

Παράλληλος και Κατανεμημένος Υπολογισμός

Εργαστήριο 3

Αμοιβαίος αποκλεισμός (συνέχεια)

1. Ξεκινήστε από τους 6 κώδικες της ερώτησης 2 του Εργαστηρίου 2 (χρήση Mutex/Lock)

Μελετήστε τους κώδικες `Code/SharedCounterSync` και παράγετε λύσεις για το ίδιο πρόβλημα με **τυπική** χρήση `synchronized`, δηλαδή με κλείδωμα μοιραζόμενου αντικειμένου.

Προσπαθείστε **να αποφύγετε** λύσεις όπως αυτές που δίνονται στους κώδικες `Code/SharedCounterSyncAlt`. Σε περίπτωση που εφαρμόσετε τέτοιες λύσεις, γράψτε σε σχόλια τους λόγους που δεν μπορείτε να χρησιμοποιήσετε την τυπική λύση.

(Σημείωση: αν χρησιμοποιείτε άλλη γλώσσα αναζητήστε μηχανισμούς που διευκολύνουν τη χρήση κλειδωμάτων, δηλαδή υποκαθιστούν το `lock-unlock` με κάποια ‘ανώτερη’ δομή).

2. Ξεκινήστε από **έναν** κώδικα της ερώτησης 2 του Εργαστηρίου 2 και **έναν** κώδικα από τους αντίστοιχους της ερώτησης 1 του Εργαστηρίου 3 (όποιους θέλετε). Μελετήστε τους κώδικες `/Code/ConcurrentCollections`.

2.1 Τροποποιείτε τους κώδικες ώστε να χρησιμοποιήσετε **λεπτομερές** κλείδωμα στη δομή δεδομένων του προβλήματος.

2.2 Προαιρετικά, προσπαθείστε να χρησιμοποιήσετε κάποια βιβλιοθήκη με ταυτόχρονες δομές δεδομένων *(Σημείωση: ανάλογα με τις βιβλιοθήκες που διαθέτει η γλώσσα που χρησιμοποιείτε).*

3. Ξεκινήστε από τον κώδικα `/Code/Dining`. Πειραματιστείτε με τις καθυστερήσεις για να ελέγξετε τη λειτουργία του.

Παράγετε τουλάχιστον μια λύση ώστε να μην έχουμε αδιέξοδα.