

Αρχικά κατασκευάζεται μια κλάση Lesson, η οποία περιέχει τα χαρακτηριστικά του μαθήματος: κωδικό, όνομα, τάξη που διδάσκεται το μάθημα και αριθμός ωρών την εβδομάδα. Πέρα από τις παραπάνω, υπάρχει και μία μεταβλητή active, η οποία χρησιμεύει στην ομοιόμορφη κατανομή των μαθημάτων. Επίσης υπάρχει μία μέθοδος reduce που μειώνει κατά ένα τις διαθέσιμες ώρες διδασκαλίας (availablehours) που διδάσκεται το μάθημα.

Μετά κατασκευάζεται η κλάση Teacher που περιέχει τα χαρακτηριστικά ενός καθηγητή: κωδικός, όνομα, μέγιστος αριθμός ωρών διδασκαλίας την εβδομάδα, έναν πίνακα τύπου int days 5 θέσεων όπου σε κάθε θέση αποθηκεύεται ο μέγιστος αριθμός ωρών διδασκαλίας την ημέρα, κι ένα ArrayList με τα μαθήματα που διδάσκει. Περιέχει και μία μέθοδο reduce που παίρνει ως όρισμα έναν ακέραιο, ο οποίος δηλώνει για ποιο κελί του πίνακα days θα μειωθεί η τιμή του κατά 1 και μειώνεται και ο μέγιστος αριθμός ωρών ανά εβδομάδα.

Μετά κατασκευάζεται η κλάση State όπου εκεί έχουμε δύο τρισδιάστατους πίνακες 5x7x9 (5 μέρες, 7 ώρες, 9 τμήματα) όπου εκεί γράφονται τα μαθήματα και οι καθηγητές που διδάσκονται και διδάσκουν αντίστοιχα κάποια μέρα και ώρα σε κάποιο τμήμα με την μέθοδο write και σβήνονται με την μέθοδο delete. Χρησιμοποιούνται οι διαστάσεις x, y, z που αναφέρονται στις διαστάσεις των πινάκων αντίστοιχα, κι αρχικά έχουν μηδενικές τιμές. Υπάρχει και μία μέθοδος isAtTheSameTime, η οποία ελέγχει αν ένας καθηγητής κάνει μάθημα κάποια συγκεκριμένη μέρα και ώρα σε περισσότερα από ένα τμήματα κι αν η μέθοδος είναι ψευδής, τότε γράφεται το μάθημα. Ακόμη υπάρχουν οι ArrayList<Teacher> teachers και ArrayList<Lesson> lessons, όπου αποθηκεύονται οι καθηγητές και τα μαθήματα αντίστοιχα. Εκτός από τα παραπάνω, υπάρχουν και μέθοδοι που

ικανοποιούν τους περιορισμούς στο πρόγραμμα που είναι διατυπωμένοι στην εκφώνηση (isEvenlySpreadTeachers, evenlySpreadDays, isTiredTeacher), εκτός του περιορισμού να μην υπάρχουν κενά και του περιορισμού με την κατανομή μαθημάτων. Ο πρώτος δεν είναι σε μέθοδο επειδή ο αρχικός σκοπός είναι να μην μένουν κενά κελιά ξεκινώντας από την Δευτέρα στο Α1 στην 1η ώρα, στη 2η κ.λπ. και στο τέλος της ημέρας γίνεται αλλαγή στην επόμενη μέρα, και στο τέλος των ημερών γίνεται αλλαγή στο επόμενο τμήμα. Ο δεύτερος δεν είναι σε μέθοδο επειδή υπάρχει η μεταβλητή active που παίρνει την τιμή false όταν γραφτεί το μάθημα που θέλουμε. Επιπλέον, ορίζεται η προτεραιότητα αναλόγως με το πόσοι περιορισμοί ικανοποιούνται (μέθοδος priority). Στην write μαπίνουν δύο ορίσματα: το μάθημα και μία boolean μεταβλητή. Αν η μεταβλητή είναι true, τότε στο μάθημα που θα γραφτεί γίνονται οι κατάλληλες μεταρρυθμίσεις και στον καθηγητή που το διδάσκει. Χρησιμοποιείται απλή ευρετική για να βρεθεί ο καλύτερος απόγονος, ο οποίος είναι αυτός που θα ικανοποιεί τους περισσότερους περιορισμούς.

Τέλος, κατασκευάζεται η mainProgram (η main της εργασίας), που αρχικά αποθηκεύει τα μαθήματα και τους καθηγητές από τα αρχεία Lessons.txt και Teachers.txt αντίστοιχα, τα οποία έχουν την εξής μορφή:

```
CODE 125
LESSON_NAME ALGEBRAA
CLASS A
NUMBER_OF_HOURS 2
```

```
CODE 123
NAME PAPADOPOULOS
```

MAX_DAY 7

MAX_WEEK 35

LESSONS ALGEBRAA,ALGEBRAB,ALGEBRAC

Τα αρχεία Lessons.txt και Teachers.txt πάνε αντίστοιχα στα args[1] και args[2]. Κάθε μάθημα ανάλογα με την τάξη που διδάσκεται τελειώνει σε A,B ή C. Μετά την αποθήκευση των καθηγητών και των μαθημάτων, με την βοήθεια των παραπάνω κλάσεων και μεθόδων, δημιουργείται το πρόγραμμα των μαθημάτων σε ένα αρχείο Schedules.txt.

Οι κώδικες είναι γραμμένοι σε Java.