

ΕΡΓΑΣΙΑ 1 ΤΕΧΝΙΤΗ ΝΟΗΜΟΣΥΝΗ

Καθηγητής: Ίων Ανδρουτσόπουλος

Συντελεστές:

ERNEST BETA (3220239)

ΠΑΝΑΓΙΩΤΗΣ ΜΠΙΧΑΛΤΖΗΣ (3220133)

ΜΕΡΟΣ Α

ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ

Για το μέρος Α, επιλέξαμε το Β θέμα που αφορά την δημιουργία ενός ρεαλιστικού σχολικού προγράμματος γυμνασίου, λαμβάνοντας υπόψη τους περιορισμούς της εκφώνησης και κατανέμοντας ομοιόμορφα τις ώρες και τα μαθήματα ώστε να επιτευχθεί βέλτιστη οργάνωση.

Οι κλάσεις που χρησιμοποιήθηκαν είναι οι εξής:

- **Teacher.java**
- **Lesson.java**
- **Timetable.java**
- **TimetableSlot.java**
- **Chromosome.java**
- **GeneticAlgorithm.java**
- **Main.java**

Χρησιμοποιηθήκαν, επίσης, 2 αρχεία τα οποία διέθεταν τα μαθήματα που διαδασκόντουσαν (lessons.txt) και τους καθηγητές του γυμνασίου (teachers.txt)

- Η μορφή του **lessons.txt** είναι η εξής:

SubjectID#Subject#Class#Hours

GR101#Γλωσσική Διδασκαλία#A,B,C#3,2,2

- Η μορφή του **teachers.txt** είναι η εξής:

T001#ΓΕΩΡΓΙΟΣ ΠΑΠΑΔΟΠΟΥΛΟΣ#GR106#5#18

ΑΝΑΛΥΣΗ

1) Teacher.java

Ιδιότητες:

- a. private String **teacherID**;
- b. private String **name**;
- c. private List<String> **subjects**;
- d. private int **maxHoursPerDay**;
- e. private int **maxHoursPerWeek**;

Getter Methods:

getTeacherID(), getName(), List<String> getSubjects(), getMaxHoursPerDay(),
getMaxHoursPerWeek()

Μέθοδοι:

boolean canTeach(String subjectID):

μέθοδος για να ελέγξουμε αν το αντικείμενο καθηγητής μπορεί να διδάξει το μάθημα
subjectID

static List<Teacher> parseTeachersFile(String filename):

καθιερωμένη μέθοδος για να διαβάσει από το teachers.txt τους καθηγητές, και να
δημιουργήσει κατάλληλα την λίστα teachers και τις προσθέτει αντικείμενα Teacher,
δίνοντάς στο καθένα τις αντίστοιχες, σύμφωνα με αυτά που διαβάζει, ιδιότητες.

2) Lesson.java

Ιδιότητες:

- a) private String **subjectID**;
- b) private String **name**;
- c) private String **classLevels**;
- d) private int[] **weeklyHours**;

Getters:

getSubjectID(), getName(), getClassLevels(), int[] getWeeklyHours()

Μέθοδοι:

static List<Lesson> parseLessonsFile(String filename):

δημιουργεί την λίστα lessons για να αποθηκεύουμε τα αντικείμενα είδος Lesson.

Δημιουργεί την λίστα weeklyHours, όπου αποθηκεύουμε σε κάθε θέση τις συνολικές ώρες του κάθε τμήματος χωριστά (3 τμήματα ισοδυναμούν σε 3 θέσεις)

3) TimetableSlot.java

Ιδιότητες:

- a) private Lesson **lesson**;
- b) private Teacher **teacher**;
- c) private String **section**;
- d) private int **day**;
- e) private int **hour**;

Getters:

getLesson(), getTeacher(), getSection(), getDay(), getHour()

4) Timetable.java

Ιδιότητες:

- a) private TimetableSlot[][] **timetable**;
- b) private static final int **SECTIONS = 9**;
- c) private static final int **DAYS = 5**;
- d) private static final int **HOURS = 7**;
- e) private static final int **TIME_SLOTS = DAYS * HOURS**;

	Monday				Tuesday	***	Friday		
(9X35)	Hour 1	Hour 2	***	Hour 7	Hour1	***	Hour1	***	Hour7
A1	[0,0]	[0,1]	***	[0,6]	[0,7]	***	[0,28]	***	[0,34]
A2	[1,0]	[1,1]			***		***		***
***	***	***			***		***		***
***	***	***			***		***		***
***	***	***			***		***		***
C3	[9,0]	[9,1]		[9,6]	[9,7]		[9,28]		[9,34]

Setters:

setSlot

Getters:

getSlot, getDayName, getSectionName

Μέθοδοι:

exportTimetable(String filename):

Γράφει το πρόγραμμα διδασκαλίας σε αρχείο κειμένου, περιλαμβάνοντας το κάθε τμήμα, την κάθε ημέρα και την κάθε ώρα συνοδευόμενη με τον διδάσκοντα
πχ:

ΗΜΕΡΑ: ΠΑΡΑΣΚΕΥΗ

Η ΩΡΑ 08:00-09:00: Ιστορία ΔΙΔΑΣΚΕΤΑΙ ΑΠΟ: ΑΘΗΝΑ ΓΕΩΡΓΑΚΗ

Η ΩΡΑ 09:00-10:00: Αγγλικά ΔΙΔΑΣΚΕΤΑΙ ΑΠΟ: ΑΧΙΛΛΕΑΣ ΣΤΕΦΑΝΙΔΗΣ

Η ΩΡΑ 10:00-11:00: Πληροφορική ΔΙΔΑΣΚΕΤΑΙ ΑΠΟ: ΕΛΕΝΗ ΠΑΝΑΓΟΥΛΗ

Η ΩΡΑ 11:00-12:00: Γλωσσική Διδασκαλία ΔΙΔΑΣΚΕΤΑΙ ΑΠΟ: ΝΕΟΠΤΟΛΕΜΟΣ
ΓΙΑΝΝΑΚΟΠΟΥΛΟΣ

Η ΩΡΑ 12:00-13:00: Γλωσσική Διδασκαλία ΔΙΔΑΣΚΕΤΑΙ ΑΠΟ: ΕΛΕΝΗ ΚΩΝΣΤΑΝΤΙΝΟΥ

Η ΩΡΑ 13:00-14:00: Μαθηματικά ΔΙΔΑΣΚΕΤΑΙ ΑΠΟ: ΠΛΟΥΤΑΡΧΟΣ ΣΑΜΑΡΑΣ

ΩΡΑ 7: Free

5) Chromosome.java

Ιδιότητες:

- a) private TimetableSlot[][] **timetable;**
- b) private double **fitness;**
- c) private static final int **SECTIONS = 9;**
- d) private static final int **DAYS = 5;**
- e) private static final int **HOURS = 7;**
- f) private static final int **TIME_SLOTS = DAYS * HOURS;**

Getters:

getTimetable

getFitness

Μέθοδοι:

calculateTotalWeeklyHours(List<Lesson> lessons, String sectionName)

για κάθε τμήμα, διασχίζει όλα τα μαθήματα μέσα από την λίστα lessons και επιστρέφει τις συνολικές ώρες διδασκαλίας

getSlot(int section, int day, int hour):

Κάνει μετατροπή το συγκεκριμένο κελί του προγράμματος που δίνεται σαν είσοδο στον αντίστοιχο δείκτη του πίνακα timetable

getClassLevelIndex(String sectionName): επιστρέφει ανάλογα με το sectionName το τμήμα

getValidLessonsForSection(List<Lesson> lessons, String sectionName):

Αυτή η συνάρτηση φιλτράρει τα μαθήματα που διδάσκονται στο συγκεκριμένο επίπεδο τάξης (Α, Β, ή Γ). Ελέγχει αν το μάθημα έχει ώρες για την τάξη που καθορίζεται και, αν ναι, το προσθέτει στη λίστα των έγκυρων μαθημάτων.

distributeHours(int totalHours):

Κατανέμει τις ώρες διδασκαλίας ισομερώς στις ημέρες του προγράμματος, εξασφαλίζοντας ότι δεν ξεπερνάται ο συνολικός αριθμός ωρών ανά ημέρα. Ανακατεύει τις ημέρες και κατανέμει τυχαία τις ώρες.

randomInitialize(List<Lesson> lessons, List<Teacher> teachers):

Η μέθοδος randomInitialize στην κλάση Chromosome γεμίζει ένα πρόγραμμα με μαθήματα και δασκάλους σε όλα τα τμήματα και τα χρονικά διαστήματα. Ξεκινάει αναγνωρίζοντας τα κατάλληλα μαθήματα για κάθε τμήμα με βάση τις απαιτήσεις του τμήματος, υπολογίζει πόσες ώρες πρέπει να προγραμματιστεί κάθε μάθημα και διανέμει αυτές τις ώρες καθ' όλη τη διάρκεια της εβδομάδας. Στη συνέχεια, τα μαθήματα ανατίθενται τυχαία σε χρονοθυρίδες μαζί με κατάλληλους δασκάλους, φροντίζοντας να τηρούνται οι περιορισμοί προγραμματισμού, ενώ οποιαδήποτε αχρονοθυρίδα παραμένει απρογραμματιστή σημειώνεται ως ελεύθερη περίοδος. Αυτή η μέθοδος εξασφαλίζει ότι η αρχική διαμόρφωση του προγράμματος κάθε χρωμοσώματος έχει μια τυχαία αλλά έγκυρη διαμόρφωση μαθημάτων και δασκάλων.

crossover(Chromosome other, List<Lesson> lessons, List<Teacher> teachers):

Εκτελεί crossover ανάμεσα στα δύο χρωμοσώματα, δημιουργώντας δύο παιδιά με βάση το τυχαίο σημείο διασταύρωσης, αντιγράφοντας τα δεδομένα του πρώτου γονέα μέχρι εκείνο το σημείο και του δεύτερου από το σημείο και μετά

calculateFitness()

υπολογίζει τη fitness ενός προγράμματος, ελέγχοντας αν πληρούνται οι περιορισμοί των καθηγητών, όπως οι συνεχόμενες ώρες διδασκαλίας και τα μαθήματα που μπορούν να διδάξουν. Στη συνέχεια, υπολογίζει την αναλογία ικανοποιημένων περιορισμών προς τον συνολικό αριθμό περιορισμών και αποδίδει μια βαθμολογία μεταξύ 0.0 και 1.0.

6) GeneticAlgorithm.java

Ιδιότητες:

- a) private List<Chromosome> **population**;
- b) private int **populationSize**;
- c) private double **mutationProbability**;
- d) private int **maxGenerations**;
- e) private double **fitnessThreshold**;

Μεθόδους:

initializePopulation(List<Lesson> lessons, List<Teacher> teachers):

Δημιουργεί έναν population από αντικείμενα τύπου Chromosome και σύμφωνα με το fitness του καθενός το προσθέτει στην λίστα πληθυσμού σε φθίνουσα σειρά, έτσι ώστε τα πιο βέλτιστα χρωμοσώματα να είναι στην αρχή της λίστας.

run(List<Lesson> lessons, List<Teacher> teachers):

εκτελεί τη διαδικασία του γενετικού αλγορίθμου για έναν καθορισμένο μέγιστο αριθμό γενεών ή μέχρι να φτάσει το στόχο που να καλύπτει τα ζητούμενα. Σε κάθε γενιά, επιλέγονται γονείς από την κορυφή του πληθυσμού, δημιουργούνται απόγονοι μέσω crossover και ενδέχεται να υπάρξει μετάλλαξη

mutateWithValidation(Chromosome chromosome, List<Lesson> lessons, List<Teacher> teachers): εκτελεί μια μετάλλαξη σε ένα χρωμόσωμα, επιλέγοντας τυχαία ένα τμήμα και μία θέση, μετά επιλέγεται τυχαία ένα μάθημα και ένας καθηγητής που έχει το δικαίωμα να το διδάξει. Αν ο καθηγητής μπορεί να διδάξει το μάθημα, η θέση στο πρόγραμμα ενημερώνεται με το μάθημα και τον καθηγητή. Στη συνέχεια, ενημερώνεται η καταλληλότητα του χρωμοσώματος.

7) Main.java

Η κλάση Main λειτουργεί ως το σημείο εισόδου για ένα σύστημα προγραμματισμού ωρολογίων προγραμμάτων που χρησιμοποιεί έναν γενετικό αλγόριθμο για να βελτιστοποιήσει την ανάθεση μαθημάτων και δασκάλων σε διάφορα τμήματα και χρονικά διαστήματα..

Ανάλυση Δεδομένων

- Μαθήματα και Δάσκαλοι:
 - Αναλύει δεδομένα μαθημάτων από το αρχείο "**lessons.txt**" και δεδομένα δασκάλων από το αρχείο "**teachers.txt**".
 - Χρησιμοποιεί προσαρμοσμένες μεθόδους ανάλυσης από τις κλάσεις Lesson και Teacher για να φορτώσει αυτά τα δεδομένα σε λίστες.

Αρχικοποίηση Γενετικού Αλγορίθμου

Παράμετροι Αλγορίθμου:

- **populationSize: 500**
- **mutationProbability: 0.05**
- **maxGenerations: 10,000,000**
- **fitnessThreshold: 0.95**

Algorithm Object:

- Δημιουργεί ένα αντικείμενο της κλάσης GeneticAlgorithm με τις παραπάνω παραμέτρους για να διαχειριστεί την εξελικτική διαδικασία.

Αρχικοποίηση και Εκτέλεση του Population

- **Αρχικοποίηση:** Δημιουργεί τον αρχικό πληθυσμό των προγραμμάτων (χρωμοσώματα) χρησιμοποιώντας τα δεδομένα των μαθημάτων και των δασκάλων που παρέχονται.
- **Εκτέλεση:** Εκτελεί τον γενετικό αλγόριθμο για να βελτιστοποιήσει τα προγράμματα και επιλέγει το πρόγραμμα με την καλύτερη απόδοση βάσει προκαθορισμένων κριτηρίων απόδοσης.

Εξαγωγή Προγράμματος

- **Κατασκευή του Τελικού Προγράμματος:**
 - Δημιουργεί ένα αντικείμενο Timetable από τα δεδομένα του προγράμματος του καλύτερου χρωμοσώματος.
 - Επαναλαμβάνει σε όλα τα τμήματα και τα χρονικά διαστήματα, τοποθετώντας μαθήματα και δασκάλους στο πρόγραμμα βάσει της βέλτιστης διαμόρφωσης του χρωμοσώματος.
- **Εξαγωγή σε Αρχείο:** Εξάγει το τελικό, βελτιστοποιημένο πρόγραμμα στο "final.txt", παρέχοντας ένα λεπτομερές πρόγραμμα που μπορεί να εξεταστεί και να χρησιμοποιηθεί έξω από το σύστημα.