

# Δεύτερη εργασία στο μάθημα Αντικειμενοστραφούς

## Προγραμματισμού

Τμήμα Πληροφορικής και Τηλεπικοινωνιών (Άρτα)

Πανεπιστήμιο Ιωαννίνων

Χρήστος Γκόγκος

4 Δεκεμβρίου 2023

### 1 Εισαγωγή

Σκοπός της εργασίας είναι η εφαρμογή του αντικειμενοστραφούς προγραμματισμού για την ανάπτυξη μιας εφαρμογής που θα αντιμετωπίζει προβλήματα προγραμματισμού ωρών έναρξης και τερματισμού εργασιών σε βιομηχανικό περιβάλλον. Έμφαση δίνεται στην υλοποίηση χρησιμοποιώντας χαρακτηριστικά του αντικειμενοστραφούς προγραμματισμού, όπως η ενθυλάκωση (απόκρυψη πληροφορίας), ο πολυμορφισμός, η σύνθεση αντικειμένων (composition) κ.α.

### 2 Περιγραφή προβλήματος

Έστω ένα εργοστάσιο που αποτελείται από ένα σύνολο μηχανών (machines),  $1, 2, \dots$ , που μπορούν να επεξεργάζονται εργασίες (jobs). Δίνεται ένα σύνολο  $N$  εργασιών,  $J_1, J_2, \dots, J_N$  με κάθε εργασία να αποτελείται από ένα σύνολο επιμέρους λειτουργιών (operations) που καθεμία εκτελείται σε ένα μηχάνημα. Για παράδειγμα, για την εργασία  $J_i$  μπορεί να υπάρχουν  $k$  λειτουργίες  $O_{i,1}, O_{i,2}, \dots, O_{i,k}$  που πρέπει να γίνουν με τη σειρά που αναφέρονται με κάθε μια εργασία να εκτελείται σε ένα μηχάνημα για γνωστό χρονικό διάστημα κατά τη διάρκεια του οποίου το συγκεκριμένο μηχάνημα είναι απασχολημένο και δεν μπορεί να χρησιμοποιηθεί από άλλες εργασίες. Για κάθε εργασία επιπλέον δίνεται ο χρόνος στον οποίο θα έπρεπε η εργασία να είχε ολοκληρωθεί (due date) και είναι επιθυμητό η εργασία να ολοκληρώνεται πριν το χρόνο αυτό. Το ζητούμενο είναι η δημιουργία ενός προγράμματος που να ελαχιστοποιεί τη συνολική παραβίαση των επιθυμητών χρόνων ολοκλήρωσης από όλες τις εργασίες μαζί.

### 3 Ζητούμενα

Ζητείται η κατασκευή ενός προγράμματος που να είναι σε θέση να διαβάζει στιγμιότυπα προβλημάτων από το <https://www.eii.uva.es/elena/JSSP/InstancesJSSP.htm> () και να τα αποθηκεύει σε κατάλληλες δομές, έτσι ώστε να συμμετέχουν στους μετέπειτα υπολογισμούς. Για παράδειγμα, το στιγμιότυπο προβλήματος 1a05 έχει τα ακόλουθα περιεχόμενα:

10

5

593

87 72 66 60 95

48 54 39 35 5

97 46 21 20 55

59 34 37 19 46  
 28 24 73 25 23  
 45 78 83 28 5  
 53 37 12 71 29  
 38 55 87 33 12  
 48 40 49 83 7  
 90 23 65 17 27  
 2 1 5 3 4  
 5 4 1 3 2  
 2 4 3 1 5  
 1 4 5 2 3  
 5 3 4 2 1  
 4 1 5 2 3  
 1 4 2 5 3  
 5 3 4 2 1  
 3 4 2 1 5  
 3 4 1 5 2

Οι πρώτες δύο γραμμές είναι το πλήθος των εργασιών (10) και των μηχανών αντίστοιχα (5). Η τρίτη γραμμή περιέχει την τιμή 593 που είναι η βέλτιστη τιμή χρόνου ολοκλήρωσης όλων των εργασιών (δεν θα ασχοληθούμε με αυτή την τιμή). Οι επόμενες 10 γραμμές είναι οι χρόνοι εκτέλεσης για κάθε εργασία από το μηχάνημα 1 μέχρι και το μηχάνημα 5. Για παράδειγμα, η εργασία 1 χρειάζεται χρόνο εκτέλεσης 87 μονάδων χρόνου στο μηχάνημα 1, 72 μονάδων χρόνου στο μηχάνημα 2 κ.ο.κ. Ομοίως, στο συγκεκριμένο παράδειγμα η εργασία 10 χρειάζεται χρόνο εκτέλεσης 90 μονάδων στο μηχάνημα 1, 23 μονάδων στο μηχάνημα 2 κ.ο.κ. Οι επόμενες 10 γραμμές αναπαριστούν την ακολουθία επίσκεψης των μηχανημάτων για κάθε εργασία. Συνεπώς, η εργασία 1 επισκέπτεται πρώτα το μηχάνημα 2 όπου θα πραγματοποιήσει επεξεργασία για 72 μονάδες χρόνου, μετά το μηχάνημα 1 για 87 μονάδες χρόνου, μετά το μηχάνημα 5 για 95 μονάδες χρόνου κ.ο.κ.

Καθώς δεν περιέχεται στο αρχείο δεδομένων η πληροφορία για τον επιθυμητό χρόνο τερματισμού, αυτή πρέπει να υπολογιστεί. Αυτό συμβαίνει ως εξής: Υπολογίζεται η συνολική διάρκεια κάθε εργασίας σε όλα τα μηχανήματα και η τιμή αυτή πολλαπλασιάζεται με το συντελεστή  $f = 1.3$ . Από το αποτέλεσμα λαμβάνεται το ακέραιο μέρος. Συνεπώς, για την εργασία 1, ο επιθυμητός χρόνος τερματισμού είναι  $1.3 * (87 + 72 + 66 + 60 + 95) = 494$ , για την εργασία 2 είναι 235 κ.ο.κ.

### 3.1 Ερωτήματα Α

Το πρόγραμμα να είναι σε θέση να εκτελεί τα ακόλουθα, εφόσον πρώτα έχει φορτωθεί ένα στιγμιότυπο προβλήματος:

1. Ο χρήστης να μπορεί να επιλέγει μια εργασία και το πρόγραμμα να εμφανίζει το χρόνο αργότερης ολοκλήρωσης του, και τους χρόνους εκτέλεσης κάθε λειτουργίας και το μηχάνημα με τη σειρά που πρέπει να εκτελεστεί η εργασία.
2. Να εμφανίζει ανάλογα με την επιλογή του χρήστη:
  - (α') Εργασίες ταξινομημένες κατά αργότερο χρόνο ολοκλήρωσης (due date).
  - (β') Εργασίες ταξινομημένες κατά συνολικό χρόνο εκτέλεσης.
  - (γ') Μηχανήματα ταξινομημένα κατά συνολικό χρόνο εκτέλεσης από όλες τις εργασίες.

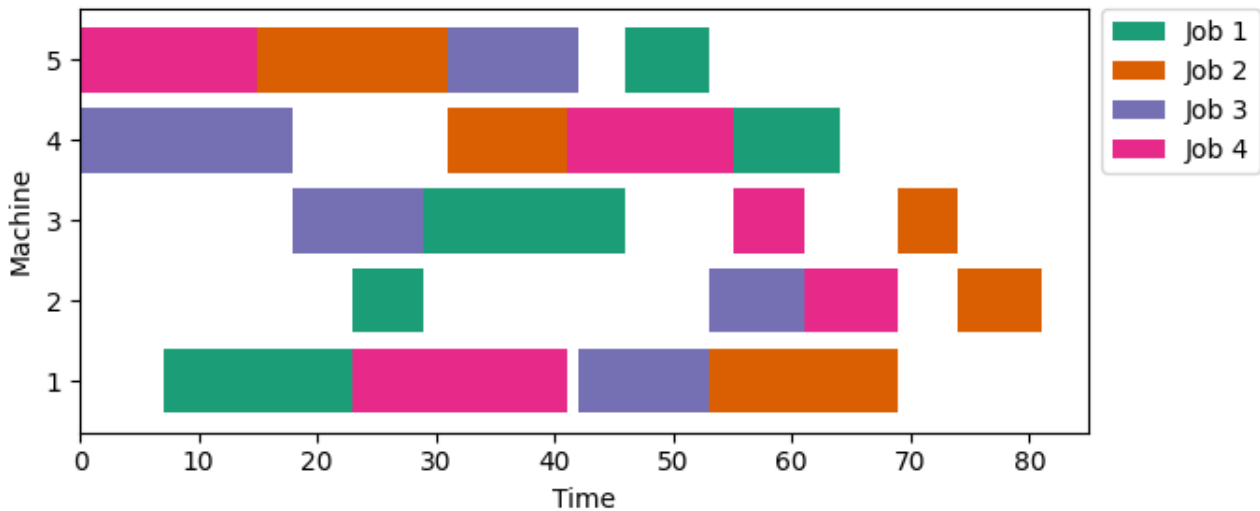
### 3.2 Ερωτήματα Β

Επιπλέον, ζητείται η δημιουργία ενός προγράμματος εργασιών που να είναι έγκυρο (να έχουν προγραμματιστεί όλες οι εργασίες και να μην υπάρχει επικάλυψη εργασιών σε οποιαδήποτε από τις μηχανές). Με τρόπο που θα επιλέξετε να εμφανίζει στην γραμμή εντολών το παραγόμενο πρόγραμμα.

### 3.3 Προαιρετικές λειτουργίες - επιπλέον βαθμολογία +20%

Οι φοιτητές και οι φοιτήτριες που αναζητούν επιπλέον προγραμματιστικές προκλήσεις μπορούν να υλοποιήσουν κάποια από τα ακόλουθα:

- Απεικόνιση του προγράμματος εργασιών γραφικά (π.χ. Σχήμα 1).
- Υλοποίηση ενός «καλού» επιλυτή των προβλημάτων.



Σχήμα 1: Ένα παράδειγμα προγραμματισμού εργασιών.

### 3.4 Κριτήρια βαθμολόγησης

Τα κριτήρια βαθμολόγησης της εργασίας είναι τα ακόλουθα:

- Ερωτήματα Α. (40%)
- Ερωτήματα Β. (30%)
- UML διάγραμμα κλάσεων με τις κλάσεις που έχουν χρησιμοποιηθεί για τη λύση του προβλήματος. (10%).
- Unit tests με το catch2 για επιμέρους λειτουργίες της εφαρμογής. (10%)
- Τεχνική αναφορά. (10%)

## 4 Υποβολή εργασίας

Σχετικά με την υποβολή της εργασίας ισχύουν τα ακόλουθα:

- Η εργασία είναι προαιρετική.
- Η εργασία μπορεί να υποβληθεί μόνο στο [ecourse: https://ecourse.uoi.gr/course/view.php?id=1944](https://ecourse.uoi.gr/course/view.php?id=1944).
- Η προθεσμία υποβολής της εργασίας είναι στις 10/1/2024.
- Η εργασία μπορεί να γίνει σε ομάδες των 2 ατόμων, σε ομάδες που θα δηλωθούν το αργότερο μέχρι τις 15/12/2023 με email στο [cgogos@uoi.gr](mailto:cgogos@uoi.gr).
- Η γλώσσα υλοποίησης της εργασίας είναι η C++.
- Παραδοτέα εργασίας (σε ένα zip αρχείο):
  - Κώδικας με επαρκή σχολιασμό.
  - Οδηγίες μεταγλώττισης και εκτέλεσης σε ένα αρχείο README.md.
  - Τεχνική αναφορά.