτε στο http://rosettacode.org/wiki/Letter_frequency#Java). Ως κείμενα μπορείτε να δοκιμάσετε το `world192.txt` από το Large Set <http://corpus.canterbury.ac.nz/descriptions/#large>

Η λύση της άσκησης είναι παρόμοια με τη 2. Μπορείτε δηλαδή να υλοποιήσετε τις 4 εκδοχές σχετικά εύκολα. Η βασική διαφορά βρίσκεται στο γεγονός ότι τώρα πρέπει να προστατευθεί ένας πίνακας από αθροίσματα (ο πίνακας `freq[]`).

3.1. Υλοποιείτε μια λύση ανάλογη της 2.3. Μετρήστε χρόνους για 1, 2, 4, 8 νήματα

3.2. Υλοποιείτε μια λύση ανάλογη της 2.4. Μετρήστε χρόνους για 1, 2, 4, 8 νήματα

3.3. Προσπαθείστε άλλες λύσεις όπως `array of locks` ή `array of atomicInteger`.

