

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή στα Πληροφοριακά Συστήματα



Τυπική Αναπαράσταση Νομοθεσίας
Θεόδωρος Παππάς

Επιβλέπων Καθηγητής
Πάυλος Πέππας

Μάιος 2012

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Τυπική Αναπαράσταση Νομοθεσίας
Θεόδωρος Παππάς

Επιβλέπων Καθηγητής
Πάυλος Πέππας

Μάιος 2012

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών
στα Πληροφοριακά Συστήματα

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Μάιος 2012

Περίληψη

Στην παρούσα εργασία γίνεται μια προσπάθεια να απαντηθούν μερικά ερωτήματα που άπτονται του πεδίου της «Τεχνητής Ευφυΐας» και συγκεκριμένα του θέματος “Law reasoning” δηλαδή της προσπάθειας αναπαράστασης σε μορφή τυπικής Γνώσης σε υπολογιστικό περιβάλλον νομοθετικών πλαισίων με στόχο την παραγωγή «νομικής» γνώσης. Ο στόχος γενικά της εργασίας είναι πέρα από την υλοποίηση με τυπική αναπαράστασης γνώσης ενός «δείγματος» νομοθεσίας η εξέταση εάν μπορεί το σύστημα να αποτελέσει εργαλείο επιδιόρθωσης του εαυτού του μέσω αναδιοργάνωσης γνώσης στην περίπτωση όπου το περιβάλλον δημιουργεί γεγονότα που τείνουν να καταργήσουν την αξιοπιστία γνώσης που παρέχεται με την εμφάνιση φαινομένων πλεονασμού ή συγκρούσεων. Το πλαίσιο νομοθεσίας που χρησιμοποιήθηκε αφορά τον χαρακτηρισμό επάρκειας φοίτησης σε σχολεία δευτεροβάθμιας εκπαίδευσης. Τα ερωτήματα στα οποία έγινε προσπάθεια να απαντηθούν είναι:

- i. Μπορεί ένα εργαλείο Legal Knowledge Based Systems να βοηθήσει ένα νομοθέτη στην προσπάθειά του να βελτιώσει ένα νόμο.

Όσον αφορά το συγκεκριμένο παράδειγμα έχουν οριστεί πάνω στο Σύστημα Γνώσης κάποια ποιοτικά κριτήρια τα οποία μπορούν να χρησιμοποιηθούν από το ίδιο το Σύστημα Γνώσης και να βοηθήσουν στην βελτίωση και την απλοποίησή του. Στο συγκεκριμένο παράδειγμα πέρα από την ορθότητα και αξιοπιστία της γνώσης εκτιμάται και το αποκαλούμενο φαινόμενο «πληθωριστικής» ή «πλεονάζουσας» γνώσης το οποίο ερμηνεύεται ως η κατάσταση κατά την οποία ένας Μαθητής χαρακτηρίζεται επαρκής λόγω πολλών διαφορετικών συνθηκών – νόμων.

- ii. Μπορεί ένα εργαλείο Legal Knowledge Based System να βοηθήσει ένα οργανισμό στην προσπάθειά του να εφαρμόσει ένα νόμο.

Το συγκεκριμένο εργαλείο που έχω αναπτύξει υλοποιεί το νομικό πλαίσιο που αφορά την φοίτηση μαθητών έχοντας την δυνατότητα να παράγει γνώση όσον αφορά τον χαρακτηρισμό φοίτησης χρησιμοποιώντας γεγονότα που αφορούν βαθμό, διαγωγή, απουσίες μαθητών καθώς και των υφισταμένων συνθηκών επάρκειας.

- iii. Πόσο εφικτό είναι να σχεδιαστεί ένα σύστημα Legal Knowledge Based στο οποίο να μπορεί να μεταβάλλεται η «γνώση» που αφορά του νόμους χωρίς να επηρεάζεται το σύστημα εξαγωγής «γνώσης» Inference System.

Στο συγκεκριμένο σύστημα υπάρχει η δυνατότητα αλλαγής της αναπαράστασης των νόμων που διέπουν τον χαρακτηρισμό φοίτησης χωρίς να επηρεάζεται το σύστημα εξόρυξης γνώσης χωρίς δηλαδή να χρειαστεί να γίνει τροποποίηση των κανόνων δηλαδή του «πυρήνα» του συστήματος Γνώσης που περιέχει τα “inference rules”.

- iv. Υπάρχουν όρια όσον αφορά την υλοποίηση τέτοιων συστημάτων.

Η πολυπλοκότητα και το μέγεθος της πληροφορίας αλλά κυρίως η προσπάθεια τυπική αναπαράστασης των νόμων που διέπουν το νομικό πλαίσιο σίγουρα θέτουν όρια.

- v. Μπορούν να υπάρξουν μετρήσιμα ποιοτικά κριτήρια τα όποια τα συστήματα θα αξιολογούν για να βελτιώνονται.

Στο συγκεκριμένο παράδειγμα προτείναμε πέρα από το κριτήριο της ορθότητας όσον αφορά τον χαρακτηρισμό επάρκειας φοίτησης αξιολογούμε το φαινόμενο εμφάνισης πλεονασμών όσον αφορά τις συνθήκες που χαρακτηρίζουν επαρκή ένα μαθητή

Summary

In this thesis we tried to answer some questions concerning artificial intelligence and law reasoning using formal language. Our goal beyond the simple implementation of a Knowledge based system using formal language upon a legislation schema was focused in the area of belief revision trying to figure out if a Knowledge Based Schema can “repair” itself from “anomalies” such as knowledge collision and redundancy. The law example upon which we made the research concerns legislation about characterization study in secondary education. More specifically we tried to answer the following questions:

- I. How could a knowledge based system be useful in law makers in order to improve legislation.

Defining some quality indicators which can be evaluated by the system in order to improve the Knowledge by removing collisions and redundancies.

- II. How could a knowledge based system be useful in law enforcement?

In our effort the system can derive all the necessary knowledge concerning characterization study with accuracy and reliability.

- III. How could a knowledge based system be always operational even though changes take place concerning Knowledge about law

In our effort we described the knowledge concerning laws about characterization study in such a way so that every change occurs does not affect the structure of the inference rules in the system.

- IV. Are there any limits concerning implementation of Knowledge base systems about law.

The complexity and the amount of information but mainly the task of formal representation of the law certainly set limits.

- V. There may be measurable quality criteria to evaluate systems in order to improve themselves.

In this example we proposed over the correctness criterion the effect of redundancy on sufficient conditions that characterize a student.

Ευχαριστίες

Θέλω να ευχαριστήσω τον υπεύθυνο καθηγητή μου Πάλο Παππά κυρίως για την έμπνευση που μου παρείχε και την οικογένειά μου για την πολύτιμη στήριξη.

Περιεχόμενα

| | |
|--------------------------------------------------------------------------------------------|-----------|
| 1.Τεχνητή Ευφυΐα Και Νομοθεσία | 10 |
| 1.1 Εισαγωγή..... | 10 |
| 1.2 Σκοπός Της Έρευνας..... | 11 |
| 1.3 Επισκόπηση..... | 11 |
| 1.4 Ιστορική Αναφορά..... | 11 |
| 1.5 Συστήματα Γνώσης για Αναπαράσταση Νόμων. | 13 |
| 1.5.1 Μεθοδολογία ανάπτυξης..... | 14 |
| 1.5.2 Μορφές Αναπαράστασης Γνώσης..... | 15 |
| 1.5.3 Μηχανισμοί Εξαγωγής Συμπερασμάτων..... | 18 |
| 1.5.4 Belief Revision (Αναθεώρηση Πεποιθήσεων)..... | 20 |
| 1.5.5 Προβλήματα που αναδεικνύονται από την αναπαράσταση νόμων με συστήματα γνώσης..... | 20 |
| 2. Περιγραφή συστήματος Γνώσης | 22 |
| 2.1 Εισαγωγή | 22 |
| 2.2 Το Νομοθέτημα Στο Οποίο Αναφέρεται. | 23 |
| 2.3 Ανάλυση..... | 26 |
| 2.4 Μεθοδολογία Σχεδίασης..... | 29 |

| | | |
|----------|------------------------------------------------|------------|
| 2.5 | Εργαλεία που χρησιμοποιήθηκαν | 54 |
| 2.6 | Περιγραφή Του Περιβάλλοντος Της Εφαρμογής..... | 56 |
| 2.6.1 | Εισαγωγή | 56 |
| 2.6.2 | Αρχικοποίηση του Συστήματος Γνώσης | 58 |
| 2.6.3 | Διαδικασίες Αναδιοργάνωσης Γνώσης..... | 58 |
| 2.6.4 | Μεταβολές Στη Γνώση | 68 |
| 2.6.5 | Βοηθητικές Λειτουργίες..... | 70 |
| 2.7 | Συνοψίζοντας..... | 72 |
| | Συμπεράσματα..... | 74 |
| | Προβληματισμοί..... | 75 |
| | Βιβλιογραφία..... | 78 |
| | Λέξεις Κλειδιά..... | 80 |
| A | Αποτύπωσης Της εφαρμογής..... | A-1 |
| A.1 | Γενικές Παρατηρήσεις..... | A-1 |
| A.2 | Περιγραφή Predicates..... | A-1 |
| A.3 | Στιγμιότυπα της Εφαρμογής..... | A-19 |
| A.4 | Πηγαία Μορφή Κώδικα Εφαρμογής..... | A-26 |
| A.5 | Ευρετήριο Αρχείων Εφαρμογής | A-26 |

Κεφάλαιο 1

Τεχνητή Ευφυΐα και Νομοθεσία

1.1 Εισαγωγή

Τα τελευταία χρόνια όλο και περισσότερο κρίνεται χρήσιμη η εφαρμογή των Συστημάτων Γνώσης (Knowledge Systems) [03] σε οργανισμούς που εμπλέκονται με σχεδίαση ή υλοποίηση ή εφαρμογή νόμων. Στόχος των προσπαθειών αποτελεί η βελτίωση της παροχής υπηρεσιών νομικής φύσεως καθώς ο όγκος της πληροφορίας που αφορά νόμους γίνεται ολοένα και μεγαλύτερος, Προβλήματα εμφανίζονται μεταξύ αντικρουόμενων πολλές φορές αποφάσεων ή και καταστάσεων όπου νόμοι πλεονάζουν ή πλέον δεν βρίσκουν πεδίο εφαρμογής. Βασικά ερωτήματα που ερευνώνται είναι τα εξής:

- Μπορούν οι μηχανές να “κατανοήσουν” την πληροφορία που υπάρχει διαθέσιμη όσον αφορά τους νόμους..
- Μπορούν να απαντήσουν σε ερωτήσεις του χρήστη σωστά και λογικά όσον αφορά αποφάσεις από εφαρμογή των νόμων.
- Μπορούν να βοηθήσουν οι μηχανές στην επίλυση διφορούμενων ή συγκρουόμενων ή πλεοναζόντων αποφάσεων με στόχο την βελτίωση και απλοποίηση των συστημάτων γνώσης.

1.2 Σκοπός Της Έρευνας

Ο σκοπός της παρούσας εργασίας είναι να θέσει κάποιους προβληματισμούς όσον αφορά τα συστήματα γνώσης νομικής φύσεως και την δυναμική που παρουσιάζουν όσον αφορά την αναθεώρηση γνώσης και των προβλημάτων που αυτή συνεπάγεται για παράδειγμα αλληλοσυγκρουόμενοι νόμοι, νόμοι επικαλυπτόμενοι νόμοι ασαφείς. Το φαινόμενο πληθωριστικών τάσεων στην νομοθεσία όπου πολλές φορές νόμοι που πλέον δεν χρησιμοποιούνται παραμένουν στο σύστημα είναι υπαρκτό..

Μπορεί η τεχνητή νοημοσύνη να βοηθήσει στην βελτίωση των υπάρχοντων συστημάτων γνώσης νομικής φύσεως κάνοντας τα συστήματα αυτά πιο απλά και πιο αξιόπιστα.

Στην παρούσα εργασία έγινε προσπάθεια ένα απλό νομοθέτημα να αναπαρασταθεί σε μια τυπική μορφή με την βοήθεια ηλεκτρονικού υπολογιστή και να οριστούν πάνω στο σύστημα ποιοτικοί μετρήσιμοι δείκτες οι οποίοι περιγράφουν την τρέχουσα κατάσταση του συστήματος όσον αφορά την ορθότητα και αξιοπιστία του. Οι συγκεκριμένοι δείκτες παρακολουθούνται από το σύστημα το οποίο μπορεί να «προτείνει» αναθεωρήσεις γνώσης για την βελτίωση του συστήματος αξιολογώντας τους συγκεκριμένους δείκτες.

1.3 Επισκόπηση

Στην επιστήμη των υπολογιστών το πεδίο της Τεχνητής ευφυΐας αποτελεί ένα ζωντανό «κομμάτι» της έρευνας. Η προσπάθεια αναπαράστασης της γνώσης που κατέχουν οι άνθρωποι με την βοήθεια των υπολογιστών και τυπικών μορφών αναπαράστασης γνώσης αποτελεί ένα «ανοικτό» πρόβλημα όσον αφορά τα οφέλη αλλά και την ρεαλιστική υλοποίησή της. Το πόσο εύκολα, και πιστά μπορεί να αναπαρασταθεί η γνώση είναι ένα ζητούμενο και πόσο μπορεί να αποτελέσει εφιαλτήριο παραγωγής «ευφυΐας» [01] απαραίτητης για επίλυση υπάρχοντων προβλημάτων είναι ένα άλλο.

1.4 Ιστορική Αναφορά

Η αναπαράσταση γνώσης με την βοήθεια ηλεκτρονικού υπολογιστή αποτέλεσε κατάκτηση πολλές φορές όταν η γνώση αφορούσε «μετρήσιμα» και «διακριτά» αντικείμενα τα οποία έχουν ορισθεί με σαφήνεια όπως για παράδειγμα η γνώση που αφορά ασθένειες και συμπτώματα που

οδήγησε στην ανάπτυξη έμπειρων συστημάτων ιατρικής διάγνωσης με αρκετές διαβαθμίσεις στα αποτελέσματά τους.

Η prolog αποτελεί μια από τις δηλωτικές γλώσσες αναπαράστασης γνώσης η οποία αναπτύχθηκε το 1972 και γνώρισε αρκετή υποστήριξη μετά το 1980 με την εμφάνιση νέων υπολογιστών συστημάτων αυξημένης υπολογιστής ισχύος. Η δυνατότητα που προσφέρει για παραγωγή συμπερασμάτων από πραγματικές αλλά και υποθετικές προτάσεις της έδωσε μεγάλη αξία.

Στον χώρο της αναπαράστασης γνώσης νομικής φύσεως έχουν γίνει αρκετές προσπάθειες [07, 08]. Τα συστήματα που προέκυψαν ταξινομούνται στις εξής κατηγορίες:

- Cased-based Legal Reasoning Models

Σε αυτά τα συστήματα η γνώση που αναπαρίσταται αφορά περιπτώσεις (Cases) δηλαδή υποθέσεις νομικής φύσεως για τις οποίες περιγράφονται γεγονότα καθώς και αποφάσεις δικαστηρίων σχετικές. Στα συγκεκριμένα συστήματα η παραγωγή συμπερασμάτων προκαλείται για μια καινούρια υπόθεση από το «ταίριασμα» με μια ήδη υπάρχουσα. Η πρώτη προσπάθεια ονομάστηκε HYPO και αποτέλεσε ένα πρόγραμμα σχεδιασμένο από τους Rissland and Ashley (1987, 1990) για ανάλυση υποθέσεων που αφορούν εμπορικό δίκαιο.

- Logic-based Legal Reasoning Models.

Σε αυτά τα συστήματα η γνώση νομικής φύσεως αναπαρίσταται σε τυπική μορφή λογικής αποδείξεως όπου κυριαρχούν νομικοί κανόνες βάσει των οποίων δημιουργούνται λογικά συμπεράσματα.

- Legal-discourse Models

Σε αυτά τα συστήματα η γνώση που αναπαρίσταται αφορά τα δομικά και σημασιολογικά χαρακτηριστικά που δίνουν σε ένα επιχείρημα την νομιμότητα.

- Legal-ontology Based Models

Σε αυτά τα συστήματα η γνώση αναπαρίσταται ως οντολογία. Μια οντολογία αποτελείται από:

- ✓ Κλάσεις ή τάξεις οντοτήτων (class): έννοιες που σχετίζονται με ένα πεδίο ή κάποιες εργασίες, οι οποίες είναι συνήθως οργανωμένες σε κάποιο ταξινομικό σύστημα.
- ✓ Σχέσεις (relations): ένας τύπος αλληλεπίδρασης μεταξύ εννοιών ενός πεδίου.
- ✓ Συναρτήσεις (functions): είναι ειδικού τύπου σχέσεις.
- ✓ Αξιώματα (axioms): προτάσεις που είναι πάντα αληθείς.
- ✓ Στιγμιότυπα ή οντότητες (instances): χρησιμοποιούνται για να δηλώσουν συγκεκριμένα στοιχεία της βάσης γνώσης (γνωστικά αντικείμενα).

Εν γένει, οι οντολογίες δεν διαφέρουν πολύ από τα σύνολα που συναντάμε στον αντικειμενοστραφή σχεδιασμό ή τα μοντέλα συσχετίσεων (Entity-Relationship, E-R) των βάσεων δεδομένων.

Γενικά θα παρατηρήσουμε ότι λόγω του «μεγέθους» της πληροφορίας αλλά και της πολυπλοκότητας [08] αλλά και της υψηλής μεταβλητότητας που παρουσιάζουν τα συστήματα αναπαράστασης νόμων είναι αρκετά δύσκολο να αναπαρασταθούν σε τυπική μορφή. Το πρόβλημα είναι να βρεθεί κάποια σημαντική ανταποδοτικότητα σε αυτή την προσπάθεια έτσι ώστε το σύστημα αναπαράστασης νομοθεσίας με έμπειρο σύστημα να αποτελέσει εργαλείο. το οποίο θα βοηθήσει τον νομοθέτη, τον δικηγόρο τον δικαστικό και γενικά όσους εμπλέκονται σε θέματα νομοθεσίας [07].

1.5 Συστήματα Γνώσης για Αναπαράσταση Νόμων

Ένα Σύστημα Γνώσης [05, 14] που περιγράφει ένα νομικό πλαίσιο, πρέπει να περιλαμβάνει τα εξής:

- Κατάλληλη κωδικοποίηση – αναπαράσταση των νόμων
- Μέθοδοι επίλυσης προβλημάτων που αφορούν ερμηνεία των νόμων

1.5.1 Μεθοδολογία Ανάπτυξης

Τα βασικά στάδια της ανάπτυξης ενός Συστήματος Γνώσης είναι:

I. Ανάλυση του προβλήματος – Απόκτηση γνώσης

Η ανάλυση αποβλέπει στον προσδιορισμό της επιθυμητής λύσης του προβλήματος. Στην φάση της απόκτησης γνώσης γίνεται η μεταφορά και μετατροπή της γνώσης από μορφές στις οποίες είναι διαθέσιμη σε μορφές οι οποίες μπορούν να αναπαρασταθούν από ένα Σύστημα Γνώσης. Συνήθως η «πηγαία» μορφή είναι ελεύθερο κείμενο πάνω στο οποίο θα αναζητηθούν κάποιες φόρμες – έννοιες.

II. Σχεδίαση.

Προσδιορίζεται η μορφή [13, 06] της αναπαράστασης της γνώσης, η συλλογιστική, καθώς και το εργαλείο ανάπτυξης του Συστήματος Γνώσης. Η γνώση ουσιαστικά αποτελείται από δυο συνιστώσες:

- Την ρητώς αποθηκευμένη γνώση αυτή δηλαδή που προκύπτει άμεσα από την αφηρημένη περιγραφή των νόμων και την
- Συμπερασματική γνώση αυτή δηλαδή που παράγεται από κανόνες συμπερασμού.

III. Υλοποίηση.

Κωδικοποιείται το μοντέλο της γνώσης χρησιμοποιώντας εργαλεία ανάπτυξης. Τα εργαλεία μπορεί να είναι δηλωτικές ή αλγοριθμικές γλώσσες προγραμματισμού ή έτοιμα εργαλεία αναπαράστασης γνώσης.

IV. Έλεγχος αξιοπιστίας – ορθότητας

Γίνεται έλεγχος της συμβατότητας του συστήματος με τις αρχικές προδιαγραφές καθώς και της πληρότητας και αξιοπιστίας στην γνώση που παρέχει.

1.5.2 Μορφές Αναπαράστασης Γνώσης

Οι επικρατέστερες μορφές αναπαράστασης γνώσης [10, 12] ταξινομούνται στις εξής:

I. Αναπαράσταση γνώσης με λογική.

Η γνώση περιγράφεται ως δηλώσεις (ή προτάσεις) και υποθέσεις (premises), από τις οποίες παράγονται άλλες δηλώσεις που ονομάζονται συμπεράσματα (conclusions). Η αναπαράσταση γνώσης με την λογική χωρίζεται στις εξής κατηγορίες::

a. Προτασιακή λογική

Η Προτασιακή Λογική (ΠΛ) είναι ένα πολύ στοιχειώδες μέσο αναπαράστασης της γνώσης που βασίζεται σε αυστηρούς ορισμούς. Όσον αφορά το συντακτικό ασχολείται με προτάσεις που κάνουν κάποιες διαπιστώσεις. Οι πιο απλές προτάσεις ονομάζονται ατομικές προτάσεις ή άτομα (atomic propositions, atoms). Οι προτάσεις αυτές συνδεόμενες με τους λογικούς συνδέσμους (logical connectives) συνθέτουν τις σύνθετες προτάσεις (compound propositions). Οι λογικοί σύνδεσμοι είναι:

- άρνηση (not),
- λογικό και, σύζευξη (and),
- Λογικό ή, διάζευξη (or),
- \leftarrow και \rightarrow απλή συνεπαγωγή
- (implies), \leftrightarrow ισοδυναμία (equivalent).

b. Κατηγορηματική λογική Α' τάξης

Η κατηγορηματική λογική (ΚΛ) ή αλλιώς λογική πρώτης τάξης (FOL: First Order Logic) εκτείνει την προτασιακή λογική έτσι ώστε να είναι δυνατή η αναπαράσταση της γνώσης που αναφέρεται σε ένα σύνολο αντικειμένων και στις σχέσεις τους, εξετάζοντας τις προτάσεις για αυτά τα αντικείμενα ως προς την αλήθεια τους. Αποτελείται από:

- Όρος (term). Ορίζεται ως μια σταθερά ή μια μεταβλητή του συντακτικού.
- Άτομο-Κατηγορημα (atom-predicates). Αν p είναι ένα κατηγορηματικό σύμβολο n -θέσεων και t_1, \dots, t_n είναι όροι τότε το $p(t_1, \dots, t_n)$ είναι άτομο ή κατηγορημα.
- Ποσοδείκτες (quantifiers). Είναι δύο σύμβολα, το \exists (Exist, υπάρχει) υπαρξιακός ποσοδείκτης και το \forall (All, για κάθε) καθολικός ποσοδείκτης.

Τα πλεονεκτήματα της ΚΛ ως μοντέλο αναπαράστασης γνώσης είναι τα εξής: η αυστηρή και καλά ορισμένη μεθοδολογία, η ικανοποιητική έκφραση ποσοτικοποίησης των εννοιών με τους κατάλληλους ποσοδείκτες, η ακρίβεια και η συνέπεια των συλλογισμών.

c. Ασαφής λογική

Η ασαφής λογική η οποία περιλαμβάνει ποσοτικές τιμές αληθείας ορισμένες στο κλειστό σύνολο $[0, 1]$. Η ασαφής λογική μπορεί να θεωρηθεί γενίκευση της συμβατικής τυπικής λογικής, αφού οι συμβατικές τιμές αληθείας είναι οι ακραίες τιμές αληθείας της ασαφούς λογικής. Τα χαρακτηριστικά της ασαφούς λογικής είναι:

- Τα ασαφή σύνολα (fuzzy sets) με στοιχεία στα οποία αποδίδονται βαθμοί συμμετοχής (membership values) στο σύνολο. Αυτές οι τιμές απεικονίζουν χαρακτηριστικά μεγέθη οντοτήτων σε ποσοτικές τιμές αληθείας μιας σχετικής με αυτά ιδιότητας.

Η ασαφής λογική μπορεί να χρησιμοποιηθεί για την αναπαράσταση ποσοτικών χαρακτηριστικών γνώσης για την οποία η καθιερωμένη τυπική λογική είναι ανεπαρκής

II. Δομημένες αναπαραστάσεις γνώσης.

Η κλασική λογική δεν μπορεί να υποστηρίξει σύνθετες δομές με αποτέλεσμα τη δημιουργία μεγάλου όγκου γνώσης (συμβόλων και εκφράσεων) για την περιγραφή ενός προβλήματος. Αυτό αποτελεί το πλεονέκτημα της δομημένης αναπαράστασης γνώσης. Οι κυριότερες μορφές της είναι οι εξής:

a. Σημασιολογικά δίκτυα

Αποτελούνται από κόμβους (nodes) και δεσμούς (links) ανάμεσα τους. Οι κόμβοι αντιπροσωπεύουν κλάσεις αντικειμένων (class), αντικείμενα (objects), έννοιες (concepts) και τιμές ιδιοτήτων (values). Οι δεσμοί ορίζουν τις σχέσεις (relations) μεταξύ αντικειμένων ή τις ιδιότητες που συνδέουν αντικείμενα με τιμές. Η αναπαράσταση γνώσης με σημασιολογικά δίκτυα είναι αρκετά διαισθητική αφού η γνώση οπτικοποιείται. Αυτό επιτυγχάνεται με την ιεραρχία των κόμβων και την υποστήριξη της κληρονομικότητας. Τα κυριότερα πλεονεκτήματα τους είναι η δυνατότητα εξαγωγής συμπερασμάτων μέσω της ιεραρχικής δομής και ότι παρέχουν συμπαγή τρόπο αποθήκευσης γνώσης εξοικονομώντας αποθηκευτικό χώρο.

b. Πλαίσια

Κάθε πλαίσιο ορίζεται ως η δυάδα $\langle \text{Name}, \text{SV} \rangle$, όπου Name είναι το μοναδικό όνομα του πλαισίου και SV: ένα σύνολο από ιδιότητες (Slots) του πλαισίου και οι τιμές τους (Values). Τα πλαίσια μπορούν να οργανώνονται σε ιεραρχίες, δηλαδή να αποτελούν κόμβους σε σημασιολογικό δίκτυο, και να αξιοποιούν έτσι την κληρονομικότητα.

c. Εννοιολογικοί γράφοι

Ένας εννοιολογικός γράφος είναι διμερής, δηλαδή περιέχει δύο ειδών κόμβους, τις έννοιες (concepts) και τις σχέσεις (relations). Επίσης είναι κατευθυνόμενος, δηλαδή τα τόξα του συνδέουν έννοιες με σχέσεις, ή σχέσεις με έννοιες. Έτσι, η ανάγνωση ενός εννοιολογικού γράφου ακολουθεί πάντα την κατεύθυνση μιας τριάδας, έννοια-σχέση-έννοια.

III. Αναπαράσταση γνώσης με κανόνες.

Οι κανόνες χαρακτηρίζονται ως η πιο προσφιλή μέθοδος αναπαράστασης γνώσης, κυρίως γιατί αναπαριστούν τη γνώση με τρόπο που πλησιάζει την ανθρώπινη γνώση. Επιπλέον, η εξαγωγή συμπερασμάτων πραγματοποιείται εύκολα (επάρκεια συνεπαγωγών).

Τα κύρια πλεονεκτήματα των κανόνων είναι τα εξής:

- Κάθε κανόνας ορίζει ένα μικρό και (σχεδόν) ανεξάρτητο τμήμα της γνώσης για ένα πρόβλημα (modularity).

- Νέοι κανόνες μπορεί να προστεθούν σε ένα σύνολο κανόνων (σχεδόν) ανεξάρτητα από άλλους υπάρχοντες κανόνες (incrementability).
- Κανόνες που ήδη υπάρχουν σε ένα σύνολο κανόνων μπορεί να αλλάξουν (σχεδόν) ανεξάρτητα από τους άλλους κανόνες (modifiability)

Ανάλογα με το είδος της γνώσης που αναπαριστούν οι κανόνες μπορούν να χωριστούν στις εξής κατηγορίες:

- Κανόνες παραγωγής (Production rules). Εκφράζουν τη γνώση που δηλώνει μια αλήθεια στον κόσμο του προβλήματος αλλά δεν αναφέρεται ρητά πότε και πως εφαρμόζεται (διαδικαστική γνώση). Η μορφή τους είναι "If συνθήκες Then ενέργειες",
- Συμπερασματικοί Κανόνες (Deductive rules). Εκφράζουν τη γνώση για το ποιες ενέργειες πρέπει να εκτελεστούν σε μια συγκεκριμένη κατάσταση (δηλωτική γνώση). Η μορφή τους είναι "If συνθήκες THEN συμπέρασμα", με σημασιολογική ερμηνεία: "αν οι συνθήκες αληθεύουν τότε αληθεύει το συμπέρασμα".
- Ενεργοί κανόνες (Active rules). Εκφράζουν και αυτοί διαδικαστική γνώση αλλά αναφέρονται με σαφήνεια το πότε ενεργοποιούνται. Η μορφή τους είναι "ON γεγονός If συνθήκες Then ενέργειες", με σημασιολογική ερμηνεία: "Όταν συμβεί το γεγονός Αν οι συνθήκες αληθεύουν τότε εκτέλεσε τις ενέργειες".

1.5.3 Μηχανισμοί Εξαγωγής Συμπερασμάτων.

Ένας μηχανισμός εξαγωγής συμπερασμάτων [01, 02, 04, 09, 11] αφορά τις διαδικασίες προσδιορισμού του αποτελέσματος που απορρέει από μια πληροφορία που έχει προστεθεί στη βάση γνώσης. Ουσιαστικά, είναι μηχανισμός που παράγει νέα γνώση από την ήδη υπάρχουσα.

Κάθε μέθοδος αναπαράστασης της γνώσης έχει τους δικούς της μηχανισμούς εξαγωγής συμπερασμάτων. Οι πιο γνωστές στρατηγικές αναζήτησης συμπερασμάτων είναι:

- I. Αναζήτηση οδηγούμενη από στόχους (goal driven ή top down).

Η διαδικασία ξεκινά από τα πιθανά συμπεράσματα και καταλήγει στα αίτια που τα στηρίζουν.

II. Οδηγούμενη από τα δεδομένα (data driven ή bottom-up).

Η διαδικασία ξεκινά από τα δεδομένα του προβλήματος και καταλήγει στο συμπέρασμα.

Οι βασικοί μηχανισμοί εξαγωγής συμπερασμάτων στην αναπαράσταση γνώσης με λογική είναι οι εξής:

- Πίνακες αληθείας (Truth Table).
- Λογική Απόδειξη (proof).

Στις δομημένες μορφές αναπαράστασης γνώσης μπορεί να χρησιμοποιηθεί οποιοσδήποτε μηχανισμός συμπερασμάτων της λογικής.

Προαπαιτούμενο για την εξαγωγή συμπερασμάτων με πλαίσια είναι η ανάκτηση της τιμής μιας ιδιότητας. Αυτό γίνεται με την κατάλληλη επιλογή ενός αλγορίθμου αναζήτησης η οποία είναι ιδιαίτερα κρίσιμη λόγω της πολλαπλής κληρονομικότητας που υποστηρίζουν τα πλαίσια.

Οι μηχανισμοί εξαγωγής συμπερασμάτων ενός ορθά δομημένου εννοιολογικού γράφου γίνονται μέσω κατάλληλων μετασχηματισμών. Οι μετασχηματισμοί αυτοί ορίζονται λαμβάνοντας υπόψη ότι κάθε εννοιολογικός γράφος έχει ισοδύναμη λογική αναπαράσταση.

Ένας μηχανισμός εξαγωγής συμπερασμάτων στην αναπαράσταση γνώσης με κανόνες, χρησιμοποιεί τη συνεπαγωγική συλλογιστική (deductive reasoning), δηλαδή έναν αλγόριθμο που υλοποιείται από την ακολουθία εκτέλεσης των κανόνων συνδυάζοντας τα δεδομένα, τους κανόνες και τα συμπεράσματα. Τα δύο είδη συνεπαγωγικής συλλογιστικής είναι:

- η ορθή ακολουθία εκτέλεσης “από αριστερά προς τα δεξιά”
- και η ανάστροφη ακολουθία εκτέλεσης “από δεξιά προς τα αριστερά”.

1.5.4 Belief Revision (Αναθεώρηση Πεποιθήσεων)

Η ανάγκη για ισχυρότερους τρόπους αναπαράστασης πληροφορίας οδήγησε στην ανάπτυξη των λογικών βάσεων γνώσεων, δηλαδή βάσεων στις οποίες η γνώση εκφράζεται με την χρήση λογικών προτάσεων και κανόνων. Τέτοιου είδους συστήματα παρέχουν μεγάλη ευκολία στην παραγωγή νέας γνώσης η οποία δεν είναι ρητώς καταχωρημένη στην Βάση [10].

Το πρόβλημα πλέον που παρουσιάζεται είναι ότι καινούρια γεγονότα που έρχονται να προστεθούν ως νέες πληροφορίες πολλές φορές έρχονται σε σύγκρουση με την ρητώς καταχωρημένη ή την παραγόμενη γνώση.

Η αναθεώρηση πεποιθήσεων έρχεται να προτείνει τρόπους προσαρμογής του υπάρχοντος συστήματος γνώσης στα νέα δεδομένα έτσι ώστε να παραμένει σε μια κατάσταση στην οποία να μην παράγεται αντιφατική ή μια ορθή πληροφορία σύμφωνα με τις ισχύουσες πεποιθήσεις.

1.5.5 Προβλήματα που αναδεικνύονται από την αναπαράσταση νόμων με συστήματα γνώσης.

Τα προβλήματα που μπορεί να συναντήσουμε στην προσπάθειά μας να αναπαραστήσουμε νόμους σε ένα Σύστημα Γνώσης ταξινομούνται στα εξής:

- Πληρότητα

Στην προσπάθεια αναπαράστασης και κωδικοποίησης τεράστιου όγκου δεδομένων πράγμα συνηθισμένο όσον αφορά νόμους η πληρότητα είναι ένα σημαντικό θέμα διότι η παράλειψη κάποιου κομματιού γνώσης οδηγεί σε μη ορθά συμπεράσματα.

- Οι ιδιομορφίες

Η γνώση σχετικά με τους νόμους παρουσιάζει υψηλή δυναμική δηλαδή μεταβάλλεται πολύ γρήγορα (καινούριοι νόμοι, καινούριες υποθέσεις κλπ.). Το σύστημα θα πρέπει να είναι συνεχώς ενημερωμένο και αν είναι δυνατό να προσπαθεί να απλοποιεί την δομή του ώστε να παραμένει λειτουργικό.

- Όγκος δεδομένων

Η τεράστια «μάζα» πληροφορίας απαιτεί «έξυπνες» τεχνικές αναπαράστασης και ισχυρά υπολογιστικά συστήματα.

- Απόκτηση και ενσωμάτωση της γνώσης

Είναι πολύ σημαντικό να περιγραφεί από ειδικούς η υπάρχουσα γνώση ώστε να γίνει η όσο το δυνατόν καλύτερη αποτύπωση της στο σύστημα. Λάθη ερμηνειών σε ένα νομικό πλαίσιο θα οδηγήσουν σε λάθη που θα πάρει το Σύστημα Γνώσης με μεγάλη δυσκολία στον εντοπισμό της αιτίας.

Κεφάλαιο 2

Περιγραφή συστήματος Γνώσης

2.1 Εισαγωγή

Για την συγκεκριμένη εργασία χρησιμοποιείται ως παράδειγμα το πλαίσιο νόμων που περιγράφει τον χαρακτηρισμό φοίτησης μαθητών δευτεροβάθμιας εκπαίδευσης. Το Σύστημα Γνώσης που αναπτύχθηκε υποστηρίζει αναδιοργάνωση της γνώσης ώστε να ανταποκρίνεται στο δυναμικό περιβάλλον.

Έχουν θεσπιστεί κάποια ποιοτικά κριτήρια τα οποία το σύστημα μέσω της αναδιοργάνωσης γνώσης πρέπει να παρακολουθεί και να βελτιστοποιεί θεωρώντας τα ως την «κρίσιμη» Γνώση η οποία αν δεν είναι πλέον ορθή το σύστημα δεν πρέπει να λειτουργεί[01]. Στο συγκεκριμένο παράδειγμα τα κριτήρια είναι :

- I. Για κάθε μαθητή που πληροί τις συνθήκες επάρκειας πρέπει το σύστημα να το προσφέρει ως συμπερασματική γνώση,
- II. Πρέπει να υπάρχουν όσο το δυνατόν λιγότερες περιπτώσεις στις οποίες μαθητής χαρακτηρίζεται επαρκής από πολλές συνθήκες. Αυτό αν συμβαίνει σημαίνει ότι γνώση πλεονάζει και πρέπει το σύστημα να γίνει πιο απλό. Το συγκεκριμένο κριτήριο θα το ονομάσουμε μέτρο πλεονάζουσας γνώσης και θα το ορίσουμε ως την απαρίθμηση των περιπτώσεων (Συνθήκη επάρκειας, μαθητής) για την οποία ο μαθητής χαρακτηρίζεται επαρκής και από άλλες συνθήκες.

2.2 Το Νομοθέτημα Στο Οποίο Αναφέρεται

Ακολουθεί μια αφηρημένη παράθεση του πλαισίου νόμων με το οποίο ασχολείται η συγκεκριμένη εφαρμογή.

Φοίτηση μαθητών

1.Χαρακτηρισμός φοίτησης

Την τελευταία ημέρα του Β τετραμήνου στα Λύκεια δεν διεξάγεται διδασκαλία μαθημάτων. Την ημέρα αυτή ο Σύλλογος των διδασκόντων κάθε σχολείου Δ.Ε., αποφασίζει για τον χαρακτηρισμό της φοίτησης των μαθητών και εκτελεί κάθε προπαρασκευαστική εργασία για την απρόσκοπτη διεξαγωγή των προβλεπομένων γραπτών εξετάσεων.

Η φοίτηση όλων των μαθητών των Γενικών Λυκείων χαρακτηρίζεται με πράξη του Συλλόγου των διδασκόντων καθηγητών την ημέρα που λήγουν τα μαθήματα ως επαρκής ή ανεπαρκής, με βάση το σύνολο των απουσιών που σημειώθηκαν κατά τη διάρκεια του διδακτικού έτους, σε συνδυασμό με την ετήσια προφορική επίδοση σε όλα τα μαθήματα.

Επαρκής χαρακτηρίζεται η φοίτηση στην περίπτωση κατά την οποία:

α) Το σύνολο των απουσιών του μαθητή, που σημειώθηκε κατά το λήγον διδακτικό έτος, δεν υπερβαίνει τις πενήντα (50), ανεξάρτητα από τους λόγους πραγματοποίησής τους.

β) Το σύνολο των απουσιών του μαθητή δεν υπερβαίνει τις εκατόν δεκατέσσερις (114), από τις οποίες οι πάνω από τις πενήντα (50) είναι δικαιολογημένες ή οφείλονται σε ασθένεια που βεβαιώνεται, «σύμφωνα με το άρθρο 3 του Π.Δ. 485/83 (ΦΕΚ184 Α) ενημέρωση γονέων-δικαιολόγηση απουσιών». Κατ' εξαίρεση, ο Σύλλογος των Διδασκόντων κάθε Σχολείου με ειδική πράξη του χαρακτηρίζει ως επαρκή τη φοίτηση μαθητή που σημείωσε από εκατόν δέκα πέντε έως εκατόν εξήντα τέσσερις (115-164) απουσίες, υπό την προϋπόθεση ότι όλες οι απουσίες πάνω από τις πενήντα (50) είναι δικαιολογημένες και οφείλονται σε ασθένεια, που βεβαιώνεται σύμφωνα με τα παραπάνω, και η επίδοσή του κρίνεται ιδιαίτερα ικανοποιητική. Ως ιδιαίτερα ικανοποιητική κρίνεται στην προκειμένη περίπτωση η επίδοση του μαθητή, όταν ο γενικός μέσος όρος της προφορικής βαθμολογίας (Γ.Μ.Ο.), όπως προκύπτει από το άθροισμα των Μ.Ο. των δύο τετραμηνιαίων βαθμών στο Λύκειο, διαιρούμενο με τον αριθμό των διδασκόμενων μαθημάτων, είναι τουλάχιστον δεκαπέντε (15) πλήρες και η διαγωγή του κοσμιωτάτη.

γ) Συντρέχουν οι λόγοι που αναφέρονται στην παρ. 6 εδ. δ του άρθρου 1 του Π.Δ. 294/1980: «Δι' αποφάσεως του ΥΠΕΠΘ δύναται να χαρακτηρίζεται ως επαρκής η φοίτησης μαθητών ανεξαρτήτως αριθμού απουσιών, εφόσον οι μαθηταί τυγχάνουν πρωταθληταί αι δε απουσίαι των οφείλονται εις την συμμετοχήν των εις διεθνείς αθλητικές εκδηλώσεις πιστοποιουμένας υπό της Γενικής Γραμματείας Αθλητισμού».

Οι μαθητές, των οποίων η φοίτηση χαρακτηρίζεται επαρκής, προσέρχονται κανονικά στις προαγωγικές και απολυτήριες εξετάσεις του Ιουνίου.

Ανεπαρκής χαρακτηρίζεται η φοίτηση μαθητή που σημείωσε πάνω από πενήντα (50) απουσίες και δεν εμπίπτει σε καμία από τις περιπτώσεις που προαναφέρθηκαν. Οι μαθητές, των οποίων η φοίτηση χαρακτηρίζεται ανεπαρκής, είναι υποχρεωμένοι να επαναλάβουν τη φοίτηση στην ίδια τάξη.

Περιπτώσεις απουσιών που ΔΕΝ ΛΑΜΒΑΝΟΝΤΑΙ υπόψη.

Όλες οι απουσίες γράφονται στο ημερήσιο δελτίο απουσιών. Κάποιες όμως δεν μεταφέρονται στο Βιβλίο Φοίτησης του Τμήματος και συνεπώς δεν λαμβάνονται υπόψη στον χαρακτηρισμό της φοίτησης του μαθητή.

Για τον χαρακτηρισμό της φοιτήσεως των μαθητών δεν λαμβάνονται υπόψη :

α. Απουσίες εκ του σχολείου κατά τις ώρες λειτουργίας αυτού μαθητών αποστελλομένων εις το εξωτερικό ή εσωτερικό για ειδική αποστολή δυνάμει εγγράφου εντολής του Υπουργού Εθνικής Παιδείας και Θρησκευμάτων ή του οικείου Νομάρχου.

β. Απουσίες μαθητών απασχολουμένων οπωσδήποτε σε εργασία ανατεθείσα σε αυτούς από το Διευθυντή ή με πράξη του Συλλόγου των Διδασκόντων.

γ. Απουσία μαθητών δια την προσέλευσή τους ενώπιον του Συμβουλίου Επιλογής Οπλιτών προσηκόντως αποδεδειγμένη.

δ. Απουσία λόγω συμμετοχής σε επίσημους αθλητικές εκδηλώσεις πιστοποιούμενη από της αρμοδίας Δημοσίας Αθλητικής Αρχής.

ε. Απουσία μαθητών: α) του Καθολικού Δόγματος κατά τις εορτές του Μνηστήρος Ιωσήφ, της Αγίας Δωρεάς και από της προτεραίας μέχρι και της επομένης του Λατινικού Πάσχα, β) του Εβραϊκού Θρησκευματος την προτεραία και την 1ην του Εβραϊκού Έτους, την ημέρα της εξελίξεως ως και την προτεραία και την ημέρα του Εβραϊκού Πάσχα, γ) του Μουσουλμανικού Θρησκευματος κατά τις ημέρες των εορτών Σεκέρ Μπαϊράμ και Κουρμπάν Μπαϊράμ και την προτεραία εκάστης τούτων.

στ. Απουσία των μαθητών λόγω μετεγγραφής σε Λύκειο ετέρας πόλεως επί τρεις ημέρας, μη περιλαμβανομένης εις αυτές της ημέρας εκδόσεως του προς αποχώρηση του μαθητού υπηρεσιακού σημειώματος.

ζ. Απουσία μαθητών απαλλασσομένων νομίμως εν μέρει ή εν συνόλω εκ της συμμετοχής αυτών εις την διδασκαλία της Φυσικής Αγωγής, των Θρησκευτικών και της Ξένης Γλώσσας.

η. Απουσίες μαθητών από το Σχολείο κατά τις ώρες λειτουργίας του οφειλόμενες σε ολιγόωρη ή ολιγοήμερη απεργία των υπεραστικών λεωφορείων των Κ.Τ.Ε.Λ που βεβαιώνεται κατά τόπους από τον οικείο Νομάρχη.

θ. Απουσίες μαθητών από το Σχολείο κατά τις ώρες λειτουργίας του, που οφείλονται στις προβλεπόμενες κυρώσεις μέχρι τριών (3) το πολύ ημερών κατά την κρίση του Συλλόγου των Διδασκόντων.

Οι αποβαλλόμενοι μαθητές δύναται να παραμένουν εις το σχολείο κατά τας ώρας διδασκαλίας των μαθημάτων απασχολούμενοι με μερίμνα του Διευθυντού του σχολείου.

Ο σύλλογος των διδασκόντων ενός σχολείου έχει την δυνατότητα να μη λάβει υπόψη του, κατά το χαρακτηρισμό και φοίτηση των μαθητών, τις απουσίες τριών (3) ημερών που οφείλονται σε πειθαρχικές κυρώσεις.

Όταν οι απουσίες αυτές ανήκουν σε περισσότερες από τρεις ημέρες τότε ο σύλλογος των καθηγητών έχει την δυνατότητα να μην υπολογίσει, από το σύνολο των απουσιών αποβολής,

όσες απουσίες πραγματοποιήθηκαν σε τρεις μόνο συγκεκριμένες ημέρες, τις οποίες ημέρες επιλέγει κατά την κρίση του.

ι. Απουσίες μαθητών μέχρι και τριών (3) το πολύ ημερών κατά μήνα, οφειλόμενες στην ανάγκη μεταγγίσεως αίματος σε ειδικό νοσηλευτικό ίδρυμα λόγω μεσογειακής ή δρεπανοκυτταρικής αναιμίας από την οποία πάσχουν. Οι απουσίες αυτές δεν μπορούν να υπερβούν αθροιστικά όλο το έτος τις απουσίες είκοσι τεσσάρων (24) ημερών.

ια. Απουσίες μαθητών οφειλόμενες σε θεραπεία αρχική ή συνεχιζόμενη σε χώρες του εξωτερικού. Στις περιπτώσεις αυτές δεν υπολογίζονται και οι απουσίες τριών (3) ημερών για τη μετάβαση και επιστροφή πέρα από τις βεβαιούμενες από τα σχετικά ιατρικά πιστοποιητικά.

ιβ. Σε όσους μαθητές μεταβούν στον τόπο καταγωγής τους για άσκηση του εκλογικού τους δικαιώματος, δεν θα υπολογιστούν για το χαρακτηρισμό της φοίτησης τους απουσίες μίας ή δύο ημερών κατά περίπτωση.

Επίσης, δεν θα υπολογιστούν για τον χαρακτηρισμό της φοίτησης οι απουσίες μιας ή δύο ημερών, κατά περίπτωση, των μαθητών που αποδεδειγμένα θα μετακινηθούν (οι ίδιοι ή οι γονείς τους) για να ψηφίσουν.

ιγ. Απουσίες μαθητών που σημειώνονται εξαιτίας της παρουσίας τους σε δικαστήριο, δεν θα λαμβάνονται υπόψη για τον χαρακτηρισμό της φοίτησης τους, προσκομίζοντας στο σχολείο σχετικό δικαιολογητικό

ιδ. Όταν τα σχολεία κλείνουν με απόφαση του νομάρχη λόγω κακοκαιρίας δεν σημειώνονται απουσίες.

ιε. Οι απουσίες των μαθητών σχολείων Δευτεροβάθμιας Εκπαίδευσης που συμμετέχουν στις προβλεπόμενες δράσεις για την κάλυψη των αναγκών του Πανελληνίου Διαγωνισμού Πληροφορικής, ο οποίος διεξάγεται κάθε χρόνο από την Ελληνική Εταιρεία Επιστημόνων και Επαγγελματιών Πληροφορικής και Επικοινωνιών (ΕΠΥ), με ανώτατο όριο τέσσερις (4) ημέρες, δε θα λαμβάνονται υπόψη για το χαρακτηρισμό της φοίτησης τους, εφόσον οι μαθητές αυτοί προσκομίσουν στο σχολείο τους βεβαίωση της Οργανωτικής Επιτροπής της ΕΠΥ, με την οποία πιστοποιείται η συμμετοχή τους στη διοργάνωση του Διαγωνισμού.

Απουσίες που ΔΕΝ ΔΙΚΑΙΟΛΟΓΟΥΝΤΑΙ

Σε καμία περίπτωση δε θεωρούνται δικαιολογημένες απουσίες μαθητών:

Όταν πραγματοποιούνται χωρίς την άδεια του Διευθυντή του σχολείου σε ώρες του ημερησίου προγράμματος λειτουργίας του σχολείου, μεμονωμένες ή συνεχείς. Για την πρώτη και την τελευταία διδακτική ώρα του ημερησίου προγράμματος λειτουργίας του σχολείου είναι δυνατόν, κατά την κρίση του συλλόγου, να δικαιολογηθούν απουσίες οφειλόμενες σε εξαιρετικά σοβαρούς αντικειμενικούς λόγους, που αναφέρονται σε ιδιαίτερες τοπικές συνθήκες καιρικές, συγκοινωνιακές και άλλες κάθε σχολείου.

Όταν οφείλονται σε ομαδική, κατά την κρίση του προϊσταμένου της Διεύθυνσης ή Γραφείου Μέσης Εκπαίδευσης, απομάκρυνση των μαθητών από το σχολείο.

Οι απουσίες μαθητών που οφείλονται σε αποχή από τα μαθήματα π.χ. λόγω κρούσματος ηπατίτιδας ή σε κατάληψη του διδακτηρίου, σε καμιά περίπτωση δεν θεωρούνται δικαιολογημένες και υπολογίζονται στο συνολικό αριθμό των απουσιών κάθε μαθητή.

Αντιθέτως δεν πρέπει να ληφθούν υπόψη (ούτε ως δικαιολογημένες) και να καταλογιστούν στους μαθητές εκείνους που με οποιονδήποτε τρόπο δήλωσαν την εμμονή τους στις μαθητικές υποχρεώσεις, δεν έλαβαν μέρος στην κατάληψη και συνεπώς εκδήλωσαν πρόθεση να ακολουθήσουν τη διδασκαλία των μαθημάτων. Για τη συνδρομή των όρων αυτών αρμόδιος είναι να αποφανθεί ο σύλλογος των καθηγητών.

Οι απουσίες μαθητή που οφείλονται σε προφυλάκιση δεν δικαιολογούνται είτε αυτός κριθεί ένοχος είτε αθώος.

Δε θεωρούνται δικαιολογημένες απουσίες λόγω ολιγόωρης ή πολύωρης ή ακόμα ολιγοήμερης ή πολυήμερης ομαδικής απομάκρυνσης των μαθητών για οποιονδήποτε λόγο από το σχολείο.

Αδικαιολόγητες είναι οι απουσίες μαθητών που πραγματοποιούνται κατά τις γιορταστικές εκδηλώσεις σε εργάσιμες ημέρες καθώς και σε μέρες που προβλέπονται από διάταξη νόμου για την πραγματοποίηση γιορταστικών εκδηλώσεων, υπολογίζονται για τον χαρακτηρισμό της φοιτήσεως τους, ενώ δεν υπολογίζονται οι απουσίες που πραγματοποιούνται σε μέρες αργίας των σχολείων. Για τους χρήστες άλλων προγραμμάτων, να γίνεται προσπάθεια να ακολουθείται στο βαθμό του δυνατού η μορφοποίηση του κειμένου όπως προκύπτει από αυτό το δείγμα.

2.3 Ανάλυση

Αναλύοντας την παραπάνω αφηρημένη περιγραφή της γνώσης και επιλέγοντας κάποιες λέξεις – έννοιες κλειδιά καταλήγουμε σε ένα σύστημα που αποτελείται από:

- **Ρητώς αποθηκευμένη γνώση η οποία θα αναπαρασταθεί σε μορφή τυπικής γλώσσας με τις εξής πληροφορίες:**
 - I. Πληροφορίες που αφορούν την κλίμακα βαθμολογίας, τα επίπεδα διαγωγής μαθητών τις τάξεις.
 - II. Πληροφορίες που αφορούν περιπτώσεις απαλλαγής απουσιών μαθητών. Για κάθε τέτοια καταγράφεται ένα μηνιαίο και ετήσιο όριο που δεν πρέπει να υπερβεί κάποιος μαθητής.
 - III. Περιπτώσεις απουσιών μαθητών που δεν δικαιολογούνται.

- IV. Πληροφορίες που αφορούν το όριο παραστατικών που μπορεί να σχετίζεται με ένα μαθητή όσον αφορά τις απουσίες του, το επιτρεπτό χρονικό διάστημα για την δικαιολόγηση μιας απουσίας
- V. Πληροφορίες που αφορούν συνθήκες χαρακτηρισμού φοίτησης μαθητή ως επαρκής βάσει βαθμού, διαγωγής και απουσιών. Πιο συγκεκριμένα μια συνθήκη αναφέρεται:
- Στην Βαθμολογία μαθητή
 - Στην Διαγωγή μαθητή
 - Στο Γενικό σύνολο απουσιών μαθητή
 - Στο σύνολο αδικαιολόγητων απουσιών
 - Στο σύνολο δικαιολογημένων απουσιών
 - Και στο σύνολο δικαιολογημένων λόγω ασθενείας.
- VI. Πληροφορίες που αφορούν την βαθμολογία την διαγωγή και τις απουσίες μαθητών.
- VII. Πληροφορίες για την επεξεργασία των απουσιών δηλαδή την αξιολόγηση και κατάταξή τους σε κάποια κατηγορία :
- Δεν μπορεί να δικαιολογηθεί.
 - Απαλλάσσεται
 - Δικαιολογημένη
 - Αδικαιολόγητη,
- VIII. Εάν υπάρχει κάποιο παραστατικό (για παράδειγμα βεβαίωση από νοσοκομείο)
- IX. Ο χρόνος που έγινε η αξιολόγηση της απουσίας.

- **Συμπερασματική Γνώση η οποία πρέπει απαραίτητως να προσφέρεται από το σύστημα και η οποία αναφέρεται στις εξής πληροφορίες:**

- I. Σύνολο δικαιολογημένων απουσιών μαθητή,
- II. Σύνολο δικαιολογημένων λόγω ασθένειας απουσιών μαθητή,
- III. Σύνολο αδικαιολογήτων απουσιών μαθητή,
- IV. Γενικό Σύνολο απουσιών μαθητή που λαμβάνονται υπόψη,
- V. Χαρακτηρισμός της επάρκειας φοίτησης ενός μαθητή,
- VI. Ποιες συνθήκες χαρακτηρίζουν επαρκή ένα μαθητή.
- VII. Ποιο είναι το μέτρο πλεονάζουσας γνώσης δηλαδή πόσες είναι οι περιπτώσεις όπου μαθητής χαρακτηρίζεται επαρκής από πολλές συνθήκες.

- **Αναδιοργάνωση Γνώσης:**

Το Σύστημα Γνώσης πρέπει να φροντίζει για την τήρηση των κριτηρίων που έχουν θεσπιστεί για την ορθότητά του. Άρα πρέπει να περιέχει διεργασίες που τροποποιούν κατάλληλα την διαθέσιμη γνώση με στόχο την διατήρηση των ποιοτικών στο σύστημα. Επίσης το σύστημα δέχεται μεταβολές που έρχονται από το περιβάλλον στα πλαίσια της δυναμικής που αυτό παρουσιάζει (Εμφάνιση νέας απουσίας μαθητή, Προσθήκη βαθμολογίας μαθητή, Προσθήκη Διαγωγής Μαθητή, Δικαιολόγηση απουσίας) τις οποίες και ενσωματώνει στην γνώση του .

2.4 Μεθοδολογία Σχεδίασης

Για την σχεδίαση του Συστήματος Γνώσης ακολουθήθηκαν τα εξής βήματα:

- **Βήμα 1ο**

Στην αρχή έγινε μια καταγραφή των εννοιών – οντοτήτων που αναφέρονται στο συγκεκριμένο νομοθέτημα. Από τα παραπάνω μετά την ανάλυση προέκυψαν οι εξής Οντότητες οι οποίες αποτελούν για την εφαρμογή γεγονότα με συγκεκριμένες ιδιότητες:

- I. Μαθητής (Αύξων Αριθμός , Όνομα, Τάξη)
- II. Μέσος Όρος Βαθμολογίας (Αύξων Αριθμός, Βαθμός)
- III. Διαγωγή (Αύξων Αριθμός, Διαγωγή),
- IV. Τάξεις (Αριθμός Τάξης, Όνομα Τάξης),
- V. Απουσία (Μαθητής, Αριθμός Απουσίας, Ώρες, Ημερομηνία),
- VI. Πράξη Δικαιολόγησης Απουσίας (Μαθητής, Αριθμός Απουσίας, Ημερομηνία, Αιτιολογία),
- VII. Περιπτώσεις Απαλλαγής Απουσιών (Αιτιολογία, Περιγραφή, Μηνιαίο Επιτρεπτό Σύνολο Απουσιών, Μηνιαίο Επιτρεπτό Σύνολο Απουσιών),
- VIII. Παραστατικό Δικαιολόγησης Απουσίας(Αύξων Αριθμός Μαθητή, Αριθμός Απουσίας),
- IX. Παραστατικό Δικαιολόγησης Απουσίας Λόγω Ασθένειας(Αύξων Αριθμός Μαθητή, Αριθμός Απουσίας, Φορέας Προέλευσης Παραστατικού),
- X. Περιπτώσεις Απουσιών Που Δεν Δικαιολογούνται (Αιτιολογία, Περιγραφή),
- XI. Περίπτωση Απουσίας Δικαιολογημένη Λόγω Ασθένειας (Αιτιολογία, Περιγραφή),
- XII. Περίπτωση Απουσίας Δικαιολογημένη (Αιτιολογία, Περιγραφή),

- XIII. Περίπτωση Απουσίας Αδικαιολόγητη (Αιτιολογία, Περιγραφή),
- XIV. Συνθήκες Επάρκειας Φοίτησης Μαθητή (Επιτρεπτή Περιοχή Συνόλου Απουσιών, Επιτρεπτή Περιοχή Συνόλου Δικαιολογημένων Απουσιών, Επιτρεπτή Περιοχή Συνόλου Δικαιολογημένων Απουσιών Λόγω Ασθένειας, Επιτρεπτή Περιοχή Συνόλου Αδικαιολόγητων Απουσιών, Επιτρεπτή Περιοχή Βαθμολογίας, Διαγωγή).
- XV. Κλίμακα Βαθμολογίας(Κάτω Όριο, Πάνω Όριο).
- XVI. Επίπεδα Διαγωγής (Ονομασία),
- XVII. Ημερήσιο Όριο Ωρών Διδασκαλίας(Όριο),
- XVIII. Όριο Ημερών Στις Οποίες Μπορεί Να Δικαιολογηθεί Μια Απουσία(Όριο),
- XIX. Όριο Παραστατικών Δικαιολόγησης Απουσίας Μαθητή(Όριο),
- XX. Μήνας Φοίτησης(Όνομα Μήνα, Αύξων Αριθμός Ημέρας Λήξης Του Μήνα Από Την Έναρξη Του Σχολικού Έτους.
- XXI. Περίοδο Διακοπών (Αύξων Αριθμός Ημέρας Έναρξης, Αύξων Αριθμός Ημέρας Λήξης)
- XXII. Σάββατο (Αύξων Αριθμός Ημέρας Πρώτου Σαββάτου Σχολικού Έτους).

Αυτά τα γεγονότα αποτελούν ουσιαστικά την «ρητώς» αποθηκευμένη γνώση του συστήματος η οποία περιγράφεται στις εικόνες που ακολουθούν:

```
student.pl
File Edit Browse Compile Prolog Pce Help
student.pl
:- dynamic student/3.
student(2, 'Μαθητής_4', 1).
student(4, 'Μαθητής_20', 1).
student(6, bbbbbbbbbbbbbbbb, 1).
student(5, aaaaaaaaaaaaaaaa, 1).
student(1, 'Μαθητής_3', 1).
student(7, 'Μαθητής_7', 1).
student(3, 'Μαθητής_15', 1).

:- dynamic mo/2.
mo(1, 20).
mo(2, 20).
mo(5, 15).
mo(7, 15).
mo(3, 15).

:- dynamic diagogh/2.
diagogh(2, μέτρια).
diagogh(4, κοσμιότητα).
diagogh(6, κοσμία).
diagogh(1, κοσμιότητα).
diagogh(3, κοσμιότητα).
diagogh(7, κοσμία).
```

Μαθητές και επίδοσή τους.

```
facts.pl
File Edit Browse Compile Prolog Pce Help
student.pl facts.pl
:- dynamic apoysia/5.
apoysia(1, 1, 2, february, 6).
apoysia(2, 1, 6, february, 6).
apoysia(3, 1, 7, february, 6).
apoysia(4, 1, 1, february, 6).
apoysia(1, 2, 4, march, 7).

:- dynamic dikaiologhsh/5.
dikaiologhsh(1, 1, february, 7, 60).
dikaiologhsh(2, 1, february, 7, 40).
dikaiologhsh(3, 1, february, 7, 35).
dikaiologhsh(4, 1, february, 7, 1).

:- dynamic parastatiko/2.

:- dynamic parastatiko_apo_forea/3.
```

Γεγονότα που αφορούν απουσίες μαθητών.

```

parameters.pl
File Edit Browse Compile Prolog Pce Help
student.pl facts.pl parameters.pl
:- dynamic hmerhsio_orio_didaskalias/1.
hmerhsio_orio_didaskalias(7).
:- dynamic arithmos_class/1.
arithmos_class(4).
:- dynamic class/2.
class(1, 'Τάξη_1').
class(2, 'Τάξη_2').
class(3, 'Τάξη_3').
class(4, 'Τάξη_4').
:- dynamic arithmos_periptwsewn_apallaghs/1.
arithmos_periptwsewn_apallaghs(2).
^
:- dynamic apallagh/4.
apallagh(1, 'Αθλητικοί αγώνες', 10, 30).
apallagh(2, 'Στρατολογία', 10, 10).
:- dynamic apoysies_poy_den_dikaiologoyntai/2.
apoysies_poy_den_dikaiologoyntai(30, 'Ωριαία αποβολή').
apoysies_poy_den_dikaiologoyntai(35, 'Ημερήσια αποβολή').
:- dynamic dikaiologhmenh_logo_astheneias/2.
dikaiologhmenh_logo_astheneias(40, d_1_a).
:- dynamic dikaiologhthei/2.
dikaiologhthei(50, d_a).
:- dynamic adikaiologhth/2.
adikaiologhth(60, a_a).
:- dynamic orio_hmerwn/1.
orio_hmerwn(10).
:- dynamic klimaka/2.
klimaka(0, 20).

```

Γεγονότα που αφορούν «βοηθητική» γνώση όπως για παράδειγμα: κατηγορίες απουσιών που δεν δικαιολογούνται, κλίμακα βαθμολογίας κ.α.

```
dates.pl
File Edit Browse Compile Prolog Pce Help
student.pl facts.pl parameters.pl dates.pl
:- dynamic mhnas/3.
:- dynamic sunday/1.
:- dynamic saturday/1.
:- dynamic diakopes/3.

mhnas(1, septembrios, 14).
mhnas(2, oktvbrios, 46).
mhnas(3, noembrios, 78).
mhnas(4, dekembrios, 109).
mhnas(5, ianoyarios, 141).
mhnas(6, febroyarios, 172).
mhnas(7, martios, 201).
mhnas(8, aprilios, 232).
mhnas(9, maios, 264).
mhnas(10, ioynios, 295).
mhnas(11, end, 326).

sunday(22).

saturday(21).

diakopes(1, 204, 207).
```

Γεγονότα που αφορούν αργίες σχολικού έτους.

- Βήμα 2ο

Καταγραφή της επιπλέον γνώσης σε μορφή «Συμπερασμάτων» που πρέπει να παράγει το σύστημα:

Η ρητώς αποθηκευμένη γνώση που περιγράφεται πιο πάνω πρέπει να εμπλουτίζεται από γνώση που προκύπτει από συμπερασματικές διεργασίες. Αυτή αποτυπώνεται ως εξής:

- I. Υπολογισμός για ένα μαθητή του συνόλου απουσιών που αξιολογούνται λαμβάνοντας υπόψη:
 - Εάν η απουσία έχει δικαιολογηθεί
 - Εάν η απουσία έχει δικαιολογηθεί σε εύλογο διάστημα όπως αυτό ορίζεται
 - Εάν η απουσία έχει δικαιολογηθεί ως απαλλακτική και δεν υπάρχει υπέρβαση του μηνιαίου ή ετήσιου ορίου,

```
% ypologimsos apoysion poy lambanontai
% ypopsh ston xarakthrismo mathiti.

synolo_a_y(Mathitis,N):-
    s_ypologismos_m(Mathitis,N1),
    ypologismos_a(Mathitis,N2),
    N is N1+N2.

% o parakatw kanonas ypologizei to synolo
% apoysiwn mathiti poy an kai apallassontai
% telika lambanontai ypopsh eksetazontas
% kai to ethsio orio

s_ypologismos_m(Mathitis,S):-
    ( \+apoysia(Mathitis,_,_,_,_) ;
    ((apoysia(Mathitis,T,_,_,_),
    \+dikaiologhsh(Mathitis,T,_,_,_))) )
    S is 0,! ,true.

s_ypologismos_m(Mathitis,S):-
    arithmos_periptwsewn_apallaghs(N),
    a_ap_poy_den_apal(Mathitis,N,S).
```

```

% o kanonas ypologismos_a paragei ton arithmo apoysivn
% mathith poy lambanontai ypopsh genika

s_a(I,_,T):-
    \+apoysia(I,_,_,_),
    T is 0,
    !,true.

s_a(Mathitis,J,T):-
    J1 is J-1,
    apoysia(Mathitis,J,X,_,_),
    dikaiologhsh(Mathitis,J,_,_,H),!,
    s_a(Mathitis,J1,T1),
    ( \+apallagh(H,_,_,_) -> R is X; R is 0),
    T is T1+R,
    !,true.

s_a(_,0,0).

ypologismos_a(Mathitis,D):-
    ( \+apoysia(Mathitis,_,_,_,_);
    ( (apoysia(Mathitis,T,_,_,_),\+dikaiologhsh(Mathitis,T,_,_,_))))
    ,D is 0,
    !,true.

ypologismos_a(Mathitis,D):-
    c(Mathitis,N),s_a(Mathitis,N,D).

```

- II. Υπολογισμός για ένα μαθητή του συνόλου δικαιολογημένων απουσιών που αξιολογούνται:

```

%o kanonas ypologismos_a_d paragei ton arithmo apoysivn
% mathith poy einai dikaiologhmenes.

s_a_d(Mathitis,J,T):-
    J1 is J-1,
    apoysia(Mathitis,J,X,_,_),s_a_d(Mathitis,J1,T1),
    ( dikaiologhmenh(Mathitis,J) ->R is X; R is 0),
    T is T1+R,!,true.

s_a_d(_,0,0).

s_a_d_y_a_p(M,I,0):-

    c(M,K),K<I,
    !,true.

^s_a_d_y_a_p(Mathitis,J,T):-

    J1 is J+1,
    s_a_d_y_a_p(Mathitis,J1,T1),
    ( apoysia(Mathitis,J,X,_,_),
    dikaiologhmenh(Mathitis,J),
    parastatiko(Mathitis,J),
    orio_parastatikwn(L),
    ( J <L;J=L))->
    (T is T1+X);
    (T is T1)),
    !,true.

ypologismos_a_d_y_p(Mathitis,D):-

    ( \+apoysia(Mathitis,_,_,_,_);
    \+((apoysia(Mathitis,T,_,_,_),dikaiologhsh(Mathitis,T,_,_,_))),parastatiko(Mathitis,T))
    ,D is 0,
    !,true.

ypologismos_a_d_y_p(Mathitis,D):-

    s_a_d_y_a_p(Mathitis,1,D).

ypologismos_a_d(Mathitis,D):-

    ( \+apoysia(Mathitis,_,_,_,_);
    ((apoysia(Mathitis,T,_,_,_),\+dikaiologhsh(Mathitis,T,_,_,_))))
    ,D is 0.

ypologismos_a_d(Mathitis,D):-

    c(Mathitis,N),s_a_d(Mathitis,N,D1),
    ypologismos_a_d_y_p(Mathitis,D2),
    D is D1-D2.

```


- III. Υπολογισμός για ένα μαθητή του συνόλου δικαιολογημένων απουσιών λόγω ασθένειας που αξιολογούνται:

```
% ο κανονας ypologismos_a_d_a paragei ton arithmo apoysivn
% mathith poy einai dikaiologhmenes logo astheneias.

s_a_d_a(Mathitis,J,T):-

    J1 is J-1,
    apoysia(Mathitis,J,X,_),
    s_a_d_a(Mathitis,J1,T1),
    ( dikaiologhmenh_l_a(Mathitis,J) -> R is X; R is 0),
    T is T1+R,
    !,true.

s_a_d_a(_,0,0):-

    !,true.

ypologismos_a_d_a(Mathitis,D):-

    ( \+apoysia(Mathitis,_,_,_) );
    ((apoysia(Mathitis,T,_,_) ,\+dikaiologhsh(Mathitis,T,_,_)))
    ,D is 0,
    !,true.

ypologismos_a_d_a(Mathitis,D):-

    c(Mathitis,N),s_a_d_a(Mathitis,N,D).
```

IV. Υπολογισμός για ένα μαθητή του συνόλου αδικαιολόγητων απουσιών που αξιολογούνται:

```

% o kanonas ypologismos_a_a paragei ton arithmo apoysivn
% mathith poy einai adikaiologhtes.
% Sto apotelesma tha prostethoyn kai oi
% dikaiologhmenes apoysies poy o arithmos
% toys yperbainei to orio poy uetei

s_a_a(Mathitis, J, T):-

    J1 is J-1,
    apoysia(Mathitis, J, X, _, _),
    dikaiologhsh(Mathitis, J, _, _, H),
    s_a_a(Mathitis, J1, T1),
    (( \+apallagh(H, _, _, _),
    \+dikaiologhmenh(I, J)
    , \+dikaiologhmenh_l_a(I, J)) -> R is X; R is 0),
    T is T1+R,
    !, true.

s_a_a(_, 0, 0).

ypologismos_a_a(Mathitis, D):-

    ( \+apoysia(Mathitis, _, _, _, _);
    ((apoysia(Mathitis, T, _, _, _), \+dikaiologhsh(Mathitis, T, _, _, _)))
    , D is 0,
    !, true.

ypologismos_a_a(Mathitis, D):-

    c(Mathitis, N), s_a_a(Mathitis, N, D),
    !, true.

```

- V. Έλεγχος εάν συγκεκριμένος μαθητής επαληθεύει κάποιες από τις συνθήκες επάρκειας φοίτησης.

```

% H check_case elegxei ean o mathiths X
% empiptei se kapoia apo tiw periptwseis

check_case(_,0):-
    !,false.

check_case(X,C):-
    ( ( test_student_aa(X,C),
    periptwsh(C,Per),write('XARAKTHRIZETAI BASH THS PERIPTWSHS:'),write(Per),nl )
    ;(C1 is C-1,
    check_case(X,C1)))
    ,!,true.

eparkis(X):-
    student(X,_,_),diagogh(X,_),mo(X,_)
    arithmos_periptwsewn_eparkeias(N),
    check_case(X,N).

test_student_aa(X,C1):-
    student(X,_,_),
    periptwsh(C1,_)
    synolo_a_y(X,M1),
    ypologismos_a_a(X,M2),
    ypologismos_a_d(X,M3),
    ypologismos_a_d_a(X,M4),
    diagogh(X,T),
    mo(X,M5),
    ( synolo(C1,N1,N2) -> F1 is 1 ; F1 is 0),
    ( synolo_a(C1,N3,N4) -> F2 is 1 ; F2 is 0),
    ( synolo_d_a(C1,N5,N6) -> F3 is 1 ; F3 is 0),
    ( synolo_d(C1,N7,N8) -> F4 is 1 ; F4 is 0),
    ( grades(C1,N9,N10) -> F5 is 1 ; F5 is 0),
    ( exei_diagogh(C1,Dia) -> F6 is 1 ; F6 is 0),

    ( ( F1==1,M1>=N1, (M1<N2;M1==N2)); F1==0),
    ( ( F2==1,M2>=N3, (M2<N4;M2==N4)); F2==0),
    ( ( F3==1,M4>=N5, (M4<N6;M4==N6)); F3==0),
    ( ( F4==1,M3>=N7, (M3<N8;M3==N8)); F4==0),
    ( ( F5==1,M5>=N9, (M5<N10;M5==N10)); F5==0),
    ( (F6==1, T==Dia); F6==0),!,true.

```

- **Βήμα 3ο**

Υλοποίηση στα πλαίσια της αναθεώρησης πεποιθήσεων και ειδικότερα της αναδιοργάνωσης γνώσης κανόνων οι οποίοι επιδρούν έτσι ώστε το σύστημα να παραμένει επίκαιρο και αξιόπιστο όσον αφορά τα κριτήρια που έχουν τεθεί. Θεωρώντας ότι τα κριτήρια που αναφερθήκαν πιο πάνω αποτελούν μέρος της γνώσης αφού ουσιαστικά εμπλέκονται στην αναδιοργάνωσή της το σύστημα έχει την δυνατότητα να:

- I. Αναζητεί τις συνθήκες επάρκειας φοίτησης για τις οποίες υπάρχουν μαθητές που επαληθεύονται ως επαρκείς από τουλάχιστον δύο συνθήκες.

```
list_dpl_case:-
```

```
  \+ (   periptwsh(C1,_),periptwsh(C2,_),\+ C1=C2,((\+ q_temp(C2));(\+ q_temp(C1))),
        student(X,_,_),test_student_aa(X,C1),test_student_aa(X,C2)),
      !,true.
```

```
list_dpl_case:-
```

```
  ▲ periptwsh(C1,_),periptwsh(C2,_), \+ C1=C2,
    \+ ( q_temp(C1), q_temp(C2)),student(X,_,_),test_student_aa(X,C1),
    test_student_aa(X,C2),
    (   \+ q_temp(C1),assert(q_temp(C1));true),
    (   \+ q_temp(C2),assert(q_temp(C2));true),
    list_dpl_case,
    !,true.
```

- II. Υπολογίζει το μέτρο πλεονασμού στο σύστημα, «γνώση» που ουσιαστικά την χρησιμοποιεί για να βελτιώσει την κατάστασή του. Το σύστημα εάν διαπιστώσει ότι το φαινόμενο πλεονασμού είναι υπαρκτό «προτείνει» αναδιοργάνωση γνώσης με στόχο την βελτίωση του μέτρου πλεονασμού χωρίς να υποβαθμίζεται το κριτήριο ορθότητας. Οι προτάσεις αφορούν συγκεκριμένες περιοχές της γνώσης στις οποίες οι προτεινόμενες αλλαγές μπορούν να μειώσουν το μέτρο πλεονασμού. Η επιλογή τους έγινε με βασικό

άξονα να διατηρείται το κριτήριο της ορθότητας με τις αλλαγές που θα γίνονται. Στην συγκεκριμένη εφαρμογή το σύστημα προτείνει αλλαγές στις εξής περιοχές της γνώσης:

- Αφαίρεση Συνθηκών Επάρκειας Φοίτησης
- Τροποποίηση Διαγωγής Μαθητή
- Διαγραφή Μαθητή.

```
:- dynamic cd/3.  
d_cd:-  
    cd(X,Y,Z),  
    retract(cd(X,Y,Z)),  
    d_cd,!,true.  
d_cd:- \+ cd(_,_,_),!,true.  
  
collision(N):-  
    \+ ( student(X,_,_),  
        periptwsh(C1,_),periptwsh(C2,_),\+ C1=C2,  
        test_student_aa(X,C1), test_student_aa(X,C2),  
        \+ cd(X,C1,_)),d_cd,N is 0,!,true.  
  
collision(N):-  
    student(X,_,_),  
    periptwsh(C1,_),periptwsh(C2,_),\+ C1=C2,  
    test_student_aa(X,C1), test_student_aa(X,C2),  
    \+ cd(X,C1,_),  
    assert(cd(X,C1,0)),  
    collision(N1),N is N1+1,!,true.
```

III. Προσθήκη μαθητή,

```
input_student(S,O,C):-  
    \+ student(_,O,C),  
    find(N),N1 is N+1,  
    class(C,_),  
    S=N1,  
    assert(student(N1,O,C)),  
    update_student,  
    ! true
```

IV. Διαγραφή Μαθητή

```
s_student(S,E):-
    S>E,!,true.

s_student(S,E):-
    (S<E;S=E),
    student(S,O,C),
    S1 is S-1,
    assert(student(S1,O,C)),
    retract(student(S,_,_)),
    ( ( mo(S,G), assert(mo(S1,G)),
    retract(mo(S,_) );true),
    ( ( diagogh(S,D), assert(diagogh(S1,D)),
    retract(diagogh(S,_) );true),

    S2 is S+1,

    s_student(S2,E),
    !,true.

delete_parastatiko(K):-
    parastatiko(K,_),
    K1 is K,
    retract(parastatiko(K,_)),

    delete_parastatiko(K1),

    !,true.

delete_parastatiko(_-):-
    !,true.

delete_parastatiko_apo_forea(K):-
    parastatiko_apo_forea(K,_,_),
    K1 is K,
    retract(parastatiko_apo_forea(K,_,_)),

    delete_parastatiko_apo_forea(K1),

    !,true.
```

```

delete_parastatiko_apo_forea(K):-
    parastatiko_apo_forea(K,_,_),
    K1 is K,
    retract(parastatiko_apo_forea(K,_,_)),
    delete_parastatiko_apo_forea(K1),
    !,true.

delete_parastatiko_apo_forea(_):-
    !,true.

delete_student(K):-
    student(K,_,_), find(N),
    delete_parastatiko(K),
    delete_parastatiko_apo_forea(K),
    K1 is K+1,
    K2 is K,
    (    retract(student(K2,_,_));true),
    (    retract(mo(K2,_));true),
    (    retract(diagogh(K2,_));true),
    s_student(K1,N),
    delete_apoysia(K2),
    delete_dikaiologhsh(K2),
    shift1_apoysies(K1,N),
    shift_dikaios(K1,N),
    update_student,
    update_facts,
    !,true.

delete_apoysia(S):-
    \+ apoysia(S,_,_,_,_),
    !,true.

delete_apoysia(S):-
    apoysia(S,_,_,_,_),
    retract(apoysia(S,_,_,_,_)),
    delete_apoysia(S),
    !,true.

```



```
delete_apoysia(S):-
```

```
    \+ apoysia(S,_,_,_,_),  
    !,true.
```

```
delete_apoysia(S):-
```

```
    apoysia(S,_,_,_,_),  
    retract(apoysia(S,_,_,_,_)),  
    delete apoysia(S),  
    !,true.▲
```

```
delete_dikaiologhsh(S):-
```

```
    dikaiologhsh(S,_,_,_,_),  
    retract(dikaiologhsh(S,_,_,_,_)),  
    delete dikaiologhsh(S),  
    !,true.
```

```
delete_dikaiologhsh(S):-
```

```
    \+ dikaiologhsh(S,_,_,_,_),  
    !,true.
```

```
shift1_apoysia(S):-
```

```
    apoysia(S,N,X,Y,Z),  
    S1 is S-1,  
    (    (    retract(apoysia(S,N,X,Y,Z)),    assert(apoysia(S1,N,X,Y,Z))) ; true),  
  
    shift1 apoysia(S),  
    !,true.
```

```
shift1_apoysia(S):-
```

```
    \+ apoysia(S,_,_,_,_),  
    !,true.
```

```
shift1_apoysies(S,E):-
```

```
    S>E,! ,true.
```

```

shift1_apoysies(S,E):-
    (S<E;S=E),
    shift1_apoysia(S),
    S1 is S+1,
    shift1_apoysies(S1,E),
    !,true.

shift1_apoysies(_,-):-
    !,true.

shift1_dikaio(S):-
    dikaiologhsh(S,N,X,Y,Z),
    S1 is S-1,
    S2 is S,
    ( ( retract(dikaiologhsh(S,N,_,_)),assert(dikaiologhsh(S1,N,X,Y,Z)),
      ( ( parastatiko(S,N),assert(parastatiko(S1,N)),retract(parastatiko(S1,_)));
      ( parastatiko_apo_forea(S,N,F),assert(parastatiko_apo_forea(S1,N,F)),retract(parastatiko_apo_forea(S1,_,_))
      ;true));true),
    shift1_dikaio(S2),
    !,true.

shift1_dikaio(S):-
    \+ dikaiologhsh(S,_,_,_),!,true.

shift_dikaio(S,E):-
    S>E,!,true.

shift_dikaio(S,E):-
    (S<E;S=E),
    shift1_dikaio(S),
    S1 is S+1,
    shift_dikaio(S1,E),
    !,true.

shift_dikaio(_,-):-
    !,true.

```

V. Διόρθωση Μαθητή

VI. Προσθήκη Βαθμολογίας

```
input_mo(S,G):-  
    student(S,_,_),  
    klimaka(L,R),  
    G >=L ,  
    ( G < R; G=R),  
    ( retract(mo(S,_));true),  
    assert(mo(S,G)),  
    update_student,  
    !,true.
```

VII. Προσθήκη διαγωγής

```
input_diagogh(S,D):-  
    student(S,_,_),  
    epipeda_diagoghs(F),  
    F=D,  
    ( retract(diagogh(S,_));true),  
    assert(diagogh(S,D)),  
    update_student,  
    !,true.
```

VIII. Προσθήκη απουσίας μαθητή.

```

input_apoysia(S, Noa, M, H) :-
    \+ apoysia(S, _, Noa, M, H),
    student(S, _, _),
    find_a_a(S, N), N1 is N+1,
    Noa>0,
    hmerhsio_orio_didaskalias(M2), (Noa<M2;Noa=M2),
    mhnas(_, M1, _), M1=M,
    day_count(M1, H, Nd),
    \+ elegxos_mh_erg(Nd),
    s_s_d_a(S, M, H, R), R1 is R+Noa, (R1<M2;R1=M2),
    assert(apoysia(S, N1, Noa, M, H)),
    update_facts,
    !, true.

input_apoysia(_, _, _, _) :-
    !, false.

```

IX. Δικαιολόγηση απουσίας

```

input_dikaiologhsh(S, A, Md, Hd, C) :-
    \+ dikaiologhsh(S, A, Md, Hd, _),
    student(S, _, _),
    apoysia(S, A, _, Ma, Ha),
    mhnas(_, Ma, Na), mhnas(_, Md, Nd), Sa is Na+Ha,
    Sd is Nd+Hd, (Sa<Sd;Sa=Sd),
    day_count(Md, Hd, N),
    \+ elegxos_mh_erg(N), day_count(Ma, Ha, N1), N2 is N-N1, orio_hmerwn(P),
    ( N2<P;N2=P),
    ( apallagh(C, _, _, _);apoysies_poy_den_dikaiologoyntai(C, _);
    dikaiologhmenh_logo_astheneias(C, _);
    dikaiologhthei(C, _);
    adikaiologhth(C, _)),
    assert(dikaiologhsh(S, A, Md, Hd, C)),
    update_facts,
    !, true.

input_dikaiologhsh(_, _, _, _, _) :-
    !, false.

```

X. Προσθήκη παραστατικού όσον αφορά την δικαιολόγηση απουσίας

XI. Αλλαγή της κλίμακας βαθμολογίας

```
menu_sel2(X,S,L,R,L1,R1):-  
    S==1111,  
    send(X,destroy),  
    retract(klimaka(L1,R1)),assert(klimaka(L,R)),update_parameters,  
    menu_2(0),!,true.
```

XII. Αλλαγή των επιπέδων διαγωγής

```
menu_sel2(X,S,Y):-  
    S==2222,  
    send(X,destroy),  
    ( retract(epipeda_diagoghs(Y))->(update_parameters,w_label3('Η διαγραφή έγινε'));  
      w_label3('Δεν υπάρχει ο τίτλος')),  
    !,true.
```

```
menu_sel2(X,S,Y):-  
    S==5555,  
    send(X,destroy),  
    ( (\+ epipeda_diagoghs(Y))->(assert(epipeda_diagoghs(Y)),update_parameters,w_label3('Η εγγραφή έγινε'));  
      w_label3('Υπάρχει ο τίτλος')),  
    !,true.
```

XIII. Προσθήκη – Αφαίρεση Τάξης.

XIV. Τροποποίηση του ημερήσιου ορίου ωρών διδασκαλίας.

XV. Τροποποίηση του χρονικού ορίου για την δικαιολόγηση απουσίας

```

menu_orio_hm(X):-
    send(X,destroy),
    new(D2,dialog('Όριο ημερών στις οποίες μπορεί να δικαιολογηθεί απουσία')),
    send(D2,size,size(400,200)),
    ( orio_hmerwn(N); N is 0),
    send(D2,append,new(T1,text_item('Όριο: ',N))),
    send(D2,append,new(B2,button(ok,message(@prolog, menu_sel2,D2,3,T1?selection)))),
    send(D2,append,new(B0,button(cancel,message(@prolog, esc_1,D2)  ))),

    get(D2,confirm,B0),
    get(D2,confirm,B2),
    !,true.

```

```

menu_sel2(X,S,P1):-

```

```

    S==3,
    send(X,destroy),
    ( retract(orio_hmerwn(_));true),
    assert(orio_hmerwn(P1)),
    update_parameters,
    w_label3('η εγγραφή έγινε'),
    !,true.

```

XVI. Τροποποίηση του ορίου αριθμού παραστατικών δικαιολόγησης απουσίας.

```

menu_parast_orio(X):-
    send(X,destroy),
    new(D2,dialog('Όριο Παραστατικών')),
    send(D2,size,size(400,200)),
    orio_parastatikwn(N),
    send(D2,append,new(T1,text_item('Όριο: ',N))),
    send(D2,append,new(B2,button(ok,message(@prolog,menu_sel2,D2,3333,T1?selection)))),
    send(D2,append,new(B0,button(cancel,message(@prolog, esc_1,D2)  ))),

    get(D2,confirm,B0),
    get(D2,confirm,B2),
    !,true.

```

```
menu_sel2(X,S,T):-
```

```
    S==3333,  
    retract(orio_parastatikwn(_)),  
    assert(orio_parastatikwn(T)),  
    send(X,destroy),  
    update_per_epa,menu_2(0),  
    !,true.
```

XVII. Τροποποίηση των κατηγοριών απαλλαγής όσον αφορά απουσίες.

XVIII. Τροποποίηση των κατηγοριών απαγόρευσης δικαιολόγησης όσον αφορά απουσίες.

XIX. Τροποποίηση των συνθηκών επάρκειας φοίτησης μαθητών.

```
input_per_eparkeias(Per,Diagwgh,Synolo1,Synolo2,Synolo_a1,Synolo_a2,Synolo_d_a1,Synolo_d_a2,Synolo_d1,Synolo_d2,Grade1,Grade2):-
```

```
    \+ periptwsh(_,Per),  
    find_p_e(N),  
    N1 is N+1,  
    klimaka(K1,Kr),  
    ( ( Grade2 >= Grade1,  
      Grade1>=K1, (Grade1<Kr;Grade1=Kr),  
      Grade2>=K1, (Grade2<Kr;Grade2=Kr));  
      ( Grade1 = -1 ,Grade2 = -1)),  
    ( ( epipeda_diagoghs(Epd),Epd=Diagwgh);Diagwgh=''),  
    arithmos_periptwsewn_eparkeias(K),  
    K1 is K+1,  
    retract(arithmos_periptwsewn_eparkeias(K)),  
    assert(arithmos_periptwsewn_eparkeias(K1)),  
    assert(periptwsh(N1,Per)),  
    ( ( \+ Diagwgh='', assert(exei_diagogh(N1,Diagwgh));true),  
      ( \+ ( Synolo1 == -1,Synolo2 == -1),assert(synolo(N1,Synolo1,Synolo2));true),  
      ( \+ ( Synolo_a1 == -1,Synolo_a2 == -1), assert(synolo_a(N1,Synolo_a1,Synolo_a2));true),  
      ( \+ ( Synolo_d1 == -1,Synolo_d2 == -1), assert(synolo_d(N1,Synolo_d1,Synolo_d2));true),  
      ( \+ ( Synolo_d_a1 == -1 ,Synolo_d_a2 == -1), assert(synolo_d_a(N1,Synolo_d_a1,Synolo_d_a2));true),  
      ( \+ ( Grade1 == -1 ,Grade2 == -1), assert(grades(N1,Grade1,Grade2));true),  
    update_per_epa,  
    !,true.
```

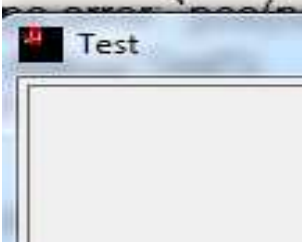
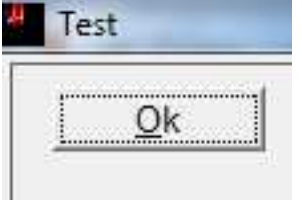
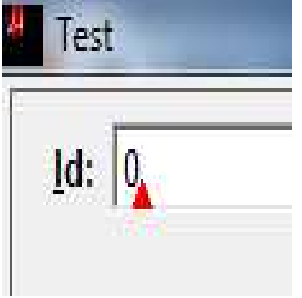
XX. Καθορισμός των αργιών.

- **Βήμα 4ο**

Υλοποίηση ενός γραφικού περιβάλλοντος επικοινωνίας το οποίο:

- Εμφανίζει τη γνώση που παράγουν οι κανόνες χρησιμοποιώντας γραφικά αντικείμενα όπως πλαίσια κειμένου, κουμπιά, λίστες επιλογής,
- Αποτελεί της διεπαφή χρήστη με το σύστημα γνώσης.

Το συγκεκριμένο περιβάλλον «κατασκευάζεται» με την χρήση ενσωματωμένων κανόνων Prolog που περιέχει το πακέτο XPCE [15]. Ενδεικτικά αναφέρω:

| Κανόνας | Αποτέλεσμα |
|------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| <pre>new(D,dialog('Test')),get(D,confirm,_).</pre> |  |
| <pre>new(D,dialog('Test')),send(D,append,new(button(ok))),get(D,confirm,_).</pre> |  |
| <pre>- new(D,dialog('Test')),send(D,append,new(text_item(id,0))),get(D,confirm,_).</pre> |  |

2.5 Εργαλεία Που Χρησιμοποιήθηκαν

Για την αναπαράσταση του συστήματος γνώσης χρησιμοποιήθηκε η δηλωτική γλώσσα προγραμματισμού PROLOG και η συγκεκριμένη έκδοσή της SWI-Prolog (Multi-threaded, 32 bits, Version 6.0.2). Οι λόγοι είναι:

- Είναι γλώσσα που ξεφεύγει από την καθαρά αλγοριθμική διαδικασία σκέψης πράγμα το οποίο ταιριάζει στην συγκεκριμένη εφαρμογή όπου οι διαδικασίες ελέγχου και αναθεωρήσεις της Γνώσης δεν επιλύονται καθορά με αλγοριθμικές προσεγγίσεις.
- Είναι γλώσσα όμως που μπορεί να υποστηρίξει στοιχειώσεις πράξεις που αφορούν Βάσεις Δεδομένων (Προσθήκη - /Διαγραφή) μέσω εντολών (assert –retract).
- Είναι γλώσσα που ταιριάζει σε εφαρμογές τεχνητής ευφυΐας.

Για την διεπαφή του συστήματος με τον χρήστη χρησιμοποιήθηκε ένα συνοδευτικό εργαλείο της SWI Prolog το XPCE που προσφέρει την δυνατότητα ανάπτυξης γραφικού περιβάλλοντος επικοινωνίας μέσω μιας πλατφόρμας με προσανατολισμό αντικειμενοστραφούς προγραμματισμού και συντακτικού εντολών παρόμοιο με αυτό της Prolog. Η συγκεκριμένη πλατφόρμα ουσιαστικά δίνει την δυνατότητα διαχείρισης στοιχείων γραφικού περιβάλλοντος επικοινωνίας (παράθυρα διαλόγου, κουμπιά, πλαίσια κειμένου) σε μορφή αντικειμένων με χρήση «ενσωματωμένων» κανόνων μορφής Prolog. Κυρίως χρησιμοποιήθηκαν οι κανόνες:

- new (Δημιουργία αντικειμένου)
- send (Διαμόρφωση – ενσωμάτωση αντικειμένων που ανήκουν ιεραρχικά σε κάποια άλλα)
- get (Διαχείριση συμβάντων πάνω σε αντικείμενα).

Ακολουθεί ένα παράδειγμα περιγραφής ενός κανόνα (m_menu) που δημιουργεί ένα μενού επιλογών που αποτελείται από κουμπιά:

```

m_menu(X):-
    ( X == 0-> start;send(X,destroy)),

    new(D1,dialog('Κεντρικό Μενού')),

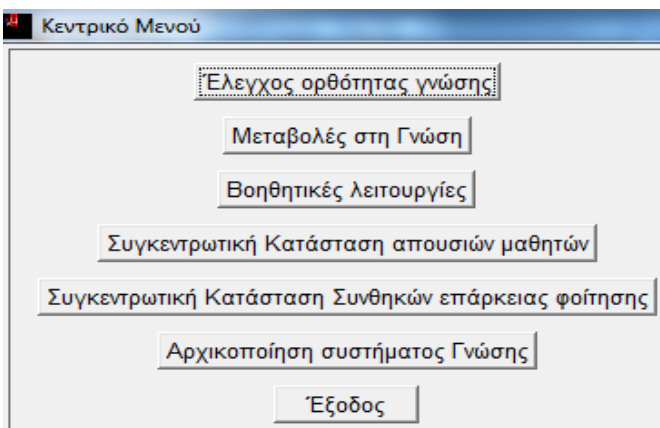
    send(D1,size,size(400,250)),
    send(D1,layout_dialog),
    send(D1,append,new(BT,dialog_group(buttons,group))),
    send(BT,gap,size(0,10)),
    send(BT,alignment,center),
    send(BT,append,new(B1,
    button('Έλεγχος ορθότητας γνώσης',message(@prolog,
    menu,D1))),below),
    send(BT,append,new(B2,
    button('Μεταβολές στη Γνώση',message(@prolog,menu_1,D1))),below),
    send(BT,append,new(B3,button('Βοηθητικές λειτουργίες',message(@prolog,
    menu_2,D1))),below),
    send(BT,append,new(B6,button('Συγκεντρωτική Κατάσταση απουσιών μαθητών',
    message(@prolog,display_database,D1))),below),
    send(BT,append,new(B7,button('Συγκεντρωτική Κατάσταση Συνθηκών επάρκειας φοίτησης',
    message(@prolog,display_database2,D1))),below),
    send(BT,append,new(B5,button('Αρχικοποίηση συστήματος Γνώσης',
    message(@prolog,new_base,D1))),below),
    send(BT,append,new(B4,button('Έξοδος',
    message(@prolog,ok1,D1))),below),

    get(D1,confirm,B1),
    get(D1,confirm,B2),
    get(D1,confirm,B3),
    get(D1,confirm,B4),
    get(D1,confirm,B5),
    get(D1,confirm,B6),
    get(D1,confirm,B7),

    !,true.

```

Και το αποτέλεσμα κλήσης του κανόνα:



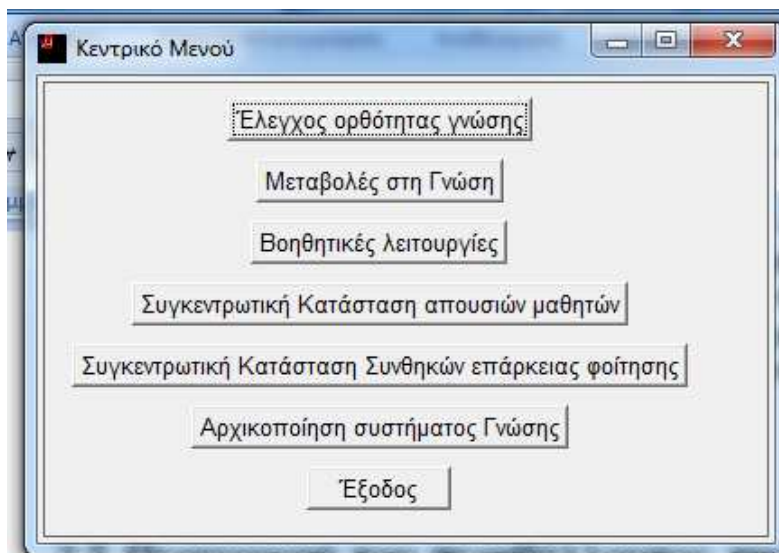
2.6 Περιγραφή Του Περιβάλλοντος Της Εφαρμογής:

2.6.1 Εισαγωγή

Γενικά στην εφαρμογή έχουμε δύο δυνατότητες:

- **Μεταβολές στη Γνώση** Τροποποίηση της υπάρχουσας γνώσης με στόχο την τήρηση του κριτηρίου ορθότητας γνώσης όσον αφορά τον συμπερασμό για την επάρκεια φοίτησης.
- **Έλεγχος ορθότητας γνώσης** Έλεγχος όσον αφορά επικείμενη αναδιοργάνωση γνώσης με δυνατότητα παραγωγής συμπερασμάτων που αφορούν προτάσεις για απλοποίηση του συστήματος επί τη εμφανίσει φαινομένων πλεονασμού γνώσης.

Το περιβάλλον της εφαρμογής υποστηρίζεται από ένα σύστημα μενού επικοινωνίας . Ουσιαστικά στην εφαρμογή υποστηρίζονται τέσσερις βασικές εργασίες. Η πλοήγηση γίνεται μέσω του κύριου μενού:



Η επιλογή **Συγκεντρωτική Κατάσταση απουσιών μαθητών** ουσιαστικά παράγει τη γνώση που αφορά τις απουσίες των μαθητών. Ακολουθεί ένα ενδεικτικό στιγμιότυπο:

Σγκεντρωτική Κατάσταση Απουσιών Μαθητών

| Αα | Μαθητή Όνομα | M.O. | Διαγωγή | Σύνολο απουσιών | Απουσίες δικαιολογημένες | Απουσίες δικαιολογημένες λόγω Ασθενείας | Απουσίες Αδικαιολόγητες | |
|----|----------------|---------------------------|------------|-----------------|--------------------------|-----------------------------------------|-------------------------|--|
| 1 | Μαθητής_3 | 20 | κοσμιότητα | 0 | 0 | 0 | 0 | |
| 2 | Μαθητής_4 | 20 | μέτρια | 6 | 0 | 6 | 0 | |
| 3 | Μαθητής_15 | 17 | κοσμιότητα | 7 | 0 | 0 | 0 | |
| 4 | Μαθητής_20 | -----Ελλιπή Στοιχεία----- | | | | | | |
| 6 | bbbbbbbbbbbbbb | -----Ελλιπή Στοιχεία----- | | | | | | |
| 5 | aaaaaaaaaaaaaa | -----Ελλιπή Στοιχεία----- | | | | | | |
| 7 | Μαθητής_7 | 15 | κοσμία | 0 | 0 | 0 | 0 | |

Στην παραπάνω εικόνα μπορούμε σύντομα να συμπεράνουμε ότι ο «Μαθητής_20» δεν αξιολογείται στο Σύστημα Γνώσης όσον αφορά την επάρκεια φοίτησης διότι λείπουν στοιχεία που τον αφορούν (βαθμός, διαγωγή). Ο «Μαθητής_4» έχει 6 απουσίες στο σύνολο οι οποίες έχουν δικαιολογηθεί λόγω ασθενείας.

Σγκεντρωτική Κατάσταση Συνθηκών επάρκειας φοίτησης

Η επιλογή **Σγκεντρωτική Κατάσταση Συνθηκών επάρκειας φοίτησης** εμφανίζει την γνώση που αφορά τις συνθήκες επάρκειας στο σύστημα:

| Αα | Όνομασία | Περιοχή Βαθμών | Διαγωγή | Σύνολο απουσιών | Σύνολο Απουσιών δικαιολογημένων | Σύνολο Απουσιών δικαιολογημένων λόγω ασθενείας | Σύνολο Απουσιών αδικαιολόγητων |
|----|-------------|------------------|------------|-----------------|---------------------------------|------------------------------------------------|--------------------------------|
| 1 | Περίπτωση_3 | 16 -- 20 | κοσμιότητα | 0 -- 161 | Χωρίς περιορισμό | 0 -- 58 | 0 -- 50 |
| 2 | Περίπτωση_2 | 0 -- 20 | κοσμία | 0 -- 25 | Χωρίς περιορισμό | Χωρίς περιορισμό | Χωρίς περιορισμό |
| 3 | Περίπτωση_1 | 0 -- 18 | κοσμία | 0 -- 51 | Χωρίς περιορισμό | Χωρίς περιορισμό | Χωρίς περιορισμό |
| 4 | Περίπτωση_4 | 1 -- 10 | μέτρια | 0 -- 24 | Χωρίς περιορισμό | Χωρίς περιορισμό | Χωρίς περιορισμό |
| 5 | Περίπτωση_8 | Χωρίς περιορισμό | κοσμιότητα | 0 -- 48 | Χωρίς περιορισμό | Χωρίς περιορισμό | Χωρίς περιορισμό |

Ενδεικτικά αναφέρουμε ότι στο συγκεκριμένο Σύστημα Γνώσης που περιγράφεται υπάρχουν 5 συνθήκες επάρκειας φοίτησης. Η περίπτωση «Περίπτωση_8» αφορά μαθητές με διαγωγή «κοσμιότητα» και σύνολο απουσιών μέχρι 48 και κανένα άλλο περιορισμό.

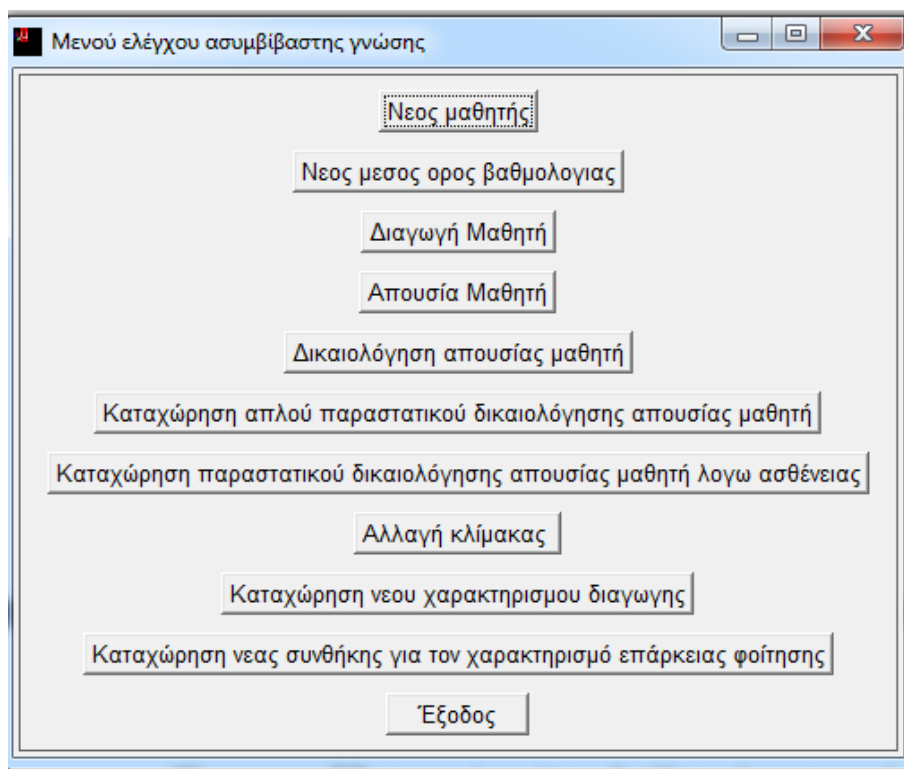
Οι τέσσερις βασικές περιοχές της εφαρμογής είναι:

2.6.2 Αρχικοποίηση Συστήματος Γνώσης

Με την επιλογή **Αρχικοποίηση συστήματος Γνώσης** ουσιαστικά το Σύστημα Γνώσης αρχικοποιείται σε μια κατάσταση που δεν υπάρχουν μαθητές ούτε συνθήκες επάρκειας φοίτησης και το σύστημα πληροί τα κριτήρια που έχουν τεθεί.

2.6.3 Διαδικασίες Αναδιοργάνωσης Γνώσης.

Με την επιλογή **Έλεγχος ορθότητας γνώσης** υποστηρίζονται διαδικασίες όπου εξετάζεται εάν μια προτεινόμενη αλλαγή στην Γνώση πρέπει να προκαλέσει και άλλες αλλαγές για να οδηγηθεί το σύστημα σε μια κατάσταση που θα πληροί τα κριτήρια που έχουν οριστεί. Η εικόνα που ακολουθεί αναφέρεται σε στιγμιότυπο της εφαρμογής που αφορά το σχετικό μενού:



Στην συγκεκριμένη περιοχή της εφαρμογής οι περιπτώσεις που ουσιαστικά αφορούν έλεγχο για αναδιοργάνωση γνώσης είναι οι επιλογές:

- Νέος Μέσος Όρος Βαθμολογίας,
- Διαγωγή Μαθητή,
- Δικαιολόγηση Απουσίας,
- Καταχώρηση Νέας Συνθήκης Για τον Χαρακτηρισμό Επάρκειας Φοίτησης.

Αυτό συμβαίνει διότι οι παραπάνω «μεταβολές» στην γνώση μπορούν να οδηγήσουν σε μεταβολή του μέτρου πλεονασμού στο σύστημα.

Πρέπει να τονίσουμε ότι το Σύστημα Γνώσης έχει σχεδιαστεί έτσι ώστε σε περίπτωση όπου οδηγείται σε μια κατάσταση όπου δεν ισχύουν τα κριτήρια που έχουν οριστεί να προτείνει αλλαγές γνώσης που το οδηγούν σε μια «καλύτερη» κατάσταση.

Παράδειγμα 1^ο:

Έστω ότι η τρέχουσα κατάσταση του συστήματος γνώσης περιγράφεται από τις παρακάτω οθόνες:

| Αα | Όνομασία | Περιοχή | Βαθμών | Διαγωγή | Σύνολο απουσιών | Σύνολο Απουσιών δικαιολογημένων | Σύνολο Απουσιών δικαιολογημένων λόγω ασθένειας | Σύνολο Απουσιών αδικαιολόγητων | |
|----|-------------|------------------|--------|---------|------------------|---------------------------------|------------------------------------------------|--------------------------------|------------------|
| 1 | Περίπτωση_3 | 16 | -- | 20 | Κοσμιότητα | 0 -- 161 | Χωρίς περιορισμό | 0 -- 58 | 0 -- 50 |
| 2 | Περίπτωση_2 | Χωρίς περιορισμό | Κοσμία | 0 -- 25 | Χωρίς περιορισμό | Χωρίς περιορισμό | Χωρίς περιορισμό | Χωρίς περιορισμό | |
| 3 | Περίπτωση_1 | Χωρίς περιορισμό | Κοσμία | 0 -- 51 | Χωρίς περιορισμό | Χωρίς περιορισμό | Χωρίς περιορισμό | Χωρίς περιορισμό | |
| 4 | Περίπτωση_4 | 1 | -- | 10 | Μέτρια | 0 -- 24 | Χωρίς περιορισμό | Χωρίς περιορισμό | Χωρίς περιορισμό |

Εικόνα 1

| Περιγραφή Συνθήκης | Αριθμός μαθητών στις οποίες εμπλέκεται |
|--------------------------------------|----------------------------------------|
| Περίπτωση_2 Μαθητές: Μαθητής_7 | 1 |
| Περίπτωση_1 Μαθητές: Μαθητής_7 | 1 |

Εικόνα 2

Οι εικόνες εικόνα1 και εικόνα2 που περιγράφουν στιγμιότυπα της εφαρμογής μας πληροφορούν ότι στο σύστημα υπάρχουν τέσσερις συνθήκες επάρκειας και εμφανίζεται το φαινόμενο πλεονασμού με μέτρο 2 (ο Μαθητής 7 χαρακτηρίζεται ως επαρκής και από δύο συνθήκες [Περίπτωση 1, Περίπτωση 2]).

Προτείνεται στο σύστημα η τροποποίηση γνώσης που περιγράφεται στην εικόνα που ακολουθεί:

Δα: 5

Όνομασία: Δοκιμή_1

Η ΤΙΜΗ -1 ΣΤΑ ΠΕΔΙΑ ΠΟΥ ΑΚΟΛΟΥΘΟΥΝ ΣΗΜΑΙΝΕΙ ΟΤΙ ΤΟ ΠΕΔΙΟ ΔΕΝ ΛΑΜΒΑΝΕΙ

Σύνολο απουσιών που λαμβάνονται υπόψη

Από: 0

Μέχρι: 30

Σύνολο απουσιών που μπορούν να δικαιολογηθούν

Από: -1

Μέχρι: -1

Σύνολο απουσιών που μπορούν να δικαιολογηθούν λόγω ασθένειας

Από: -1

Μέχρι: -1

Σύνολο απουσιών που είναι αδικαιολόγητες

Από: -1

Μέχρι: -1

Περιοχή βαθμολογίας

Από: -1

Μέχρι: -1

Διαγωγή: μέτρια

Εικόνα 3

Η εικόνα 3 περιγράφει την διαδικασία εντολής προς το Σύστημα Γνώσης να υλοποιήσει πρόταση αναδιοργάνωσης γνώσης στην περίπτωση που στο σύστημα προστίθεται η γνώση:

Νέα συνθήκη επάρκειας Δοκιμή_1 η οποία περιέχει τους εξής περιορισμούς:

- Σύνολο απουσιών μέχρι 30
- Διαγωγή μαθητή Μέτρια.

Η γνώση που παράγει το σύστημα περιγράφεται από την εικόνα που ακολουθεί:

| Κατάσταση συνθηκών που δημιουργούν Πλεονασμό | | | |
|------------------------------------------------------------------------------------------------------------------|--------------------|----------------------|--------------------------------------|
| Συνθήκη | Αριθμός Πλεονασμών | | |
| Δοκιμή_1 | 2 | | |
| κατάσταση υπολοίπων συνθηκών που εμπλέκονται και ελέγχεται η επίδραση της αποβολής τους: | | | |
| Συνθήκη | Νέα Κατάσταση | | |
| Περίπτωση_2 | Αριθμός Πλεονασμών | 0 | |
| Περίπτωση_1 | Αριθμός Πλεονασμών | 0 | |
| κατάσταση μαθητών που εμπλέκονται και ελέγχεται η επίδραση της αποβολής τους στο μέτρο Πλεονασμών: | | | |
| A. Μαθητή | Όνομα μαθητή | Αριθμός Πλεονασμών | |
| 7 | Μαθητής_7 | 0 | |
| κατάσταση μαθητών που εμπλέκονται και ελέγχεται η επίδραση της αλλαγής της διαγωγής τους στον αριθμό Πλεονασμών: | | | |
| A. Μαθητή | Όνομα μαθητή | προτεινόμενη διαγωγή | Αριθμός Πλεονασμών |
| 7 | Μαθητής_7 | κοσμία | Μεταβολή στον χαρακτηρισμό επάρκειας |
| 7 | Μαθητής_7 | κοσμιότατη | Μεταβολή στον χαρακτηρισμό επάρκειας |
| 7 | Μαθητής_7 | μέτρια | 0 |
| 7 | Μαθητής_7 | Κακή | Μεταβολή στον χαρακτηρισμό επάρκειας |

Εικόνα 4

Σύμφωνα με την εικόνα 4 που αποτελεί στιγμιότυπο της εφαρμογής με την προσθήκη της συνθήκης «Δοκιμή_1» το σύστημα παραμένει σε μέτρο πλεονασμού 2 και βέβαια προτείνει την αφαίρεση γνώσης είτε της συνθήκης Περίπτωση_2 είτε της συνθήκης Περίπτωση_1 είτε της

τροποποίησης της διαγωγής του μαθητή Μαθητή_7 σε «μέτρια» είτε της αποβολής του μαθητή «Μαθητής_7» ώστε το Σύστημα Γνώσης να ελαχιστοποιήσει το μέτρο πλεονασμού.

- Πρέπει να τονίσουμε ότι η αποβολή του μαθητή «Μαθητής_7» μπορεί να οδηγήσει σε περιορισμό της ορθότητας στη Γνώση σε σχέση όμως με την προηγούμενη κατάσταση και όχι με την τρέχουσα. Από την άλλη όμως απλοποιεί το σύστημα όσον αφορά την εναπομείνουσα γνώση. Επίσης, η πρόταση για αλλαγή της διαγωγής του μαθητή «Μαθητής_7» σε (κοσμία, κοσμιότητα ή κακή) παραβιάζει το κριτήριο της ορθότητας αφού πλέον μαθητής που κρινόταν επαρκής πλέον κρίνεται μη επαρκής.

Παράδειγμα 2^ο:

Έστω ότι προτείνεται στο υπάρχον Σύστημα Γνώσης η προσθήκη της συνθήκης που περιγράφεται στην εικόνα που ακολουθεί:

Νέα συνθήκη για τον χαρακτηρισμό επάρκειας φοίτησης

Αα: 5

Όνομασία: Δρκιμή_2

Η ΤΙΜΗ -1 ΣΤΑ ΠΕΔΙΑ ΠΟΥ ΑΚΟΛΟΥΘΟΥΝ ΣΗΜΑΪΝΕΙ ΌΤΙ ΤΟ ΠΕΔΙΟ ΔΕΝ ΛΑΜΒΑΝΕΤΑΙ ΥΠΟΨΗ

Σύνολο απουσιών που λαμβάνονται υπόψη

Από: -1

Μέχρι: -1

Σύνολο απουσιών που μπορούν να δικαιολογηθούν

Από: -1

Μέχρι: -1

Σύνολο απουσιών που μπορούν να δικαιολογηθούν λόγω ασθένειας

Από: -1

Μέχρι: -1

Σύνολο απουσιών που είναι αδικαιολόγητες

Από: -1

Μέχρι: -1

Περιοχή βαθμολογίας

Από: -1

Μέχρι: -1

Διαγωγή: κοσμία

Ok Cancel

Εικόνα 5

Η εικόνα 5 αναφέρεται σε συνθήκη επάρκειας Δοκιμή_2 κα μοναδικό περιορισμό:

- Διαγωγή κοσμία

Η γνώση που παράγει το σύστημα περιγράφεται από την εικόνα που ακολουθεί:

| Κατάσταση συνθηκών που δημιουργούν Πλεονασμό | | | |
|------------------------------------------------------------------------------------------------------------------|--------------------|----------------------|----------------------------------------|
| Συνθήκη | Αριθμός Πλεονασμών | | |
| Δοκιμή_2 | 3 | | |
| κατάσταση υπολοίπων συνθηκών που εμπλέκονται και ελέγχεται η επίδραση της αποβολής τους: | | | |
| Συνθήκη | Νέα Κατάσταση | | |
| Περίπτωση_2 | Αριθμός Πλεονασμών | 2 | |
| Περίπτωση_1 | Αριθμός Πλεονασμών | 2 | |
| κατάσταση μαθητών που εμπλέκονται και ελέγχεται η επίδραση της αποβολής τους στο μέτρο Πλεονασμών: | | | |
| A. Μαθητή | Όνομα μαθητή | Αριθμός Πλεονασμών | |
| 7 | Μαθητής_7 | 0 | |
| κατάσταση μαθητών που εμπλέκονται και ελέγχεται η επίδραση της αλλαγής της διαγωγής τους στον αριθμό Πλεονασμών: | | | |
| A. Μαθητή | Όνομα μαθητή | προτεινόμενη διαγωγή | Αριθμός Πλεονασμών |
| 7 | Μαθητής_7 | κοσμιότητα | 0 |
| 7 | Μαθητής_7 | μέτρια | Μεταβολή στον χαρακτηρισμό επάρκειας |
| 7 | Μαθητής 7 | κακή | Μεταβολή στον γαοακτοποισιού επάρκειας |

Εικόνα 6

Σύμφωνα με την εικόνα 6 που αποτελεί στιγμιότυπο της εφαρμογής με την προσθήκη της συνθήκης δοκιμή_2 το σύστημα εμφανίζει πλέον μέτρο πλεονασμού 3 και βέβαια προτείνει την αφαίρεση γνώσης είτε της συνθήκης «Περίπτωση_2» είτε της συνθήκης «Περίπτωση_1» είτε τροποποίηση της διαγωγής του σε κοσμιότητα είτε αφαίρεση του μαθητή από το Σύστημα Γνώσης.

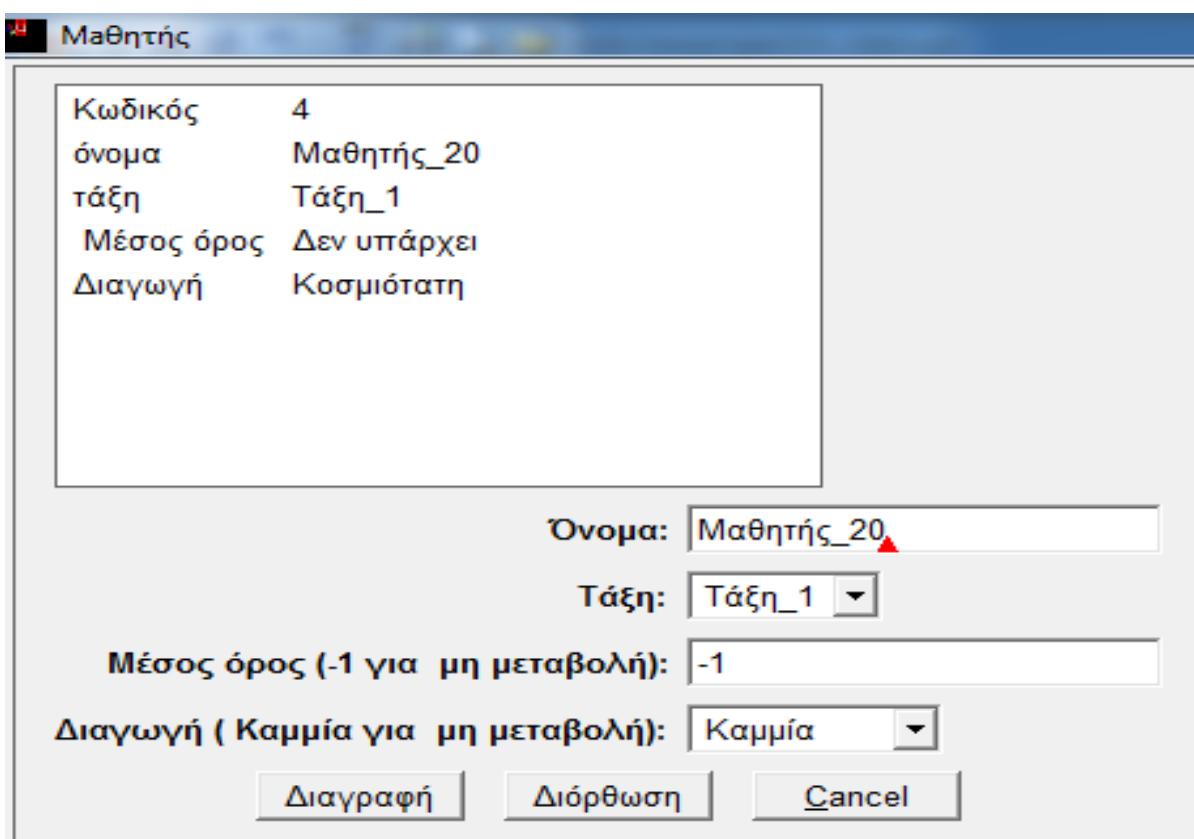
Πρέπει να τονίσουμε ότι η αφαίρεση της γνώσης που αφορά τον μαθητή «Μαθητής_7» ουσιαστικά δημιουργεί αναξιόπιστη γνώση αφού ο μαθητής_7 πλέον δεν αποτελεί μέρος της

γνώσης αλλά όσον αφορά την κατάσταση του συστήματος αυτό πληροί πλέον τα κριτήρια ορθότητας.

- Το ερώτημα που τίθεται είναι αν το τελικό αποτέλεσμα είναι καλύτερο δηλαδή αν η επιλεκτική αφαίρεση κάποιας γνώσης από το σύστημα η οποία ναι μεν θα μεταβάλλει κάποιες πεποιθήσεις αλλά θα οδηγήσει το σύστημα σε μια καλύτερη κατάσταση όσον αφορά κάποια κριτήρια που έχουν τεθεί είναι προτιμότερη από την επιλογή να αφήσουμε το σύστημα στην νέα κατάσταση όπου κάποιο ή κάποια κριτήρια που έχουμε θέσει δεν πληρούνται.

Παράδειγμα 3^ο:

Ακολουθεί ένα άλλο στιγμιότυπο (Εικόνα 7) της εφαρμογής:



The screenshot shows a window titled "Μαθητής" (Student) with the following content:

| | |
|------------|-------------|
| Κωδικός | 4 |
| όνομα | Μαθητής_20 |
| τάξη | Τάξη_1 |
| Μέσος όρος | Δεν υπάρχει |
| Διαγωγή | Κοσμιότητα |

Below the table, there are input fields and dropdown menus:

- Όνομα: Μαθητής_20
- Τάξη: Τάξη_1
- Μέσος όρος (-1 για μη μεταβολή): -1
- Διαγωγή (Καμμία για μη μεταβολή): Καμμία

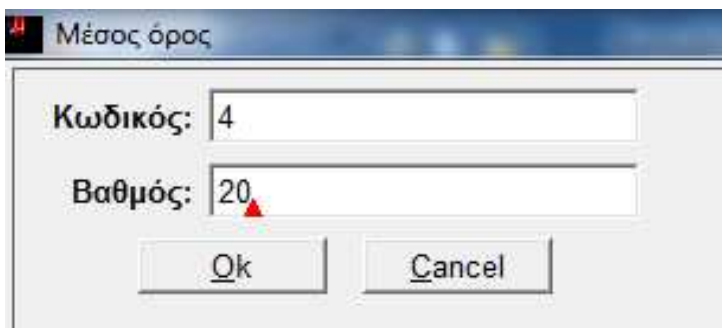
At the bottom, there are three buttons: Διαγραφή, Διόρθωση, and Cancel.

Εικόνα 7

Σύμφωνα με την εικόνα 7 ο μαθητής_20 έχει ελλιπή στοιχεία (δεν του έχει αποδοθεί μέσος όρος)

Θα χρησιμοποιήσουμε το σύστημα για τον έλεγχο αναθεώρησης γνώσης «Όρισε βαθμό στον μαθητή_20».

Η διαδικασία περιγράφεται από την οθόνη:



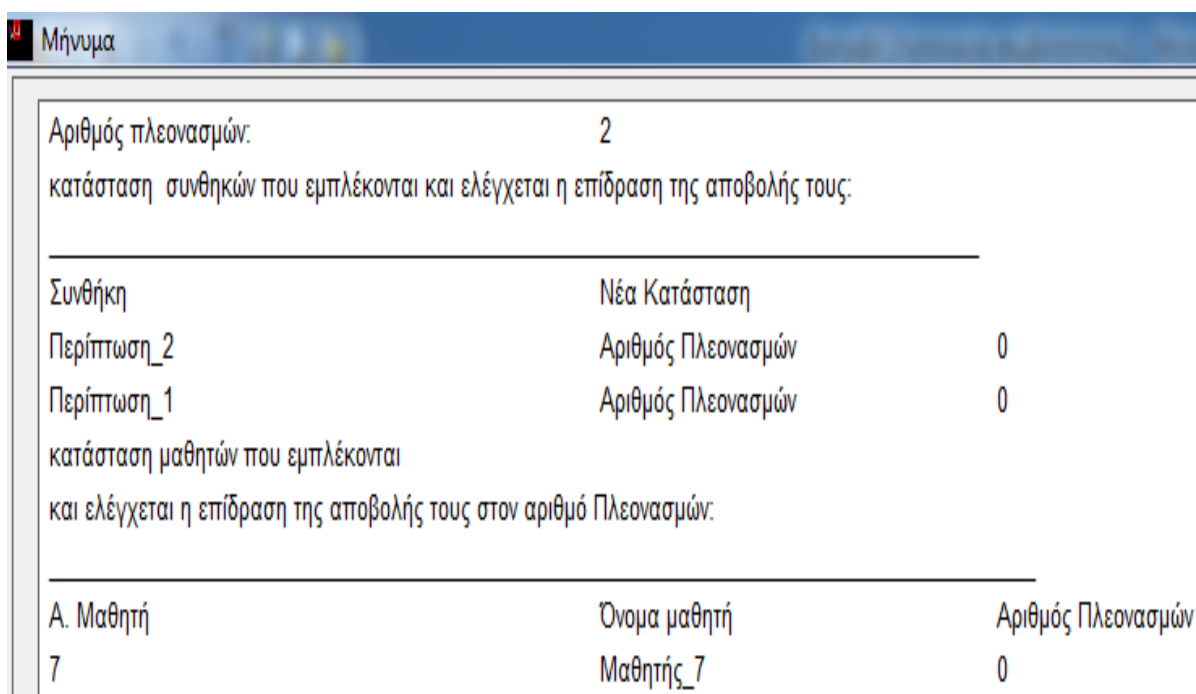
Μέσος όρος

Κωδικός: 4

Βαθμός: 20

Ok Cancel

Το αποτέλεσμα περιγράφεται στο στιγμιότυπο που ακολουθεί:



Μήνυμα

Αριθμός πλεονασμών: 2

κατάσταση συνθηκών που εμπλέκονται και ελέγχεται η επίδραση της αποβολής τους:

| Συνθήκη | Νέα Κατάσταση | |
|-------------|--------------------|---|
| Περίπτωση_2 | Αριθμός Πλεονασμών | 0 |
| Περίπτωση_1 | Αριθμός Πλεονασμών | 0 |

κατάσταση μαθητών που εμπλέκονται και ελέγχεται η επίδραση της αποβολής τους στον αριθμό Πλεονασμών:

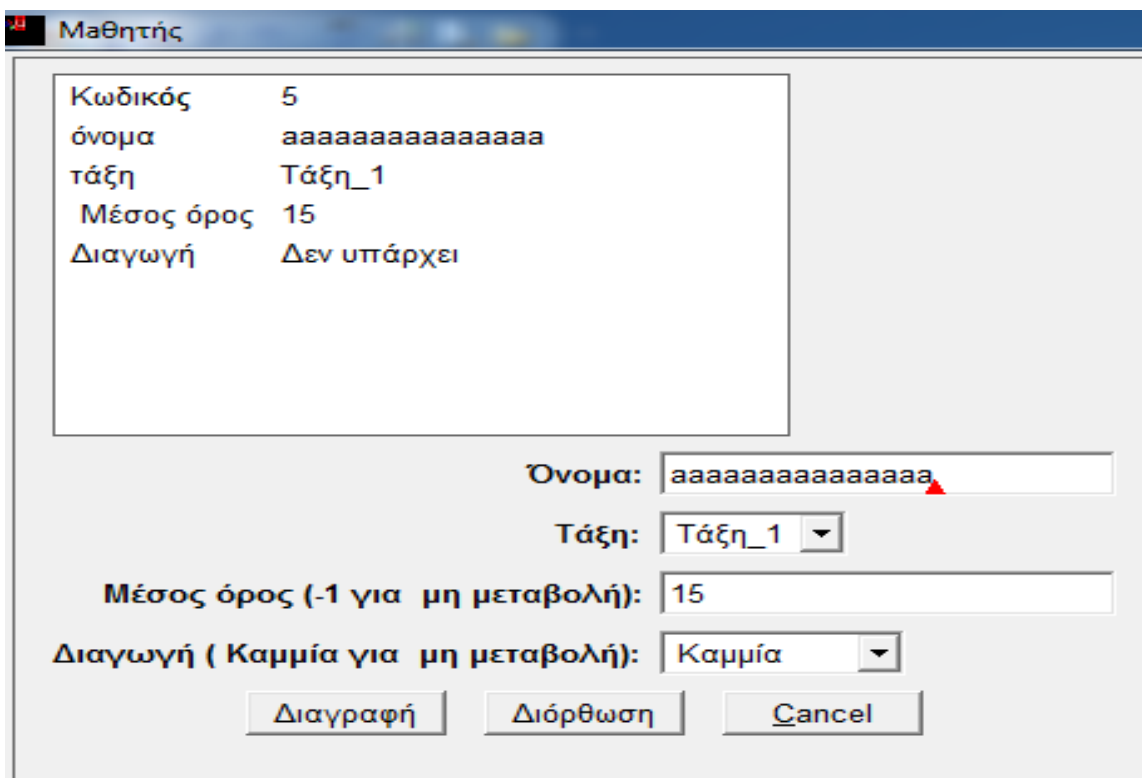
| A. Μαθητή | Όνομα μαθητή | Αριθμός Πλεονασμών |
|-----------|--------------|--------------------|
| 7 | Μαθητής_7 | 0 |

Εικόνα 8

Στην εικόνα 8 ουσιαστικά δεν υπάρχει μεταβολή στην κατάσταση του συστήματος γνώσης και ισχύουν ότι περιγράφεται στο παράδειγμα 2.

Παράδειγμα 4^ο :

Ακολουθεί ένα άλλο στιγμιότυπο (Εικόνα 9) της εφαρμογής:



The screenshot shows a window titled "Μαθητής" (Student) with a list of student records and a form for editing. The list contains the following data:

| | |
|------------|--------------------|
| Κωδικός | 5 |
| όνομα | αααααααααααααααααα |
| τάξη | Τάξη_1 |
| Μέσος όρος | 15 |
| Διαγωγή | Δεν υπάρχει |

Below the list, there are input fields for editing the student's information:

- Όνομα: αααααααααααααααααα
- Τάξη: Τάξη_1
- Μέσος όρος (-1 για μη μεταβολή): 15
- Διαγωγή (Καμμία για μη μεταβολή): Καμμία

At the bottom, there are three buttons: Διαγραφή (Delete), Διόρθωση (Edit), and Cancel.

Εικόνα 9

Σύμφωνα με την εικόνα 9 ο μαθητής «αααααααααααααααααα» έχει ελλιπή στοιχεία (δεν του έχει αποδοθεί διαγωγή)

Θα χρησιμοποιήσουμε το σύστημα για τον έλεγχο αναθεώρησης γνώσης «Όρισε Διαγωγή στον αααααααααααααααααα».

Η διαδικασία περιγράφεται από την οθόνη:

Διαγωγή μαθητή

Κωδικός:

Διαγωγή:

Το αποτέλεσμα περιγράφεται στο στιγμιότυπο που ακολουθεί:

Μήνυμα

Αριθμός πλεονασμών: 4
κατάσταση συνθηκών που εμπλέκονται και ελέγχεται η επίδραση της αποβολής τους:

| Συνθήκη | Νέα Κατάσταση | |
|-------------|--------------------|---|
| Περίπτωση_2 | Αριθμός Πλεονασμών | 0 |
| Περίπτωση_1 | Αριθμός Πλεονασμών | 0 |

κατάσταση μαθητών που εμπλέκονται και ελέγχεται η επίδραση της αποβολής τους στον αριθμό Πλεονασμών:

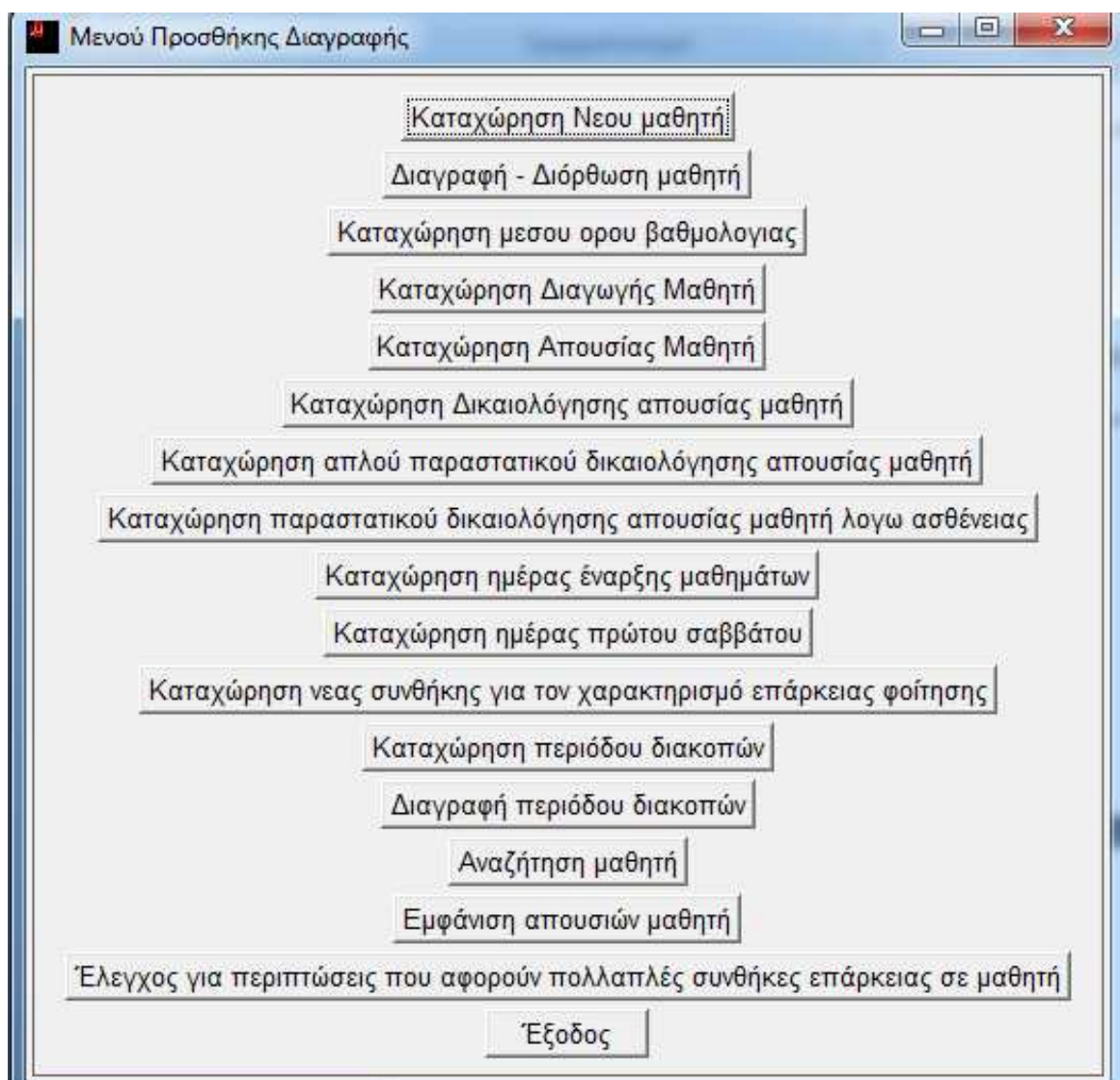
| A. Μαθητή | Όνομα μαθητή | Αριθμός Πλεονασμών |
|-----------|--------------|--------------------|
| 7 | Μαθητής_7 | 2 |

Εικόνα 10

Στην εικόνα 10 ουσιαστικά έχουμε μεταβολή του μέτρου πλεονασμού αυξήθηκε σε τέσσερα και ουσιαστικά η αφαίρεση μια εκ των συνθηκών «Περίπτωση_1» και «Περίπτωση_2» οδηγεί σε μηδενισμό του κριτηρίου «μέτρο πλεονασμού».

2.6.4 Μεταβολές στην Γνώση

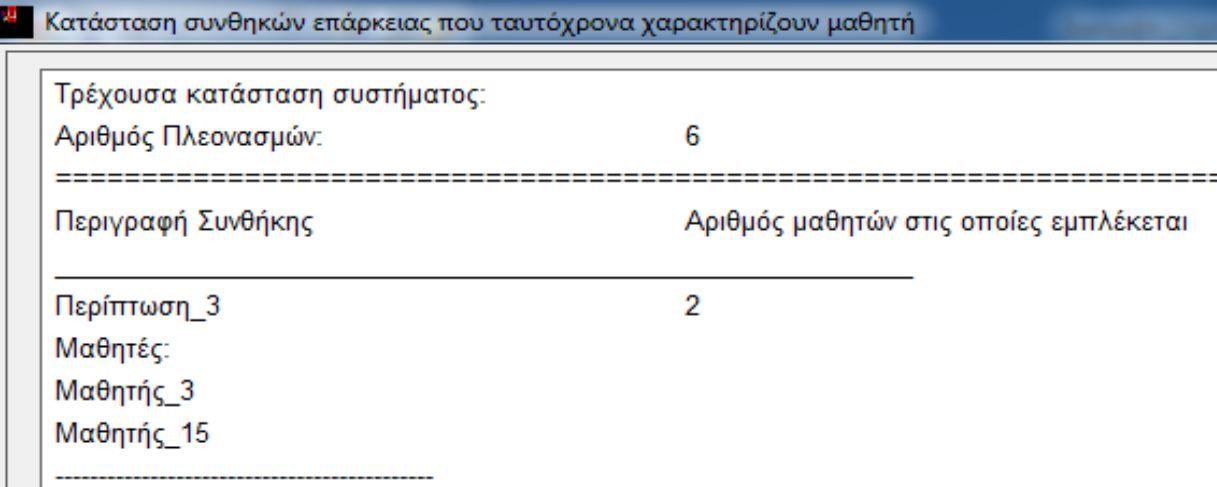
Με την επιλογή **Μεταβολές στη Γνώση** έχουμε την δυνατότητα να μεταβάλλουμε την υπάρχουσα γνώση του συστήματος μέσω του Μενού προσθήκης - διαγραφής όπου υλοποιούνται βασικές μεταβολές γνώσης πάνω στο σύστημα. Μέσα από το συγκεκριμένο μενού το Σύστημα Γνώσης υπόκειται τροποποίηση γνώσης με στόχο την τήρηση του 1^{ου} κριτηρίου δηλαδή την ορθότητα όσον αφορά τον συμπερασμό για την επάρκεια φοίτησης κάποιου μαθητή αλλά χωρίς να έχουμε προτάσεις για εμφάνιση πλεονασμών και απλοποίηση του συστήματος. Ο λόγος ουσιαστικά αυτής της επιλογής είναι η δυνατότητα που πρέπει να έχουμε στην προσαρμογή υπάρχουσας γνώσης στην οποία μπορεί ήδη να υπάρχουν προβλήματα μη ορθότητας ή πλεονασμών στο νέο σύστημα που αναπτύσσεται. Είναι δουλειά μετά του νέου συστήματος να προχωρήσει σε αναδιοργάνωση της γνώσης για την βελτιστοποίησή του.



Εικόνα 11,

Στην εικόνα 11 που αποτελεί στιγμιότυπο της εφαρμογής επισημαίνουμε τις επιλογές:

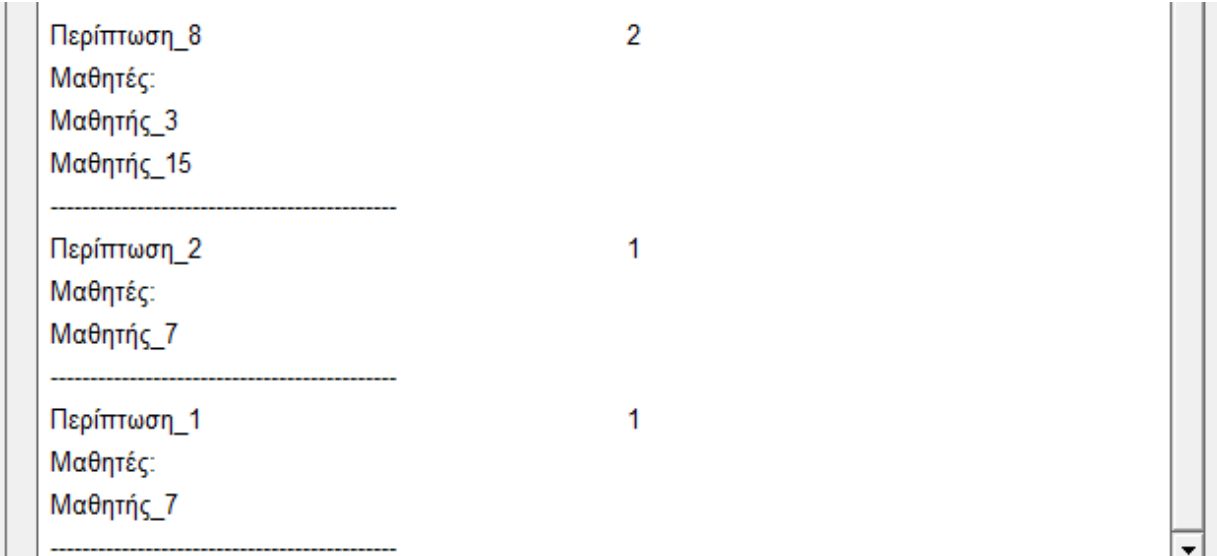
- Έλεγχος για περιπτώσεις που αφορούν πολλαπλές συνθήκες επάρκειας σε μαθητή η οποία ουσιαστικά παράγει γνώση όσον αφορά τα φαινόμενα πλεονασμού που υπάρχουν στο σύστημα . Οι εικόνες που ακολουθούν περιγράφουν ένα παράδειγμα αυτής της αποτύπωσης :



Κατάσταση συνθηκών επάρκειας που ταυτόχρονα χαρακτηρίζουν μαθητή

| Τρέχουσα κατάσταση συστήματος: | |
|--------------------------------|----------------------------------------|
| Αριθμός Πλεονασμών: | 6 |
| ===== | |
| Περιγραφή Συνθήκης | Αριθμός μαθητών στις οποίες εμπλέκεται |
| Περίπτωση_3 | 2 |
| Μαθητές: | |
| Μαθητής_3 | |
| Μαθητής_15 | |
| ----- | |

Εικόνα 12



| | |
|-------------|---|
| Περίπτωση_8 | 2 |
| Μαθητές: | |
| Μαθητής_3 | |
| Μαθητής_15 | |
| ----- | |
| Περίπτωση_2 | 1 |
| Μαθητές: | |
| Μαθητής_7 | |
| ----- | |
| Περίπτωση_1 | 1 |
| Μαθητές: | |
| Μαθητής_7 | |
| ----- | |

Εικόνα 13

Οι εικόνες 12 και 13 ουσιαστικά περιγράφουν την Γνώση: ότι στο συγκεκριμένο σύστημα το μέτρο πλεονασμού είναι 6 αφού σε δύο μαθητές «Μαθητής_3», «Μαθητής_15» έχουμε χαρακτηρισμό φοίτησης επάρκειας από τις συνθήκες «Περίπτωση_3» και «Περίπτωση_16» ενώ ο μαθητής «Μαθητής_7» χαρακτηρίζεται ως επαρκής από τις συνθήκες «Περίπτωση_2» και «Περίπτωση_1»

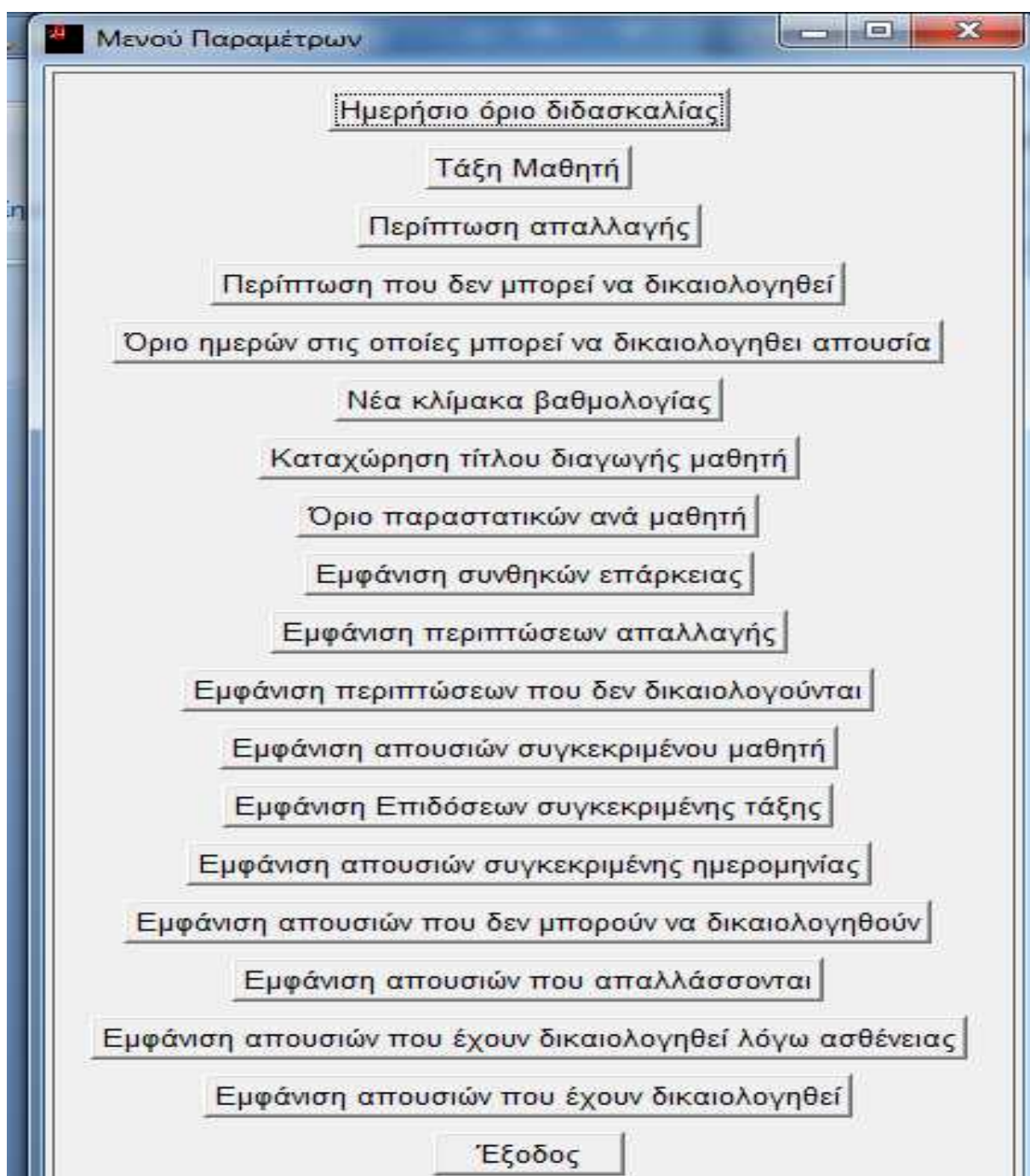
- **Εμφάνιση απουσιών μαθητή** η οποία εμφανίζει την γνώση για τις απουσίες ενός μαθητή:

| Κατάσταση Απουσιών | | | | | | | |
|--------------------|--------------|------|---|-------------------|------------|---------------|-------------|
| Όνομα μαθητή | Μαθητής_3 | Ωρες | | Ημ. δικαιολόγησης | Αιτιολογία | | παραστατικό |
| Αα | Ημ. απουσίας | | | | | | |
| 1 | febroyarios | 6 | 2 | febroyarios | 7 | Αδικαιολόγητη | δεν υπάρχει |
| 2 | martios | 7 | 4 | | | | |

Στο στιγμιότυπο της εφαρμογής που περιγράφει η πιο πάνω εικόνα μπορούμε να διακρίνουμε μια απουσία (Febroyarios 6) που έχει αξιολογηθεί ως αδικαιολόγητη και μια (Martios 7) η οποία δεν έχει αξιολογηθεί δηλαδή δεν έχει χαρακτηριστεί ακόμη.

2.6.5 Βοηθητικές Λειτουργίες

Η επιλογή **Βοηθητικές Λειτουργίες** ουσιαστικά αποτελεί ένα σύνολο από βοηθητικές εργασίες. Κάποιες Αφορούν μεταβολή στη Γνώση που αναπαρίσταται και κάποιες αφορούν συμπεράσματα. Στη συγκεκριμένη περιοχή της εφαρμογής δεν μπορεί να προκληθεί αναθεώρηση γνώσης η οποία θα οδηγήσει σε παραβίαση του κριτηρίου της ορθότητας. Μπορούμε να πούμε ότι στη συγκεκριμένη περιοχή οι μεταβολές που υφίσταται το Σύστημα Γνώσης δεν μπορούν να προκαλέσουν «υποβάθμιση» της ορθότητας και αξιοπιστίας όπως αυτές περιγράφονται από τα κριτήρια που έχουμε ορίσει. Ακολουθεί εικόνα σχετική με την συγκεκριμένη περιοχή:



Ενδεικτικά αναφέρουμε ότι κάποιες επιλογές οδηγούν σε περαιτέρω αλληλεπίδραση όπως για παράδειγμα η επιλογή η οποία οδηγεί σε μια αλληλεπίδραση όπου υπάρχει η δυνατότητα προβολής ή διαγραφής μιας συνθήκης επάρκειας.

Ακολουθεί ένα στιγμιότυπο αυτής της αλληλεπίδρασης:

Συνθήκη για τον χαρακτηρισμό επάρκειας φοίτησης

Αα: 1

Όνομασία: Περίπτωση_3

Σύνολο απουσιών που λαμβάνονται υπόψη

Από: 0

Μέχρι: 161

Σύνολο απουσιών που μπορούν να δικαιολογηθούν λόγω ασθένειας

Από: 0

Μέχρι: 58

Σύνολο απουσιών που είναι αδικαιολόγητες

Από: 0

Μέχρι: 50

Περιοχή βαθμολογίας

Από: 16

Μέχρι: 20

Διαγωγή: κοσμιότητα

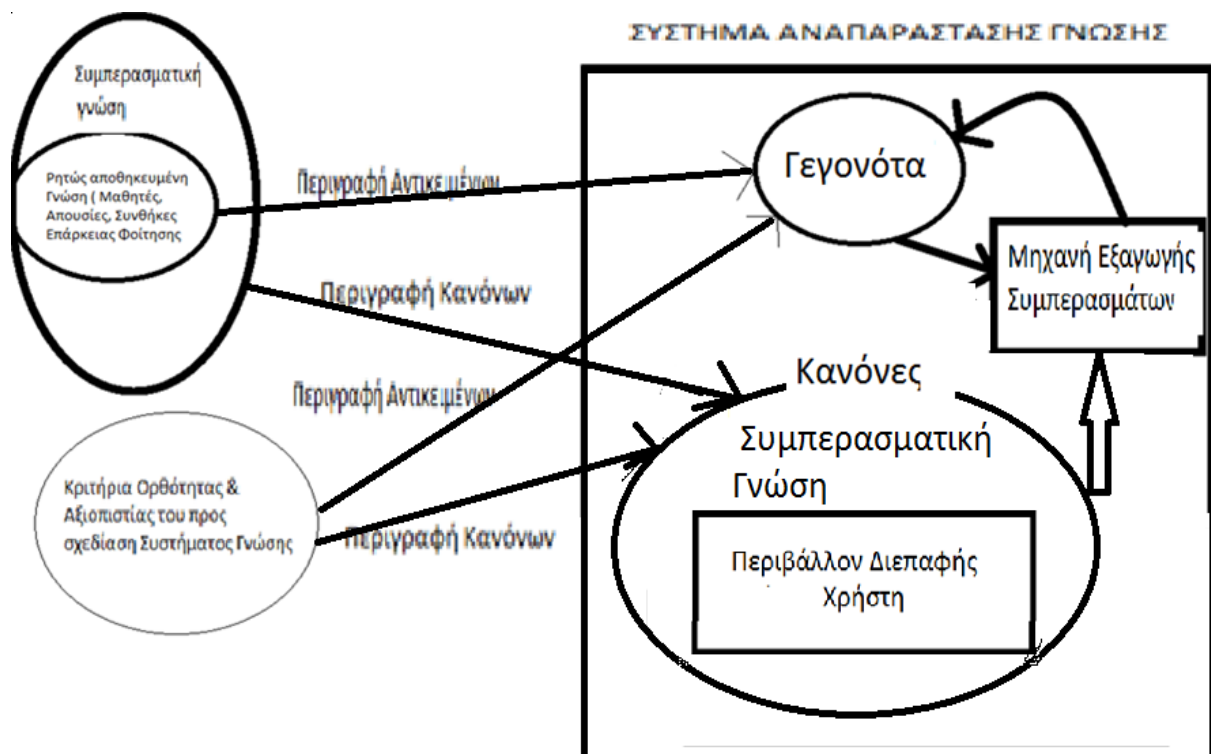
Επόμενη Διαγραφή Έξοδος

Εικόνα 14

Στην εικόνα 14 έχουμε το συμπέρασμα για την συνθήκη «Περίπτωση_3» και τις προδιαγραφές της αλλά επίσης υπάρχει η δυνατότητα αποβολής της από το Σύστημα Γνώσης. Πρέπει να παρατηρήσουμε ότι αυτό θεωρείται αντίφαση με το γεγονός ότι σε αυτή την περιοχή της εφαρμογής δεν συντελείται αναθεώρηση πεποιθήσεων αφού υπάρχει η περίπτωση η αφαίρεση μιας συνθήκης να οδηγήσει ένα μαθητή που πριν ήταν επαρκής σε μη επάρκεια δηλαδή η γνώση «άλλαξε». Αυτό που πρέπει να τονιστεί όμως είναι ότι ορθώς άλλαξε σε σχέση με την τρέχουσα κατάσταση του συστήματος στο οποίο η συγκεκριμένη συνθήκη επάρκειας δεν αποτελεί μέρος της γνώσης του και ο μαθητής κρίνεται πλέον ανεπαρκής.

2.7 Συνοψίζοντας.

Συνοψίζοντας μπορούμε να πούμε ότι η συγκεκριμένη εργασία της οποίας η μεθοδολογία αποτυπώνεται στο γράφημα που ακολουθεί:



αποτελέσει μια προσπάθεια αναπαράστασης σε τυπική μορφή Γνώσης ενός πλαισίου νόμου και της διαδικασίας εφαρμογής του. Αφού ταξινομήθηκε αρχικά η ρητή γνώση του συστήματος η οποία αναπαραστάθηκε σε γεγονότα της γλώσσας Prolog στη συνέχεια αποτυπώθηκε η συμπερασματική Γνώση ή οποία αναπαραστάθηκε σε κανόνες Prolog. Στην συνέχεια διατυπώθηκαν κάποια κριτήρια τα οποία ουσιαστικά θα περιγράφουν την «κατάσταση» του προς υλοποίηση συστήματος τα οποία ενσωματώθηκαν ως γνώση κανόνων και γεγονότων στο σύστημα. Στο τέλος, αναπτύχθηκε ένα γραφικό περιβάλλον επικοινωνίας (XPCE) μέσω του οποίου ο χρήστης έχει πρόσβαση στην γνώση του συστήματος.

Το στοιχείο που ουσιαστικά διαφοροποιεί την συγκεκριμένη προσπάθεια από μια κλασική εφαρμογή Βάσεων Δεδομένων είναι η δυνατότητα αναδιοργάνωσης της γνώσης που υλοποιείται στα πλαίσια κάποιων ποιοτικών κριτηρίων που αφορούν κυρίως όχι την ορθότητά της αλλά την «βελτίωσή» της όσον αφορά φαινόμενα πλεονασμού ή αντιφάσεων που εμφανίζονται στην παραγόμενη πληροφορία.

Συμπεράσματα

Στην παρούσα εργασία έγινε προσπάθεια να αντιμετωπισθεί το πρόβλημα της αναπαράστασης νομοθεσίας σε μορφή τυπικής γλώσσας. Το εργαλείο που χρησιμοποιήθηκε ήταν η Prolog ως γλώσσα τυπικής αναπαράστασης γνώσης.

Οι στόχοι ήταν:

- i. Να εξευρεθεί ένα πεδίο νόμων πάνω στο οποίο να αναπτυχθεί ένα Σύστημα Γνώσης με την βοήθεια υπολογιστικής μηχανής και τυπικής αναπαράστασης γνώσης.
- ii. Να τεθούν πάνω στο σύστημα κάποιες προδιαγραφές τις οποίες το σύστημα θα πρέπει να υποστηρίζει για να θεωρείται αξιόπιστο. Οι συγκεκριμένες προδιαγραφές έχουν να κάνουν κυρίως με ζητήματα απόδοσης και ορθότητας τα οποία προκύπτουν όταν το σύστημα υφίσταται αναδιοργάνωση γνώσης. Το σύστημα τότε λόγω των κριτηρίων που έχουν τεθεί (προδιαγραφές) πρέπει να «αντιδράσει» προτείνοντας πλαίσιο αλλαγών στην γνώση του είτε προχωρώντας μέσω αυτοματοποιημένης διαδικασίας στις προτεινόμενες μεταβολές
- iii. Να υλοποιηθεί το Σύστημα Γνώσης και να δοκιμαστεί όσον αφορά την αξιοπιστία του.
- iv. Να αξιολογηθεί η συνεισφορά του στη συγκεκριμένη εφαρμογή.

Η ανταπόκριση στους στόχους ήταν η εξής:

- i. Επιλέχθηκε το νομικό πλαίσιο που διέπει τον χαρακτηρισμό φοίτησης μαθητών δευτεροβάθμιας εκπαίδευσης. Η επιλογή έγινε διότι πιστεύω ότι αποτελεί ένα αντιπροσωπευτικό δείγμα νομοθεσίας η οποία μπορεί να αναπαρασταθεί σε τυπική μορφή και στην οποία εν δυνάμει συμβαίνουν συχνά μεταβολές οι οποίες θεωρητικά μπορούν να οδηγήσουν σε φαινόμενα πλεονασμού.
- ii. Το Σύστημα Γνώσης υποστηρίζει κυρίως την παραγόμενη γνώση για τον χαρακτηρισμό φοίτησης μαθητών. Θεωρώντας όμως ότι στο περιβάλλον του συμβαίνουν συχνά αλλαγές οι οποίες γενικά μπορεί να επηρεάσουν την απόδοσή (π.χ. φαινόμενα πλεονασμών), έχουμε επιπλέον ορίσει ένα κριτήριο που αφορά τον περιορισμό του φαινομένου των πλεονασμών στο σύστημα. Ως πλεονασμό ορίσαμε την κατάσταση

στην οποία μαθητής χαρακτηρίζεται επαρκής από πολλές συνθήκες επάρκειας και ορίσαμε ως μέτρο του τον αριθμό εμφανίσεων ζευγαριών (μαθητής, συνθήκη) που εμπλέκονται σε καταστάσεις πλεονασμού.

- iii. Το σύστημα υλοποιήθηκε και παράγει αξιόπιστη γνώση όσον αφορά τον χαρακτηρισμό φοίτησης. Επίσης το σύστημα μπορεί να αξιολογήσει το φαινόμενο πλεονασμού και να προτείνει «λύσεις» που περιλαμβάνουν αναδιοργάνωση γνώσης με στόχο την βελτίωση των κριτηρίων που τέθηκαν.

Ένα γενικό συμπέρασμα όσον αφορά τη συγκεκριμένη εφαρμογή περιέχει δυο σκέλη:

- Όσον αφορά την υλοποίηση του συστήματος για παραγωγή γνώσης σχετικά με την επάρκεια φοίτησης το έργο με εργαλείο μια γλώσσα τεχνητής ευφυΐας είναι πιο δύσκολο σε σχέση με την επιλογή ενός αλγοριθμικού εργαλείου και μιας σχεσιακής βάσης δεδομένων.
- Όσον αφορά την λειτουργία γενικά της αναθεώρησης πεποιθήσεων και ειδικά της τροποποίησης της γνώσης όπου το ίδιο το σύστημα πρέπει να «αναδιοργανώνεται» και να βελτιώνεται όπως για παράδειγμα με το να απλοποιείται, το έργο υλοποιείται πιο εύκολα με γλώσσες τυπικής αναπαράστασης γνώσης.

Προβληματισμοί

Αυτό που θέλω να επισημάνω είναι ότι η τυπική αναπαράσταση νομοθεσίας με εργαλεία τεχνητής ευφυΐας μπορεί να βοηθήσει στην απλοποίηση αλλά και βελτίωση της ορθότητας και αξιοπιστίας στα νομικά συστήματα γνώσης. Είναι γεγονός ότι το φαινόμενο των επικαλυπτόμενων νόμων και αρμοδιοτήτων αλλά και πολλές φορές των ανενεργών είναι συχνό και πάντα η ύπαρξη ενός εργαλείου που θα βοηθούσε στην απομάκρυνση αυτού του φαινομένου θα ήταν ευπρόσδεκτη. Αυτό όμως που αποτελεί ζήτημα σημαντικό είναι η θέσπιση των κατάλληλων κριτηρίων που πρέπει να ορισθούν και να ενσωματωθούν στο Σύστημα Γνώσης πέρα από την ρητώς αποθηκευμένη γνώση που αφορά τους νόμους και τις ερμηνείες τους.

Επίσης σε ένα γενικότερο πλαίσιο μελέτης θα μπορούσαμε να πούμε ότι βάσει των κριτηρίων που θεσπίζουμε πρέπει να ταξινομούμε τις περιοχές του συστήματος γνώσης που επηρεάζονται (μέσω αναδιοργάνωσης) σε επίπεδα σπουδαιότητας. Έτσι η εφαρμογή ενός κριτηρίου πρέπει πάντα να επηρεάζει πρώτα την περιοχή του συστήματος γνώσης με την χαμηλότερη σπουδαιότητα. Ακολουθεί ένα παράδειγμα:

Στην εφαρμογή που έχω αναπτύξει και στο στιγμιότυπο που περιγράφεται στην εικόνα 4 (σελ. 61) ουσιαστικά έχουμε τέσσερις διαφορετικές προτάσεις όσον αφορά το κριτήριο του πλεονασμού και τον μηδενισμό του μέτρου:

- I. Διαγραφή του μαθητή «Μαθητής_7»,
- II. Τροποποίηση της διαγωγής του μαθητή «Μαθητής_7», σε «μέτρια».
- III. Διαγραφή της συνθήκης επάρκειας φοίτησης «Περίπτωση_1»,
- IV. Διαγραφή της συνθήκης επάρκειας φοίτησης «Περίπτωση_1».

Η διαγραφή μαθητή είναι πολύ πιο «σπουδαίο» γεγονός για το Σύστημα Γνώσης από την μεταβολή της διαγωγής του χωρίς να μεταβληθεί ο χαρακτηρισμός επάρκειάς του και σίγουρα από την διαγραφή κάποιας από τις δυο συνθήκες επάρκειας για τις οποίες δεν έχουμε καμία ουσιαστική μεταβολή στην εναπομείνασα γνώση του συστήματος.

Μπορούμε έτσι να συμπεράνουμε ότι για το κριτήριο πλεονασμού μεγαλύτερη σπουδαιότητα παρουσιάζει η οντότητα μαθητής και μικρότερη η συνθήκη επάρκειας φοίτησης.

Τέλος, θα μπορούσαμε να προτείνουμε το σύστημα να προχωρά μόνο του σε αναδιοργάνωση γνώσης με βάσει τα συγκεκριμένα κριτήρια ιεραρχώντας κριτήρια ποιότητας και περιοχές γνώσης που επηρεάζονται.

Γενικά μπορούμε να πούμε ότι μέσα από την συγκεκριμένη εφαρμογή διαπιστώνεται ότι σίγουρα η τεχνητή ευφυΐα και τα συστήματα αναπαράστασης γνώσης μπορούν να βοηθήσουν στην βελτίωση των υπάρχοντων νομικών πλαισίων μέσω απλοποιήσεων και γενικά παρεμβάσεων με πεδίο εφαρμογής την αναθεώρηση πεποιθήσεων (belief revision). Είναι σίγουρο ότι ένα πλήρως αυτοματοποιημένο σύστημα είναι κάτι αρκετά δύσκολο να υλοποιηθεί κυρίως λόγω της πολυπλοκότητας και της συχνής μεταβλητότητας που παρουσιάζει. Μπορεί

επίσης να τεθεί και το ζήτημα της «αναδιοργάνωσης» και των ίδιων των κριτηρίων που παρακολουθεί το σύστημα για να «βελτιώνεται».

Βιβλιογραφία

- [01]. (Iyad Rahwan, Guillermo R. Simari), "Argumentation in Artificial Intelligence", Springer-Verlag New York Incm 2009.
- [02]. (Thomas F. Gordon, Douglas Walton), "Legal Reasoning with Argumentation Schemes", Proceedings of the 12th International Conference on Artificial Intelligence, 2009
- [03]. (Jamal Bentahar , Bernard Moulin , Micheline Bélanger), "A taxonomy of argumentation models used for knowledge representation", Artificial Intelligence Review, 2010 - Springer
- [04]. (Ulrich Reimer, Andreas Margelisc), "A Hybrid Knowledge Representation Approach to Reusability of Legal Knowledge Bases", Proceeding ICAIL '95 Proceedings of the 5th international conference on Artificial intelligence and law
- [05]. (TREVOR BENCH-CAPON, HENRY PRAKKEN), "Introducing the Logic and Law Corner", Journal of logic and computation, 2008 - Oxford Univ Press
- [06]. (Robert Kozualski), "THE TREATMENT OF NEGATION IN LOGIC PROGRAMS FOR REPRESENTING LEGISLATION", Proceeding ICAIL '89 Proceedings of the 2nd international conference on Artificial intelligence and law
- [07]. (Ken Satoh, Kento Asai, Takamune Kogawa, Masahiro Kubota, Megumi Nakamura, Yoshiaki Nishigai, Kei Shirakawa, and Chiaki Takano), "PROLEG: An Implementation of the Presupposed Ultimate Fact Theory of Japanese Civil Code by PROLOG Technology", New Frontiers in Artificial Intelligence Lecture Notes in Computer Science, 2011, Volume 6797/2011, 153-164, DOI: 10.1007/978-3-642-25655-4_14
- [08]. (Guido Boella, Guido Governatori, Antonino Rotolo, Leendert van der Torre), "A Logical Understanding of Legal Interpretation", Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2010)
- [09]. (Giovanni Sartor), "Defeasibility in Legal Reasoning", EUI working paper LAW 2009/02.

- [10]. (Revision Raymundo), "Models for Belief", *Philosophical Issues* Vol. 2, Rationality in Epistemology (1992), pp. 227-247
- [11]. (Guido Boella, Guido Governatori, Antonino Rotolo, Leendert van der Torre), "A formal study on legal compliance and interpretation", 2010
- [12]. (Monica Palmirani, Guido Governatori, Giuseppe Contissa)," Modelling Temporal Legal Rules", Proceedings of the 13th International Conference on Artificial Intelligence and Law. June 6–10 2011, Pittsburgh, PA, USA.
- [13]. (Alexander Boer)," Legal Theory, Sources of Law & the Semantic Web", IOS Press Nieuwe Hemweg 6B 1013 BG Amsterdam Netherlands ISBN 978-1-60750-003-2 Library of Congress Control Number 2009924846
- [14]. (Laurens Mommers Z Wim Voermans Z Wouter Koelewijn Z Hugo Kielman), "Understanding the law: improving legal knowledge dissemination by translating the contents of formal sources of law", Springer Science+Business Media B.V. 2009.
- [15]. (Jan Wielemaker and Anjo Anjewierden), "An Architecture for Making Object-Oriented Systems Available from Prolog", Jan Wielemaker and Anjo Anjewierden Social Science Informatics (SWI), University of Amsterdam, August 2, 2002

Λέξεις Κλειδιά

Αναπαράσταση Γνώσης (Knowledge Representation)

Αναθεώρηση Πεποιθήσεων (Belief Revision)

Συλλογισμός (Reasoning)

Αλλαγή Πεποιθήσεων (Belief Change)

Ενημέρωση Πεποιθήσεων (Belief Updating)

Rule-Based Reasoning,

Case-Based Reasoning.

Artificial Intelligence And Law;

Legal Reasoning;

Defeasible Argumentation

Case-Based Model

Logic-Based Model

Legal Discourse Model

Παράρτημα Α

Αποτύπωση της εφαρμογής

A.1 Γενικές παρατηρήσεις

Η εφαρμογή έχει σχεδιαστεί σε Prolog. Αποτελείται από κανόνες και γεγονότα. Κάποια από αυτά είναι δυναμικά και ουσιαστικά αναπαριστούν την ρητώς αποθηκευμένη γνώση η οποία πρέπει να μεταβάλλεται υποστηρίζοντας τις αναθεωρήσεις πεποιθήσεων. Το αρχείο εκκίνησης της εφαρμογής είναι το `m_menu.pl`. Η κλήση του predicate `m_menu(0)` ξεκινά το γραφικό περιβάλλον της εφαρμογής.

A.2 Περιγραφή Predicates

Η πηγαία μορφή της εφαρμογής έχει ταξινομηθεί σε ξεχωριστά αρχεία με την εξής κατανομή:

- 1. Αρχεία που αναπαριστούν την διαθέσιμη Γνώση. Αποτελούνται από γεγονότα που αναπαριστούν πέντε διαφορετικές περιοχές:**

- I. Μαθητές

Το αρχείο student.pl με περιεχόμενο:

| Όνομα Predicate | Περιγραφή |
|-----------------|--------------------------------------------------------------------------------------|
| student | Περιέχει ιδιότητες που αφορούν κωδικό μαθητή, όνομα μαθητή, κωδικό τάξης που ανήκει. |
| mo | Μέσος όρος βαθμολογίας για συγκεκριμένο μαθητή |
| diagogh | Διαγωγή συγκεκριμένου μαθητή |

II. Απουσίες

Το αρχείο facts.pl με περιεχόμενο:

| Όνομα Predicate | Περιγραφή |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| apoyisia | Περιέχει ιδιότητες που αφορούν για συγκεκριμένο μαθητή, και συγκεκριμένη απουσία τις ώρες διδασκαλίας που χάθηκαν, και την ημερομηνία |
| dikaiologhsh | Περιέχει ιδιότητες που αφορούν για συγκεκριμένο μαθητή, και συγκεκριμένη απουσία την αξιολόγησή της δηλαδή εάν θεωρείται: Δικαιολογημένη, Δικαιολογημένη λόγω ασθένειας, Απαλλάσσεται επειδή ανήκει σε κάποια κατηγορία. Δεν μπορεί να δικαιολογηθεί επειδή ανήκει σε κάποια κατηγορία. Αδικαιολόγητη, |

| | |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| | Καταγράφεται επίσης ο χρόνος στον οποίο η απουσία αξιολογήθηκε. |
| parastatiko | Εμφάνιση κηδεμόνα για την δικαιολόγηση συγκεκριμένης απουσίας συγκεκριμένου μαθητή. |
| parastatiko_apo_forea | Επισύναψη παραστατικού από συγκεκριμένο φορέα (π.χ. Νοσοκομείο) για την δικαιολόγηση συγκεκριμένης απουσίας λόγω ασθένειας συγκεκριμένου μαθητή. |

III. Συνθήκες που αφορούν τον χαρακτηρισμό φοίτησης

Το αρχείο per_eparkeias.pl με περιεχόμενο:

| Όνομα Predicate | Περιγραφή |
|-----------------|-------------------------------------------------------------------------------------------------------------------|
| periptwsh | Περιέχει ιδιότητες που αφορούν την ονομασία συγκεκριμένης συνθήκης που αφορά χαρακτηρισμό φοίτησης. |
| exei_diagogh | Περιέχει ιδιότητες που αφορούν για συγκεκριμένη συνθήκη το επίπεδο διαγωγής μαθητή που επιτρέπεται |
| synolo | Περιέχει ιδιότητες που αφορούν για συγκεκριμένη συνθήκη το σύνολο απουσιών μαθητή που επιτρέπεται |
| synolo_a | Περιέχει ιδιότητες που αφορούν για συγκεκριμένη συνθήκη το σύνολο αδικαιολογήτων απουσιών μαθητή που επιτρέπεται |
| synolo_d | Περιέχει ιδιότητες που αφορούν για συγκεκριμένη συνθήκη το σύνολο δικαιολογημένων απουσιών μαθητή που επιτρέπεται |

| | |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| synolo_d_a | Περιέχει ιδιότητες που αφορούν για συγκεκριμένη συνθήκη το σύνολο δικαιολογημένων απουσιών λόγω ασθενείας μαθητή που επιτρέπεται |
| grades | Περιέχει ιδιότητες που αφορούν για συγκεκριμένη συνθήκη τις επιτρεπτές τιμές βαθμολογίας μαθητή που επιτρέπεται |
| arithmos_periptwsewn_eparkeias | Περιέχει ιδιότητες που αφορούν το πλήθος συνθηκών επάρκειας. |

IV. Αναπαράσταση κατηγοριών απουσιών μαθητή.

Το αρχείο parameters.pl με περιεχόμενο:

| Όνομα Predicate | Περιγραφή |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| arithmos_periptwsewn_apallagh | Περιέχει ιδιότητες που αφορούν το πλήθος των περιπτώσεων απαλλαγής. |
| apallagh | Περιέχει ιδιότητες που αφορούν την περιγραφή συγκεκριμένης κατηγορίας απαλλαγής απουσίας με ιδιότητες που αφορούν το μηνιαίο και ετήσιο όριο για συγκεκριμένο μαθητή. |
| apoyxies_poy_den_dikaiologoyntai | Περιέχει ιδιότητες που αφορούν την περιγραφή συγκεκριμένης κατηγορίας απουσίας που δεν προβλέπεται δικαιολόγηση. |
| dikaiologhthei | Περιέχει ιδιότητες που αφορούν την κατηγορία «απουσία μαθητή δικαιολογημένη» |
| dikaiologhmenh_logo_astheneias | Περιέχει ιδιότητες που αφορούν την κατηγορία «απουσία μαθητή δικαιολογημένη λόγω ασθενείας» |

| | |
|-------------------|----------------------------------------------------------------------------------------------------------------|
| adikaiologhth | Περιέχει ιδιότητες που αφορούν την κατηγορία «απουσία μαθητή αδικαιολόγητη» |
| orio_hmerwn | Περιέχει ιδιότητες που αφορούν το μέγιστο επιτρεπτό όριο χρονικό στο οποίο πρέπει να δικαιολογηθεί μια απουσία |
| klimaka | Περιέχει ιδιότητες που αφορούν την κλίμακα βαθμολογίας μαθητών |
| epipeda_diagoghs | Περιέχει ιδιότητες που αφορούν τους χαρακτηρισμούς που αποδίδονται στην διαγωγή μαθητή. |
| orio_parastatikwn | Περιέχει ιδιότητες που αφορούν το όριο παραστατικών δικαιολόγησης απουσίας για ένα μαθητή |

V. Παράμετρους λειτουργίας Σχολικού έτους.

Το αρχείο dates.pl με περιεχόμενο:

| Όνομα Predicate | Περιγραφή |
|---------------------------|---------------------------------------------------------------------------------------------------|
| klimaka | Περιέχει ιδιότητες που αφορούν την κλίμακα βαθμολογίας μαθητών |
| epipeda_diagoghs | Περιέχει ιδιότητες που αφορούν τους χαρακτηρισμούς που αποδίδονται στην διαγωγή μαθητή. |
| hmerhsio_orio_didaskalias | Περιέχει ιδιότητα που αφορά τον μέγιστο αριθμό ωρών απουσίας που μπορούμε να έχουμε σε μια ημέρα. |
| arithmos_class | Περιέχει ιδιότητα που αφορά τον αριθμό τάξεων της σχολικής μονάδας. |

| | |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| class | Περιέχει ιδιότητες που αφορούν για συγκεκριμένη συνθήκη το όνομα συγκεκριμένης τάξης |
| mhnas | Περιέχει ιδιότητες που αφορούν για συγκεκριμένο μήνα του αύξοντα αριθμού ημέρας στην οποία τελειώνει ο μήνας με σημείο αναφορά την έναρξη σχολικού έτους. |
| sunday | Περιέχει ιδιότητα που αφορά τον αύξοντα αριθμό ημέρας του πρώτου Σαββάτου της σχολική; Χρονιάς. |
| saturday | Περιέχει ιδιότητα που αφορά τον αύξοντα αριθμό ημέρας της πρώτης Κυριακής της σχολική; Χρονιάς. |
| diakopes | Περιέχει ιδιότητες που αφορούν συγκεκριμένη περίοδο διακοπών οριοθετημένη με τους αύξοντες αριθμούς ημερών αρχής και τέλους της χρονικής περιόδου. |

2. Αρχεία που υποστηρίζουν το γραφικό περιβάλλον επικοινωνίας. Αποτελούνται κυρίως από κανόνες που χρησιμοποιούν βιβλιοθήκες της πλατφόρμας XPCF:

Το αρχείο sub_menus.pl το οποίο διαχειρίζεται τα μενού της εφαρμογής:

| Όνομα Predicate | Περιγραφή |
|-----------------|-----------------------------------------------------------------------------------------------------------|
| menu_1 | Διαχείριση του μενού λειτουργικών που αφορούν «αναθεωρήσεις» στο Σύστημα Γνώσης. |
| menu | Διαχείριση του μενού λειτουργικών που αφορούν ελέγχους πάνω στο Σύστημα Γνώσης για «πιθανές» αναθεωρήσεις |
| menu_2 | Διαχείριση του μενού λειτουργικών που αφορούν αναθεωρήσεις σε παραμέτρους του συστήματος γνώσης |

Το αρχείο user_interface.pl το οποίο διαχειρίζεται βοηθητικά στοιχεία για το user interface όπως μηνύματα ελέγχου:

| Όνομα Predicate | Περιγραφή |
|-------------------------------------------------------------------------------|------------------------------------------------------------------------|
| w_label4, w_label2, disp_message d_mes, w_label3, new_p, ok | Κανόνες που παράγουν μηνύματα ελέγχου σε μορφή παραθύρων διαλόγου. |
| s_day, s_mhnas, n_list, pop_student, pop_diagogh, m_class | Δημιουργία βοηθητικών πλαισίων λίστας για επιλογή. |
| d_q2, d_q8, dq_temp, | Κανόνες διαγραφής των εμφανίσεων συγκεκριμένων δυναμικών κατηγορημάτων |

| | |
|-------|------------------------------------------|
| d_q5, | |
| ok | Ο κανόνας που «τερματίζει» την εφαρμογή. |

Το αρχείο m_menu.pl το οποίο διαχειρίζεται την εκκίνηση της εφαρμογής συστήματος γνώσης:

| Όνομα Predicate | Περιγραφή |
|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| start | Ο κανόνας που εκτελείται με την εκκίνηση της εφαρμογής και ουσιαστικά μεταγλωττίζει όλα τα αρχεία prolog της εφαρμογής. |
| m_menu | Ο κανόνας που διαχειρίζεται το εισαγωγικό μενού εργασιών της εφαρμογής |
| display_database, display_database1 | Κανόνες που εμφανίζουν την «συνολική εικόνα» του συστήματος γνώσης όσον αφορά τις απουσίες μαθητών. |
| display_database1, display_database_cases1 | Κανόνες που εμφανίζουν την «συνολική εικόνα» του συστήματος γνώσης όσον αφορά τις συνθήκες χαρακτηρισμού επάρκειας φοίτησης μαθητών. |

3. Αρχεία που υποστηρίζουν κανόνες αναδιοργάνωσης του συστήματος γνώσης:

Το αρχείο input_delete_menu.pl το οποίο διαχειρίζεται κανόνες που σχετίζονται με την μεταβολή γνώσης στο σύστημα χρησιμοποιώντας γραφική διεπαφή χρήστη:

| Όνομα Predicate | Περιγραφή |
|-----------------|-------------------------------------------------------------|
| menu_student1 | Ο κανόνας που διαχειρίζεται την προσθήκη μαθητή στο Σύστημα |

| | |
|----------------|-------------------------------------------------------------------------------------------------------------------------|
| | Γνώσης. |
| menu_mo1 | Ο κανόνας που διαχειρίζεται την προσθήκη μέσου όρου βαθμολογίας μαθητή στο Σύστημα Γνώσης. |
| menu_student01 | Ο κανόνας που διαχειρίζεται την διόρθωση ή διαγραφή μαθητή στο Σύστημα Γνώσης. |
| menu_dia1 | Ο κανόνας που διαχειρίζεται την προσθήκη χαρακτηρισμού διαγωγής μαθητή στο Σύστημα Γνώσης. |
| menu_ap1 | Ο κανόνας που διαχειρίζεται την προσθήκη απουσίας μαθητή στο Σύστημα Γνώσης. |
| menu_di1 | Ο κανόνας που διαχειρίζεται την προσθήκη πράξης δικαιολόγησης απουσίας μαθητή στο Σύστημα Γνώσης. |
| menu_parast1 | Ο κανόνας που διαχειρίζεται την προσθήκη γνώσης «παραστατικό δικαιολόγησης απουσίας» στο Σύστημα Γνώσης. |
| menu_parast_f1 | Ο κανόνας που διαχειρίζεται την προσθήκη γνώσης «παραστατικό δικαιολόγησης απουσίας λόγω ασθενείας» στο Σύστημα Γνώσης. |
| menu_mhnas1 | Ο κανόνας που διαχειρίζεται την προσθήκη του γεγονότος ημέρα έναρξης μαθημάτων. |
| menu_saturday1 | Ο κανόνας που διαχειρίζεται την προσθήκη του γεγονότος ,Αριθμός ημέρας πρώτης αργίας Σαββάτου. |
| menu_epa1 | Ο κανόνας που διαχειρίζεται την προσθήκη μιας νέας συνθήκης χαρακτηρισμού επάρκειας φοίτησης μαθητή στο σύστημα.. |
| menu_diakop | Ο κανόνας που διαχειρίζεται την προσθήκη μιας νέας περιόδου διακοπών μαθημάτων στο σύστημα. |

| | |
|-----------------------|---------------------------------------------------------------------------------------------------------------|
| menu_diakop_d | Ο κανόνας που διαχειρίζεται την αφαίρεση μιας περιόδου διακοπών μαθημάτων στο σύστημα. |
| menu_search | Ο κανόνας που εμφανίζει συγκεντρωτικά στοιχεία μαθητή |
| menu_search_apoysies, | Ο κανόνας που εμφανίζει αναλυτικά στοιχεία απουσιών μαθητή |
| menu_dpl_case | Ο κανόνας που εμφανίζει καταστάσεις όπου μαθητές χαρακτηρίζονται επαρκής βάσει περισσοτέρων από μια συνθηκών. |

Το αρχείο input.pl το οποίο διαχειρίζεται κανόνες που σχετίζονται με την μεταβολή γνώσης στο σύστημα οι οποίοι αποτελούν ουσιαστικά την «low level» δραστηριότητα όσον αφορά τις μεταβολές στο Σύστημα Γνώσης :

| Όνομα Predicate | Περιγραφή |
|--------------------|-----------------------------------------------------------------------------------------------------|
| input_student | Ο κανόνας που προσθέτει ένα μαθητή στο Σύστημα Γνώσης |
| delete_student | Ο κανόνας που υλοποιεί την αφαίρεση γνώσης ενός μαθητή στο Σύστημα Γνώσης |
| input_mo | Ο κανόνας που προσθέτει την βαθμολογία ενός μαθητή στο Σύστημα Γνώσης |
| input_diagogh | Ο κανόνας που προσθέτει τον χαρακτηρισμό διαγωγής ενός μαθητή στο Σύστημα Γνώσης |
| input_apoysia | Ο κανόνας που προσθέτει τα στοιχεία μιας απουσίας μαθητή στο Σύστημα Γνώσης |
| input_dikaiologhsh | Ο κανόνας που προσθέτει τα στοιχεία που αφορούν την δικαιολόγηση απουσίας μαθητή στο Σύστημα Γνώσης |

| | |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| input_parastatiko | Ο κανόνας που προσθέτει τα στοιχεία που αφορούν την προσκόμιση παραστατικού δικαιολόγησης απουσίας μαθητή στο Σύστημα Γνώσης |
| input_parastatiko_apo_forea | Ο κανόνας που προσθέτει τα στοιχεία που αφορούν την προσκόμιση παραστατικού δικαιολόγησης απουσίας λόγω ασθενείας μαθητή στο Σύστημα Γνώσης |
| input_per_eparkeias | Ο κανόνας που προσθέτει τα στοιχεία που αφορούν μια συνθήκη χαρακτηρισμού επαρκούς φοίτησης μαθητή στο Σύστημα Γνώσης |
| input_mhnas | Ο κανόνας που προσθέτει τα στοιχεία που αφορούν τους μήνες του σχολικού έτους. |
| input_diakopes | Ο κανόνας που προσθέτει τα στοιχεία που αφορούν περιόδους διακοπών του σχολικού έτους. |
| reset_diakopes | Ο κανόνας που αφορά την αφαίρεση όλης της Γνώσης σχετικά με περιόδους διακοπών του σχολικού έτους. |

Το αρχείο `update_knowledge.pl` το οποίο διαχειρίζεται κανόνες που σχετίζονται με την μεταβολή των δυναμικών κατηγορημάτων που αναπαριστούν την γνώση.

| Όνομα Predicate | Περιγραφή |
|-----------------|------------------------------------------------------------------|
| update_student | Ο κανόνας που ενημερώνει το αρχείο <code>student.pl</code> |
| update_facts | Ο κανόνας που ενημερώνει το αρχείο <code>facts.pl</code> |
| update_per_epa | Ο κανόνας που ενημερώνει το αρχείο <code>per_eparkeias.pl</code> |
| update_dates | Ο κανόνας που ενημερώνει το αρχείο <code>dates.pl</code> |

| | |
|-------------------|--------------------------------------------------|
| update_parameters | Ο κανόνας που ενημερώνει το αρχείο parameters.pl |
|-------------------|--------------------------------------------------|

Το αρχείο param_menu.pl το οποίο διαχειρίζεται κανόνες που σχετίζονται με βοηθητικές εργασίες πάνω στο Σύστημα Γνώσης.

| Όνομα Predicate | Περιγραφή |
|-----------------|----------------------------------------------------------------------------------------|
| menu_a_d | Εμφάνιση απουσιών που έχουν δικαιολογηθεί |
| menu_a_d_a | Εμφάνιση απουσιών που έχουν δικαιολογηθεί λόγω ασθενείας. |
| menu_a_a | Κατάσταση απουσιών που έχουν απαλλαγεί. |
| menu_a_d_d | Κατάσταση απουσιών που απαγορεύεται να δικαιολογηθούν. |
| menu_a_h | Εμφάνιση απουσιών συγκεκριμένης ημερομηνίας. |
| menu_e_c | Εμφάνιση απουσιών συγκεκριμένης τάξης |
| menu_a_l | Κατάσταση απουσιών συγκεκριμένου μαθητή |
| menu_h_orio | Αναθεώρηση του μέγιστου ημερήσιου αριθμού ωρών διδασκαλίας |
| menu_klimaka | Αναθεώρηση της κλίμακας βαθμολογίας. |
| menu_class | Αναθεώρηση της γνώσης όσον αφορά τις τάξεις μαθητών. |
| menu_diag | Αναθεώρηση της γνώσης όσον αφορά τα επίπεδα χαρακτηρισμού διαγωγής μαθητή. |
| menu_apallagh | Αναθεώρηση της γνώσης όσον αφορά τις κατηγορίες περιπτώσεων απαλλαγής απουσίας μαθητή. |

| | |
|------------------|----------------------------------------------------------------------------------------------------------------------------|
| menu_den_d | Αναθεώρηση της γνώσης όσον αφορά τις κατηγορίες περιπτώσεων που δεν επιτρέπεται δικαιολόγηση απουσίας μαθητή. |
| menu_parast_orio | Αναθεώρηση της γνώσης όσον αφορά τον μέγιστο αριθμό παραστατικών δικαιολόγησης απουσίας μαθητή. |
| menu_orio_hm | Αναθεώρηση της γνώσης όσον αφορά το χρονικό όριο στο οποίο μπορεί να δικαιολογηθεί μια απουσία. |
| menu_epa2 | Εμφάνιση και δυνατότητα αναθεώρησης την γνώσης που αφορά συνθήκες επάρκειας φοίτησης που υπάρχουν στο Σύστημα Γνώσης. |
| menu_apa2 | Εμφάνιση και δυνατότητα αναθεώρησης των περιπτώσεων απαλλαγής απουσιών μαθητών που υπάρχουν στο Σύστημα Γνώσης. |
| menu_p_d_d2 | Εμφάνιση και δυνατότητα αναθεώρησης των περιπτώσεων απουσιών μαθητών που δεν μπορούν να δικαιολογηθούν στο Σύστημα Γνώσης. |

4. Αρχεία που υποστηρίζουν κανόνες που περιγράφουν την μεταβολή της κατάστασης του συστήματος γνώσης που προκαλούν πιθανές αναθεωρήσεις.

Το αρχείο belief_revision.pl το οποίο περιέχει κανόνες που ελέγχουν πιθανές αναθεωρήσεις πάνω στο Σύστημα Γνώσης και τι αυτές συνεπάγονται για τη ορθότητα και αξιοπιστία του συστήματος γνώσης. Προσφέροντας μια γραφική διεπαφή χρήστη:

| Όνομα Predicate | Περιγραφή |
|-----------------|-----------|
| | |

| | |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Menu_epa | Ο κανόνας που διαχειρίζεται τον έλεγχο όσον αφορά την δυνατότητα προσθήκης μιας νέας συνθήκης χαρακτηρισμού επαρκής φοίτησης στο Σύστημα Γνώσης εξετάζοντας και παρουσιάζοντας την κατάσταση στην οποία θα βρεθεί το σύστημα .Επίσης μέσω του κανόνα αναδεικνύονται και προτάσεις που αφορούν την βελτίωση της κατάστασης του συστήματος στην περίπτωση που το νέο γεγονός πρέπει να προστεθεί στο Σύστημα Γνώσης. |
| meniu_mo | Ο κανόνας που διαχειρίζεται τον έλεγχο στην περίπτωση προσθήκης βαθμολογίας συγκεκριμένου μαθητή και την παρουσίαση της κατάστασης στην οποία θα βρεθεί το Σύστημα Γνώσης.. |
| menu_student | Ο κανόνας που διαχειρίζεται τον έλεγχο στην περίπτωση προσθήκης ενός νέου μαθητή και την παρουσίαση της κατάστασης στην οποία θα βρεθεί το Σύστημα Γνώσης.. |
| menu_dia | Ο κανόνας που διαχειρίζεται τον έλεγχο στην περίπτωση προσθήκης επιπέδου διαγωγής σε μαθητή και την παρουσίαση της κατάστασης στην οποία θα βρεθεί το Σύστημα Γνώσης.. |
| menu_ap | Ο κανόνας που διαχειρίζεται τον έλεγχο στην περίπτωση προσθήκης συγκεκριμένης απουσίας σε μαθητή και την παρουσίαση της κατάστασης στην οποία θα βρεθεί το Σύστημα Γνώσης.. |
| menu_di | Ο κανόνας που διαχειρίζεται τον έλεγχο στην περίπτωση προσθήκης γεγονότος δικαιολόγησης απουσίας σε μαθητή και την παρουσίαση της κατάστασης στην οποία θα βρεθεί το Σύστημα Γνώσης.. |
| menu_parast_f1 | Ο κανόνας που διαχειρίζεται την προσθήκη γνώσης «παραστατικό δικαιολόγησης απουσίας λόγω ασθένειας» στο Σύστημα Γνώσης. |

| | |
|-----------------------|-------------------------------------------------------------------------------------------------------------------|
| menu_mhnas1 | Ο κανόνας που διαχειρίζεται την προσθήκη του γεγονότος ημέρα έναρξης μαθημάτων. |
| menu_saturday1 | Ο κανόνας που διαχειρίζεται την προσθήκη του γεγονότος ,Αριθμός ημέρας πρώτης αργίας Σαββάτου. |
| menu_epa1 | Ο κανόνας που διαχειρίζεται την προσθήκη μιας νέας συνθήκης χαρακτηρισμού επάρκειας φοίτησης μαθητή στο σύστημα.. |
| menu_diakop | Ο κανόνας που διαχειρίζεται την προσθήκη μιας νέας περιόδου διακοπών μαθημάτων στο σύστημα. |
| menu_diakop_d | Ο κανόνας που διαχειρίζεται την αφαίρεση μιας περιόδου διακοπών μαθημάτων στο σύστημα. |
| menu_search | Ο κανόνας που εμφανίζει συγκεντρωτικά στοιχεία μαθητή |
| menu_search_apoysies, | Ο κανόνας που εμφανίζει αναλυτικά στοιχεία απουσιών μαθητή |
| menu_dpl_case | Ο κανόνας που εμφανίζει καταστάσεις όπου μαθητές χαρακτηρίζονται επαρκής βάσει περισσότερων από μια συνθηκών. |

Το αρχείο new_state1.pl το οποίο περιέχει κανόνες που ελέγχουν πιθανές αναθεωρήσεις πάνω στο Σύστημα Γνώσης και τι αυτές συνεπάγονται για τη ορθότητα και αξιοπιστία του συστήματος γνώσης. Στο συγκεκριμένο αρχείο περιγράφεται το συμπέρασμα «σύγκρουση» το οποίο ορίζεται ως η κατάσταση στην οποία ένας μαθητής χαρακτηρίζεται ως επαρκής από δύο ή και παραπάνω συνθήκες επάρκειας φοίτησης.

| Όνομα Predicate | Περιγραφή |
|-----------------|----------------------------------------------------------------------------------------------------------------------|
| collision | Ο κανόνας που παράγει το συμπέρασμα «αριθμός συγκρούσεων στο σύστημα» το οποίο είναι ουσιαστικά το πλήθος εμφανίσεων |

| | |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | μαθητών σε αλληλοσυμπληρούμενες συνθήκες επάρκειας. |
| new_state | Ο κανόνας που παράγει το συμπέρασμα ένα μια συγκεκριμένη αναθεώρηση οδηγεί το σύστημα σε μια κατάσταση όπου παραβιάζεται η ορθότητα και αξιοπιστία της γνώσης που περιέχει. |

Το αρχείο test_cons.pl το οποίο περιέχει κανόνες που ελέγχουν την υφιστάμενη κατάσταση του συστήματος γνώσης και εμφανίζουν διάφορα συμπεράσματα.

| Όνομα Predicate | Περιγραφή |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| test_incons_about_cases | Ο κανόνας που παράγει το συμπέρασμα «συνθήκες επάρκειας φοίτησης που αλληλοσυμπληρώνονται» |
| test_incons_about_apoysies | Ο κανόνας που παράγει το συμπέρασμα της μη ορθότητας γνώσης που αφορά απουσίες μαθητών. Παράδειγμα μη ορθής γνώσης αποτελεί το ότι μια απουσία δικαιολογήθηκε μετά το επιτρεπτό χρονικό όριο. |
| test_student1 | Ο κανόνας που συμπεραίνει εάν υπάρχουν μαθητές χωρίς βαθμούς |
| test_student2 | Ο κανόνας που συμπεραίνει εάν υπάρχουν μαθητές χωρίς να έχουν διαγωγή. |
| test_student3 | Ο κανόνας που συμπεραίνει εάν υπάρχουν μαθητές με βαθμό εκτός κλίμακας. |
| test_student | Ο κανόνας που συμπεραίνει εάν υπάρχουν μαθητές με χαρακτηρισμό διαγωγής άγνωστο. |
| test_student_aa | Ο κανόνας που συμπεραίνει εάν συγκεκριμένος μαθητής χαρακτηρίζεται επαρκής από συγκεκριμένη συνθήκη |

5. Το αρχείο `rules.pl` το οποίο περιέχει τους βασικούς κανόνες που χρησιμοποιούνται στην παραγωγή συμπερασμάτων και στην αναθεώρηση γνώσης.

| Όνομα Predicate | Περιγραφή |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>dikaiologhmenh</code> | Ο κανόνας που παράγει το συμπέρασμα εάν συγκεκριμένη απουσία είναι δικαιολογημένη |
| <code>dikaiologhmenh_l_a</code> | Ο κανόνας που παράγει το συμπέρασμα εάν συγκεκριμένη απουσία είναι δικαιολογημένη λόγω ασθενείας |
| <code>s_ypologismos_m</code> | Ο κανόνας που παράγει το συμπέρασμα για τον αριθμό απουσιών μαθητή που αν και απαλλάσσονται λόγω υπέρβασης του μηνιαίου ή ετήσιου ορίου λαμβάνονται υπόψη. |
| <code>ypologismos_a_a</code> | Ο κανόνας που παράγει το συμπέρασμα «σύνολο αδικαιολογήτων απουσιών» για συγκεκριμένο μαθητή. |
| <code>ypologismos_a_d_a</code> | Ο κανόνας που παράγει το συμπέρασμα «σύνολο δικαιολογημένων απουσιών λόγω ασθενείας» για συγκεκριμένο μαθητή. |
| <code>synolo_a_y</code> | Ο κανόνας που παράγει το συμπέρασμα «σύνολο απουσιών που λαμβάνονται υπόψη» για συγκεκριμένο μαθητή. |
| <code>aksiologhsh</code> | Ο κανόνας που αξιολογηθεί ένα μαθητή ως επαρκή ανάλογα με τις συνθήκες που υπάρχουν καθώς και αν υπάρχουν εκκρεμότητες όσον αφορά απουσίες που δεν έχουν δικαιολογηθεί ή δεν υπάρχει βαθμός ή διαγωγή μαθητή. |
| <code>report</code> | Ο κανόνας που συμπεραίνει τα στοιχεία που αφορούν τις απουσίες μαθητή. |
| <code>kartella</code> | Ο κανόνας που συμπεραίνει αναλυτικά στοιχεία όσον αφορά |

| | |
|--|-------------------------------------|
| | απουσίες και χαρακτηρισμό φοίτησης. |
|--|-------------------------------------|

6. Το αρχείο `dyn_pred.pl` το οποίο ορίζει τα δυναμικά κατηγορήματα της εφαρμογής.
7. Το αρχείο `reset.pl` το οποίο περιέχει τον ομώνυμο κανόνα `reset` ο οποίος προκαλεί την «αρχικοποίηση» του συστήματος γνώσης.

A.3 Στιγμιότυπα της Εφαρμογής

Νέα συνθήκη για τον χαρακτηρισμό επάρκειας φοίτησης

Αα: 6

Όνομασία:

Η ΤΙΜΗ -1 ΣΤΑ ΠΕΔΙΑ ΠΟΥ ΑΚΟΛΟΥΘΟΥΝ ΣΗΜΑΙΝΕΙ ΌΤΙ ΤΟ ΠΕΔΙΟ ΔΕΝ ΛΑΜΒΑΝΕΤΑΙ ΥΠΟΨΗ

Σύνολο απουσιών που λαμβάνονται υπόψη

Από: -1

Μέχρι: -1

Σύνολο απουσιών που μπορούν να δικαιολογηθούν

Από: -1

Μέχρι: -1

Σύνολο απουσιών που μπορούν να δικαιολογηθούν λόγω ασθένειας

Από: -1

Μέχρι: -1

Σύνολο απουσιών που είναι αδικαιολόγητες

Από: -1

Μέχρι: -1

Περιοχή βαθμολογίας

Από: -1

Μέχρι: -1

Διαγωγή:

Ok Cancel

Εισαγωγή συνθήκης επάρκειας φοίτησης.

Μαθητής

| | |
|------------|-----------|
| Κωδικός | 2 |
| όνομα | Μαθητής_4 |
| τάξη | Τάξη_1 |
| Μέσος όρος | 20 |
| Διαγωγή | μέτρια |

Όνομα:

Τάξη:

Μέσος όρος (-1 για μη μεταβολή):

Διαγωγή (Καρμιά για μη μεταβολή):

Ενημέρωση στοιχείων μαθητή

Καταχώρηση Απουσίας μαθητή

Κωδικός Μαθητή:

Αριθμός απουσιών:

Μήνες:

Ημερα:

Καταχώρηση απουσίας μαθητή.

Δικαιολόγηση Απουσίας μαθητή

Υπάρχουσες Απουσίες

| Όνομα Μαθητή | Αα Απουσίας | Ημερομηνία |
|--------------|-------------|------------|
| Μαθητής_3 | 2 | martios 7 |

Αα Μαθητή: Μαθητής_4

Αα απουσίας: 0

Μήνες: Ιανουαριος

Ημερα: 1

Περιπτώσεις απαλλαγής:: όχι

Περιπτώσεις που δεν δικαιολογούνται:: όχι

Περιπτώσεις που δικαιολογούνται:: όχι

Ok Cancel

Δικαιολόγηση απουσίας μαθητή

Κατάσταση Απουσιών

| Αα | όνομα μαθητή | Μαθητής_3 | Ημ. απουσίας | Ωρες | Ημ. δικαιολόγησης | Αιτιολογία | παραστατικό |
|----|--------------|-----------|--------------|-------------|-------------------|---------------|-------------|
| 1 | febroyarios | 6 | 2 | febroyarios | 7 | Αδικαιολόγητη | δεν υπάρχει |
| 2 | martios | 7 | 4 | ----- | | | |

Κατάσταση απουσιών μαθητή

Στοιχεία Μαθητή

Ok

Αα: 7 ▲

Όνομα: Μαθητής_7

Τάξη: Τάξη_1

Βαθμός: 15

Διαγωγή: κοσμία

Σύνολο απουσιών που εμπίπτουν σε περίπτωση απαλλαγής αλλά υπερβαίνουν όριο: 0

Σύνολο απουσιών που δεν λαμβάνονται υπόψη: 0

Σύνολο απουσιών που λαμβάνονται υπόψη στον χαρακτηρισμό φοίτησης: 0

Σύνολο απουσιών που έχουν δικαιολογηθεί λόγω ασθένειας: 0

Σύνολο απουσιών που έχουν δικαιολογηθεί: 0

Επίδοση: eparkhs

Συνθήκες που χαρακτηρίζουν τον μαθητή επαρκή

| Αα συνθήκης | Περίπτωση |
|-------------|-------------|
| 2 | Περίπτωση_2 |

Καρτέλα επίδοσης μαθητή

| Συγκεντρωτική Κατάσταση Συνθηκών επάρκειας φοίτησης | | | | | | | | | | | | | |
|-----------------------------------------------------|-------------|---------|--------|---------|-----------------|---------------------------------|------------------------------------------------|--------------------------------|------------------|--|--|--|--|
| Αα | Όνομασία | Περιοχή | Βαθμών | Διαγωγή | Σύνολο απουσιών | Σύνολο Απουσιών δικαιολογημένων | Σύνολο Απουσιών δικαιολογημένων λόγω ασθενείας | Σύνολο Απουσιών αδικαιολόγητων | | | | | |
| 1 | Περίπτωση_3 | 16 | -- | 20 | κοσμιότατη | 0 -- 161 | Χωρίς περιορισμό | 0 -- 58 | 0 -- 50 | | | | |
| 2 | Περίπτωση_2 | 0 | -- | 15 | κοσμία | 0 -- 25 | Χωρίς περιορισμό | Χωρίς περιορισμό | Χωρίς περιορισμό | | | | |
| 3 | Περίπτωση_1 | 16 | -- | 18 | κοσμία | 0 -- 51 | Χωρίς περιορισμό | Χωρίς περιορισμό | Χωρίς περιορισμό | | | | |
| 4 | Περίπτωση_4 | 1 | -- | 10 | μέτρια | 0 -- 24 | Χωρίς περιορισμό | Χωρίς περιορισμό | Χωρίς περιορισμό | | | | |
| 5 | Περίπτωση_9 | 0 | -- | 13 | κοσμιότατη | Χωρίς περιορισμό | Χωρίς περιορισμό | Χωρίς περιορισμό | Χωρίς περιορισμό | | | | |

Συγκεντρωτική κατάσταση συνθηκών επάρκειας φοίτησης.

| Συγκεντρωτική Κατάσταση Απουσιών Μαθητών | | | | | | | | |
|------------------------------------------|----------------|---------------------------|------------|-----------------|--------------------------|-----------------------------------------|-------------------------|--|
| Αα Μαθητή | Όνομα | M.O. | Διαγωγή | Σύνολο απουσιών | Απουσίες δικαιολογημένες | Απουσίες δικαιολογημένες Λόγω Ασθενείας | Απουσίες Αδικαιολόγητες | |
| 2 | Μαθητής_4 | 20 | μέτρια | 6 | 0 | 6 | 0 | |
| 4 | Μαθητής_20 | -----Ελλιπή Στοιχεία----- | | | | | | |
| 6 | bbbbbbbbbbbbbb | -----Ελλιπή Στοιχεία----- | | | | | | |
| 5 | aaaaaaaaaaaaaa | -----Ελλιπή Στοιχεία----- | | | | | | |
| 1 | Μαθητής_3 | 20 | κοσμιότατη | 0 | 0 | 0 | 0 | |
| 7 | Μαθητής_7 | 15 | κοσμία | 0 | 0 | 0 | 0 | |
| 3 | Μαθητής_15 | 15 | κοσμιότατη | 7 | 0 | 0 | 0 | |

Συγκεντρωτική κατάσταση απουσιών μαθητών

Κατάσταση απουσιών που απαλλάσσονται

| Όνομα μαθητή | Τάξη | Ημερομηνία απουσίας | Ώρες | Ημερομηνία δικαιολόγησης | Κατηγορία |
|--------------|--------|---------------------|------|--------------------------|--------------------|
| Μαθητής_20 | Τάξη_1 | febroyarios | 6 | 1 febroyarios | 7 Αθλητικοί αγώνες |

Εμφάνιση απουσιών που απαλλάσσονται.

Κατάσταση απουσιών που δεν μπορούν να δικαιολογηθούν

| Όνομα μαθητή | Τάξη | Ημερομηνία απουσίας | Ώρες | Ημερομηνία δικαιολόγησης | Κατηγορία |
|--------------|--------|---------------------|------|--------------------------|--------------------|
| Μαθητής_15 | Τάξη_1 | febroyarios | 6 | 7 febroyarios | 7 Ημερήσια αποβολή |

Εμφάνιση απουσιών που δεν μπορούν να δικαιολογηθούν.

Κατάσταση απουσιών που έχουν δικαιολογηθεί λόγω ασθένειας

| Όνομα μαθητή | Τάξη | Ημερομηνία απουσίας | Ώρες | Ημερομηνία δικαιολόγησης | Παραστατικό |
|--------------|--------|---------------------|------|--------------------------|-------------|
| Μαθητής_4 | Τάξη_1 | febroyarios | 6 | 6 febroyarios | 7 Όχι |

Εμφάνιση απουσιών που έχουν δικαιολογηθεί λόγω ασθένειας.

| Κατάσταση συνθηκών που δημιουργούν Πλεονασμό | | | |
|------------------------------------------------------------------------------------------------------------------|--------------------|----------------------|--------------------|
| Συνθήκη | Αριθμός Πλεονασμών | | |
| aaaaaaaaaaaa | 4 | | |
| κατάσταση υπολοίπων συνθηκών που εμπλέκονται και ελέγχεται η επίδραση της αποβολής τους: | | | |
| Συνθήκη | Νέα Κατάσταση | | |
| Περίπτωση_3 | Αριθμός Πλεονασμών | 2 | |
| Περίπτωση_2 | Αριθμός Πλεονασμών | 2 | |
| κατάσταση μαθητών που εμπλέκονται και ελέγχεται η επίδραση της αποβολής τους στο μέτρο Πλεονασμών: | | | |
| A. Μαθητή | Όνομα μαθητή | Αριθμός Πλεονασμών | |
| 1 | Μαθητής_3 | 2 | |
| 7 | Μαθητής_7 | 2 | |
| κατάσταση μαθητών που εμπλέκονται και ελέγχεται η επίδραση της αλλαγής της διαγωγής τους στον αριθμό Πλεονασμών: | | | |
| A. Μαθητή | Όνομα μαθητή | προτεινόμενη διαγωγή | Αριθμός Πλεονασμών |
| 1 | Μαθητής_3 | κοσμία | 2 |
| 1 | Μαθητής_3 | μέτρια | 2 |
| 1 | Μαθητής_3 | κακή | 2 |
| 7 | Μαθητής_7 | κοσμιότητα | 2 |
| 7 | Μαθητής_7 | μέτρια | 2 |
| 7 | Μαθητής_7 | κακή | 2 |

Οθόνη προτάσεων περιορισμού των πλεονασμών μέσω αναδιοργάνωση γνώσης.

A.4 Πηγαία Μορφή Κώδικα Εφαρμογής

Ακολουθεί ο κώδικας της εφαρμογής που υλοποιεί το Σύστημα Γνώσης. Κάποιες γενικές παρατηρήσεις σχετικά με την δομή του είναι:

- Έχει χρησιμοποιηθεί αρκετά η δυνατότητα αναδρομής στην περιγραφή κανόνων.
- Για πολλές διαδικασίες παραγωγής συμπερασμάτων έχουν χρησιμοποιηθεί δυναμικής μορφής predicates τα οποία εξυπηρετούν προσωρινά τις ανάγκες παραγωγής ενός συμπεράσματος.
- Η ρητώς αποθηκευμένη γνώση του συστήματος περιγράφεται από δυναμικά κατηγορήματα στα οποία επενεργούν κανόνες και τα μεταβάλλουν ανάλογα με τις απαιτήσεις της αναθεώρησης πεποιθήσεων.

A.5 Ευρετήριο Αρχείων Εφαρμογής

| Όνομα αρχείου προγράμματος Prolog | Περιγραφή | Σελίδα |
|-----------------------------------|-----------------------------------------------------------------------------|--------|
| belief_revision.pl | Προτάσεις για αναδιοργάνωση γνώσης και βελτιστοποίηση του συστήματος γνώσης | A-28 |
| dates.pl | Αντικείμενα που σχετίζονται με περιόδους αργιών στην σχολική χρονιά | A-30 |
| dyn_pred.pl | Βοηθητικά δυναμικά γεγονότα για την παραγωγή γνώσης μέσω κανόνων | A-31 |
| facts.pl | Γεγονότα που σχετίζονται με τις απουσίες μαθητών | A-32 |
| input.pl | Κανόνες που υλοποιούν αναδιοργάνωση | A-33 |

| | | |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------|
| | γνώσης στο σύστημα | |
| input_delete_menu.pl | Κανόνες που διαμορφώνουν το γραφικό περιβάλλον επικοινωνίας που αφορά μεταβολές στην γνώση. | A-42 |
| m_menu.pl | Κανόνες που διαμορφώνουν το εισαγωγικό περιβάλλον της εφαρμογής. Κανόνας εκκίνησης: m_menu(0). | A-61 |
| new_state1.pl | Κανόνες που ελέγχουν αν προτεινόμενη αναδιοργάνωση γνώσης οδηγεί το σύστημα σε ορθή ή μη ορθή κατάσταση | A-65 |
| param_menu.pl | Κανόνες που σχετίζονται με βοηθητικές λειτουργίες στην εφαρμογή. | A-71 |
| parameters.pl | Γεγονότα τα οποία αναπαριστούν «βοηθητική» γνώση όπως για παράδειγμα: Κατηγορίες απουσιών που απαλλάσσονται, Κλίμακα βαθμολογίας. | A-91 |
| per_eparkeias | Γεγονότα τα οποία αναπαριστούν τις συνθήκες επάρκειας φοίτησης | A-93 |
| reset.pl | Κανόνες που υλοποιούν την αρχικοποίηση του συστήματος γνώσης. | A-94 |
| rules.pl | Κανόνες που παράγουν τη γνώση για την επάρκεια φοίτησης μαθητών | A-96 |
| student.pl | Γεγονότα που αφορούν τη γνώση για τους μαθητές | A-111 |
| sub_menus.pl | Κανόνες που υλοποιούν τα υπομενού της εφαρμογής. | A- 113 |
| test_cons.pl | Κανόνες που ελέγχουν την παρούσα κατάσταση του συστήματος γνώσης | A-117 |
| update_knowledge.pl | Κανόνες που υλοποιούν την ενημέρωση των δυναμικών γεγονότων Prolog της εφαρμογής. | A-125 |
| user_interface.pl | Κανόνες που υλοποιούν τα βοηθητικά παράθυρα διαλόγου της εφαρμογής. | A-127 |

belief_revision.pl

list_dpl_case:-

```
\+ ( periptwsh(C1,_),periptwsh(C2,_),\+ C1=C2,((\+ q_temp(C2));(\+ q_temp(C1))),
      student(X,_),test_student_aa(X,C1),test_student_aa(X,C2)),
      !,true.
```

list_dpl_case:-

```
periptwsh(C1,_),periptwsh(C2,_), \+ C1=C2,
\+ ( q_temp(C1), q_temp(C2)),student(X,_),test_student_aa(X,C1),
test_student_aa(X,C2),
( \+ q_temp(C1),assert(q_temp(C1));true),
( \+ q_temp(C2),assert(q_temp(C2));true),
list_dpl_case,
!,true.
```

insert_st_q2:-

```
q_temp(C),student(X,_),test_student_aa(X,C),\+ q2(X),
assert(q2(X),diagogh(X,D),
insert_st_q_diag2(X,D),
insert_st_q2,
!,true.
```

insert_st_q2:-

```
\+ ( q_temp(C),student(X,_),test_student_aa(X,C),\+ q2(X)),
!,true.
```

insert_st_q3(Y):-

dates.pl

`:- dynamic mhnas/3.`
`:- dynamic sunday/1.`
`:- dynamic saturday/1.`
`:- dynamic diakopes/3.`

`mhnas(1, septembrios, 14).`
`mhnas(2, oktvbrios, 46).`
`mhnas(3, noembrios, 78).`
`mhnas(4, dekembrios, 109).`
`mhnas(5, ianoyarios, 141).`
`mhnas(6, febroyarios, 172).`
`mhnas(7, martios, 201).`
`mhnas(8, aprilios, 232).`
`mhnas(9, maios, 264).`
`mhnas(10, ioynios, 295).`
`mhnas(11, end, 326).`

`sunday(22).`

`saturday(21).`

`diakopes(1, 204, 207).`

dyn_pred.pl

:- dynamic q2/1.
:- dynamic q8/2.
:- dynamic q_temp/1.
:- dynamic q_temp1/2.
:- dynamic q_temp2/2.
:- dynamic qc/1.

facts.pl

`:- dynamic apoysia/5.`

`apoysia(1, 1, 2, february, 6).`

`apoysia(2, 1, 6, february, 6).`

`apoysia(3, 1, 7, february, 6).`

`apoysia(4, 1, 1, february, 6).`

`apoysia(1, 2, 4, march, 7).`

`:- dynamic dikaiologhsh/5.`

`dikaiologhsh(1, 1, february, 7, 60).`

`dikaiologhsh(2, 1, february, 7, 40).`

`dikaiologhsh(3, 1, february, 7, 35).`

`dikaiologhsh(4, 1, february, 7, 1).`

`:- dynamic parastatiko/2.`

`:- dynamic parastatiko_apo_forea/3.`

input.pl

find_parast(S,N):-

```
\+ (parastatiko(S,I),\+q2(I)),
N is 0,d_q2,
!,true.
```

find_parast(S,N):-

```
parastatiko(S,I),\+q2(I),assert(q2(I)),
find_parast_apo_forea(S,N1),N is N1+1,
!,true.
```

find_parast_apo_forea(S,N):-

```
\+ (parastatiko_apo_forea(S,I,_),\+q2(I)),
N is 0,
d_q2,
!,true.
```

find_parast_apo_forea(S,N):-

```
parastatiko_apo_forea(S,I,_),\+q2(I),
assert(q2(I)),
find_parast_apo_forea(S,N1),
N is N1+1,
!,true.
```

find(N):-

```
\+ student(_,_),
N is 0,
!,true.
```

find(N):-

```
student(N,_),
(\+ (student(I,_),I>N) ),
!,true.
```

find_c(N):-

```
\+ class(_),N is 0,
!,true.
```

find_c(N):-

```
class(N,_)
,(\+ (class(I,_) ,I>N) ),
!,true.
```

find_diakopes(N):-

```
\+ diakopes(____),N is 0,!,true.
```

find_diakopes(N):-

```
diakopes(N,_)
,(\+ (diakopes(I,_) ,I>N) ),!,true.
```

find_p_e(N):-

```
\+ periptwsh(____),N is 0,!,true.
```

find_p_e(N):-

```
periptwsh(N,_)
,(\+ (periptwsh(I,_) ,I>N) ),!,true.
```

find_a_a(N,A):-

```
\+ apoysia(N,____),A is 0,!,true.
```

find_a_a(N,A):-

```
apoysia(N,A,____)
,(\+ (apoysia(N,I,____) ,I>A) ),!,true.
```

input_student(S,O,C):-

```
\+ student(_O,C),
find(N),N1 is N+1,
class(C,_)
S=N1,
assert(student(N1,O,C)),
update_student,
!,true.
```

s_student(S,E):-

```
S>E,!,true.
```

s_student(S,E):-

```
(S<E;S=E),
student(S,O,C),
S1 is S-1,
assert(student(S1,O,C)),
retract(student(S,_)),
( ( mo(S,G), assert(mo(S1,G)),
retract(mo(S,_)));true),
( ( diagogh(S,D), assert(diagogh(S1,D)),
retract(diagogh(S,_)));true),
```

S2 is S+1,

```
s_student(S2,E),
!,true.
```

delete_parastatiko(K):-

```
parastatiko(K,_),
K1 is K,
retract(parastatiko(K,_)),
```

delete_parastatiko(K1),

!,true.

delete_parastatiko(_):-

!,true.

delete_parastatiko_apo_forea(K):-

```
parastatiko_apo_forea(K,_),
K1 is K,
retract(parastatiko_apo_forea(K,_)),
```

delete_parastatiko_apo_forea(K1),

!,true.

delete_parastatiko_apo_forea(_):-

!,true.

delete_student(K):-

```
student(K,_),find(N),
```

```

delete_parastatiko(K),
delete_parastatiko_apo_forea(K),
K1 is K+1,
K2 is K,
( retract(student(K2,_));true),
( retract(mo(K2,_));true),
( retract(diagogh(K2,_));true),
s_student(K1,N),
delete_apoysia(K2),
delete_dikaiologhsh(K2),
shift1_apoysies(K1,N),
shift_dikaios(K1,N),
update_student,
update_facts,
!,true.

```

delete_apoysia(S):-

```

\+ apoysia(S,_),
!,true.

```

delete_apoysia(S):-

```

apoysia(S,_),
retract(apoysia(S,_)),
delete_apoysia(S),
!,true.

```

delete_dikaiologhsh(S):-

```

dikaiologhsh(S,_),
retract(dikaiologhsh(S,_)),
delete_dikaiologhsh(S),
!,true.

```

delete_dikaiologhsh(S):-

```

\+ dikaiologhsh(S,_),
!,true.

```

shift1_apoysia(S):-

```

apoysia(S,N,X,Y,Z),
S1 is S-1,
( ( retract(apoysia(S,N,X,Y,Z)), assert(apoysia(S1,N,X,Y,Z)));true),
shift1_apoysia(S),

```

```

!,true.

shift1_apoysia(S):-
  \+ apoysia(S,_,_,_),
  !,true.

shift1_apoysies(S,E):-

  S>E,!,true.

shift1_apoysies(S,E):-

  (S<E;S=E),
  shift1_apoysia(S),
  S1 is S+1,
  shift1_apoysies(S1,E),
  !,true.

shift1_apoysies(_):-

  !,true.

shift1_dikaio(S):-

  dikaiologhsh(S,N,X,Y,Z),
  S1 is S-1,
  S2 is S,

  ( ( retract(dikaiologhsh(S,N,_,_)),assert(dikaiologhsh(S1,N,X,Y,Z)),
    ( ( parastatiko(S,N),assert(parastatiko(S1,N)),retract(parastatiko(S1,_)));
    (
parastatiko_apo_forea(S,N,F),assert(parastatiko_apo_forea(S1,N,F)),retract(parastatiko_apo_f
orea(S1,_)))
;true));true),
  shift1_dikaio(S2),
  !,true.

shift1_dikaio(S):-

  \+ dikaiologhsh(S,_,_,_),!,true.

shift_dikaios(S,E):-

  S>E,!,true.

shift_dikaios(S,E):-

  (S<E;S=E),
  shift1_dikaio(S),
  S1 is S+1,

```

shift_dikaios(S1,E),
!,true.

shift_dikaios(_):-

!,true.

input_mo(S,G):-

student(S,_),
klimaka(L,R),
G >= L,
(G < R; G = R),
(retract(mo(S,_));true),
assert(mo(S,G)),
update_student,
!,true.

input_diagogh(S,D):-

student(S,_),
epipeda_diagoghs(F),
F = D,
(retract(diagogh(S,_));true),
assert(diagogh(S,D)),
update_student,
!,true.

s_s_d_a(S,M,D,R):-

\+ (apoysia(S,A,_M,D),\+ q2(A)),d_q2,
R is 0,
!,true.

s_s_d_a(S,M,D,R):-

apoysia(S,A,X,M,D),\+ q2(A),
assert(q2(A)),
s_s_d_a(S,M,D,R1),
R is R1+X,
!,true.

input_apoysia(S,Noa,M,H):-

\+ apoysia(S,_Noa,M,H),
student(S,_),
find_a_a(S,N),N1 is N+1,
Noa > 0,
hmerhsio_orio_didaskalias(M2),(Noa < M2; Noa = M2),
mhnas(_M1,_),M1 = M,


```

day_count(M1,H,Nd),
\+ elegxos_mh_erg(Nd),
s_s_d_a(S,M,H,R), R1 is R+Noa,(R1<M2;R1=M2),
assert(apoysia(S,N1,Noa,M,H)),
update_facts,
!,true.

```

input_apoysia(____):-

```
!,false.
```

input_dikaiologhsh(S,A,Md,Hd,C):-

```

\+ dikaiologhsh(S,A,Md,Hd,_),
student(S,_),
apoysia(S,A,_Ma,Ha),
mhnas(_Ma,Na), mhnas(_Md,Nd), Sa is Na+Ha,
Sd is Nd+Hd, (Sa<Sd;Sa=Sd),
day_count(Md,Hd,N),
\+ elegxos_mh_erg(N), day_count(Ma,Ha,N1), N2 is N-N1,orio_hmerwn(P),
( N2<P;N2=P),
( apallagh(C,____);apoysies_poy_den_dikaiologoyntai(C,_);
dikaiologhmenh_logo_astheneias(C,_);
dikaiologhthei(C,_);
adikaiologhth(C,_),
assert(dikaiologhsh(S,A,Md,Hd,C)),
update_facts,
!,true.

```

input_dikaiologhsh(____):-

```
!,false.
```

input_parastatiko(S,A):-

```

\+ parastatiko(S,A),
student(S,_),
apoysia(S,A,____),
dikaiologhsh(S,A,____E),
dikaiologhthei(E,_),
find_parast(S,Np),
orio_parastatikwn(Npl),
Np1 is Np+1,
( Np1 < Npl;Np1=Npl),
assert(parastatiko(S,A)),
update_facts,
!,true.

```

input_parastatiko_apo_forea(S,A,F):-

```

\+ parastatiko_apo_forea(S,A,F),
student(S,_),
apoysia(S,A,_),
dikaiologhsh(S,A,_E),
dikaiologhmenh_logo_astheneias(E,_),
find_parast_apo_forea(S,Np),
orio_parastatikwn(Npl),
Np1 is Np+1,
( Np1 < Npl;Np1=Npl),
assert(parastatiko_apo_forea(S,A,F)),
update_facts,
!,true.

```

test_range(L1,R1,L2,R2):-

```

( L1>= L2,(L1<R2;L1=R2));
( L2>=L1,(L2<R1;L2=R1));
( L1<L2,R1>R1);
( L2<L1,R2>R1).

```

input_per_eparkeias(Per,Diagwgh,Synolo1,Synolo2,Synolo_a1,Synolo_a2,Synolo_d_a1,Synolo_d_a2,Synolo_d1,Synolo_d2,Grade1,Grade2):-

```

\+ periptwsh(_Per),
find_p_e(N),
N1 is N+1,
klimaka(Kl,Kr),
( ( Grade2 >= Grade1,
Grade1>=Kl,(Grade1<Kr;Grade1==Kr),
Grade2>=Kl,(Grade2<Kr;Grade2==Kr));
( Grade1== -1,Grade2 == -1)),
( ( epipeda_diagoghs(Epd),Epd==Diagwgh);Diagwgh==" ),
arithmos_periptwsewn_eparkeias(K),
K1 is K+1,
retract(arithmos_periptwsewn_eparkeias(K)),
assert(arithmos_periptwsewn_eparkeias(K1)),
assert(periptwsh(N1,Per)),
( ( \+ Diagwgh==" , assert(exei_diagogh(N1,Diagwgh)));true),
( ( \+ ( Synolo1 == -1,Synolo2 == -1),assert(synolo(N1,Synolo1,Synolo2)));true),
( ( \+ ( Synolo_a1 == -1,Synolo_a2 == -1),
assert(synolo_a(N1,Synolo_a1,Synolo_a2)));true),
( ( \+ ( Synolo_d1 == -1,Synolo_d2 == -1),
assert(synolo_d(N1,Synolo_d1,Synolo_d2)));true),
( ( \+ ( Synolo_d_a1 == -1 ,Synolo_d_a2 == -1),
assert(synolo_d_a(N1,Synolo_d_a1,Synolo_d_a2)));true),
( ( \+ ( Grade1 == -1,Grade2 == -1), assert(grades(N1,Grade1,Grade2)));true),

update_per_epa,
!,true.

```

input_mhnas(I,_):-

```
I>11,  
update_dates,  
!,true.
```

input_mhnas(I,S,Disekto):-

```
mhnas(I,P,X),  
retract(mhnas(I,P,X)),  
assert(mhnas(I,P,S)),  
I1 is I+1,(  
  ( I=3;I=5;I=7;I=9;I=10;I=11 ), S1 is S+31 );  
  ((I=2;I=4;I=8;I=1), S1 is S+32);  
  ( I=6,Disekto=1,S1 is S+30);  
  ( I=6,Disekto=0,S1 is S+29)  
),  
input_mhnas(I1,S1,Disekto),  
!,true.
```

input_sunday(X):-

```
( retract(sunday(_));true),  
assert(sunday(X))  
,update_dates,  
!,true.
```

input_saturday(X):-

```
( retract(saturday(_));true),  
assert(saturday(X)),  
X1 is X+1,  
( retract(sunday(_));true),  
assert(sunday(X1)),  
update_dates,  
!,true.
```

input_diakopes(X1,X2):-

```
find_diakopes(N),  
N1 is N+1,  
( X2>X1;X2=X1),  
\+ ( diakopes(_Y1,Y2),  
  ( ( X1>Y1,X1<Y2);  
    ( X2>Y1,X2<Y2))),  
assert(diakopes(N1,X1,X2)),  
update_dates,  
!,true.
```

reset_diakopes:-

```
\+ diakopes(_ _),  
update_dates,  
!,true.
```

reset_diakopes:-

```
diakopes(X,_),  
retract(diakopes(X,_)),  
reset_diakopes,  
!,true.
```

Input_delete_menu.pl

```
disp_ap_student(A,_):-
```

```
    \+ apoysia(A,_,_,_),
    w_label2('Ο μαθητής δεν έχει απουσίες'),
    !,true.
```

```
disp_ap_student(A,_):-
```

```
    \+ (apoysia(A,B,_,_), \+ q2(B)),
    d_q2,
    !,true.
```

```
disp_ap_student(A,T):-
```

```
    apoysia(A,B,H,M,D),\+ q2(B),write(B),
    send(T,append(B)),
    send(T,append(M)),
    send(T,append(D)), send(T,append(H)),
    ( dikaiologhsh(A,B,M1,D1,T1)->
      ( send(T,append(M1)),
        send(T,append(D1)),
        (      (      apallagh(T1,_,_),send(T,append('Απαλλαγή')));
          (      apoysies_poy_den_dikaiologoyntai(T1,_,_) ,send(T,append('Δεν δικαιολογείται')));
          (      dikaiologhmenh_logos_astheneias(T1,_,_) ,send(T,append('Δικαιολογημένη λόγω
ασθενείας')));
          (      dikaiologhthei(T1,_,_) ,send(T,append('Δικαιολογημένη')));
          (      adikaiologhth(T1,_,_) ,send(T,append('Αδικοιολόγητη')))),
        (      [      parastatiko(A,B); parastatiko_apo_forea(A,B,_) ,
          send(T,append('υπάρχει'));send(T,append(' δεν υπάρχει')));
          send(T,append('-----',colspan:=2))),
          send(T,next_row),assert(q2(B)),disp_ap_student(A,T),
          !,true.
```

```
count_st_case(C,N):-
```

```
    \+ (      student(Y,_,_),
    test_student_aa(Y,C),
    \+ q8(Y,_,_),
    d_q8,
    N is 0,
    !,true.
```

```
count_st_case(C,N):-
```

```
    student(Y,_,_),
```

```

test_student_aa(Y,C),
\+ q8(Y,_),
assert(q8(Y,C)),
count_st_case(C,N1),
N is N1+1,
!,true.

```

list_dpl_case(T):-

```

\+ (   periptwsh(C1,_),periptwsh(C2,_),\+ C1=C2,((\+ q2(C2));(\+ q2(C1))),
      student(X,_),test_student_aa(X,C1),test_student_aa(X,C2)),
send_dpl_report(T),
!,true.

```

list_dpl_case(T):-

```

periptwsh(C1,_),periptwsh(C2,_),\+ C1=C2,
\+ ( q2(C1), q2(C2)),student(X,_),test_student_aa(X,C1),
test_student_aa(X,C2),
( \+ q2(C1),assert(q2(C1));true),
( \+ q2(C2),assert(q2(C2));true),
list_dpl_case(T),
!,true.

```

send_dpl_report(_):-

```

\+ q2(_),
!,true.

```

send_dpl_report(T):-

```

q2(C),periptwsh(C,Cper),
count_st_case(C,N),send(T,append(Cper)),send(T,append(N)),
send(T,next_row),
send(T,append('Μαθητές:')),
send(T,next_row),
find_students(C),
send_dpl_report1(T),
send(T,next_row),
send(T,append('-----',colspan:=2)),
send(T,next_row),
retract(q2(C)),
write(C),
send_dpl_report(T),
!,true.

```

find_students(X):-

```
\+ ( student(C,_) , \+ q_temp(C), test_student_aa(C,X)),  
!,true.
```

find_students(X):-

```
student(C,_) , \+ q_temp(C), test_student_aa(C,X), assert(q_temp(C)),  
find_students(X),  
!,true.
```

send_dpl_report1(_):-

```
\+ q_temp(_),  
!,true.
```

send_dpl_report1(T):-

```
q_temp(C), student(C,O,_) ,  
send(T,append(O)),  
send(T,next_row),  
retract(q_temp(C)),  
send_dpl_report1(T),  
!,true.
```

d_k_apallaghs(X):-

```
apallagh(C,P,_) , \+ q2(C),  
send(X,append,new(_menu_item(C,@default,P))),  
assert(q2(C)),  
d_k_apallaghs(X),  
!,true.
```

d_k_apallaghs(_):-

```
\+ (apallagh(C,_,_) , \+ q2(C)), d_q2,  
!,true.
```

d_k_den_dikaio(X):-

```
apoysies_poy_den_dikaiologoyntai(C,P) , \+ q2(C),  
send(X,append,new(_menu_item(C,@default,P))),  
assert(q2(C)),  
d_k_den_dikaio(X),  
!,true.
```

d_k_den_dikaio(_):-

```
\+ (apoysies_poy_den_dikaiologoyntai(C,_) , \+ q2(C)), d_q2,  
!,true.
```

d_k_dikaio(X):-

```

dikaiologhmenh_logo_astheneias(C1,_),
dikaiologhthei(C2,_),
send(X,append,new(_menu_item(0,@default,'οχι'))),
send(X,append,new(_menu_item(C1,@default,'δικαιολόγηση λόγω ασθένειας'))),
send(X,append,new(_menu_item(C2,@default,'δικαιολόγηση'))),
!,true.

```

menu_dpl_case(X):-

```

send(X,destroy),
new(BT,dialog('Κατάσταση συνθηκών επάρκειας που ταυτόχρονα χαρακτηρίζουν
μαθητή')),
send(BT,size,size(650,500)),
send(BT,append,new(P,picture("))),
send(P,size,size(600,400)),
send(P,display,new(T,tabular)),
send(T,border,1),
collision(N),
send(T,append('Τρέχουσα κατάσταση συστήματος:')), send(T,next_row),
send(T,append('Αριθμός Πλεονασμών:')), send(T,append(N)), send(T,next_row),
send(T,append('=====
=====','colspan:=2)),
send(T,next_row),
send(T,append('Περιγραφή Συνθήκης')),
send(T,append('Αριθμός μαθητών στις οποίες εμπλέκεται')),
send(T,next_row),
send(T,append('_____','colspan:=2)),
send(T,next_row),
list_dpl_case(T),
send(BT,append,new(B2,button(ok,message(@prolog,
menu_1,BT) ))),

get(BT,confirm,B2),
!,true.

```

menu_search(X):-

```

send(X,destroy),
new(D2,dialog('Αναζήτηση Μαθητή')),
send(D2,append,new(T1,menu('Αα μαθητή','cycle'))),
pop_student(T1),
send(D2,append,new(B2,button(ok,message(@prolog,
menu_sel1,D2,60,T1?selection) ))),
send(D2,append,new(B0,button(cancel,message(@prolog, esc_2,D2) ))),

get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.

```

menu_search_apoysies(X):-


```

send(X,destroy),
new(D2,dialog('Εμφάνιση απουσιών Μαθητή')),
send(D2,append,new(T1,menu('Αα μαθητή ',cycle))),
pop_student(T1),
send(D2,append,new(B2,button(ok,message(@prolog.menu_sel1,D2,600,T1?selection)
))),
send(D2,append,new(B0,button(cancel,message(@prolog.esc_2,D2)  )),

get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.

```

menu_mhnas1(X):-

```

send(X,destroy),
new(D2,dialog('Ημερομηνία έναρξης σχολικής χρονιάς')),
send(D2,append,new(T1,text_item('αα ημέρας Σεπτεμβρίου',0))),
send(D2,append,new(T2,text_item('Δίσεκτο (1/0)',0))),
send(D2,append,new(B2,button(ok,message(@prolog.menu_sel1,D2,50,T1?selection,T2?
selection)))),
send(D2,append,new(B0,button(cancel,message(@prolog.esc_2,D2)  )),

get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.

```

menu_diakop(X):-

```

send(X,destroy),
new(D2,dialog('Περίοδο διακοπών')),
find_diakopes(N),N1 is N+1,
send(D2,append,new(_text_item('αα ',N1))),
send(D2,append,new(label(a,'Από'))),
send(D2,append,new(M1,menu('Μήνες',cycle))),
s_mhnas(M1),
send(D2,append,new(T2,menu('αα ημέρας',cycle))),
s_day(1,T2),
send(D2,append,new(label(b,'Μέχρι'))),
send(D2,append,new(M2,menu('Μήνες',cycle))),
s_mhnas(M2),
send(D2,append,new(T3,menu('αα ημέρας',cycle))),
s_day(1,T3),

send(D2,append,new(B2,button(ok,message(@prolog,
menu_sel1,D2,40,M1?selection,T2?selection,M2?selection,T3?selection))))),
send(D2,append,new(B0,button(cancel,message(@prolog.esc_2,D2)  )),

get(D2,confirm,B0),

```

```
get(D2,confirm,B2),
!,true.
```

menu_diakop_d(X):-

```
send(X,destroy),
reset_diakopes,
new(D2,dialog('Μήνυμα')),
send(D2,append,new(label(a,'Οι περιοδοι διεγραφησαν'))),
send(D2,append,new(B2,button(ok,message(@prolog,menu_1,D2))))),
send(D2,append,new(B0,button(cancel,message(@prolog,esc_2,D2) ))),
```

```
get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.
```

menu_epa1(X):-

```
send(X,destroy),
new(D2,dialog('Νεα συνθήκη για τον χαρακτηρισμό επάρκειας φοίτησης')),
find_p_e(N),N1 is N+1,
send(D2,append,new(T0,text_item('Aa',N1))),
send(D2,append,new(T1,text_item("Όνομασία","))),
send(D2,append,new(label(t,' Η ΤΙΜΉ -1 ΣΤΑ ΠΕΔΙΑ ΠΟΥ ΑΚΟΛΟΥΘΟΥΝ ΣΗΜΑΪΝΕΙ
'ΟΤΙ ΤΟ ΠΕΔΙΟ ΔΕΝ ΛΑΜΒΑΝΕΤΑΙ ΥΠΟΨΗ'))),
send(D2,append,new(label(l1,'Σύνολο απουσιών που λαμβάνονται υπόψη'))),
send(D2,append,new(T2,menu('Από',cycle))),
n_list(-1,T2),
send(D2,append,new(T3,menu('Μέχρι',cycle))),
n_list(-1,T3),
send(D2,append,new(label(l2,'Σύνολο απουσιών που μπορούν να δικαιολογηθούν'))),
send(D2,append,new(T4,menu('Από',cycle))),
n_list(-1,T4),
send(D2,append,new(T5,menu('Μέχρι',cycle))),
n_list(-1,T5),
send(D2,append,new(label(l3,'Σύνολο απουσιών που μπορούν να δικαιολογηθούν λόγω
ασθένειας'))),
send(D2,append,new(T6,menu('Από',cycle))),
n_list(-1,T6),
send(D2,append,new(T7,menu('Μέχρι',cycle))),
n_list(-1,T7),
send(D2,append,new(label(l4,'Σύνολο απουσιών που είναι αδικαιολόγητες'))),
send(D2,append,new(T8,menu('Από',cycle))),
n_list(-1,T8),
send(D2,append,new(T9,menu('Μέχρι',cycle))),
n_list(-1,T9),
send(D2,append,new(label(l4,'Περιοχή βαθμολογίας'))),
send(D2,append,new(T10,menu('Από',cycle))),
n_list(-1,T10),
send(D2,append,new(T11,menu('Μέχρι',cycle))),
n_list(-1,T11),
```

```

send(D2,append,new(T12,menu('Διαγωγή:',cycle))),
send(T12,append,new(_menu_item("@default,"))),
pop_diagogh(T12),
send(D2,append,new(B2,button(ok,message(@prolog,
menu_sel1,D2,10,T0?selection,T1?selection,T2?selection,T3?selection,T4?selection,T5?sel
ection,
T6?selection,T7?selection,T8?selection,T9?selection,T10?selection,T11?selection,T12?sel
ection )))),
send(D2,append,new(B0,button(cancel,message(@prolog,esc_2,D2) )),

get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.

```

menu_parast_f1(X):-

```

send(X,destroy),
new(D2,dialog('Παραστατικό από φορέα')),
win_disp_ap_la(D2),
send(D2,append,new(T1,menu(κωδικός_μαθητή,cycle))),
pop_student(T1),
send(D2,append,new(T2,text_item(αα_απουσίας,0))),
send(D2,append,new(T3,text_item('φορέας δικαιολόγησις,'))),
send(D2,append,new(B2,button(ok,message(@prolog,
menu_sel1,D2,7,T1?selection,T2?selection,T3?selection)))),
send(D2,append,new(B0,button(cancel,message(@prolog,esc_2,D2) )),

get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.

```

menu_parast1(X):-

```

send(X,destroy),
new(D2,dialog('Παραστατικό')),
win_disp_ap_d(D2),
send(D2,append,new(T1,menu(κωδικός_μαθητή,cycle))),
pop_student(T1),
send(D2,append,new(T2,text_item(αα_απουσίας,0))),
send(D2,append,new(B2,button(ok,message(@prolog,
menu_sel1,D2,6,T1?selection,T2?selection)))),
send(D2,append,new(B0,button(cancel,message(@prolog,esc_2,D2) )),

get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.

```

menu_saturday1(X):-

```
send(X,destroy),
new(D2,dialog('Ημερομηνία πρώτου σαββάτου')),
send(D2,append,new(T1,menu('αα ημέρας',cycle))),
s_day(1,T1),
send(D2,append,new(B2,button(ok,message(@prolog,
menu_sel1,D2,21,T1?selection)))),
send(D2,append,new(B0,button(cancel,message(@prolog, esc_2,D2)  ))),

get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.
```

menu_student1(X):-

```
send(X,destroy),
new(D2,dialog('Μαθητής')),
find(N),N1 is N+1,
send(D2,append,new(T1,text_item(κωδικός,N1))),
send(D2,append,new(T3,text_item(όνομα,"))),
send(D2,append,new(T2,menu('τάξη',cycle))),
m_class(T2),
send(D2,append,new(B2,button(ok,message(@prolog,menu_sel1,D2,1,T1?selection,T3?s
election,T2?selection)))),
send(D2,append,new(B0,button(cancel,message(@prolog,esc_2,D2)  ))),

get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.
```

menu_student01(X):-

```
send(X,destroy),
new(D2,dialog('Μαθητής προς διαγραφή - διόρθωση')),
send(D2,append,new(T1,menu(κωδικός,cycle))),
pop_student(T1),
send(D2,append,new(B2,button(ok,message(@prolog,
menu_sel1,D2,20,T1?selection)))),
send(D2,append,new(B0,button(cancel,message(@prolog,esc_2,D2)  ))),

get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.
```

menu_mo1(X):-

```
send(X,destroy),
new(D3,dialog('Μέσος όρος')),
send(D3,append,new(T1,menu(κωδικός,cycle))),
pop_student(T1),
send(D3,append,new(T2,text_item(βαθμός,0))),
send(D3,append,new(B2,button(ok,message(@prolog,
menu_sel1,D3,2,T1?selection,T2?selection)))),
send(D3,append,new(B0,button(cancel,message(@prolog,esc_2,D3)  ))),

get(D3,confirm,B0),
get(D3,confirm,B2),
!,true.
```

menu_dia1(X):-

```
send(X,destroy),
new(D4,dialog('διαγωγή μαθητή')),
send(D4,append,new(T1,menu('κωδικός',cycle))),
pop_student(T1),
send(D4,append,new(T2,menu('διαγωγή',cycle))),
pop_diagogh(T2),
send(D4,append,new(B2,button(ok,message(@prolog,
menu_sel1,D4,3,T1?selection,T2?selection)))),
send(D4,append,new(B0,button(cancel,message(@prolog,esc_2,D4)  ))),

get(D4,confirm,B0),
get(D4,confirm,B2),
!,true.
```

menu_ap1(X):-

```
send(X,destroy),
new(D4,dialog('Καταχώρηση Απουσίας μαθητή')),
send(D4,append,new(T1,menu('Κωδικός Μαθητή',cycle))),
pop_student(T1),
send(D4,append,new(T2,text_item('Αριθμός_απουσιών',0))),
send(D4,append,new(D,menu('Μήνες',cycle))),
s_mhnas(D),
send(D4,append,new(T4,menu('Ημερα',cycle))),
s_day(1,T4),
send(D4,append,new(B2,button(ok,message(@prolog,
menu_sel1,D4,4,T1?selection,T2?selection,D?selection,T4?selection)))),
send(D4,append,new(B0,button(cancel,message(@prolog,esc_2,D4)  ))),

get(D4,confirm,B0),
get(D4,confirm,B2),
```

!,true.

menu_di1(X):-

```
send(X,destroy),
new(D5,dialog('Δικαιολογηση Απουσίας μαθητή')),
win_disp_ap(D5),
send(D5,append,new(T1,menu('Αα_Μαθητή',cycle))),
pop_student(T1),
send(D5,append,new(T2,text_item('Αα_απουσίας',0))),
send(D5,append,new(D,menu('Μήνες',cycle))),
s_mhnas(D),
send(D5,append,new(T4,menu('Ημερα',cycle))),
s_day(1,T4),
send(D5,append,new(L1,menu('Περιπτώσεις απαλλαγής:',cycle))),
send(L1,append,new(_menu_item(0,@default,'όχι'))),
d_k_apallaghs(L1),
send(D5,append,new(L2,menu('Περιπτώσεις που δεν δικαιολογούνται:',cycle))),
send(L2,append,new(_menu_item(0,@default,'όχι'))),
d_k_den_dikaio(L2),
send(D5,append,new(L3,menu('Περιπτώσεις που δικαιολογούνται:',cycle))),
d_k_dikaio(L3),
send(D5,append,new(B2,button(ok,message(@prolog,
menu_sel1,D5,5,T1?selection,T2?selection,D?selection,T4?selection,L1?selection,L2?select
ion,L3?selection))))),
send(D5,append,new(B0,button(cancel,message(@prolog,esc_2,D5)  ))),

get(D5,confirm,B0),
get(D5,confirm,B2),
!,true.
```

menu_sel11(X,S,K,O,C,G,D):-

```
S==2,
send(X,destroy),
retract(student(K,_)),assert(student(K,O,C)),
((
\+
G
==
-
1, klimaka(L,R),G>=L,(G<R;G==R),(retract(mo(K,_));true),assert(mo(K,G));true),
(\+
D
==
'Καμμιά',(retract(diagogh(K,_));true),assert(diagogh(K,D));true),update_student,
w_label2('Η διόρθωση έγινε'),
!,true.
```

menu_sel1(X,Ss,A):-

```
Ss==60,
send(X,destroy),
new(D2,dialog('Στοιχεία Μαθητή')),
send(D2,area(area(1,1,1000,900))),
```

```

kartella(Onoma,G,D,A,Cname,S,N1,N2,Epidosh),
s_ypologismos_m(A,S1),
synolo_a_y(A,S2),

send(D2,append,new(_text_item('Aa',A))),
send(D2,append,new(_text_item('Όνομα',Onoma))),
send(D2,append,new(_text_item('Τάξη',Cname))),
send(D2,append,new(_text_item('Βαθμός',G))),
send(D2,append,new(_text_item('Διαγωγή',D))),
send(D2,append,new(_text_item('Σύνολο απουσιών που εμπίπτουν σε περίπτωση
απαλλαγής αλλά υπερβαίνουν όριο',S1))),
send(D2,append,new(_text_item('Σύνολο απουσιών που δεν λαμβάνονται υπόψη',S))),
send(D2,append,new(_text_item('Σύνολο απουσιών που λαμβάνονται υπόψη στον
χαρακτηρισμό φοίτησης',S2))),
send(D2,append,new(_text_item('Σύνολο απουσιών που έχουν δικαιολογηθεί λόγω
ασθένειας',N1))),
send(D2,append,new(_text_item('Σύνολο απουσιών που έχουν δικαιολογηθεί',N2))),
send(D2,append,new(T1,text_item('Επίδοση:',Epidosh))),
send(T1,area(area(1,1,400,20))),
win_disp_case(D2,A),
send(D2,append,new(B2,button(ok,message(@prolog, menu_1,D2) )),

get(D2,confirm,B2),
!,true.

```

menu_sel1(X,S,P1):-

```

S==21,
( \+ number(P1)-> X1 is 0;X1 is P1),
input_saturday(X1), send(X,destroy),
w_label2('η εγγραφή έγινε'),
!,true.

```

menu_sel1(X,S,P1):-

```

S==20,
send(X,destroy),
new(D3,dialog('Μαθητής')),
send(D3,size,size(450,750)),
student(P1,O,C),class(C,Cn),
(mo(P1,G);G is -1),(diagogh(P1,D);D="),
send(D3,append,new(P,picture(""))),
send(P,size,size(300,200)),
send(P,display,new(T,tabular)),
send(T,border,1),
send(T,append('Κωδικός')), send(T,append(P1)),send(T,next_row),

```

```

send(T,append('όνομα')), send(T,append(O)),send(T,next_row),
send(T,append('τάξη')), send(T,append(Cn)),send(T,next_row),
send(T,append(' Μέσος όρος')),((mo(P1,G), send(T,append(G)));send(T,append('Δεν
υπάρχει')),send(T,next_row),
send(T,append('Διαγωγή')),((diagogh(P1,D), send(T,append(D)));send(T,append('Δεν
υπάρχει')),send(T,next_row),
send(D3,append,new(T3,text_item(όνομα,O)),below),
send(D3,append,new(T2,menu('τάξη',cycle))),
m_class(T2),
send(D3,append,new(T4,text_item('Μέσος όρος (-1 για μη μεταβολή)',G)),below),
send(D3,append,new(T5,menu('Διαγωγή ( Καμμία για μη μεταβολή)',cycle))),
send(T5,append,new(_menu_item('Καμμία',@default,'Καμμία'))),pop_diagogh(T5),
send(D3,append,new(B2,button('Διαγραφή',message(@prolog,menu_sel11,D3,1,P1))))),
send(D3,append,new(B3,button('Διόρθωση',message(@prolog,
menu_sel11,D3,2,P1,T3?selection,T2?selection,T4?selection,T5?selection))))),
send(D3,append,new(B0,button(cancel,message(@prolog, esc_2,D3) ))),

get(D3,confirm,B0),
get(D3,confirm,B3),
get(D3,confirm,B2),

!,true.

```

menu_sel11(X,S,K):-

```

S==1,
send(X,destroy),
delete_student(K),
w_label2('Η διαγραφή έγινε'),
!,true.

```

menu_sel1(_Ss,A):-

```

Ss==60,
\+ kartella(_Ss,A,_____),
new(D2,dialog('Στοιχεία Μαθητή')),
send(D2,append,new(label(l,'ελλιπή στοιχεία'))),
send(D2,append,new(B2,button(ok,message(@prolog, menu_1,D2) ))),

get(D2,confirm,B2),
!,true.

```

menu_sel1(X,Ss,A):-

```

Ss==600,
send(X,destroy),
student(A,O,_),
new(D0,dialog('Κατάσταση Απουσιών')),

send(D0,size,size(1200,600)),

```



```

send(D0,append,new(P,picture("")),
send(P,size,size(1050,500)),
send(P,display,new(T,tabular)),
send(T,border,1),
send(T,append('όνομα μαθητή')),    send(T,append(0)),    send(T,next_row),
send(T,append('Αα')),
send(T,append('Ημ. απουσίας',colspan:=2)),send(T,append('Ωρες')),
send(T,append('Ημ.                                δικαιολόγησης',colspan:=2)),
send(T,append('Αιτιολογία',colspan:=2)),
send(T,append('παραστατικό'),    send(T,next_row),
send(T,append('_____
_____','colspan:=8)),
send(T,next_row),

disp_ap_student(A,T),
send(T,next_row),

send(D0,append,new(B0,button(ok,message(@prolog,menu_1,D0))))),

get(D0,confirm,B0),
!,true.

```

menu_sel1(X,S,P1,P2):-

```

S==50,
( P1>0,P1<31),S1 is 30-P1+1,
input_mhnas(1,S1,P2),
send(X,destroy),
w_label2('η εγγραφή έγινε'),
!,true.

```

menu_sel1(X,S,P1,P2):-

```

S==3,
( \+ number(P1)-> X1 is 0;X1 is P1),send(X,destroy),
(input_diagogh(X1,P2)->w_label2('η εγγραφή έγινε');w_label2('η εγγραφή δεν έγινε')),
!,true.

```

menu_sel1(X,S,P1,P2):-

```

S==2,
( \+ number(P2)-> X2 is 0;X2 is P2),
( \+ number(P1)-> X1 is 0;X1 is P1),

```

```
send(X,destroy),
(input_mo(X1,X2)->w_label2('η εγγραφή έγινε');w_label2('η εγγραφή δεν έγινε')),
!,true.
```

```
menu_sel1(X,S,P1,P2):-
  S==6,
  ( \+ number(P2)-> X2 is 0;X2 is P2),
  ( \+ number(P1)-> X1 is 0;X1 is P1),send(X,destroy),
  (input_parastatiko(X1,X2)-> w_label2('η εγγραφή έγινε');w_label2('η εγγραφή δεν
έγινε')),
!,true.
```

```
menu_sel1(X,S,P1,O,P2):-

  S==1,
  number(P2),number(P1),send(X,destroy),
  (input_student(P1,O,P2)->w_label2('η εγγραφή έγινε');w_label2('η εγγραφή δεν έγινε')),
!,true.
```

```
menu_sel1(X,S,P1,P2,P3):-

  S==7,
  ( \+ number(P2)-> X2 is 0;X2 is P2),
  ( \+ number(P1)-> X1 is 0;X1 is P1),send(X,destroy),
  (input_parastatiko_apo_forea(X1,X2,P3)->w_label2('η εγγραφή
έγινε');w_label2('η
εγγραφή δεν έγινε')),
!,true.
```

```
menu_sel1(X,S,P1,P2,P3,P4,P5,P6,P7):-

  S==5,
  ( \+ number(P1)-> X1 is 0;X1 is P1),
  ( \+ number(P2)-> X2 is 0;X2 is P2),send(X,destroy),
  (( ( \+ number(P4)-> X4 is 0;X4 is P4),
  ( ( P5=0,P6=0,\+ P7=0, X5 is P7);
  ( \+ P5=0,P6=0, P7=0,X5 is P5);( P5=0,\+ P6=0, P7=0,X5 is P6);
  ( P5=0,P6=0,P7=0,adikaiologhth(N,_),X5 is N)),
  input_dikaiologhsh(X1,X2,P3,X4,X5))-> w_label2('η εγγραφή έγινε');w_label2('η εγγραφή
δεν έγινε')),
!,true.
```

```
menu_sel1(X,S,P1,P2,P3,P4):-

  S==4,
  ( \+ number(P2)-> X2 is 0;X2 is P2),
  ( \+ number(P4)-> X4 is 0;X4 is P4),send(X,destroy),
```

```
(input_apoysia(P1,X2,P3,X4)->w_label2('η εγγραφή έγινε');w_label2('η εγγραφή δεν έγινε')),  
!,true.
```

```
menu_sel1(X,S,P1,P2,P3,P4):-
```

```
S==40,  
send(X,destroy),  
(( number(P2),number(P4),  
day_count(P1,P2,T1),  
day_count(P3,P4,T2),  
T2>=T1,  
\+ (diakopes(_L,R),((T1>=L,(T1<R;T1=R));(T2>=L,(T2<R;T2=R))))),  
input_diakopes(T1,T2))-> w_label2('η εγγραφή έγινε');w_label2('η εγγραφή δεν έγινε')),  
!,true.
```

```
menu_sel1(X,S,P0,P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11,P12):-
```

```
S==10,  
( \+ number(P0)-> X0 is 0;X0 is P0),  
( \+ number(P2)-> X2 is 0;X2 is P2),  
( \+ number(P3)-> X3 is 0;X3 is P3),  
( \+ number(P4)-> X4 is 0;X4 is P4),  
( \+ number(P5)-> X5 is 0;X5 is P5),  
( \+ number(P6)-> X6 is 0;X6 is P6),  
( \+ number(P7)-> X7 is 0;X7 is P7),  
( \+ number(P8)-> X8 is 0;X8 is P8),  
( \+ number(P9)-> X9 is 0;X9 is P9),  
( \+ number(P10)-> X10 is 0;X10 is P10),  
( \+ number(P11)-> X11 is 0;X11 is P11),  
send(X,destroy),  
(input_per_eparkeias(P1,P12,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11) ->w_label2('η εγγραφή έγινε');  
w_label2('η εγγραφή δεν έγινε')),  
!,true.
```

```
menu_sel1(X,S):-
```

```
S==0,  
send(X,destroy),  
!,true.
```

```
esc_2(X):-
```

```
send(X,destroy)  
,menu_1(0),  
!,true.
```

win_disp_ap(D):-

```
send(D,append,new(P,picture("))),
send(P,size,size(400,400)),
send(P,display,new(T,tabular)),
send(T,border,1),
send(T,append('Υπάρχουσες Απουσίες')),send(T,next_row),
send(T,append('_____','colspan:=4)),send(T,next_row),
send(T,append('Όνομα Μαθητή')),send(T,append('Αα Απουσίας')),
send(T,append('Ημερομηνία','colspan:=2)), send(T,next_row),
win_disp_ap1(T)
!,true.
```

win_disp_ap1(_):-

```
\+ (apoysia(X,A,_,_), \+ dikaiologhsh(X,A,_,_), \+ q8(X,A)),
d_q8,
!,true.
```

win_disp_ap1(T):-

```
student(X,O,_),apoysia(X,A,_M,D), \+ dikaiologhsh(X,A,_,_), \+ q8(X,A),
send(T,append(O)),
send(T,append(A)),send(T,append(M)),send(T,append(D)),send(T,next_row),
assert(q8(X,A)),
win_disp_ap1(T),
!,true.
```

win_disp_case(D,X):-

```
send(D,append,new(P,picture("))),
send(P,size,size(300,200)),
send(P,display,new(T,tabular)),
send(T,border,1),
send(T,append('Συνθήκες που χαρακτηρίζουν τον μαθητή
επαρκή','colspan:=2)),send(T,next_row),
send(T,append('_____','colspan:=2)),send(T,next_row),
send(T,append('Αα συνθήκης')),send(T,append('Περίπτωση')), send(T,next_row),
win_disp_case1(T,X)
!,true.
```

win_disp_case1(_X):- \+ (periptwsh(Y,_), test_student_aa(X,Y), \+ q2(Y)),
d_q2,
!,true.

win_disp_case1(T,X):-

```

periptwsh(Y,A), test_student_aa(X,Y), \+ q2(Y),
send(T,append(Y)), send(T,append(A)),send(T,next_row),
assert(q2(Y)),
win_disp_case1(T,X),
!,true.

```

win_disp_ap_l_a(D):-

```

send(D,append,new(P,picture("))),
send(P,size,size(600,400)),
send(P,display,new(T,tabular)),
send(T,border,1),
send(T,append('Υπάρχουσες Απουσίες που έχουν δικαιολογηθεί λόγω
ασθένειας',colspan:=4)),send(T,next_row),
send(T,append('_____','colspan:=4)),send(T,next_row),

send(T,append("Όνομα Μαθητή")),
send(T,append('Αα Απουσίας')),send(T,append('Ημερομηνία',colspan:=2)),
send(T,next_row),
win_disp_ap_l_a1(T)
!,true.

```

win_disp_ap_l_a1(_):-

```

\+ (apoysia(X,A,_), dikaiologhsh(X,A,_T),
dikaiologhmenh_logo_astheneias(T,_),\+ parastatiko_apo_forea(X,A,_),
\+ q8(X,A)),
d_q8,
!,true.

```

win_disp_ap_l_a1(T):-

```

student(X,O,_),apoysia(X,A,_M,D),
dikaiologhsh(X,A,_T1),dikaiologhmenh_logo_astheneias(T1,_),
\+ parastatiko_apo_forea(X,A,_),
\+ q8(X,A),
send(T,append(O)),
send(T,append(A)),send(T,append(M)),send(T,append(D)),send(T,next_row),
assert(q8(X,A)),
win_disp_ap_l_a1(T),
!,true.

```

win_disp_ap_d(D):-

```

send(D,append,new(P,picture("))),
send(P,size,size(600,400)),
send(P,display,new(T,tabular)),
send(T,border,1),
send(T,append('Υπάρχουσες           Απουσίες           που           έχουν
δικαιολογηθεί',colspan:=4)),send(T,next_row),
send(T,append('_____','colspan:=4)),send(T,next_row),
send(T,append("Όνομα Μαθητή")),send(T,append('Αα Απουσίας')),
send(T,append('Ημερομηνία',colspan:=2)), send(T,next_row),
win_disp_ap_d1(T)
!,true.

```

win_disp_ap_d1(_):-

```

\+ (apoysia(X,A,_,_), dikaiologhsh(X,A,_,T),dikaiologhthei(T,_),
\+ parastatiko(X,A),\+ q8(X,A)),
d_q8,
!,true.

```

win_disp_ap_d1(T):-

```

student(X,O,_),apoysia(X,A,_,M,D), dikaiologhsh(X,A,_,T1),dikaiologhthei(T1,_),
\+ parastatiko(X,A),
\+ q8(X,A),
send(T,append(O)),
send(T,append(A)),send(T,append(M)),send(T,append(D)),send(T,next_row),
assert(q8(X,A)),
win_disp_ap_d1(T),
!,true.

```

m_menu.pl

statistics:-

```
tell('statistic.pl'),  
listing,  
told.
```

start:-

```
consult(dyn_pred),  
retractall(student(_,_)),  
retractall(mo(_)),  
retractall(diagogh(_)),  
retractall(apoysia(_,_,_)),  
retractall(dikaiologhsh(_,_,_,_)),  
retractall(parastatiko(_)),  
retractall(parastatiko_apo_forea(_,_)),  
retractall(hmerhsio_orio_didaskalias(_)),  
retractall(arithmos_class(_)),  
retractall(class(_)),  
retractall(arithmos_periptwsewn_apallaghs(_)),  
retractall(apallagh(_,_,_)),  
retractall(apoysies_poy_den_dikaiologoyntai(_)),  
retractall(dikaiologhmenh_logo_astheneias(_)),  
retractall(dikaiologhthei(_)),  
retractall(adikaiologhth(_)),  
retractall(orio_hmerwn(_)),
```

```
retractall(epipeda_diagoghs(_)),  
retractall(orio_parastatikwn(_)),  
retractall(mhnas(_,_)),  
retractall(sunday(_)),  
retractall(saturday(_)),  
retractall(diakopes(_,_)),  
retractall(klimaka(_)),  
consult(student),  
consult(facts),  
consult(input),  
consult(new_state1),  
consult(parameters),  
consult(per_eparkeias),  
consult(rules),  
consult(test_cons),  
consult(facts),  
consult(dates),
```

```

consult(belief_revision),
consult(input_delete_menu),
consult(param_menu),
consult(reset),
consult(update_knowledge),
consult(user_interface),
consult(sub_menus),
!,true.

```

m_menu(X):-

```

( X == 0-> start;send(X,destroy)),

new(D1,dialog('Κεντρικό Μενού')),

send(D1,size,size(400,250)),
send(D1,layout_dialog),
send(D1,append,new(BT,dialog_group(buttons,group))),
send(BT,gap,size(0,10)),
send(BT,alignment,center),
send(BT,append,new(B1,
button("Έλεγχος ορθότητας γνώσης',message(@prolog,
menu,D1))),below),
send(BT,append,new(B2,
button('Μεταβολές στη Γνώση',message(@prolog,menu_1,D1))),below),
send(BT,append,new(B3,button('Βοηθητικές λειτουργίες',message(@prolog,
menu_2,D1))),below),
send(BT,append,new(B6,button('Συγκεντρωτική Κατάσταση απουσιών μαθητών',
message(@prolog,display_database,D1))),below),
send(BT,append,new(B7,button('Συγκεντρωτική Κατάσταση Συνθηκών επάρκειας
φοίτησης',
message(@prolog,display_database2,D1))),below),
send(BT,append,new(B5,button('Αρχικοποίηση συστήματος Γνώσης',
message(@prolog,new_base,D1))),below),
send(BT,append,new(B4,button("Έξοδος',
message(@prolog,ok1,D1))),below),

get(D1,confirm,B1),
get(D1,confirm,B2),
get(D1,confirm,B3),
get(D1,confirm,B4),
get(D1,confirm,B5),
get(D1,confirm,B6),
get(D1,confirm,B7),

!,true.

```

new_base(X):-


```

send(X,destroy),

new(D1,dialog('Προσοχή!!!!')),
send(D1,append,new(label(t,'Είσαι σίγουρος'))),
send(D1,append,new(B1,button('Ναι',
message(@prolog,new_p,D1))),below),
send(D1,append,new(B2,button("Όχι",
message(@prolog,m_menu,D1))),below),

get(D1,confirm,B1),
get(D1,confirm,B2),

send(D1,destroy),
!,true.

```

display_database(X):-

```

send(X,destroy),
new(D0,dialog('Σγκεντρωτική Κατάσταση Απουσιών Μαθητών')),

send(D0,size,size(1200,600)),
send(D0,append,new(P,picture("))),
send(P,size,size(1050,500)),
send(P,display,new(T,tabular)),
send(T,border,1),
send(T,append('Αα Μαθητή')),
send(T,append("Όνομα")),send(T,append('Μ.Ο.')),
send(T,append('Διαγωγή')),send(T,append('Σύνολο απουσιών')),
send(T,append('Απουσίες δικαιολογημένες')),
send(T,append('Απουσίες δικαιολογημένες Λόγω Ασθενείας')),
send(T,append('Απουσίες Αδικαιολόγητες')),
send(T,next_row),
send(T,append('_____','colspan:=8)),
_____
send(T,next_row),
display_database1(T),

send(D0,append,new(B0,button(ok,
message(@prolog,m_menu,D0)))),

get(D0,confirm,B0),
!,true.

```

display_database1(_):-

```

\+ (student(X,_), \+ q2(X)),d_q2,

```

```

!,true.

display_database1(T):-
    student(X,O,_)\,+ q2(X),send(T,append(X)),send(T,append(O)),
    ( (mo(X,G),diagogh(X,D)) ->
    ( ypologismos_a_d(X,S1), ypologismos_a_a(X,S2), ypologismos_a_d_a(X,S3),
synolo_a_y(X,S4),
send(T,append(G)),send(T,append(D)),send(T,append(S4)),send(T,append(S1)),send(T,append(S3)),
    send(T,append(S2)));
    send(T,append('-----Ελλιπή Στοιχεία-----
-----',colspan:=6))),send(T,next_row),assert(q2(X)),
    display_database1(T),
    !,true.

```

```
display_database_cases1(_):-
```

```

\+ (periptwsh(X,_), \+ q2(X)),d_q2,
!,true.

```

```
display_database_cases1(T):-
```

```

periptwsh(X,O)\,+ q2(X),
send(T,append(X)),send(T,append(O)),
( grades(X,L1,R1) -> (
send(T,append(L1)),send(T,append('--')),send(T,append(R1)));
send(T,append('Χωρίς περιορισμό',colspan:=3))),
( exei_diagogh(X,D1) -> send(T,append(D1));
send(T,append('Χωρίς περιορισμό')),
( synolo(X,L2,R2) -> (send(T,append(L2)),send(T,append('--
')),send(T,append(R2)));
send(T,append('Χωρίς περιορισμό',colspan:=3))),
( synolo_d(X,L3,R3) -> (send(T,append(L3)),send(T,append('--
')),send(T,append(R3)));
send(T,append('Χωρίς περιορισμό',colspan:=3))),
( synolo_d_a(X,L4,R4) -> (send(T,append(L4)),send(T,append('--
')),send(T,append(R4)));
send(T,append('Χωρίς περιορισμό',colspan:=3))),
( synolo_a(X,L5,R5) -> (send(T,append(L5)),send(T,append('--
')),send(T,append(R5)));
send(T,append('Χωρίς περιορισμό',colspan:=3))),send(T,next_row),

assert(q2(X)),

```


new_state1.pl

`:- dynamic cd/3.`

`d_cd:-`

`cd(X,Y,Z),
 retract(cd(X,Y,Z)),
 d_cd,!true.`

`d_cd:- \+ cd(_,_),!true.`

`collision(N):-`

`\+ (student(X,_),
 periptwsh(C1,_),periptwsh(C2,_),\+ C1=C2,
 test_student_aa(X,C1), test_student_aa(X,C2),
 \+ cd(X,C1,_)),d_cd,N is 0,!true.`

`collision(N):-`

`student(X,_),
 periptwsh(C1,_),periptwsh(C2,_),\+ C1=C2,
 test_student_aa(X,C1), test_student_aa(X,C2),
 \+ cd(X,C1,_),
 assert(cd(X,C1,0)),
 collision(N1),N is N1+1,!true.`

`new_state(student(X,_Z)):-`

`find(N),N1 is N+1,N1=X,
 class(Z,_),
 !true.`

`new_state(mo(X,G)):-`

`student(X,_),
 \+ (mo(X1,_), X1=X),
 klimaka(L,R),
 G>=L,(G<R;G=R),!true.`

`new_state(diagogh(X,D)):-`

student(X,_),

\+ (diagogh(X1,_), X1=X),
epipeda_diagoghs(D1),
D1=D,!true.

new_state(apoysia(S,N,M,D)):-

student(S,_),

\+ apoysia(S,_N,M,D),
N>0,
hmerhsio_orio_didaskalias(M2),(N<M2;N=M2),
mhnas(_M1,_),M1=M,
day_count(M1,D,Nd),
\+ elegxos_mh_erg(Nd),
s_d(S,M,D,T),
T1 is T+N,
(T1<M2;T1=M2),!true.

new_state(dikaiologhsh(S,A,M,D,C)):-

apoysia(S,A,_M1,D1),

\+ dikaiologhsh(S,A,_,_),
mhnas(_M1,Na), mhnas(_M,Nd), Sa is Na+D1,
Sd is Nd+D, (Sa<Sd;Sa=Sd),
day_count(M,D,N),
\+ elegxos_mh_erg(N),
R is Sd-Sa,
orio_hmerwn(L),
(R<L;R=L),
(apallagh(C,_,_));
apoysies_poy_den_dikaiologoyntai(C,_);
dikaiologhmenh_logo_astheneias(C,_);
dikaiologhthei(C,_);
adikaiologhth(C,_),

!,true.

new_state(parastatiko(S,A)):-

\+ parastatiko(S,A),
apoysia(S,A,_,_),
dikaiologhsh(S,A,_,_E),
dikaiologhthei(E,d_a),

```

find_parast(S,N),
orio_parastatikwn(M),
N1 is N+1,
( N1<M ;N1=M),
!,true.

```

new_state(parastatiko_apo_forea(S,A,F)):-

```

\+ parastatiko_apo_forea(S,A,F),
apoysia(S,A,_,_),
dikaiologhsh(S,A,_,E),
dikaiologhmenh_logo_astheneias(E,d_l_a),
find_parast(S,N),
orio_parastatikwn(M),
N1 is N+1,
( N1<M ;N1=M),
!,true.

```

new_state(klimaka(L,R)):-

```

\+ (mo(_G),(G<L;G>R)),
\+ (synolo(_L1,R1),(L1<L;R1>R)),
\+ (synolo_d(_L2,R2),(L2<L;R2>R)),
\+ (synolo_a(_L3,R3),(L3<L;R3>R)),
\+ (synolo_d_a(_L4,R4),(L4<L;R4>R)),
\+ (grades(_L5,R5),(L5<L;R5>R)),
\+ (mo(_G),(G<L;G>R)),!,true.

```

new_state(epipeda_diagoghs(X)):-

```

\+ (epipeda_diagoghs(R), R=X),!,true.

```

new_state(periptwsh(Y,X)):-

```

\+ periptwsh(Y,_),
find_p_e(N),N1 is N+1,N1=Y,
assert(periptwsh(Y,X)),
collision(M),retract(periptwsh(Y,X)),M=0,d_cd,write(M),
!,true.

```

new_state(exei_diagogh(X,D)):-

```

periptwsh(X,_),

\+ exei_diagogh(X,_),
epipeda_diagoghs(D),
assert(exei_diagogh(X,D)),
collision(N),retract(exei_diagogh(X,D)),N=0,d_cd,write(N),    N=0
!,true.

```

new_state(synolo(X,L,R)):-

```
    periptwsh(X,_),
    R>=L,
    \+ synolo(X,_),
    \+ ( klimaka(L1,R1), (L<L1;R>R1)),
    assert(synolo(X,L,R)),
    collision(N),d_cd,      retract(synolo(X,L,R)),N=0
,!,true.
```

new_state(synolo_a(X,L,R)):-

```
    periptwsh(X,_),
    R>=L,
    \+ synolo_a(X,_),
    \+ ( klimaka(L1,R1), (L<L1;R>R1)),
    assert(synolo_a(X,L,R)),
    collision(N),d_cd,      retract(synolo_a(X,L,R)),N=0
,!,true.
```

new_state(synolo_d(X,L,R)):-

```
    periptwsh(X,_),
    R>=L,
    \+ synolo_d(X,_),
    \+ ( klimaka(L1,R1), (L<L1;R>R1)),
    assert(synolo_d(X,L,R)),
    collision(N),d_cd, retract(synolo_d(X,L,R)),N=0
,!,true.
```

new_state(synolo_d_a(X,L,R)):-

```
    periptwsh(X,_),
    R>=L,
    \+ synolo_d_a(X,_),
    \+ ( klimaka(L1,R1), (L<L1;R>R1)),
    assert(synolo_d_a(X,L,R)),
    collision(N),d_cd,      retract(synolo_d_a(X,L,R)),N=0
,!,true.
```

new_state(grades(X,L,R)):-

```
    periptwsh(X,_),
    R>=L,
    \+ grades(X,_),
    \+ ( klimaka(L1,R1), (L<L1;R>R1)),
```

```
assert(grades(X,L,R),  
collision(N),d_cd,N=0, retract(synolo_d_a(X,L,R)),N=0  
,!,true.
```


param_menu.pl

list_apoysia(S,X):-

```
apoysia(S,A,H,M,D),
\+ q2(A),
send(X,append(M)),
send(X,append(D)),
send(X,append(H)),
assert(q2(A)),(
( dikaiologhsh(S,A,M1,D1,T),
send(X,append(M1)),
send(X,append(D1)),
(
(apallagh(T,_,_),Per='Δεν λαμβάνεται υπόψη');
(apoysies_poy_den_dikaiologoyntai(T,_),Per='Δεν μπορεί να δικαιολογηθεί');
(dikaiologhmenh_logo_astheneias(T,_),Per='Δικαιολογημένη λόγω ασθένειας');
(dikaiologhthei(T,_),Per='Δικαιολογημένη');
Per='Αδικαιολόγητη'),
send(X,append(Per)));(
send(X,append('Δεν έχει αξιολογηθεί')),
send(X,append(")),
send(X,append(")),
send(X,next_row),
list_apoysia(S,X),
!,true.
```

list_apoysia(S,_):-

```
\+ (apoysia(S,C,_,_),\+ q2(C)),
d_q2,
!,true.
```

list_day(M,D,X):-

```
apoysia(S,A,H,M,D),
student(S,O,C),
\+ q5(S,A),
class(C,Cper),
send(X,append(O)),
send(X,append(Cper)),
send(X,append(H)),
assert(q5(S,A)),(
```

```
( dikaiologhsh(S,A,M1,D1,T),
send(X,append(M1)),
send(X,append(D1)),
(
(apallagh(T,_,_),Per='Δεν λαμβάνεται υπόψη');
(apoysies_poy_den_dikaiologoyntai(T,_),Per='Δεν μπορεί να δικαιολογηθεί');
(dikaiologhmenh_logo_astheneias(T,_),Per='Δικαιολογημένη λόγω ασθένειας');
(dikaiologhthei(T,_),Per='Δικαιολογημένη');
Per='Αδικαιολόγητη'),
send(X,append(Per));(
send(X,append('Δεν έχει αξιολογηθεί')),
send(X,append("")),send(X,append(""))
)),
send(X,next_row),
list_day(M,D,X),
!,true.
```

list_day(M,D,-):-

```
\+ (apoysia(S,C,_,M,D),student(S,_,_),\+ q5(S,C)),
d_q5,
!,true.
```

list_a_d_d(X):-

```
apoysia(S,A,H,M,D),
student(S,O,C),
dikaiologhsh(S,A,M1,D1,T),
apoysies_poy_den_dikaiologoyntai(T,Tper),
\+ q5(S,A),
class(C,Cper),
send(X,append(O)),
send(X,append(Cper)),
send(X,append(M)),
send(X,append(D)),
send(X,append(H)),
assert(q5(S,A)),
send(X,append(M1)),
send(X,append(D1)),
send(X,append(Tper))
,send(X,next_row),
list_a_d_d(X),
!,true.
```

list_a_d_d(-):-

```
\+ (apoysia(S,C,_,_),student(S,_,_),dikaiologhsh(S,C,_,_T),
apoysies_poy_den_dikaiologoyntai(T,_),\+ q5(S,C)),
d_q5,
```

!,true.

list_a_d_a(X):-

```
apoysia(S,A,H,M,D),
student(S,O,C),
dikaiologhsh(S,A,M1,D1,T),
dikaiologhmenh_logo_astheneias(T,_),
\+ q5(S,A),
class(C,Cper),
send(X,append(O)),
send(X,append(Cper)),
send(X,append(M)),
send(X,append(D)),
send(X,append(H)),

assert(q5(S,A)),

send(X,append(M1)),
send(X,append(D1)),
( ( parastatiko_apo_forea(S,A,Tper),
send(X,append(Tper)));(send(X,append("Οχι"))))
,send(X,next_row),

list_a_d_a(X),
!,true.
```

list_a_d_a(_):-

```
\+
(apoysia(S,C,_,_),student(S,_,_),dikaiologhsh(S,C,_,T),dikaiologhmenh_logo_astheneias(T,_),\
+ q5(S,C)),
d_q5,
!,true.
```

list_a_d(X):-

```
apoysia(S,A,H,M,D),
student(S,O,C),
dikaiologhsh(S,A,M1,D1,T),
dikaiologhthei(T,_),
\+ q5(S,A),
class(C,Cper),
send(X,append(O)),
send(X,append(Cper)),
send(X,append(M)),
send(X,append(D)),
send(X,append(H)),
```

```

assert(q5(S,A)),

send(X,append(M1)),
send(X,append(D1)),
( ( parastatiko(S,A),
send(X,append("Υπάρχει')));(send(X,append("Όχι'))))
,send(X,next_row),
list_a_d(X),
!,true.

```

list_a_d(_):-

```

\+ (apoysia(S,C,_,_),student(S,_,_),dikaiologhsh(S,C,_,_T),dikaiologhthei(T,_,_)\+ q5(S,C)),
d_q5,
!,true.

```

list_a_a(X):-

```

apoysia(S,A,H,M,D),
student(S,O,C),
dikaiologhsh(S,A,M1,D1,T),
apallagh(T,Tper,_,_),
\+ q5(S,A),
class(C,Cper),
send(X,append(O)),
send(X,append(Cper)),
send(X,append(M)),
send(X,append(D)),
send(X,append(H)),
assert(q5(S,A)),
send(X,append(M1)),
send(X,append(D1)),
send(X,append(Tper))
,send(X,next_row),
list_a_a(X),
!,true.

```

list_a_a(_):-

```

\+ (apoysia(S,C,_,_),student(S,_,_),dikaiologhsh(S,C,_,_T), apallagh(T,_,_),\+ q5(S,C)),
d_q5,
!,true.

```

list_class(S,X):-

```

student(C,O,S),
aksiologhsh(C,O,Epidosh),
\+ q2(C),

```

```

send(X,append(C)),
send(X,append(O)),
send(X,append(Epidosh)),
( ( check_case(C,F),periptwsh(F,Fper),send(X,append(Fper)));true),
assert(q2(C)),
send(X,next_row),
list_class(S,X),
!,true.

```

list_class(S,-):-

```

\+ (student(C,_S),\+ q2(C)), d_q2,!,true.

```

menu_a_d(X):-

```

send(X,destroy),
new(BT,dialog('Κατάσταση απουσιών που έχουν δικαιολογηθεί')),
send(BT,size,size(750,500)),
send(BT,append,new(P,picture(""))),
send(P,size,size(700,400)),
send(P,display,new(T,tabular)),
send(T,border,1),
send(T,append("Όνομα μαθητή")),
send(T,append("Τάξη")),
send(T,append('Ημερομηνία απουσίας',colspan:=2)),
send(T,append("Ωρες")),
send(T,append('Ημερομηνια δικαιολόγησης',colspan:=2)),
send(T,append('Παραστατικό')),
send(T,next_row),
list_a_d(T),

send(BT,append,new(B2,button(ok,message(@prolog, menu_2,BT) ))),
get(BT,confirm,B2),
!,true.

```

menu_a_d_a(X):-

```

send(X,destroy),
new(BT,dialog('Κατάσταση απουσιών που έχουν δικαιολογηθεί λόγω ασθένειας')
),
send(BT,size,size(750,500)),

send(BT,append,new(P,picture(""))),
send(P,size,size(700,400)),
send(P,display,new(T,tabular)),
send(T,border,1),
send(T,append("Όνομα μαθητή")),
send(T,append("Τάξη")),
send(T,append('Ημερομηνία απουσίας',colspan:=2)),

```

```

send(T,append("Ωρες")),
send(T,append('Ημερομηνια δικαιολόγησης',colspan:=2)),
send(T,append('Παραστατικό')),
send(T,next_row),
list_a_d_a(T),
send(BT,append,new(B2,button(ok,message(@prolog,menu_2,BT) ))),

get(BT,confirm,B2),
!,true.

```

menu_a_a(X):-

```

send(X,destroy),
new(BT,dialog('Κατάσταση απουσιών που απαλλάσσονται')),
send(BT,size,size(750,500)),
send(BT,append,new(P,picture("))),
send(P,size,size(700,400)),
send(P,display,new(T,tabular)),
send(T,border,1),
send(T,append("Όνομα μαθητή")),
send(T,append("Τάξη")),
send(T,append('Ημερομηνία απουσίας',colspan:=2)),
send(T,append("Ωρες")),
send(T,append('Ημερομηνια δικαιολόγησης',colspan:=2)),
send(T,append('Κατηγορία')),
send(T,next_row),
list_a_a(T),
send(BT,append,new(B2,button(ok,message(@prolog,menu_2,BT) ))),

get(BT,confirm,B2),
!,true.

```

menu_a_d_d(X):-

```

send(X,destroy),
new(BT,dialog('Κατάσταση απουσιών που δεν μπορούν να δικαιολογηθούν')),
send(BT,size,size(750,500)),
send(BT,append,new(P,picture("))),
send(P,size,size(700,400)),
send(P,display,new(T,tabular)),
send(T,border,1),
send(T,append("Όνομα μαθητή")),
send(T,append("Τάξη")),
send(T,append('Ημερομηνία απουσίας',colspan:=2)),
send(T,append("Ωρες")),
send(T,append('Ημερομηνια δικαιολόγησης',colspan:=2)),
send(T,append('Κατηγορία')),

```

```
send(T,next_row),
list_a_d_d(T),
send(BT,append,new(B2,button(ok,message(@prolog, menu_2,BT) )),
```

```
get(BT,confirm,B2),
!,true.
```

menu_a_h(X):-

```
send(X,destroy),
new(D2,dialog('Επέλεξε ημερομηνία')),
send(D2,size,size(500,200)),
send(D2,append,new(T1,menu('Μήνας',cycle))),
s_mhnas(T1),
send(D2,append,new(T2,menu('Ημέρα',cycle))),s_day(1,T2),
```

```
send(D2,append,new(B2,button(ok,message(@prolog,menu_sel2,D2,333,T1?selection,T2
?selection) )),
send(D2,append,new(B0,button(cancel,message(@prolog, esc_1,D2) )),
```

```
get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.
```

menu_e_c(X):-

```
send(X,destroy),
new(D2,dialog('Επέλεξε Τάξη')),
send(D2,size,size(500,200)),
send(D2,append,new(T1,menu('Τάξη',cycle))),
m_class(T1),
send(D2,append,new(B2,button(ok,message(@prolog, menu_sel2,D2,222,T1?selection)
))),
send(D2,append,new(B0,button(cancel,message(@prolog, esc_1,D2) )),
```

```
get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.
```

menu_a_l(X):-

```
send(X,destroy),
new(D2,dialog('Επέλεξε μαθητή')),
send(D2,size,size(500,200)),
send(D2,append,new(T1,menu('Μαθητής',cycle))),
pop_student(T1),
```

```

send(D2,append,new(B2,button(ok,message(@prolog menu_sel2,D2,111,T1?selection)
))),
send(D2,append,new(B0,button(cancel,message(@prolog esc_1,D2) )),

get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.

```

menu_sel2(X,S,C):-

```

S==111,
student(C,O,_),
send(X,destroy),
new(BT,dialog('Κατάσταση απουσιών')),
send(BT,size,size(750,500)),
send(BT,append,new(P,picture(''))),
send(P,size,size(700,400)),
send(P,display,new(T,tabular)),
send(T,border,1),
send(T,append(C,colspan:=2)),
send(T,append(O,colspan:=3)),
send(T,next_row),
send(T,append('Ημερομηνια απουσιας',colspan:=2)),
send(T,append("Ωρες")),
send(T,append('Ημερομηνια δικαιολόγησης',colspan:=2)),
send(T,append('Αιτιολογία')),
send(T,next_row),
list_apoysia(C,T),
send(BT,append,new(B2,button(ok,message(@prolog menu_2,BT) )),
send(BT,append,new(B0,button(cancel,message(@prolog esc_1,BT) )),

get(BT,confirm,B0),
get(BT,confirm,B2),
!,true.

```

menu_sel2(X,S,C):-

```

S==222,
class(C,Cper),
send(X,destroy),
new(BT,dialog(Cper)),
send(BT,size,size(750,500)),
send(BT,append,new(P,picture(''))),
send(P,size,size(700,400)),
send(P,display,new(T,tabular)),
send(T,border,1),
send(T,append('Κατάσταση Απουσιών',colspan:=3)),
send(T,next_row),

```



```

send(T,append('Κωδικός μαθητή')),
send(T,append("Όνομα μαθητή")),
send(T,append('Επίδοση')),
send(T,next_row),
list_class(C,T),
send(BT,append,new(B2,button(ok,message(@prolog,menu_2,BT) ))),

get(BT,confirm,B2),
!,true.

```

menu_sel2(X,S,Y):-

```

S==2222,

send(X,destroy),
( retract(epipeda_diagoghs(Y))->(update_parameters,w_label3('Η διαγραφή έγινε'));
w_label3('Δεν υπάρχει ο τίτλος')),
!,true.

```

menu_sel2(X,S,Y):-

```

S==5555,
send(X,destroy),
( \+ epipeda_diagoghs(Y))->
(assert(epipeda_diagoghs(Y)),update_parameters,w_label3('Η εγγραφή έγινε'));
w_label3('Υπάρχει ο τίτλος')),
!,true.

```

menu_sel2(X,S,T):-

```

S==3333,
retract(orio_parastatikwn(_)),
assert(orio_parastatikwn(T)),
send(X,destroy),
update_parameters,menu_2(0),
!,true.

```

menu_sel2(X,S,P1):-

```

S==1,
\+ (apoysia(_,_C,_),C>P1),
send(X,destroy),
( retract(hmerhsio_orio_didaskalias(_));true),
assert(hmerhsio_orio_didaskalias(P1)),

```

```
update_parameters,  
w_label3('η εγγραφή έγινε'),  
!,true.
```

menu_sel2(X,S,P1):-

```
S==22,  
send(X,destroy),  
( ( class(P1,_), \+ (student(_,P1)),  
retract(class(P1,_)),  
arithmos_class(N),  
( retract(arithmos_class(_));true),  
N1 is N-1,  
assert(arithmos_class(N1)),  
update_parameters)-> w_label3('η διαγραφή έγινε'); w_label3('η διαγραφή δεν έγινε')),  
!,true.
```

menu_sel2(X,S,P1):-

```
S==55,  
  
apallagh(P1,_,_),  
\+ (dikaiologhsh(____P1)),  
send(X,destroy),  
retract(apallagh(P1,_,_)),  
arithmos_periptwsewn_apallaghs(N),  
( retract(arithmos_periptwsewn_apallaghs(_));true),  
N1 is N-1,  
assert(arithmos_periptwsewn_apallaghs(N1)),  
update_parameters,  
w_label3('η διαγραφή έγινε'),  
!,true.
```

menu_sel2(X,S,P1):-

```
S==77,  
  
apoysies_poy_den_dikaiologoyntai(P1,_),  
\+ (dikaiologhsh(____P1)),  
send(X,destroy),  
retract(apoysies_poy_den_dikaiologoyntai(P1,_)),  
update_parameters,  
w_label3('η διαγραφή έγινε'),  
!,true.
```

menu_sel2(X,S,P1):-

```
S==3,  
send(X,destroy),  
( retract(orio_hmerwn(_));true),  
assert(orio_hmerwn(P1)),  
update_parameters,  
w_label3('η εγγραφή έγινε'),  
!,true.
```

menu_sel2(X,S,_):-

```
S==77,  
send(X,destroy),  
w_label3('η διαγραφή δεν έγινε'),  
!,true.
```

menu_h_orio(X):-

```
send(X,destroy),  
new(D2,dialog('Ημερήσιο όριο διδασκαλίας')),  
send(D2,size,size(500,200)),  
(hmerhsio_orio_didaskalias(N); N is 0),  
send(D2,append,new(T1,text_item("Όριο: ",N))),  
send(D2,append,new(B2,button(ok,message(@prolog, menu_sel2,D2,1,T1?selection) ))),  
send(D2,append,new(B0,button(cancel,message(@prolog, esc_1,D2) ))),  
  
get(D2,confirm,B0),  
get(D2,confirm,B2),!,true.
```

menu_klimaka(X):-

```
send(X,destroy),  
new(D2,dialog('Νέα κλίμακα βαθμολογίας')),  
send(D2,size,size(500,200)),  
klimaka(L1,R1),  
send(D2,append,new(T1,text_item('Από: ',L1))),  
send(D2,append,new(T2,text_item('Μέχρι: ',R1))),  
send(D2,append,new(B2,button(ok,message(@prolog,  
menu_sel2,D2,1111,T1?selection,T2?selection,L1,R1) ))),  
send(D2,append,new(B0,button(cancel,message(@prolog, esc_1,D2) ))),  
  
get(D2,confirm,B0),  
get(D2,confirm,B2),
```

!,true.

```
menu_class(X):-
  send(X,destroy),
  new(D2,dialog('Νέα Τάξη')),
  send(D2,size,size(500,200)),
  find_c(N),N1 is N+1,
  send(D2,append,new(T1,text_item('αα Τάξης:',N1))),
  send(D2,append,new(T2,text_item('Περιγραφή',''))),
  send(D2,append,new(B2,button(ok,message(@prolog,
menu_sel2,D2,2,T1?selection,T2?selection))))),
  send(D2,append,new(B3,button(delete,message(@prolog,
menu_sel2,D2,22,T1?selection))))),
  send(D2,append,new(B0,button(cancel,message(@prolog, esc_1,D2) )),

  get(D2,confirm,B0),
  get(D2,confirm,B2),
  get(D2,confirm,B3),
  !,true.
```

```
menu_diag(X):-
  send(X,destroy),
  new(D2,dialog('Τίτλος διαγωγής μαθητή')),
  send(D2,size,size(500,400)),
  send(D2,append,new(T1,text_item('Περιγραφή',''))),
  send(D2,append,new(B2,button('Καταχώρηση',message(@prolog,menu_sel2,D2,5555,T
1?selection))))),
  send(D2,append,new(B3,button('Διαγραφή',message(@prolog,menu_sel2,D2,2222,T1?se
lection))))),
  send(D2,append,new(B0,button(cancel,message(@prolog, esc_1,D2) )),

  get(D2,confirm,B0),
  get(D2,confirm,B2),
  get(D2,confirm,B3),
  !,true.
```

```
menu_apallagh(X):-

  send(X,destroy),
  new(D2,dialog('Περιπτώσεις για τις οποίες απουσίες δεν λαμβάνονται υπόψη')),
  send(D2,size,size(500,400)),
  ( arithmos_periptwsewn_apallaghs(N);N is 0),
  N1 is N+1,
  send(D2,append,new(T1,text_item('αα Περίπτωσης απαλλαγής:',N1))),
  send(D2,append,new(T2,text_item('Περιγραφή',''))),
  send(D2,append,new(T3,text_item('Μηνιαίο όριο που απαλλάσσεται:',0))),
  send(D2,append,new(T4,text_item('Ετήσιο όριο που απαλλάσσεται:',0))),
  send(D2,append,new(B2,button(ok,message(@prolog,menu_sel2,D2,5,T1?selection,
T2?selection,T3?selection,T4?selection))))),
```

```

send(D2,append,new(B3,button(delete,message(@prolog,
menu_sel2,D2,55,T1?selection))))),
send(D2,append,new(B0,button(cancel,message(@prolog, esc_1,D2) )),

```

```

get(D2,confirm,B0),
get(D2,confirm,B2),
get(D2,confirm,B3),
!,true.

```

```

menu_den_d(X):-

```

```

send(X,destroy),
new(D2,dialog('Περιπτώσεις για τις οποίες απουσίες δεν μπορούν να δικαιολογηθούν')),
send(D2,size,size(500,400)),
send(D2,append,new(T1,text_item('αα Περίπτωσης[21-39]:',0))),
send(D2,append,new(T2,text_item('Περιγραφή',''))),
send(D2,append,new(B2,button(ok,message(@prolog,
menu_sel2,D2,7,T1?selection,T2?selection))))),
send(D2,append,new(B3,button(delete,message(@prolog,
menu_sel2,D2,77,T1?selection))))),
send(D2,append,new(B0,button(cancel,message(@prolog, esc_1,D2) )),

```

```

get(D2,confirm,B0),
get(D2,confirm,B2),
get(D2,confirm,B3),
!,true.

```

```

menu_parast_orio(X):-

```

```

send(X,destroy),
new(D2,dialog("Όριο Παραστατικών")),
send(D2,size,size(400,200)),
orio_parastatikwn(N),
send(D2,append,new(T1,text_item("Όριο: ',N)")),
send(D2,append,new(B2,button(ok,message(@prolog,menu_sel2,D2,3333,T1?selection))
)),
send(D2,append,new(B0,button(cancel,message(@prolog, esc_1,D2) )),

```

```

get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.

```

```

menu_orio_hm(X):-

```

```

send(X,destroy),
new(D2,dialog("Όριο ημερών στις οποίες μπορεί να δικαιολογηθεί απουσία')),
send(D2,size,size(400,200)),
( orio_hmerwn(N); N is 0),
send(D2,append,new(T1,text_item("Όριο: ',N)")),
send(D2,append,new(B2,button(ok,message(@prolog, menu_sel2,D2,3,T1?selection))))),
send(D2,append,new(B0,button(cancel,message(@prolog, esc_1,D2) )),

```

```
get(D2,confirm,B0),
get(D2,confirm,B2),
!,true.
```

```
menu_sel2(X,S,M,D):-
```

```
    S==333,
    send(X,destroy),
    new(BT,dialog('Κατάσταση απουσιών')),
    send(BT,size,size(750,500)),
    send(BT,append,new(P,picture("))),
    send(P,size,size(700,400)),
    send(P,display,new(T,tabular)),
    send(T,border,1),
    send(T,append(D,colspan:=2)),
    send(T,append(M,colspan:=3)),
    send(T,next_row),
    send(T,append("Όνομα μαθητή")),
    send(T,append("Τάξη")),
    send(T,append("Ωρες")),
    send(T,append('Ημερομηνια δικαιολόγησης',colspan:=2)),
    send(T,append('Αιτιολογία')),
    send(T,next_row),
    list_day(M,D,T),
    send(BT,append,new(B2,button(ok,message(@prolog,menu_2,BT) ))),
    send(BT,append,new(B0,button(cancel,message(@prolog,esc_1,BT) ))),
```

```
get(BT,confirm,B0),
get(BT,confirm,B2),
!,true.
```

```
menu_sel2(X,S,P1,P2):-
```

```
    S==7,
    send(X,destroy),
    ( (
        \+ apoysies_poy_den_dikaiologoyntai(P1,_),
        \+ apoysies_poy_den_dikaiologoyntai(_P2),
        P1>20,P1<40,
        assert(apoysies_poy_den_dikaiologoyntai(P1,P2)),
        update_parameters-> w_label3('η εγγραφή έγινε');
        w_label3('η εγγραφή δεν έγινε')),
    !,true.
```

menu_sel2(X,S,P1,P2):-

```
S==2,  
send(X,destroy),  
(find_c(N),N1 is N+1, N1=P1,  
\+ class(_P2),  
assert(class(P1,P2)),  
( retract(arithmos_class(_));true),  
assert(arithmos_class(N1)),  
update_parameters)-> w_label3('η εγγραφή έγινε');  
w_label3('η εγγραφή δεν έγινε'),  
!,true.
```

menu_sel2(X,S,L,R,L1,R1):-

```
S==1111,  
send(X,destroy),  
retract(klimaka(L1,R1)),assert(klimaka(L,R)),update_parameters,  
menu_2(0),!,true.
```

menu_sel2(X,S,P1,P2,P3,P4):-

```
S==5,  
send(X,destroy),  
( ( ( arithmos_periptwsewn_apallaghs(N);N is 0),  
N1 is N+1,N1=P1,  
\+ apallagh(_P2,_),  
assert(apallagh(P1,P2,P3,P4)),  
( retract(arithmos_periptwsewn_apallaghs(_));true),  
assert(arithmos_periptwsewn_apallaghs(N1)),  
update_parameters)-> w_label3('η εγγραφή έγινε');  
w_label3('η εγγραφή δεν έγινε'),  
!,true.
```

menu_sel2(_S,_):-

```
S==55,  
w_label3('η διαγραφή δεν έγινε'),  
!,true.
```

menu_epa2(X,_):-

```

find_p_e(M),
M==0,
send(X,destroy),
w_label3('Δεν υπάρχουν συνθήκες'),
!,true.

```

menu_epa2(X,N):-

```

find_p_e(M),
(N>M -> C is 1; C is N),
send(X,destroy),
periptwsh(C,P),
new(D2,dialog('Συνθήκη για τον χαρακτηρισμό επάρκειας φοίτησης')),
send(D2,size,size(600,800)),
send(D2,append,new(_text_item('Aa',C))),
send(D2,append,new(_text_item('Ονομασία',P))),
( ( synolo(C,N1,N2),
    send(D2,append,new(label(11,'Σύνολο απουσιών που λαμβάνονται υπόψη'))),
    send(D2,append,new(_text_item('Από',N1))),
    send(D2,append,new(_text_item('Μέχρι',N2)))));true),
( ( synolo_d(C,N3,N4),
    send(D2,append,new(label(12,'Σύνολο απουσιών που μπορούν να δικαιολογηθούν'))),
    send(D2,append,new(_text_item('Από',N3))),
    send(D2,append,new(_text_item('Μέχρι',N4)))));true),
( ( synolo_d_a(C,N5,N6),
    send(D2,append,new(label(13,'Σύνολο απουσιών που μπορούν να δικαιολογηθούν
λόγω ασθένειας'))),
    send(D2,append,new(_text_item('Από',N5))),
    send(D2,append,new(_text_item('Μέχρι',N6)))));true),

( ( synolo_a(C,N7,N8),
    send(D2,append,new(label(14,'Σύνολο απουσιών που είναι αδικαιολόγητες'))),
    send(D2,append,new(_text_item('Από',N7))),
    send(D2,append,new(_text_item('Μέχρι',N8)))));true),
( (grades(C,N9,N10),
    send(D2,append,new(label(14,'Περιοχή βαθμολογίας'))),
    send(D2,append,new(_text_item('Από',N9))),
    send(D2,append,new(_text_item('Μέχρι',N10)))));true),
( (exei_diagogh(C,D),
    send(D2,append,new(_text_item('Διαγωγή:',D)))));true),
C1 is C+1,
send(D2,append,new(B2,button('επόμενη',message(@prolog,menu_epa2,D2,C1      )))),
send(D2,append,new(B4,button('Διαγραφή',message(@prolog,menu_epa_d2,D2,C))))),
send(D2,append,new(B3,button('έξοδος',message(@prolog, menu_2,D2))))),

get(D2,confirm,B2),
get(D2,confirm,B4),
get(D2,confirm,B3),
!,true.

```


menu_epa_d2(X,X0):-

```
send(X,destroy),
(retract(periptwsh(X0,_));true),
(retract(synolo(X0,_));true),
(retract(synolo_d(X0,_));true),
(retract(synolo_d_a(X0,_));true),
(retract(synolo_a(X0,_));true),
(retract(grades(X0,_));true),
(retract(exei_diagogh(X0,_));true),
find_p_e(L,arithmos_periptwsewn_eparkeias(K),K1 is K-1,
retract(arithmos_periptwsewn_eparkeias(_)),
assert(arithmos_periptwsewn_eparkeias(K1)),
X1 is X0+1,
shift2_eparkeia(X1,L),
w_label3('Η διαγραφή έγινε'),
!,update_per_epa,true.
```

shift2_eparkeia(L,R):-

```
L>R,update_per_epa,
!,true.
```

shift2_eparkeia(L,R):-

```
L1 is L-1,
periptwsh(L,Z),
assert(periptwsh(L1,Z)),retract(periptwsh(L,_)),

( ( synolo(L,A,B),assert(synolo(L1,A,B)),retract(synolo(L,_));true),
( ( synolo_a(L,C,D),assert(synolo_a(L1,C,D)),retract(synolo_a(L,_));true),
( ( synolo_d(L,E,F),assert(synolo_d(L1,E,F)),retract(synolo_d(L,_));true),
( ( synolo_d_a(L,G,H),assert(synolo_d_a(L1,G,H)),retract(synolo_d_a(L,_));true),
( ( exei_diagogh(L,I),assert(exei_diagogh(L1,I)),retract(exei_diagogh(L,_));true),
( ( grades(L,A1,A2),assert(grades(L1,A1,A2)),retract(grades(L,_));true),

Next is L+1,
shift2_eparkeia(Next,R),
!,true.
```

menu_apa2(X,_):-

```
\+ apallagh(_,_),
send(X,destroy),
w_label3('δεν υπάρχουν περιπτώσεις'),
!,true.
```

menu_apa2(X,N):-

```

arithmos_periptwsewn_apallaghs(M),
(N>M -> C is 1; C is N),
send(X,destroy),
apallagh(C,P,N1,N2),
new(D2,dialog('Περιπτώσεις απαλλαγής απουσιών')),
send(D2,size,size(500,500)),
send(D2,append,new(_text_item('Aa',C))),
send(D2,append,new(_text_item('Περίπτωση:',P))),
send(D2,append,new(_text_item('Μηνιαίο όριο απουσιών',N1))),
send(D2,append,new(_text_item('Ετήσιο όριο απουσιών',N2))),
C1 is C+1,
send(D2,append,new(B2,button('επόμενη',message(@prolog,menu_apa2,D2,C1)))),
send(D2,append,new(B4,button('Διαγραφή',message(@prolog,menu_apa_d2,D2,C1)))),
send(D2,append,new(B3,button('έξοδος',message(@prolog,menu_2,D2,C1))))),

get(D2,confirm,B2),
get(D2,confirm,B4),
get(D2,confirm,B3),
!,true.

```

menu_apa_d2(X,X0):-

```

send(X,destroy),(
    \+ dikaiologhsh(____,X0),
    retract(apallagh(X0,____)),
    arithmos_periptwsewn_apallaghs(L),
    X1 is X0+1,arithmos_periptwsewn_apallaghs(K),K1 is K-1,
    retract(arithmos_periptwsewn_apallaghs(K)),
    assert(arithmos_periptwsewn_apallaghs(K1)),

    shift2_apallagh(X1,L),w_label3('Η διαγραφή έγινε');
    ( w_label3('Η διαγραφή δεν έγινε λόγω αναφοράς'))),

!,true.

```

shift2_apallagh(L,R):-

```

L>R,
update_parameters,
!,true.

```

shift2_apallagh(L,R):-

```

L1 is L-1,
apallagh(L,Y,Z,W),assert(apallagh(L1,Y,Z,W)),retract(apallagh(L,Y,____)),
Next is L+1,
shift2_apallagh(Next,R),

```

!,true.

menu_p_d_d2(X):-

```
\+ apoysies_poy_den_dikaiologoyntai(_,_),
send(X,destroy),
w_label3('δεν υπάρχουν περιπτώσεις')
!,true.
```

menu_p_d_d2(X):-

```
apoysies_poy_den_dikaiologoyntai(C,P)\+ q2(C),assert(q2(C)),
send(X,destroy),
new(D2,dialog('Περιπτώσεις απουσιών που δεν δικαιολογούνται')),
send(D2,size,size(500,300)),
send(D2,append,new(_text_item('Aa',C))),
send(D2,append,new(_text_item('Περίπτωση:',P))),
send(D2,append,new(B2,button('επόμενη',message(@prolog,menu_p_d_d2,D2)))),
send(D2,append,new(B4,button('Διαγραφή',message(@prolog,menu_d_d_d2,D2,C)))),
send(D2,append,new(B3,button('έξοδος',message(@prolog,menu_2,D2
))),

get(D2,confirm,B2),
get(D2,confirm,B4),
get(D2,confirm,B3),
!,true.
```

menu_p_d_d2(X):-

```
d_q2,
send(X,destroy),
menu_2(0),
!,true.
```

menu_d_d_d2(X,X0):-

```
send(X,destroy),(
( \+ dikaiologhsh(_,_,_X0),
retract(apoysies_poy_den_dikaiologoyntai(X0,_)),update_parameters,
w_label3('Η διαγραφή έγινε'));
( w_label3('Η διαγραφή δεν έγινε λόγω αναφοράς'))),

!,true.
```

esc_1(X):-

send(X,destroy),
menu_2(0),
!,true.

parameters.pl

```
:- dynamic hmerhsio_orio_didaskalias/1.

hmerhsio_orio_didaskalias(7).

:- dynamic arithmos_class/1.

arithmos_class(4).

:- dynamic class/2.

class(1, 'Τάξη_1').
class(2, 'Τάξη_2').
class(3, 'Τάξη_3').
class(4, 'Τάξη_4').

:- dynamic arithmos_periptwsewn_apallaghs/1.

arithmos_periptwsewn_apallaghs(2).

:- dynamic apallagh/4.

apallagh(1, 'Αθλητικοί αγώνες', 10, 30).
apallagh(2, 'Στρατολογία', 10, 10).

:- dynamic apoysies_poy_den_dikaiologoyntai/2.

apoysies_poy_den_dikaiologoyntai(30, 'Ωριαία αποβολή').
apoysies_poy_den_dikaiologoyntai(35, 'Ημερήσια αποβολή').

:- dynamic dikaiologhmenh_logo_astheneias/2.

dikaiologhmenh_logo_astheneias(40, d_l_a).

:- dynamic dikaiologhthei/2.

dikaiologhthei(50, d_a).

:- dynamic adikaiologhth/2.

adikaiologhth(60, a_a).

:- dynamic orio_hmerwn/1.
```

orio_hmerwn(10).

:- dynamic klimaka/2.

klimaka(2, 20).

:- dynamic epipeda_diagogs/1.

epipeda_diagogs(κοσμία).

epipeda_diagogs(κοσμιότητα).

epipeda_diagogs(μέτρια).

epipeda_diagogs(κακή).

:- dynamic orio_parastatikwn/1.

orio_parastatikwn(11).

per_eparkeias.pl

`:- dynamic periptwsh/2.`

```
periptwsh(1, 'Περίπτωση_3').  
periptwsh(2, 'Περίπτωση_2').  
periptwsh(3, 'Περίπτωση_1').  
periptwsh(4, 'Περίπτωση_4').  
periptwsh(5, 'Περίπτωση_8').
```

`:- dynamic exei_diagogh/2.`

```
exei_diagogh(1, κοσμιότητα).  
exei_diagogh(2, κοσμία).  
exei_diagogh(3, κοσμία).  
exei_diagogh(4, μέτρια).  
exei_diagogh(5, κοσμιότητα).
```

`:- dynamic synolo/3.`

```
synolo(1, 0, 161).  
synolo(2, 0, 25).  
synolo(3, 0, 51).  
synolo(4, 0, 24).  
synolo(5, 0, 48).
```

`:- dynamic synolo_d/3.`

`:- dynamic synolo_a/3.`

```
synolo_a(1, 0, 50).
```

`:- dynamic synolo_d_a/3.`

```
synolo_d_a(1, 0, 58).
```

`:- dynamic grades/3.`

```
grades(1, 16, 20).  
grades(4, 1, 10).  
grades(2, 0, 20).  
grades(3, 0, 18).
```

`:- dynamic arithmos_periptwsewn_eparkeias/1.`

```
arithmos_periptwsewn_eparkeias(5).
```

reset.pl

```
reset:-    tell('facts.pl'),
           write(':- dynamic apoysia/5.').nl,
           write(':- dynamic dikaiologhsh/5.').nl,
           write(':- dynamic parastatiko_apo_forea/3.').nl,
           write(':- dynamic parastatiko/2.').nl,
           told,

           tell('parameters.pl'),
           write(':- dynamic hmerhsio_orio_didaskalias/1.').nl,
           write('hmerhsio_orio_didaskalias(0)').nl,
           write(':- dynamic arithmos_class/1.').nl,
           write('arithmos_class(0)').nl,
           write(':- dynamic class/2.').nl,
           write(':- dynamic arithmos_periptwsewn_apallaghs/1.').nl,
           write('arithmos_periptwsewn_apallaghs(0)').nl,
           write(':- dynamic apallagh/4.').nl,
           write(':- dynamic apoysies_poy_den_dikaiologoyntai/2.').nl,
           write(':- dynamic dikaiologhmenh_logo_astheneias/2.').nl,
           write('dikaiologhmenh_logo_astheneias(40, d_l_a)').nl,
           write(':- dynamic dikaiologhthei/2.').nl,
           write('dikaiologhthei(50, d_a)').nl,
           write(':- dynamic adikaiologhth/2.').nl,
           write('adikaiologhth(60, a_a)').nl,
           write(':- dynamic orio_hmerwn/1.').nl,
           write('orio_hmerwn(0)').nl,
           write(':- dynamic klimaka/2.').nl,
           write('klimaka(1, 20)').nl,
           write(':-dynamic epipeda_diagoghs/1.').nl,
           write(':- dynamic orio_parastatikwn/1.').nl,
           write('orio_parastatikwn(0)').nl,

           told,

           tell('dates.pl'),
           write(':- dynamic mhnas/3.').nl,
           write('mhnas(1, septembrios, 28)').nl,
           write('mhnas(2, oktvbrios, 60)').nl,
           write('mhnas(3, noembrios, 92)').nl,
           write('mhnas(4, dekembrios, 123)').nl,
           write('mhnas(5, ianoyarios, 155)').nl,
           write('mhnas(6, febroyarios, 186)').nl,
           write('mhnas(7, martios, 215)').nl,
           write('mhnas(8, aprilios, 246)').nl,
           write('mhnas(9, maios, 278)').nl,
           write('mhnas(10, ioynios, 309)').nl,
```



```
write('mhnas(11, end, 340).'),nl,
write(':- dynamic sunday/1. '),nl,
write('sunday(6).'),nl,
write(':- dynamic saturday/1. '),nl,
write('saturday(5).'),nl,
write(':- dynamic diakopes/3. '),nl,
told,
```

```
tell('per_eparkeias.pl'),
write(':- dynamic periptwsh/2. '),nl,
write(':- dynamic exei_diagogh/2. '),nl,
write(':- dynamic synolo/3. '),nl,
write(':- dynamic synolo_d/3. '),nl,
write(':- dynamic synolo_a/3. '),nl,
write(':- dynamic synolo_d_a/3. '),nl,
write(':- dynamic grades/3. '),nl,
write(':- dynamic arithmos_periptwsewn_eparkeias/1. '),nl,
write('arithmos_periptwsewn_eparkeias(0). '),nl,
```

told,

```
tell('student.pl'),
write(':- dynamic student/3. '),nl,
write(':- dynamic mo/2. '),nl,
write(':- dynamic diagogh/2. '),nl,
told,
```

```
tell('dyn_pred.pl'),
write(':- dynamic q2/1. '),nl,
write(':- dynamic q8/2. '),nl,
write(':- dynamic q_temp/1. '),nl,
write(':- dynamic q_temp1/2. '),nl,
write(':- dynamic q_temp2/2. '),nl,
write(':- dynamic qc/1. '),nl,
told.
```

rules.pl

```
% o parakatw kanonas dikaiologhmenh elegxei an mia apoysia
% poy mporei na dikaiologhtei exei
% dikaiologhtei apo gonea mesa se 10 ergasimes
% hmeres

% o parakatw kanonas elegxei ean mia hmera
% antistoixei se argia

elegxos_mh_erg2(N1):-
    (saturday(N1);sunday(N1)),
    !,true.

elegxos_mh_erg2(N):-
    saturday(N1),sunday(N2),N<N1,N<N2,
    !,false.

elegxos_mh_erg2(N):-
    N1 is N-7,
    elegxos_mh_erg2(N1),
    !,true.

elegxos_mh_erg1(N):-
    diakopes(_N1,N2),
    N11 is N1-1, N22 is N2+1,
    N>N11,N<N22,
    !,true.

elegxos_mh_erg(N):-
    ( elegxos_mh_erg1(N);elegxos_mh_erg2(N)),
    !,true.

% o parakatw kanonas ypologizei gia sygkekrimenh hmeromhnia ton
% ayksonta arithmo hmeras ston opoio antistoixei.

day_count(Mhnas,Day,N):-
    mhnas(_Mhnas,X),
    N is X+Day,
    !,true.
```

```

% o parakatw kanonas ypologizei sto M
% ton arithmo ergasimwn hmerwn metajy
% dyo hmeromhniwn poy antistoixoyh
% se ayksontes arithmoys N1,N2

```

```

ergasimes_count(N1,N2,N,M):-

```

```

    N2<N1,M is N,!,true.

```

```

ergasimes_count(N1,N2,N,M):-

```

```

    N3 is N1+1,
    (elegxos_mh_erg(N1)->N7 is N;N7 is N+1),
    ergasimes_count(N3,N2,N7,M).

```

```

% o parakatw kanonas ypologizei ton arithmo ergasimwn
% hmervn metaksy dyo hmeromhniwn poy
% perigrafontai me mhna kai arithmo hmeras

```

```

ypologismos_erg(Mhnas_a,Hmer_a,Mhnas_t,Hmer_t,N):-

```

```

    mhnas(_Mhnas_a,X1),mhnas(_Mhnas_t,X2),

```

```

    N1 is X1+Hmer_a,N2 is X2+Hmer_t,

```

```

    ergasimes_count(N1,N2,0,N).

```

```

% parakatw kanonas elegxei ean mia apoysia
% exei dikaiologhthei mesa sto eylogo
% diasthma

```

```

dikaiologhmenh(Student,Aa):-

```

```

    apoysia(Student,Aa,_M1,H1),
    dikaiologhsh(Student,Aa,M2,H2,X),
    dikaiologhthei(X,d_a),
    ypologismos_erg(M1,H1,M2,H2,N),
    orio_hmerwn(K),
    ( N<K;N==K).

```

```

% parakatw kanonas elegxei ean mia apoysia
% exei dikaiologhthei mesa sto eylogo

```

```

% diasthma logw astheneias

dikaiologhmenh_l_a(Student,Aa):-

    apoysia(Student,Aa,_M1,H1),
    dikaiologhsh(Student,Aa,M2,H2,X),
    dikaiologhmenh_logo_astheneias(X,d_l_a),
    ypologismos_erg(M1,H1,M2,H2,N),
    orio_hmerwn(K),
    ( N<K;N==K).

% o kanonas C paragei ton teleytaio auksonta
% arithmo apoysias gia sygkekrimeno mathith.
%
c(Mathitis,Aa):-

    apoysia(Mathitis,Aa,_,_),
    \+ (apoysia(Mathitis,M,_,_),
    M>Aa),!,true.

% o parakatw kanonas ypologizei
% gia sygkekrimeno mathith mhna kai periptvsh
% sthn opoia empitoyn to synolo arithmoy apoysiwn (T)

s_m(I,Mhnas,Periptosh,J,T):-

    J1 is J-1,
    apoysia(I,J,X,Mhnas1,_),
    dikaiologhsh(I,J,_,Periptosh1),
    s_m(I,Mhnas,Periptosh,J1,T1),
    ( Periptosh1=Periptosh,Mhnas1==Mhnas ->R is X; R is 0),
    T is T1+R,
    !,true.

s_m(____,0,0).

% o parakatw kanonas ypologizei
% gia sygkekrimeno mathith mhna kai periptvsh apallaghs
% sthn opoia empitoyn toa arithmo apoysiwn (S)
% poy telika den apallassontai dioti yperbainoyn to
% mhniaio orio.

s_m_oxi_apallagh(Mathitis,Mhnas,Periptosh,S):-

    mhnas(_,Mhnas,_),
    c(Mathitis,N),

```

```
s_m(Mathitis,Mhnas,Periptosh,N,X),
apallagh(Periptosh,_Lm,_),
(X>Lm,\+Lm==0 -> S is X-Lm ; S is 0).
```

```
% o parakatw kanonas ypologizei
% gia sygkekrimeno mathith kai periptvsh apallaghs
% sthn opoia empitoyn to arithmo apoysiwn (S)
% poy telika den apallassontai dioti yperbainoyn to
% mhniaio orio se kathe mhna.
```

```
ypologismos_m(Mathitis,Periptosh,S):-
  s_m_oxi_apallagh(Mathitis,ianoyarios,Periptosh,S1),
  s_m_oxi_apallagh(Mathitis,febroyarios,Periptosh,S2),
  s_m_oxi_apallagh(Mathitis,martios,Periptosh,S3),
  s_m_oxi_apallagh(Mathitis,apriliios,Periptosh,S4),
  s_m_oxi_apallagh(Mathitis,maios,Periptosh,S5),
  s_m_oxi_apallagh(Mathitis,ioynios,Periptosh,S6),
  s_m_oxi_apallagh(Mathitis,oktvbrios,Periptosh,S7),
  s_m_oxi_apallagh(Mathitis,noembrios,Periptosh,S8),
  s_m_oxi_apallagh(Mathitis,dekembrios,Periptosh,S9),
  S is S1+S2+S3+S4+S5+S6+S7+S8+S9.
```

```
% o parakatw kanonas ypologizei to synolo
% apoysiwn mathiti poy an kai apallassontai
% telika lambanontai ypopsh eksetazontas
% kai to ethsio orio gia thn sygkekrimenh
% periptwsh apallaghs me kwdiko N
```

```
a_ap_poy_den_apal(_0,0).
```

```
a_ap_poy_den_apal(Mathitis,N,S):-
```

```
  N1 is N-1,
  ypologismos_m(Mathitis,N,S11),
  apallagh(N,_S12),
  ( S11>S12 -> ST1 is S12; ST1 is S11),
  a_ap_poy_den_apal(Mathitis,N1,S1),
  S is S1+ST1,
  !,true.
```

```
% o parakatw kanonas ypologizei to synolo
% apoysiwn mathiti poy an kai apallassontai
% telika lambanontai ypopsh eksetazontas
% kai to ethsio orio
```

s_ypologismos_m(Mathitis,S):-

```
( \+apoysia(Mathitis,___);  
((apoysia(Mathitis,T,___),  
 \+dikaiologhsh(Mathitis,T,___)))) ,  
S is 0,!,true.
```

s_ypologismos_m(Mathitis,S):-

```
arithmos_periptwsewn_apallaghs(N),  
a_ap_poy_den_apal(Mathitis,N,S).
```

%o kanonas ypologismos_a_d paragei ton arithmo apoysivn
% mathith poy einai dikaiologhmenes.

s_a_d(Mathitis,J,T):-

```
J1 is J-1,  
apoysia(Mathitis,J,X,_) ,s_a_d(Mathitis,J1,T1),  
( dikaiologhmenh(Mathitis,J) ->R is X; R is 0),  
T is T1+R,!,true.
```

s_a_d(_,0,0).

s_a_d_y_a_p(M,I,0):-

```
c(M,K),K<I,  
!,true.
```

s_a_d_y_a_p(Mathitis,J,T):-

```
J1 is J+1,  
s_a_d_y_a_p(Mathitis,J1,T1),  
( (apoysia(Mathitis,J,X,_) ,  
dikaiologhmenh(Mathitis,J),  
parastatiko(Mathitis,J),  
orio_parastatikwn(L),  
( J <L;J=L))->  
(T is T1+X);  
(T is T1)),  
!,true.
```

ypologismos_a_d_y_p(Mathitis,D):-

```
( \+apoysia(Mathitis,___);
\+((apoysia(Mathitis,T,___),dikaiologhsh(Mathitis,T,___))),parastatiko(Mathitis,T))
,D is 0,
!,true.
```

ypologismos_a_d_y_p(Mathitis,D):-

```
s_a_d_y_a_p(Mathitis,1,D).
```

ypologismos_a_d(Mathitis,D):-

```
( \+apoysia(Mathitis,___);
((apoysia(Mathitis,T,___),\+dikaiologhsh(Mathitis,T,___))))
,D is 0.
```

ypologismos_a_d(Mathitis,D):-

```
c(Mathitis,N),s_a_d(Mathitis,N,D1),
ypologismos_a_d_y_p(Mathitis,D2),
D is D1-D2.
```

```
% o kanonas ypologismos_a_a paragei ton arithmo apoysivn
% mathith poy einai adikaiologhtes.
% Sto apotelesma tha prostethoyn kai oi
% dikaiologhmenes apoysies poy o arithmos
% toys yperbainei to orio poy uetei
```

s_a_a(Mathitis,J,T):-

```
J1 is J-1,
apoysia(Mathitis,J,X,___),
dikaiologhsh(Mathitis,J,___H),
s_a_a(Mathitis,J1,T1),
((\+apallagh(H,___),
\+dikaiologhmenh(I,J))
,\+dikaiologhmenh_l_a(I,J)) -> R is X; R is 0),
T is T1+R,
!,true.
```

s_a_a(_,0,0).

ypologismos_a_a(Mathitis,D):-

```
( \+apoysia(Mathitis,___);
((apoysia(Mathitis,T,___),\+dikaiologhsh(Mathitis,T,___))))
```

```

,D is 0,
!,true.

ypologismos_a_a(Mathitis,D):-

    c(Mathitis,N),s_a_a(Mathitis,N,D),
    !,true.

% o kanonas ypologismos_a_d_a paragei ton arithmo apoysivn
% mathith poy einai dikaiologhmenes logo astheneias.

s_a_d_a(Mathitis,J,T):-

    J1 is J-1,
    apoysia(Mathitis,J,X,_),
    s_a_d_a(Mathitis,J1,T1),
    ( dikaiologhmenh_l_a(Mathitis,J) -> R is X; R is 0),
    T is T1+R,
    !,true.

s_a_d_a(_0,0):-

    !,true.

ypologismos_a_d_a(Mathitis,D):-

    ( \+apoysia(Mathitis,_,_,_);
    ((apoysia(Mathitis,T,_,_),\+dikaiologhsh(Mathitis,T,_,_))))
    ,D is 0,
    !,true.

ypologismos_a_d_a(Mathitis,D):-

    c(Mathitis,N),s_a_d_a(Mathitis,N,D).

% o kanonas ypologismos_a paragei ton arithmo apoysivn
% mathith poy lambanontai ypopsh genika

s_a(I,_T):-

    \+apoysia(I,_,_,_),
    T is 0,
    !,true.

s_a(Mathitis,J,T):-

```



```

J1 is J-1,
apoyisia(Mathitis,J,X,_),
dikaiologhsh(Mathitis,J,_H),!,
s_a(Mathitis,J1,T1),
( \+apallagh(H,_)) -> R is X; R is 0),
T is T1+R,
!,true.

s_a(_0,0).

ypologismos_a(Mathitis,D):-

( \+apoyisia(Mathitis,_));
( (apoyisia(Mathitis,T,_),\+dikaiologhsh(Mathitis,T,_)))
,D is 0,
!,true.

ypologismos_a(Mathitis,D):-

c(Mathitis,N),s_a(Mathitis,N,D).

% ypologimsos apoyision poy lambanontai
% ypopsh ston xarakthrismo mathiti.

synolo_a_y(Mathitis,N):-

s_ypologismos_m(Mathitis,N1),
ypologismos_a(Mathitis,N2),
N is N1+N2.

% H check_case elegxei ean o mathiths X
% empiptei se kapoia apo tiw periptwseis

check_case(_0):-

!,false.

check_case(X,C):-

( ( test_student_aa(X,C),
periptwsh(C,Per),write('XARAKTHRIZETAI BASH THS PERIPTWSHS:'),write(Per),nl )
;(C1 is C-1,
check_case(X,C1)))
!,true.

```

eparkis(X):-

```
student(X,_),diagogh(X,_),mo(X,_),
arithmos_periptwsewn_eparkeias(N),
check_case(X,N).
```

```
% to parakatv afora thn periptwsh opoy
% gia mauhth den exoyme xarakthrismo
% se kapoia apoysia toy
```

aksiologhsh(Mathitis,Onoma,Y):-

```
apoysia(Mathitis,Aa,_),
\+dikaiologhsh(Mathitis,Aa,_),
student(Mathitis,Onoma,_),
Y= 'ekremhs_aksiologhsh_apoysiwn',
!, true.
```

aksiologhsh(Mathitis,Onoma,Y):-

```
(eparkis(Mathitis)-> Z is 0;Z is 1),
( ( student(Mathitis,Onoma,C),Z==0,Y=eparkhs,!,true)
;(student(Mathitis,Onoma,C),Z==1,Y=aneparkhs,!,true))
.
```

aksiologhsh(Mathitis,_Y):-

```
\+student(Mathitis,_),Y=oxi_mathiths,!,true.
```

report(N):-

```
mo(N,G),diagogh(N,D),
write('mesos oros bathmologias:'),write(G),nl,
write('Diagwgh:'),write(D),nl,
write('apoysies poy lambanontai ypopsh:'),
ypologismos_a(N,N1),write(N1),nl,
write('apoysies poy lambanontai ypopsh logw yperbashs orioy :'),
write('apoysiwn poy den lambanontai ypopsh:'),
s_ypologismos_m(N,N2),write(N2),nl,
write('apoysies poy exoyn dikaiologhthei:'),
ypologismos_a_d(N,N3),write(N3),nl,
write('apoysies poy exoyn dikaiologhthei logw astheneias:'),
ypologismos_a_d_a(N,N4),write(N4),
```

```

!,true.

report(_):-

    nl,write('yparxoyrn eleipseis sta stoixeia mathith'),nl.

lista_mathitwn(N,_):-

    N==0,!,true.

lista_mathitwn(N,C):-

    ( ( student(N,_C1),C1=C,
aksiologhsh(N,Onoma,Epidosh)
,write('====='),nl,write(Onoma),
write('--'),write(Epidosh),nl,
report(N),nl,nl,!,true);
(\+ student(N,_));(\+ c1==C)
, N1 is N-1,lista_mathitwn(N1,C).

kartella(Aa,Found):-

    \+ student(Aa,_),write('den yparxei mathitis'),
Found is 0 ,nl,
!,true.

kartella(Onoma,G,D,Aa,Cname,S,N1,N2,Epidosh):-

    student(Aa,Onoma,C),class(C,Cname),
mo(Aa,G),
diagogh(Aa,D),
synolo_a_y(Aa,S),ypologismos_a_d_a(Aa,N1),ypologismos_a_d(Aa,N2),
aksiologhsh(Aa,Onoma,Epidosh)
,!,true.

lista_class(C):-

    class(C,Cname),
write('-----taksh:'),write(Cname),write('---'),
nl,
find(N),lista_mathitwn(N,C),
C1 is C-1,
lista_class(C1).

telika_apotelesmata:-

    arithmos_class(M),lista_class(M).

report(_0):-

```

```

!,true.
report_a(S,N):-
    nl,
    apoysia(S,N,C,M,D),
    write(C),write(' apoysies'),
    write(' stis '),write(D),write('-'),
    write(M),nl,
    ( (\+ dikaiologhsh(S,N,_,_),write('den exei aksiologhthei'));
    ( dikaiologhsh(S,N,M1,D1,E),
      write('aksiologhthike stis '),write(D1),write('-'), write(M1),nl,
      (
        ( apallagh(E,Per,_,_),write('apallasetai
logw
'),write(Per));(dikaiologhmenh_logo_astheneias(E,_),
      write('einai dikaiologhmenh_logo_astheneias ');
        (dikaiologhthei(E,_),
      write('exei dikaiologhthei '));
        (adikaiologhth(E,_),
      write('den exei dikaiologhthei '))
      )),nl,nl,
    N1 is N-1,
    report_a(S,N1),!,true.

```

report_apoysies(S):-

```

\+ student(S,_,_),write('oxi mathiths'),
nl,
!,true.

```

report_apoysies(S):-

```

student(S,O,C),class(C,Cper),c(S,N),write('O mathiths me onoma '),write(O),
write(' kai takshn'),write(Cper),write(' exeitiw ekshs apoysies'),nl,
report_a(S,N).

```

report_day(D,H):-

```

\+ (
apoysia(X,P,_,D,H),
\+ q(X,P)),e_q,
!,true.

```

report_day(D,H):-

```

\+ q(X,P),
apoysia(X,P,N,D,H),
student(X,O,C),class(C,Cper),

```

```

write('O mathiths '),write(O),write(' thw takshs '),write(Cper),nl,
write('ekane '),write(N),write(' apoysies'),nl,
( ( dikaiologhsh(X,P,_) ,write(' exei aksiologhthei'));
( \+ dikaiologhsh(X,P,_) ,write('den exei aksiologhthei'))),nl,
assert(q(X,P)),
report_day(D,H),
!,true.

```

```

e_q:-
\+ q(_),
!,true.

```

```

e_q:-
q(X,Y),
retract(q(X,Y)),
e_q,
!,true.

```

```

report_case_apallagh(Perc):-

```

```

\+ apallagh(_Perc,_),
write('den yparxei tetoia periptwsh'),
nl,!,true.

```

```

report_case_apallagh(Perc):-

```

```

apallagh(A,Perc,_),
\+ dikaiologhsh(,_A),
write('den yparxei apoyisa me sxetikh periptwsh'),
nl,!,true.

```

```

report_case_apallagh(Perc):-

```

```

apallagh(A,Perc,_),
\+ (apoysia(X,P,_) ,
dikaiologhsh(X,P,_A),
\+ q(X,P)),
e_q,
!,true.

```

```

report_case_apallagh(Perc):-

```

```

apallagh(A,Perc,_),
\+ q(X,P),(
( apoysia(X,P,_M1,H1),

```

```

student(X,O,C),class(C,Cper), dikaiologhsh(X,P,_,A),
write('O mathiths '),write(O),write(' thw takshs '),write(Cper),nl,
write('apallasetai gia thn apoysia stis '),write(M1),write('-'),write(H1),nl);
( apoysia(X,P,_,_);dikaiologhsh(X,P,_,_)))
,assert(q(X,P)),
report_case_apallagh(Perc),
!,true.

```

report_all_case_ap(N):-

```

arithmos_periptwsewn_apallaghs(M),
N>M,
!,true.

```

report_all_case_ap(N):-

```

apallagh(N,O,_,_),
write('PERIPTWSH: '),
write(O),nl,
write('====='),nl,
report_case_apallagh(O),
N1 is N+1,
report_all_case_ap(N1),!,true.

```

report_case_poy_den_dikaiol(Perc):-

```

\+ apoysies_poy_den_dikaiologyntai(_,Perc),
write('den yparxei tetoia periptwsh'),
nl,
!,true.

```

report_case_poy_den_dikaiol(Perc):-

```

apoysies_poy_den_dikaiologyntai(A,Perc) ,
\+ dikaiologhsh(____A),
write('den yparxei apoysia me sxetikh periptwsh mh dikaiologhshs'),
nl,
!,true.

```

report_case_poy_den_dikaiol(Perc):-

```

apoysies_poy_den_dikaiologyntai(A,Perc),
apoysia(X,P,_,M1,H1),
student(X,O,C),class(C,Cper), dikaiologhsh(X,P,_,A),
write('Gia ton mathith '),write(O),write(' thw takshs '),write(Cper),nl,
write(' den mporei na dikaiologhthei h apoysia stis '),write(M1),write('-'),
write(H1),nl,!,true.

```

report_case_adikaio:-

```
adikaiologhth(A,_),
\+ dikaiologhsh(____A),
write('den yparxei apoysia poy exei xarakthristei adikaiologhth'),
nl,!true.
```

report_case_adikaio:-

```
adikaiologhth(A,_),
apoysia(X,P,_M1,H1),
student(X,O,C),class(C,Cper), dikaiologhsh(X,P,____A),
write('Gia ton mathith '),write(O),write(' thw takshs '),write(Cper),nl,
write(' thewreitai adikaiologhth h apoysia stis '),write(M1),write('-'),write(H1),nl,!true.
```

report_case_dikaio:-

```
dikaiologhthei(A,_),
\+ dikaiologhsh(____A),
write('den yparxei apoysia poy na exei dikaiologhthei'),
nl,!true.
```

report_case_dikaio:-

```
dikaiologhthei(A,_),
apoysia(X,P,_M1,H1),
student(X,O,C),class(C,Cper), dikaiologhsh(X,P,____A),
write('Gia ton mathith '),write(O),write(' thw takshs '),write(Cper),nl,
write(' exei dikaiologhthei h apoysia stis '),write(M1),write('-'),write(H1),nl,
( ( parastatiko(X,P),write(' exei episynafthei parastatiko'));true),nl.
```

report_case_dikaio_l_a:-

```
dikaiologhmenh_logo_astheneias(A,_),
\+ dikaiologhsh(____A),
write('den yparxei apoysia poy na exei dikaiologhthei logw astheneias'),
nl,!true.
```

report_case_dikaio_l_a:-

```
dikaiologhmenh_logo_astheneias(A,_),
apoysia(X,P,_M1,H1),
student(X,O,C),class(C,Cper), dikaiologhsh(X,P,____A),
write('Gia ton mathith '),write(O),write(' thw takshs '),write(Cper),nl,
```

```
write(' exei dikaiologhthei logw astheneias h apoysia stis '),write(M1),write('-  
'),write(H1),nl,  
( ( parastatiko_apo_forea(X,P,_),write('exei episynafthei parastatiko apo foreas'));true),nl.
```


student.pl

`:- dynamic student/3.`

```
student(1, 'Μαθητής_3', 1).  
student(2, 'Μαθητής_4', 1).  
student(3, 'Μαθητής_15', 1).  
student(4, 'Μαθητής_20', 1).  
student(6, bbbbbbbbbbbbbbbb, 1).  
student(5, aaaaaaaaaaaaaaaa, 1).  
student(7, 'Μαθητής_7', 1).
```

`:- dynamic mo/2.`

```
mo(1, 20).  
mo(2, 20).  
mo(3, 17).  
mo(5, 15).  
mo(7, 15).
```

`:- dynamic diagogh/2.`

```
diagogh(1, κοσμότητα).  
diagogh(2, μέτρια).  
diagogh(3, κοσμότητα).  
diagogh(4, κοσμότητα).  
diagogh(6, κοσμία).  
diagogh(7, κοσμία).
```

sub_menus.pl

```
menu_1(X):-
  ( X == 0-> true;send(X,destroy)),

  new(D1,dialog('Μενού Προσθήκης Διαγραφής')),
  send(D1,layout_dialog),
  send(D1,append,new(BT,dialog_group(buttons,group))),
  send(BT,gap,size(0,4)),
  send(BT,append,new(B1,

  button('Καταχώρηση Νεου_μαθητή',message(@prolog,
    menu_student1,D1))),below),
  send(BT,append,new(B01,
  button('Διαγραφή - Διόρθωση μαθητή',message(@prolog,
    menu_student01,D1))),below),
  send(BT,append,new(B2,
  button('Καταχώρηση μεσου ορου βαθμολογιας',message(@prolog,
    menu_mo1,D1))),below),
  send(BT,append,new(B4,
  button('Καταχώρηση Διαγωγής Μαθητή',message(@prolog,
    menu_dia1,D1))),below),
  send(BT,append,new(B5,
  button('Καταχώρηση Απουσίας Μαθητή',message(@prolog,
    menu_ap1,D1))),below),
  send(BT,append,new(B6,
  button('Καταχώρηση Δικαιολόγησης απουσίας μαθητή',message(@prolog,
    menu_di1,D1))),below),
  send(BT,append,new(B7,
  button('Καταχώρηση απλού παραστατικού δικαιολόγησης απουσίας
μαθητή',message(@prolog,
    menu_parast1,D1))),below),
  send(BT,append,new(B8,
  button('Καταχώρηση παραστατικού δικαιολόγησης απουσίας μαθητή λογω
ασθένειας',message(@prolog,
    menu_parast_f1,D1))),below),
  send(BT,append,new(B9,
  button('Καταχώρηση ημέρας έναρξης μαθημάτων',message(@prolog,
    menu_mhnas1,D1))),below),
  send(BT,append,new(B10,
  button('Καταχώρηση ημέρας πρώτου σαββάτου',message(@prolog,
    menu_saturday1,D1))),below),
  send(BT,append,new(B11,
  button('Καταχώρηση νεας συνθήκης για τον χαρακτηρισμό επάρκειας φοίτησης'
,message(@prolog,
    menu_epa1,D1))),below),
  send(BT,append,new(B12,
```

```

button('Καταχώρηση περιόδου διακοπών',message(@prolog,
  menu_diakop,D1))),below),
send(BT,append,new(B13,
button('Διαγραφή περιόδου διακοπών',message(@prolog,
  menu_diakop_d,D1))),below),
send(BT,append,new(B14,
button('Αναζήτηση μαθητή',message(@prolog,
  menu_search,D1))),below),
  send(BT,append,new(B60,
button('Εμφάνιση απουσιών μαθητή',message(@prolog,
  menu_search_apoysies,D1))),below),
send(BT,append,new(B50,
button('Έλεγχος για περιπτώσεις που αφορούν πολλαπλές συνθήκες επάρκειας σε
μαθητή',message(@prolog,
  menu_dpl_case,D1))),below),
  send(BT,append,new(B3,
button('Έξοδος',message(@prolog,
  m_menu,D1))),below),get(D1,confirm,B2),
get(D1,confirm,B1),  get(D1,confirm,B01),  get(D1,confirm,B3),  get(D1,confirm,B4),
get(D1,confirm,B5),
get(D1,confirm,B6),
get(D1,confirm,B7),
get(D1,confirm,B8),
get(D1,confirm,B9),
get(D1,confirm,B10),
get(D1,confirm,B11),
get(D1,confirm,B12),
get(D1,confirm,B13),
get(D1,confirm,B14),
get(D1,confirm,B50),  get(D1,confirm,B60),
!,true.

```

menu(X):-

```
( X==0 -> true;send(X,destroy)),
```

```
new(D1,dialog('Μενού ελέγχου ασυμβίβαστης γνώσης')),
```

```
send(D1,layout_dialog),
```

```
send(D1, append, new(BT, dialog_group(buttons, group))),
```

```
send(BT, gap, size(0, 10)),
```

```
send(BT,append,new(B1,
```

```
button('Νεος μαθητής',message(@prolog,
  menu_student,D1))),below),
```

```
send(BT,append,new(B2,
```

```
button('Νεος μεσος ορος βαθμολογιας',message(@prolog,
  menu_mo,D1))),below),
```

```

send(BT,append,new(B4,
button('Διαγωγή Μαθητή',message(@prolog,
menu_dia,D1))),below),
send(BT,append,new(B5,
button('Απουσία Μαθητή',message(@prolog,
menu_ap,D1))),below),
send(BT,append,new(B6,
button('Δικαιολόγηση απουσίας μαθητή',message(@prolog,
menu_di,D1))),below),
send(BT,append,new(B7,
button('Καταχώρηση απλού παραστατικού δικαιολόγησης απουσίας
μαθητή',message(@prolog,
menu_parast,D1))),below),
send(BT,append,new(B8,
button('Καταχώρηση παραστατικού δικαιολόγησης απουσίας μαθητή λόγω
ασθένειας',message(@prolog,
menu_parast_f,D1))),below),
send(BT,append,new(B9,
button('Αλλαγή κλίμακας ',message(@prolog,
menu_klimaka1,D1))),below),
send(BT,append,new(B10,
button('Καταχώρηση νεου χαρακτηρισμου διαγωγης',message(@prolog,
menu_e_dia,D1))),below),
send(BT,append,new(B11,
button('Καταχώρηση νεας συνθήκης για τον χαρακτηρισμό επάρκειας φοίτησης'
,message(@prolog,
menu_epa,D1))),below),
send(BT,append,new(B3,
button('Έξοδος',message(@prolog,
m_menu,D1))),below),
get(D1,confirm,B1),
get(D1,confirm,B2),
get(D1,confirm,B3),
get(D1,confirm,B4),
get(D1,confirm,B5),
get(D1,confirm,B6),
get(D1,confirm,B7),
get(D1,confirm,B8),
get(D1,confirm,B9),
get(D1,confirm,B10),
get(D1,confirm,B11),

```

!,true.

menu_2(X):-

```
( X == 0 ->true;send(X,destroy)),
```

```

new(D1,dialog('Μενού Παραμέτρων')),
send(D1,size,size(400,800)),
send(D1,layout_dialog),

```

```

send(D1,append,new(BT,dialog_group(buttons,group))),

send(BT,append,new(B1,

button('Ημερήσιο όριο διδασκαλίας',message(@prolog,
menu_h_orio,D1))),below),
send(BT,append,new(B2,

button('Τάξη Μαθητή',message(@prolog,
menu_class,D1))),below),
send(BT,append,new(B3,
button('Περίπτωση απαλλαγής',message(@prolog,
menu_apallagh,D1))),below),
send(BT,append,new(B31,
button('Περίπτωση που δεν μπορεί να δικαιολογηθεί',message(@prolog,
menu_den_d,D1))),below),

send(BT,append,new(B4,
button('Όριο ημερών στις οποίες μπορεί να δικαιολογηθεί απουσία',message(@prolog,
menu_orio_hm,D1))),below),
send(BT,append,new(B5,
button('Νέα κλίμακα βαθμολογίας',message(@prolog,
menu_klimaka,D1))),below),
send(BT,append,new(B20,
button('Καταχώρηση τίτλου διαγωγής μαθητή',message(@prolog,
menu_diag,D1))),below),
send(BT,append,new(B30,
button('Όριο παραστατικών ανά μαθητή',message(@prolog,
menu_parast_orio,D1))),below),

send(BT,append,new(B7,
button('Εμφάνιση συνθηκών επάρκειας',message(@prolog,
menu_epa2,D1,1))),below),
send(BT,append,new(B8,
button('Εμφάνιση περιπτώσεων απαλλαγής',message(@prolog,
menu_apa2,D1,1))),below),
send(BT,append,new(B9,
button('Εμφάνιση περιπτώσεων που δεν δικαιολογούνται',message(@prolog,
menu_p_d_d2,D1))),below),
send(BT,append,new(B10,
button('Εμφάνιση απουσιών συγκεκριμένου μαθητή',message(@prolog,
menu_a_l,D1))),below),
send(BT,append,new(B11,
button('Εμφάνιση Επιδόσεων συγκεκριμένης τάξης',message(@prolog,
menu_e_c,D1))),below),
send(BT,append,new(B12,
button('Εμφάνιση απουσιών συγκεκριμένης ημερομηνίας',message(@prolog,
menu_a_h,D1))),below),
send(BT,append,new(B13,
button('Εμφάνιση απουσιών που δεν μπορούν να δικαιολογηθούν',message(@prolog,

```

```

menu_a_d_d,D1))),below),
send(BT,append,new(B14,
button('Εμφάνιση απουσιών που απαλλάσσονται',message(@prolog,
menu_a_a,D1))),below),
send(BT,append,new(B15,
button('Εμφάνιση απουσιών που έχουν δικαιολογηθεί λόγω
ασθένειας',message(@prolog,
menu_a_d_a,D1))),below),

send(BT,append,new(B16,
button('Εμφάνιση απουσιών που έχουν δικαιολογηθεί',message(@prolog,
menu_a_d,D1))),below),
send(BT,append,new(B6,
button('Εξοδος',message(@prolog,
m_menu,D1))),below),
get(D1,confirm,B1),
get(D1,confirm,B2),
get(D1,confirm,B3),
get(D1,confirm,B31),
get(D1,confirm,B4),
get(D1,confirm,B5),
get(D1,confirm,B6),
get(D1,confirm,B7),
get(D1,confirm,B8),
get(D1,confirm,B9),
get(D1,confirm,B10),
get(D1,confirm,B11),
get(D1,confirm,B12),
get(D1,confirm,B13),
get(D1,confirm,B14),
get(D1,confirm,B15),
get(D1,confirm,B16),
get(D1,confirm,B20),
get(D1,confirm,B30),
!,true.

```

test_cons.pl

```
d_qc:-  
  \+ qc(_),!,true.
```

```
d_qc:-  
  qc(X),  
  retract(qc(X)),  
  d_qc!,true.
```

```
test_incons_about_cases(P1):-  
  \+ periptwsh(_P1),  
  write('den yparxei h synuthkh eparkeias'),!,true.
```

```
test_incons_about_cases(P1):-  
  periptwsh(X,P1),  
  periptwsh(Y,Per2),  
  \+ qc(Y),  
  
  \+X = Y,  
  exe_i_diagogh(X,D1),  
  exe_i_diagogh(Y,D2),  
  D1=D2,  
  synolo(X,L1,R1),  
  synolo(Y,L2,R2),  
  ( test_range(L1,R1,L2,R2);test_range(L2,R2,L1,R1))  
  ,  
  synolo_a(X,L3,R3),  
  synolo_a(Y,L4,R4),  
  ( test_range(L3,R3,L4,R4);test_range(L4,R4,L3,R3))  
  ,  
  synolo_d_a(X,L5,R5),  
  synolo_d_a(Y,L6,R6),  
  ( test_range(L5,R5,L6,R6);test_range(L6,R6,L5,R5))  
  ,  
  synolo_d(X,L7,R7),  
  synolo_d(Y,L8,R8),  
  ( test_range(L7,R7,L8,R8);test_range(L8,R8,L7,R7))  
  ,  
  grades(X,L9,R9),  
  grades(Y,L10,R10),  
  ( test_range(L9,R9,L10,R10);test_range(L10,R10,L9,R9))  
  ,  
  write(Per2),nl,
```

```

assert(qc(Y)),
test_incons_about_cases(P1),!,true.

test_incons_about_cases(_):- d_qc,!,true.

test_incons_about_apoysies:-
  apoysia(X,A,_,_),
  \+ ( student(X1,_,_),X1 = X ),
  write('H apoysia ME KWDIKO MATHITH:'),write(X),nl,
  write('kai ayksonta arithmo:'),write(A),nl,
  write('den antistoixei se mathith'),nl.

test_incons_about_apoysies:-
  apoysia(X,A,M,_,_),
  hmerhsio_orio_didaskalias(N),M>N,
  student(X,O,_,_),
  write('H apoysia TOY MATHITH:'),write(O),nl,
  write('kai ayksonta arithmo:'),write(A),nl,
  write('yperbainei to hmerhsio orio wrvn didaskalias'),nl.

test_incons_about_apoysies:-
  apoysia(X,A,P1,M1,D1),
  apoysia(X,A,P2,M2,D2),
  ( \+P1=P2; \+M1=M2;\+D1=D2),
  student(X,O,_,_),
  write('Ston MATHITH me onoma :'),write(O),nl,
  write('yparxoyn panw apo mia apoysia me ayksonta arithmo:'),write(A),nl,!,true.

test_incons_about_apoysies:-
  student(X,_,_),
  apoysia(X,Y1,_,_,_),
  apoysia(X,Y2,_,_,_),Z is Y2-Y1,Z>1,
  \+ (apoysia(X,Y3,_,_,_),Y3>Y1,Y3<Y2),
  write('MATHITIS:'),write(X),nl,
  write('LEIPOYN APOYSIES ME AYKSONTES ARITHMOYS:'),
  Z1 is Y1+1,Z2 is Y2-1,write(Z1),write('ews'),write(Z2),
  nl.

test_incons_about_apoysies:-
  apoysia(X,A,_,_,_),
  \+ ( dikaiologhsh(X,A,_,_,_)),

  write('H apoysia ME KWDIKO MATHITH:'),write(X),nl,
  write('kai ayksonta arithmo:'),write(A),nl,
  write('den exei aksiologhthei'),nl.

```



```

test_incons_about_apoysies:-
  dikaiologhsh(X,A,_,_),
  \+ ( apoysia(X,A,_,_)),

  write('Den yparxei apoysia ME KWDIKO MATHITH:')
  ,write(X),nl,
  write('kai ayksonta arithmo:'),write(A),nl,
  write('h opoia exei aksiologhthei'),nl.

```

```

test_incons_about_apoysies:-
  apoysia(X,A,_,M,_)
  \+ ( mhnas(,X1,_)X1=M),

  write('Gia thn apoysia ME KWDIKO MATHITH:'),
  write(X),nl,
  write('kai ayksonta arithmo:'),write(A),nl,
  write('den antistoixei gnvstos mhnas'),nl.

```

```

test_incons_about_apoysies:-
  dikaiologhsh(X,A,M,_,_),
  \+ ( mhnas(,X1,_)X1=M),

  write('Gia thn dikaiologhsh apoysias ME KWDIKO MATHITH:'),
  write(X),nl,
  write('kai ayksonta arithmo:'),write(A),nl,
  write('den antistoixei gnvstos mhnas'),nl.

```

```

test_incons_about_apoysies:-
  dikaiologhsh(X,A,_,_,K),
  apoysia(X,A,_,_),
  \+ ( apallagh(K,_,_));
  apoysies_poy_den_dikaiologoyntai(K,_)
  dikaiologhmenh_logo_astheneias(K,_)
  dikaiologhthei(K,_)
  adikaiologhth(K,_)
  write('Gia thn dikaiologhsh apoysias ME KWDIKO MATHITH:'),
  write(X),nl,
  write('kai ayksonta arithmo:'),write(A),nl,
  write('einai agnvstos o kwdikos dikaiologhshs'),nl.

```

```

test_incons_about_apoysies:-
  apoysia(X,A,B1,C1,K1),
  apoysia(X,A,B2,C2,K2),
  ( \+B1=B2; \+C1=C2; \+K1=K2),
  write('H apoysia ME KWDIKO MATHITH:'),
  write(X),nl,
  write('kai ayksonta arithmo:'),write(A),nl,
  write('emfanizetai polles fores'),nl,!true.

```

```

test_incons_about_apoysies:-

```

```
dikaiologhsh(X,A,B1,C1,K1),
dikaiologhsh(X,A,B2,C2,K2),
( \+B1=B2; \+C1=C2; \+K1=K2),
```

```
write('H dikaiologhsh sthn apoysia ME KWDIKO MATHITH:'),
write(X),nl,
write('kai ayksonta arithmo:'),write(A),nl,
write('emfanizetai polles fores'),nl,! ,true.
```

```
test_incons_about_apoysies:-
dikaiologhsh(X,A,Md,Hd,_),
apoysia(X,A,_Ma,Ha),
mhnas(_Ma,Na), mhnas(_Md,Nd), Sa is Na+Ha,
Sd is Nd+Hd, Sa>Sd,
```

```
write('Sthn apoysia ME KWDIKO MATHITH:')
,write(X),nl,
write('kai ayksonta arithmo:'),write(A),nl,
write('exei dikaiologhthei se xrono progenestero'),
write('ths emfanishs'),nl.
```

```
test_incons_about_apoysies:-
apoysia(X,A,_Ma,Ha),
day_count(Ma,Ha,N),
elegxos_mh_erg(N),
write('Sthn apoysia ME KWDIKO MATHITH:')
,write(X),nl,
write('kai ayksonta arithmo:'),write(A),nl,
write('exei kataxwrhthei hmeromhnia se mh ergasimh hmera'),
nl.
```

```
test_incons_about_apoysies:-
dikaiologhsh(X,A,Md,Hd,_),
day_count(Md,Hd,N),
elegxos_mh_erg(N),
write('Sthn dikaiologhsh apoysias ME KWDIKO MATHITH:')
,write(X),nl,
write('kai ayksonta arithmo:'),write(A),nl,
write('exei kataxwrhthei hmeromhnia se mh ergasimh hmera'),
nl.
```

```
test_incons_about_apoysies:-
apoysia(X,A,_Md,Hd),
mhnas(K,Md,Sd),
K1 is K+1,
mhnas(K1,_Sd1),
R is Sd+Hd,
R>Sd1,
write('Sthn katagrafh apoysias ME KWDIKO MATHITH:')
```

```
,write(X),nl,
write('kai ayksonta arithmo:'),write(A),nl,
write('exei kataxwrthei mh epitreptos arithmos hmeras'),
nl.
```

```
test_incons_about_apoysias:-
dikaiologhsh(X,A,Md,Hd,_),
mhnas(K,Md,Sd),
K1 is K+1,
mhnas(K1,_Sd1),
R is Sd+Hd,
R>Sd1,
write('Sthn dikaiologhsh apoysias ME KWDIKO MATHITH:')
,write(X),nl,
write('kai ayksonta arithmo:'),write(A),nl,
write('exei kataxwrthei mh epitreptos arithmos hmeras'),
nl.
```

```
test_student1:-
student(X,N,_),
\+mo(X,_),
\+qc(X),
write('O mathiths me onoma:'),
write(N),nl,
write('kai kwdiko:'),write(X),nl,
write('den exei meso oro bathmologias'),nl,
assert(qc(X)),
test_student1,!true.
```

```
test_student1:- d_qc,!true.
```

```
test_student2:-
student(X,N,_),
\+diagogh(X,_),
\+qc(X),
write('O mathiths me onoma:'),
write(N),nl,
write('kai kwdiko:'),write(X),nl,
write('den toy exei apodothei diagwgh'),nl,
assert(qc(X)),
test_student2,!true.
```

```
test_student2:- d_qc,!true.
```

```
test_student3:-
student(X,N,_),
mo(X,G),klimaka(E,M),\+qc(X),
( G<E;G>M),
write('O mathiths me onoma:'),
```

```

write(N),nl,
write('kai kwdiko:'),write(X),nl,
write('exei meso oro bathmologias ektos klimakas'),nl,
assert(qc(X)),
test_student3,!true.

```

```

test_student3:- d_qc,!true.

```

```

test_student:-
  student(X,N,_),
  diagogh(X,D),
  \+ ( epipeda_diagoghs(Ed),Ed=D ),
  write('O mathiths me onoma:'),
  write(N),nl,
  write('kai kwdiko:'),write(X),nl,
  write(' toy exei apodothei diagwgh agnwstoy epipedoy'),nl.

```

```

test_student_a(X,P1):-
  student(X,_),
  periptwsh(C1,P1),
  test_student_aa(X,C1),!,true.

```

```

test_student_aa(X,C1):-
  student(X,_),
  periptwsh(C1,_),
  synolo_a_y(X,M1),
  ypologismos_a_a(X,M2),
  ypologismos_a_d(X,M3),
  ypologismos_a_d_a(X,M4),
  diagogh(X,T),
  mo(X,M5),
  ( synolo(C1,N1,N2) -> F1 is 1 ; F1 is 0),
  ( synolo_a(C1,N3,N4) -> F2 is 1 ; F2 is 0),
  ( synolo_d_a(C1,N5,N6) -> F3 is 1 ; F3 is 0),
  ( synolo_d(C1,N7,N8) -> F4 is 1 ; F4 is 0),
  ( grades(C1,N9,N10) -> F5 is 1 ; F5 is 0),
  ( exei_diagogh(C1,Dia) -> F6 is 1 ; F6 is 0),

  ( ( F1==1,M1>=N1,(M1<N2;M1==N2)); F1==0),
  ( ( F2==1,M2>=N3,(M2<N4;M2==N4)); F2==0),
  ( ( F3==1,M4>=N5,(M4<N6;M4==N6)); F3==0),
  ( ( F4==1,M3>=N7,(M3<N8;M3==N8)); F4==0),
  ( ( F5==1,M5>=N9,(M5<N10;M5==N10)); F5==0),
  ( (F6==1, T==Dia); F6==0),!,true.

```

```

dupl_case(_):- \+ (periptwsh(X,_),\+ qc(X)),d_qc!,true.
dupl_case(C):- C=0,d_qc!,true.
dupl_case(C):-
    (
        periptwsh(C,Per),periptwsh(C1,Per1),\+
        student(X,_),test_student_aa(X,C),test_student_aa(X,C1),
        ( ( \+ qc(C1),assert(qc(C1)),write(Per1),nl);
        ( \+ qc(C),assert(qc(C)),write(Per),nl)); true),
        C2 is C-1,dupl_case(C2),!,true.
    )
    C==C1,

```

```

report_cases:-
    write('periptwseis oi opoies perigrafoyn taytoxrona eparkeia:'),nl,
    arithmos_periptwsewn_eparkeias(N),
    dupl_case(N),!,true.

```

```

test_incons_about_klimaka:-
    periptwsh(P,Per),
    grades(P,L,R),
    klimaka(L1,R1),
    ( L<L1;R>R1),
    write('H periptwsh'),nl,
    write(Per),nl,
    write('anaferetai se orio vathmwn ektos klimakas'),
    nl.

```

```

test_incons_about_klimaka:-
    klimaka(L1,R1),
    student(S,O,_),mo(S,G),
    ( G<L1;G>R1),
    write('O mathiths'),nl,
    write(O),nl,
    write('exei meso oro vathmwn ektos klimakas'),
    nl.

```

```

test_incons_about_diagogh:-
    periptwsh(P,Per),
    exeidiagogh(P,Per),
    \+ (epipeda_diagoghs(Per1),Per1=Per),
    write('H periptwsh'),nl,
    write(Per),nl,
    write('anaferetai se agnwsto epipedo diagwghs'),
    nl.

```

```

test_incons_about_diagogh:-
    student(S,O,_),diagogh(S,P),
    \+ (epipeda_diagoghs(Per1),Per1=P),

```

```
write('O mathiths'),nl,  
write(O),nl,  
write('exei agnwsto epipedo diagvghs'),  
nl.
```

```
s_d(S,M,D,Sum):-  
  \+ ( apoysia(S,A,M,D), \+ qc(A) ),  
  Sum is 0,d_qc!,true.
```

```
s_d(S,M,D,Sum):-  
  apoysia(S,A,N,M,D),  
  \+ qc(A),  
  assert(qc(A)),  
  
  s_d(S,M,D,S1), Sum is S1+N!,true.
```

update_knowledge.pl

```
update_student:-  
    tell('student.pl'),  
    listing(student),  
    listing(mo),  
    listing(diagogh),  
    told,!,true.
```

```
update_facts:-  
    tell('facts.pl'),  
    listing(apoysia),  
    listing(dikaiologhsh),  
    listing(parastatiko),  
    listing(parastatiko_apo_forea),  
    told,!,true.
```

```
update_per_epa:-  
    tell('per_eparkeias.pl'),  
    listing(periptwsh),  
    listing(exei_diagogh),  
    listing(synolo),  
    listing(synolo_d),  
    listing(synolo_a),  
    listing(synolo_d_a),  
    listing(grades),  
    listing(arithmos_periptwsewn_eparkeias),  
    told,!,true.
```

```
update_dates:-  
    tell('dates.pl'),  
    listing(mhnas),  
    listing(sunday),  
    listing(saturday),  
    listing(diakopes),  
    told,!,true.
```

```
update_parameters:-  
    tell('parameters.pl'),  
    listing(hmerhsio_orio_didaskalias),  
    listing(arithmos_class),  
    listing(class),
```

listing(arithmos_periptwsewn_apallaghs),
listing(apallagh),
listing(apoysies_poy_den_dikaiologoyntai),
listing(dikaiologhmenh_logo_astheneias),
listing(dikaiologhthei),
listing(adikaiologhth),
listing(orio_hmerwn),
listing(klimaka),
listing(epipeda_diagoghs),
listing(orio_parastatikwn),
told!,true.

user_interface.pl

```
:- dynamic q2/1.
:- dynamic q_temp/1.
:- dynamic q_temp1/2.
:- dynamic q_temp2/2.
:- dynamic q8/2.
:- dynamic q5/2.
:- dynamic q/2.

:- use_module(library(tabular)).
:- use_module(library(autowin)).

d_q2:-
    q2(_),retract(q2(_)),
    d_q2,
    !,true.

d_q2:-
    !,true.

d_q8:-
    q8(_),retract(q8(_)),
    d_q8,
    !,true.

d_q8:-
    !,true.

w_label4(X):-
    new(D0,dialog('Μήνυμα')),
    send(D0,append,new(label(t1,X))),
    send(D0,append,new(B0,button(ok,message(@prolog,
    m_menu,D0))))),
    get(D0,confirm,B0),
    free(D0),
    !,true.

ok1(X):-
    send(X,destroy),
    abort,
    !,true.
```

new_p(X):-

```
    send(X,destroy),
    reset,
    w_label4('Η αρχικοποίηση ολοκληρώθηκε'),
    !,true.
```

w_label2(X):-

```
    new(D7,dialog('Μήνυμα')),
    send(D7,append,new(label(t1,X))),
```

```
send(D7,append,new(B0,button(ok,message(@prolog,
    menu_1,D7))))),
```

```
    get(D7,confirm,B0),
    !,true.
```

disp_message(X):-

```
    new(D0,dialog('Μήνυμα')),
    send(D0,append,new(label(t1,X))),
    send(D0,append,new(B0,button(ok,message(@prolog,
    ok,D0))))),
```

```
    get(D0,confirm,B0),
    !,true.
```

ok(X):-

```
    send(X,destroy)
    ,menu(0),
    !,true.
```

d_mes(X):-

```
    X == 0,
    new(D0,dialog('Μήνυμα')),
    send(D0,append,new(label(t2,'Μη ορθή γνώση'))),
    send(D0,append,new(B1,button(check,message(@prolog,menu,D0) ) ) ),
    get(D0,confirm,B1)
    ,!,true.
```

d_mes(X):-

```
    X == 2,
    new(D0,dialog('Μήνυμα')),
    send(D0,append,new(label(t2,'Δεν δημιουργεί σύγκρουση'))),
    send(D0,append,new(B1,button(check,message(@prolog,menu,D0) ) ) ),
    get(D0,confirm,B1)
    ,!,true.
```

```
d_mes(X,S,X1,X2,X3,X4,X5):-
    X == 1,
    ( ( S == 1,assert(mo(X1,X2)));
      ( S == 2,assert(diagogh(X1,X2)));
      ( S == 3,assert(dikaiologhsh(X1,X2,X3,X4,X5 ) ) ) ),
    collision(N),
    w_label11(N,S,X1,X2,X3,X4,X5),
```

```
!,true.
```

```
s_day(I,-):-
```

```
    I==32,
    !,true.
```

```
s_day(I,D):-
```

```
    send(D,append,new(_menu_item(I))),
    I1 is I+1,
    s_day(I1,D),
    !,true.
```

```
m_class(X):-
```

```
    class(C,P),\+ q2(C),
    send(X,append,new(_menu_item(C,@default,P))),assert(q2(C)),
    m_class(X),
    !,true.
```

```
m_class(_):-
```

```
    \+ ( class(C,_),\+ q2(C)),d_q2,
    !,true.
```

```
s_mhnas(D):-
```

```
    send(D,append,new(_menu_item('ianoyarios'))),
    send(D,append,new(_menu_item('febroyarios'))),
    send(D,append,new(_menu_item('martios'))),
    send(D,append,new(_menu_item('aprilios'))),
    send(D,append,new(_menu_item('maios'))),
    send(D,append,new(_menu_item('ioynios'))),
    send(D,append,new(_menu_item('septembrios'))),
    send(D,append,new(_menu_item('oktwbrios'))),
    send(D,append,new(_menu_item('noembrios'))),
    send(D,append,new(_menu_item('dekembrios'))),
    !,true.
```

```
dq_temp:-
```

```
dq_temp(_),retract(q_temp(_)),
dq_temp,
!,true.
```

```
dq_temp:-
!,true.
```

```
d_q5:-
q5(_),retract(q5(_)),
d_q5, !,true.
```

```
d_q5:-
!,true.
```

```
n_list(I,_):-
```

```
I==300,
!,true.
```

```
n_list(I,D):-
```

```
send(D,append,new(_menu_item(I))),
I1 is I+1,
n_list(I1,D),
!,true.
```

```
pop_student(X):-
```

```
student(C,O,_),
\+ q2(C),assert(q2(C)),
send(X,append,new(_menu_item(C,@default,O))),
pop_student(X),
!,true.
```

```
pop_student(_):-
```

```
\+ (student(C,_),\+ q2(C)),
d_q2,
!,true.
```

```
pop_diagogh(X):-
```

```
epipeda_diagoghs(C),
```

```
\+ q2(C),assert(q2(C)),
send(X,append,new(_menu_item(C,@default,C))),
pop_diagogh(X),
!,true.
pop_diagogh(_):-
```

```
\+ ( epipeda_diagoghs(C),\+ q2(C)),
d_q2,
!,true.
```

```
w_label3(X):-
```

```
new(D0,dialog('Μήνυμα')),
send(D0,append,new(label(t1,X))),
send(D0,append,new(B0,button(ok,message(@prolog.menu_2,D0))))),

get(D0,confirm,B0),
!,true.
```