

Εργασία εξαμήνου στο μάθημα Αντικειμενοστραφής

Προγραμματισμός

Τμήμα Πληροφορικής και Τηλεπικοινωνιών (Άρτα)

Πανεπιστήμιο Ιωαννίνων

Χρήστος Γκόγκος

29 Οκτωβρίου 2025

1 Εισαγωγή

Σκοπός της εργασίας είναι η εξοικείωση με τον αντικειμενοστραφή προγραμματισμό και με τον τρόπο που η γλώσσα προγραμματισμού C++ χρησιμοποιείται στα πλαίσια του αντικειμενοστραφούς προγραμματισμού.

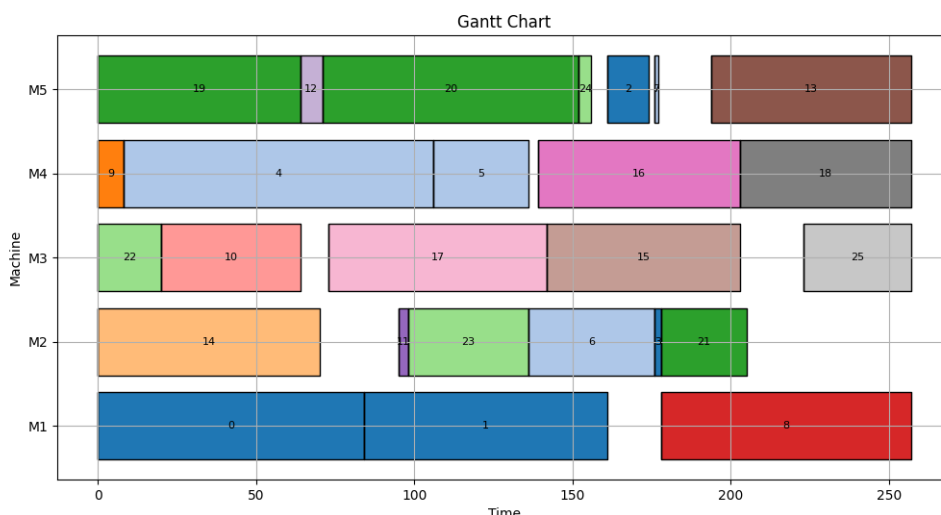
2 Εκφώνηση εργασίας

Ζητούμενο της εργασίας είναι η κατασκευή μιας εφαρμογής που θα είναι σε θέση να επιλύει προβλήματα χρονοπρογραμματισμού εργασιών (Job Shop Scheduling Problem - JSSP). Το πρόβλημα JSSP αφορά τον προσδιορισμό της βέλτιστης ακολουθίας με την οποία ένα σύνολο εργασιών, καθεμία αποτελούμενη από μια σειρά διαδοχικών λειτουργιών, πρέπει να εκτελεστεί σε ένα σύνολο μηχανών, όπου κάθε λειτουργία απαιτεί τη χρήση μιας συγκεκριμένης μηχανής για προκαθορισμένο χρόνο. Κάθε μηχανή μπορεί να εκτελεί μόνο μία λειτουργία τη φορά και η σειρά εκτέλεσης των λειτουργιών για κάθε εργασία πρέπει να διατηρείται. Ο στόχος είναι η ελαχιστοποίηση του συνολικού χρόνου ολοκλήρωσης όλων των εργασιών (makespan).

Μπορείτε να δείτε παραδείγματα σχετικών εφαρμογών στα ακόλουθα:

- https://github.com/ai-for-decision-making-tue/Job_Shop_Scheduling_Benchmark_Environments_and_Instances
- <https://github.com/thomasWeise/jsspInstancesAndResults>
- https://github.com/Pabloo22/job_shop_lib
- https://github.com/dothinking/jsp_framework
- <https://github.com/tamy0612/JSPLIB>
- <https://github.com/wsgisler/job-shop-scheduling>

Περιγραφή της εφαρμογής Ζητείται να κατασκευάσετε μια εφαρμογή που να επιτρέπει στους χρήστες να φορτώνουν προβλήματα JSSP, να επιλέγουν πρόβλημα, να επιλύουν το πρόβλημα υλοποιώντας μια οποιαδήποτε μέθοδο που δίνει έγκυρη λύση και να απεικονίζουν τη λύση του προβλήματος με τη μορφή ενός διαγράμματος Gantt, παρόμοιο με το Σχήμα 1.



Σχήμα 1: Ένα παράδειγμα οπτικοποίησης λύσης για το JSSP από το https://github.com/ai-for-decision-making-tue/Job_Shop_Scheduling_Benchmark_Environments_and_Instances.

Λειτουργίες εργασίας Η εφαρμογή πρέπει να υλοποιηθεί με αντικειμενοστραφή προσέγγιση και να περιλαμβάνει τις παρακάτω λειτουργίες:

- Ανάγνωση λίστας προβλημάτων: Θα διαβάζει αρχεία προβλημάτων που θα βρίσκονται σε έναν φάκελο και θα εμφανίζει λίστα με τα ονόματα τους.
- Επιλογή προβλήματος: Επιλογή ενός προβλήματος, εμφάνιση χαρακτηριστικών των προβλημάτων (π.χ. όνομα προβλήματος, πλήθος εργασιών, πλήθος μηχανών, λειτουργίες ανά εργασία κ.α.).
- Επίλυση προβλήματος: Επίλυση του επιλεχθέντος προβλήματος με οποιοδήποτε τρόπο δίνει έγκυρη λύση, εμφάνιση χαρακτηριστικών λύσης, εξαγωγή λύσης προβλήματος σε μορφή αρχείου λύσης (με ελεύθερη επιλογή του τρόπου αποθήκευσης της λύσης, π.χ. αρχείο απλού κειμένου, JSON, XML).
- Οπτικοποίηση λύσης: Εμφάνιση λύσης που είτε μόλις έχει παραχθεί, είτε φορτώνεται από αρχείο στο οποίο έχει αποθηκευτεί.

Τεχνικές απαιτήσεις Οι τεχνικές απαιτήσεις της εφαρμογής είναι οι ακόλουθες:

- Αντικειμενοστραφής σχεδίαση: Ορίστε κατάλληλες κλάσεις και αντικείμενα για την υλοποίηση της λύσης σας στο πρόβλημα JSSP και για τη φόρτωση, αποθήκευση και οπτικοποίηση της λύσης.
- Μοναδιαίοι έλεγχοι (unit tests): Προσθέστε μοναδιαίους ελέγχους που να ελέγχουν όλη τη λειτουργικότητα της εφαρμογής.
- Διεπαφή χρήστη: Δημιουργήστε διεπαφή έτσι ώστε η εφαρμογή να μπορεί να χρησιμοποιηθεί εύκολα και διαισθητικά. Η διεπαφή μπορεί να είναι είτε γραφική (Graphical User Interface - GUI) είτε γραμμής εντολών (Command Line Interface - CLI). Αν επιλέξετε GUI, μπορείτε να χρησιμοποιήσετε βιβλιοθήκες όπως η wxWidgets¹ ή μπορεί η διεπαφή μόνο της εφαρμογής να γίνει σε άλλη γλώσσα προγραμματισμού ή τεχνολογία (π.χ. Python, Javascript).
- Διάγραμμα κλάσεων: Σχεδιάστε ένα διάγραμμα κλάσεων που να απεικονίζει όλες τις κλάσεις της εφαρμογής και τις συσχετίσεις μεταξύ τους.

2.1 Κριτήρια βαθμολόγησης

Τα κριτήρια βαθμολόγησης της εργασίας είναι τα ακόλουθα:

¹<https://www.wxwidgets.org/>

- Ορθή εφαρμογή αρχών αντικειμενοστραφούς σχεδίασης και προγραμματισμού με τη C++ (30%).
- Λειτουργικότητα και αξιοπιστία (30%).
- Σχεδίαση και υλοποίηση διεπαφής χρήστη (10%).
- Έγερξη unit tests με το catch2² ή με άλλο testing framework της C++, για τις επιμέρους λειτουργίες της εφαρμογής (10%).
- Διαγράμματα UML (10%).
- Παρουσίαση εφαρμογής (10%).

2.2 Προαιρετικές λειτουργίες - επιπλέον βαθμολόγηση μέχρι +20%

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

- Ιδιωτικό αποθετήριο στο github³ με τον κώδικα της εργασίας (διαμοιρασμός με τον χρήστη cgogos@uoi.gr). Το αποθετήριο μπορεί να γίνει δημόσιο μετά την ολοκλήρωση βαθμολόγησης της εργασίας.
- Εκτέλεση της εφαρμογής σε Docker⁴ container.
- Υλοποίηση πρωτότυπου και αποδοτικού τρόπου εύρεσης λύσης των προβλημάτων JSSP.

3 Υποβολή εργασίας

Σχετικά με την υποβολή της εργασίας ισχύουν τα ακόλουθα:

- Η εργασία μπορεί να υποβληθεί μόνο στο ecourse: <https://ecourse.uoi.gr/course/view.php?id=1944>.
- Η εργασία είναι ατομική.
- Η προθεσμία υποβολής της εργασίας είναι η 11/1/2026.
- Η γλώσσα υλοποίησης της εργασίας είναι η C++.
- Υποχρέωση υποβολής βίντεοσκοπημένης παρουσίασης διάρκειας περίπου 10 λεπτών με περιγραφή της αντικειμενοστραφούς σχεδίασης της εφαρμογής και του τρόπου υλοποίησής της.
- Παραδοτέα εργασίας (σε ένα zip αρχείο):
 - Κώδικας με επαρκή σχολιασμό.
 - Οδηγίες μεταγλώττισης και εκτέλεσης σε ένα αρχείο README.md.
 - Έγγραφο με screenshots από την εκτέλεση της εφαρμογής.
 - Αρχείο βίντεοσκοπημένης παρουσίασης.

²<https://github.com/catchorg/Catch2>

³<https://github.com/>

⁴<https://www.docker.com/>