



Αντικειμενοστρεφής Προγραμματισμός

Ενότητα 13: Ταυτόχρονος Προγραμματισμός

Γρηγόρης Τσουμάκας, Επικ. Καθηγητής
Τμήμα Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΙΚΑ
ΜΑΘΗΜΑΤΑ



Ταυτόχρονος Προγραμματισμός



Τα παραδείγματα κώδικα που χρησιμοποιούνται σε κάποιες από τις ακόλουθες διαφάνειες μπορούν να βρεθούν στον παρακάτω σύνδεσμο:
<http://users.auth.gr/greg/oop.zip>

Ταυτόχρονο Λογισμικό

- Λειτουργικά συστήματα.
 - Προγραμματίζουμε, ακούμε μουσική, γράφουμε έγγραφα, εκτυπώνουμε, κατεβάζουμε αρχεία, κ.α.
- Εφαρμογή διαδικτυακού ραδιοφώνου.
 - Ανάγνωση δεδομένων από το δίκτυο, αποσυμπίεση, αναπαραγωγή, ενημέρωση διεπαφής.

Χρονομερίδια



Διαδικασίες και Νήματα

- Διαδικασίες (processes).
 - Έχουν το δικό τους περιβάλλον εκτέλεσης, όπως χώρο μνήμης και ανοιχτά αρχεία.
 - Συνήθως μια διαδικασία είναι ένα πρόγραμμα.
- Νήματα (threads).
 - Τα νήματα μοιράζονται τους πόρους της διαδικασίας στην οποία ανήκουν.
 - Μια διαδικασία έχει ένα ή παραπάνω νήματα.



Ταυτόχρονος Προγραμματισμός

- Πολυνηματικές εφαρμογές.
 - Ανάπτυξη εφαρμογών που μπορούν να εκτελούν ταυτόχρονα πολλά νήματα.
- Οφέλη.
 - Δημιουργία ταυτόχρονου λογισμικού.
 - Επιτάχυνση εκτέλεσης εργασιών μέσω της εκμετάλλευσης πολλαπλών πυρήνων.



Δημιουργία και Εκτέλεση Νήματος

- Τρόπος Α.
 - Υποσκέλιση της μεθόδου *run()* της κλάσης *Thread*.
 - Δημιουργία αντικειμένου της νέας κλάσης και κλήση της μεθόδου *start()* της κλάσης *Thread*.
- Τρόπος Β.
 - Πέρασμα στον κατασκευαστή της *Thread* ενός αντικειμένου μιας κλάσης που υλοποιεί τη διεπαφή *Runnable*, η οποία περιέχει μέθοδο *run()*.
 - Κλήση της *start()*.

HelloRunnable.java, HelloThread.java



Αναστολή Εκτέλεσης

- Αναστολή εκτέλεσης.
 - `Thread.sleep(long millis)`.
 - `Thread.sleep(long millis, int nanos)`.
- Απαιτεί χειρισμό της *InterruptedException*.
 - Σε περίπτωση που κάποιο άλλο νήμα το ενημερώσει να διακόψει την αναστολή εκτέλεσης.

Sleep.java



Διακοπή (Interrupt)

- Διακοπή.
 - Ενημέρωση ενός νήματος ότι θα πρέπει να σταματήσει αυτό που κάνει και να κάνει κάτι άλλο.
 - Ο προγραμματιστής καθορίζει το τι ακριβώς.
- Ενημέρωση του νήματος.
 - Κλήση της μεθόδου *interrupt()* του νήματος.
 - Κλήση της μεθόδου *Thread.interrupted()* από το νήμα για να ελέγξει αν του έχει σταλεί διακοπή.

Interrupt.java



Αναμονή Ολοκλήρωσης

- Αναμονή ολοκλήρωσης ενός νήματος.
 - Κλήση της μεθόδου *join()* του νήματος που θέλουμε να περιμένουμε να τελειώσει.
- Απαιτεί χειρισμό της *InterruptedException*.
 - Σε περίπτωση που κάποιο άλλο νήμα το ενημερώσει να διακόψει την αναμονή ολοκλήρωσης.

Join.java



Συγχρονισμός

- Προβλήματα συγχρονισμού.
 - Μπέρδεμα νημάτων.
- Αντιμετώπιση.
 - Συγχρονισμένες μέθοδοι.
 - Συγχρονισμένες προτάσεις.



Μπέρδεμα Νημάτων

- Μπέρδεμα νημάτων.
 - Δύο λειτουργίες που περιλαμβάνουν πολλαπλά βήματα και αφορούν στην ίδια δομή δεδομένων.
 - Αν τα βήματα εκτελεστούν ταυτόχρονα, ενδέχεται το αποτέλεσμα να μην είναι το επιθυμητό.

CounterThread.java, Counter.java



Συγχρονισμένες Μέθοδοι

- Προσθήκη *synchronized* στη δήλωση.
 - Δεν επιτρέπεται να εκτελεστούν ταυτόχρονα δύο συγχρονισμένες μέθοδοι του ίδιου αντικειμένου.
 - Άλλες κλήσεις προς συγχρονισμένες μεθόδους περιμένουν μέχρι να τελειώσει η τρέχουσα.

SynchronizedCounterThread.java, SynchronizedCounter.java



Εγγενείς Κλειδαριές

- Εγγενής κλειδαριά.
 - Κάθε αντικείμενο έχει και μια κλειδαριά.
 - Όταν ένα νήμα καλεί μια συγχρονισμένη μέθοδο, αυτομάτως κλειδώνει το αντικείμενο.
 - Όταν τελειώσει με την εκτέλεση της μεθόδου, τότε το αντικείμενο ξεκλειδώνεται.



Συγχρονισμένες Προτάσεις (1/2)

- Συγχρονισμένες προτάσεις.
 - Μπλοκ συγχρονισμένου κώδικα που καθορίζουν και το αντικείμενο που κλειδώνουν.
- Παράδειγμα.

```
public void addName(String name) {  
    synchronized(this) {  
        lastName = name;  
        nameCount++;  
    }  
    // κώδικας που δεν απαιτεί συγχρονισμό  
}
```



Συγχρονισμένες Προτάσεις (2/2)

```
public class Locks {  
    private long c1 = 0;  
    private long c2 = 0;  
    private Object lock1 = new Object();  
    private Object lock2 = new Object();  
  
    public void incl1() {  
        synchronized(lock1) { c1++; }  
    }  
    public void decl1() {  
        synchronized(lock1) { c1--; }  
    }  
    public void inc2() {  
        synchronized(lock2) { c2++; }  
    }  
    public void dec2() {  
        synchronized(lock2) { c2--; }  
    }  
}
```



Αμετάλλακτα Αντικείμενα

- Υπενθύμιση.
 - Αντικείμενα των οποίων η κατάσταση δεν μπορεί να αλλάξει.
 - Π.χ. αντικείμενα της κλάσης String.
- Χρήσιμα στον ταυτόχρονο προγραμματισμό.
 - Δεν πρόκειται να βρεθούν σε «ανεπιθύμητη» κατάσταση λόγω προβλημάτων συγχρονισμού.



Παράδειγμα (1/2)

```
public class SynchronizedRGB {
    private int red;

    private int green;

    private int blue;

    public SynchronizedRGB(int r, int g, int b) {
        red = r; green = g; blue = b;
    }

    public synchronized void set(int r, int g, int b) {
        red = r; green = g; blue = b;
    }

    public synchronized void invert() {

        red = 255 - red;
        green = 255 - green;
        blue = 255 - blue;
    }
}
```



Παράδειγμα (2/2)

```
public class ImmutableRGB {
    private int red;

    private int green;

    private int blue;

    public ImmutableRGB(int r, int g, int b) {
        red = r; green = g; blue = b;
    }

    public ImmutableRGB invert() {

        return new ImmutableRGB(255-red, 255-green, 255-blue);
    }
}
```



Νήματα και Γραφικές Διεπαφές

- Νήμα αποστολής γεγονότων.
 - Νήμα στο οποίο τρέχει ο κώδικας διαχείρισης γεγονότων στη Swing.
 - Απαραίτητο για αποφυγή προβλημάτων συγχρονισμού.
- Τι θα συμβεί όμως αν μια λειτουργία καθυστερήσει να τελειώσει;
 - Η αλληλεπίδραση θα «παγώσει».

Concurrency.java



Παράλληλο Άθροισμα Πίνακα

- Παράδειγμα αξιοποίησης νημάτων για επιτάχυνση της εκτέλεσης σε περιβάλλοντα με πολλούς πυρήνες.

ParallelSum.java, PartialSum.java





Τέλος Ενότητας

Επεξεργασία: Εμμανουήλ Ρήγας
Θεσσαλονίκη, Εαρινό Εξάμηνο 2013-2014



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ