



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Τίτλος: Ανάπτυξη Συστήματος Γνώσης για Διάγνωση Αναιμίας

Δέσποινα Μπασδέκη (ΑΜ:3186)

Επιβλέπων Καθηγητής: Δρ. Μαρακάκης Εμμανουήλ

Επιτροπή Αξιολόγησης: Κονδυλάκης Χαρίδημος
Μαρακάκης Εμμανουήλ
Παπαδάκης Νικόλαος

Ημερομηνία Παρουσίασης: 19/07/2016

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου Δρ. Μανόλη Μαρακάκη για την υπομονή που έκανε κατά τη διάρκεια υλοποίησης της πτυχιακής εργασίας. Όπως επίσης και για την πολύτιμη βοήθεια και καθοδήγησή του, για την επίλυση διάφορων θεμάτων. Ιδιαίτερες ευχαριστίες εκφράζω στον μεταπτυχιακό φοιτητή κ. Άρη Παπακωνσταντίνου για την ανιδιοτελή προσφορά του και για το χρόνο που αφιέρωσε και τις πληροφορίες που μου μετέδωσε καθ' όλη τη διάρκεια της συγγραφής αυτής γιατί χωρίς τη βοήθεια του δεν θα ήταν δυνατή η πραγματοποίηση της εργασίας αυτής.

Σε αυτό το σημείο οφείλω να επισημάνω την ευγνωμοσύνη μου στους γονείς μου για όλα όσα μου έχουν προσφέρει στη διάρκεια αυτών των χρόνων και την αμέριστη υποστήριξη τους σε κάθε μου επιλογή και την φροντίδα τους για την καλύτερη δυνατή μόρφωσή μου.

Abstract

This project presents the development of a Knowledge System (KS) which makes diagnosis of anemia. Initially, an introduction to knowledge systems is presented, with emphasis in expert systems and tools which are used for their development. Special emphasis is given to tools which have been used for the development of this KS. That is, Prolog which has been used for the implementation of the reasoning engine and the knowledge base of KS and Java which has been used for the implementation of the interface. Then, an introduction is made to anemia. Next, the architectural design of the KS is presented and the diagnostic method which the system follows. Next, the implementation of the diagnostic system of anemia (DiSA) follows and the implementation of its interface. Next, sample session of the system illustrate its main functions. The study of the performance of DiSA is discussed next. The main part of the report discusses conclusions from the development of this major project are future extensions. There are four annexes. The first appendix shows the installation of the appropriate supporting software in order to run DiSA. The second appendix presents the terminology used and the abbreviations of medical names. In the third appendix, the Prolog code of the system is presented. In the third appendix the Java code is shown.



Σύνοψη

Αυτή η πτυχιακή εργασία παρουσιάζει την ανάπτυξη ενός Συστήματος Γνώσης (ΣΓ) το οποίο κάνει διάγνωση της αναιμίας. Αρχικά γίνεται μια εισαγωγή στα συστήματα γνώσης και ιδιαίτερα στα έμπειρα συστήματα καθώς και στα εργαλεία που χρησιμοποιούνται για την ανάπτυξη τους. Γίνεται ιδιαίτερη αναφορά στα εργαλεία που χρησιμοποιήθηκαν στην ανάπτυξη αυτού του ΣΓ, δηλαδή στη Prolog στην οποία υλοποιήθηκε η μηχανή συλλογισμών και η βάση γνώση του ΣΓ και στη Java στην οποία υλοποιήθηκε η διεπαφή του. Στη συνέχεια, γίνεται εισαγωγική παρουσίαση του θέματος της αναιμίας. Ακολουθεί, η παρουσίαση του αρχιτεκτονικού σχεδιασμού του ΣΓ καθώς και η μέθοδος διάγνωσης την οποία εφαρμόζει το διαγνωστικό σύστημα. Ακολουθεί, η περιγραφή της υλοποίησης του διαγνωστικού συστήματος αναιμίας (ΔιΣΑ) καθώς και η υλοποίηση της διεπαφή του. Ακολουθούν υποδείγματα τα οποία επιδεικνύουν τις κυριότερες λειτουργίες του ΔιΣΑ. Στη συνέχεια εξετάζεται η απόδοση του διαγνωστικού συστήματος. Το κύριο μέρος της πτυχιακής ολοκληρώνεται με τα συμπεράσματα από την ανάπτυξη αυτής της πτυχιακής εργασίας. Υπάρχουν 4 παραρτήματα. Στο πρώτο παράρτημα περιγράφεται ο τρόπος εγκατάστασης του αναγκαίου λογισμικού για να τρέξει το σύστημα ΔιΣΑ. Το δεύτερο παράρτημα παρουσιάζει την ορολογία που χρησιμοποιείται και τις συντομογραφίες των ιατρικών ονομάτων. Στο τρίτο παράρτημα, παρουσιάζεται ο κώδικας Prolog του συστήματος. Στο παράρτημα 4, παρουσιάζεται ο κώδικας Java.

Καλως Ήρθατε στο σύστημα διάγνωσης της Αναιμίας

Διαγνωστικό Σύστημα Αναιμίας (ΔΙΣΑ)

Έξοδος



Copyright by Despoina Mpasdeki
TEI CRETE

ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή	9
2. Συστήματα Γνώσης και Εργαλεία ανάπτυξης τους	10
2.1. Εισαγωγικές Έννοιες για τα Συστήματα Γνώσης.....	10
2.2. Συστήματα Γνώσης στην Ιατρική.....	11
2.3. Συστήματα Γνώσης στη διάγνωση της Αναιμίας.....	14
2.4. Ένα Σύστημα Γνώσης για τη Διάγνωση Αναιμίας.....	15
3. Η Prolog ως Εργαλείο Ανάπτυξης Συστημάτων Γνώσης.....	17
3.1. Προτάσεις και Γεγονότα σε Prolog.....	17
3.2. SWI-Prolog.....	20
4. Η Java ως Εργαλείο Κατασκευής Διεπικοινωνίας ενός Συστήματος Γνώσης.....	27
4.1. Εισαγωγή και Βασικά Χαρακτηριστικά τηςJava.....	27
4.2. Η Java ως εργαλείο διεπαφής για διαδικτυακές εφαρμογές σε Prolog.....	29
5. Εισαγωγικές Έννοιες της Αναιμίας.....	31
6. Σχεδιαστική περιγραφή του Διαγνωστικού Συστήματος Αναιμίας (ΔιΣΑ)	44
6.1. Αναπαράσταση του μηχανισμού διάγνωσης ως δέντρο αποφάσεων.....	44
6.2. Αρχιτεκτονική περιγραφή του ΔιΣΑ.....	51
6.3. Περιγραφή της διεπαφής του ΔιΣΑ.....	52
6.4. Περιγραφή της Αναπαράστασης της Βάσης Γνώσης του ΔιΣΑ.....	59
7. Υλοποίηση του Συστήματος.....	62
7.1. Περιγραφή της Υλοποίησης της Βάσης Γνώσης του ΔιΣΑ σε Prolog.....	62
7.2. Περιγραφή της υλοποίησης της διαδικασίας Διάγνωσης του ΔιΣΑ σε Prolog.....	63
7.3. Περιγραφή της υλοποίησης της διαδικασίας ενημέρωσης της βάσης γνώσης.....	64
7.4. Περιγραφή της υλοποίησης της διεπαφής του ΔιΣΑ σε Java.....	65
7.5. Περιγραφή ΣύνδεσηςJava-Prolog.....	72
8. Υποδείγματα με τις κύριες λειτουργίες του συστήματος ΔιΣΑ.....	74
9. Απόδοση του διαγνωστικού συστήματος ΔιΣΑ.....	90
10. Συμπεράσματα και Μελλοντικές Επεκτάσεις	91
11. Βιβλιογραφία	93
11.1. Βιβλία.....	93
11.2. Τοποθεσίες και άρθρα Web.....	93
12. Παράρτημα Α: Εγκατάσταση Βασικών Προγραμμάτων	95
12.1. Εγκατάσταση Netbeans.....	95
12.2. Εγκατάσταση SWI-Prolog.(*).....	100
12.3. Βήματα.....	102
13. Παράρτημα Β: Ορισμοί μεταβλητών του συστήματος	104
14. Παράρτημα Γ: Κώδικας Prolog	105
15. Παράρτημα Δ: ΚώδικαςJava.....	118

Λίστα Εικόνων

Εικόνα 1:Σύστημα γνώσης.....	13
Εικόνα 2: Πάνελ επιλογών ΔιΣΑ	16
Εικόνα 3:SWI μήνυμα καλωσορίσματος	21
Εικόνα 4:φόρτωση αρχείου anemia_diagnosis.pl	22
Εικόνα 5: Παράδειγμα ερώτησης σε Prolog	23
Εικόνα 6:Ιχνηλάτιση(tracing).....	24
Εικόνα 7:Γραφικό ιχνηλάτισης(tracing)	26
Εικόνα 8: Πεδίο Ονοματεπώνυμο	27
Εικόνα 9: Μόριο αιμοσφαιρίνης	31
Εικόνα 10:πιθανότητα πάθησης τέκνων όταν ο ένας γονέας είναι υγιής και ο άλλος φορέας β μεσογειακής αναιμίας.....	40
<i>Εικόνα 11: πιθανότητα πάθησης τέκνων όταν και οι δύο γονείς είναι φορείς β μεσογειακής αναιμίας.....</i>	<i>40</i>
Εικόνα 12:Ασθενής με μεσογειακή αναιμία	41
Εικόνα 13: Διάγραμμα	45
Εικόνα 14: Αρχιτεκτονική Συστήματος ΔιΣΑ	51
Εικόνα 15:Αρχική Οθόνη ΔιΣΑ	53
Εικόνα 16:Παράθυρο Συμπτωμάτων	54
Εικόνα 17:Παράθυρο Διάγνωσης.....	54
Εικόνα 18:Παράθυρο Αναζήτησης	55
Εικόνα 19:Παράθυρο ενημέρωσης βάσης κανόνων	55
Εικόνα 20: Παράθυρο εξετάσεων	56
Εικόνα 21: Παράθυρο έξτρα εξετάσεων	57
Εικόνα 22: Παράθυρο στοιχείων ΔιΣΑ.....	68
Εικόνα 23: Παράθυρο Εξετάσεων	70
Εικόνα 24: Παράθυρο έξτρα εξετάσεων	70
Εικόνα 25: κουτί επιλογής	71
Εικόνα 26: Στοιχεία ασθενή	74
Εικόνα 27:Δεδομένα εξετάσεων	75
Εικόνα 28: Συμπτώματα ασθενούς	76
Εικόνα 29: Διάγνωση ασθενούς.....	76
Εικόνα 30: Θεραπεία ασθενούς.....	77
Εικόνα 31: Δεδομένα για αλλαγή κανόνα.....	77
Εικόνα 32: Επιτυχής αλλαγή κανόνα	78
Εικόνα 33: Αναζήτηση Ασθενούς.....	78
Εικόνα 34: Επιτυχής αναζήτηση ασθενούς.....	79
Εικόνα 35: Εμφάνιση δεδομένων ασθενή	79
Εικόνα 36: Σιδηροπενική_Αναιμία:Στοιχεία Ασθενή.....	80
Εικόνα 37: Σιδηροπενική_Αναιμία:Εξετάσεις ΓΕΑ	80
Εικόνα 38: Σιδηροπενική_Αναιμία:Συμπτώματα	81
Εικόνα 39: Σιδηροπενική_Αναιμία:Διάγνωση.....	81
Εικόνα 40: Στίγμα_Μεσογειακής:Στοιχεία Ασθενή	82
Εικόνα 41: Στίγμα_Μεσογειακής:Εξετάσεις	82
Εικόνα 42: Στίγμα_Μεσογειακής:Συμπτώματα.....	83
Εικόνα 43: Στίγμα_Μεσογειακής:Διάγνωση	83
Εικόνα 44: Μεσογειακή_Αναιμία:Στοιχεία Ασθενή.....	84
Εικόνα 45: Μεσογειακή_Αναιμία:Εξετάσεις ΓΕΑ	84

Εικόνα 46: Μεσογειακή_Αναιμία:Εξτρα εξετάσεις.....	85
Εικόνα 47: : Μεσογειακή_Αναιμία:Συμπτώματα	85
Εικόνα 48: Μεσογειακή_Αναιμία:Διάγνωση.....	86
Εικόνα 49: B12_Αναιμία:Στοιχεία Ασθενή	87
Εικόνα 50: B12_Αναιμία:Εξετάσεις ΓΕΑ.....	87
Εικόνα 51: B12_Αναιμία: Εξτρα εξετάσεις	88
Εικόνα 52: B12_Αναιμία: Συμπτώματα.....	88
Εικόνα 53: B12_Αναιμία:Διάγνωση	89
Εικόνα 54:JDK εγκατάσταση.....	95
Εικόνα 55: Ρυθμίσεις για προχωρημένους.....	96
Εικόνα 56:JavaPath(μονοπάτι).....	97
Εικόνα 57:JDK Netbeans εγκατάσταση.....	98
Εικόνα 58: Glassfish εγκατάσταση(συνέχεια Netbeans)	98
Εικόνα 59Netbeansεγκατάσταση	99
Εικόνα 60Netbeans Συντόμευση.....	99
Εικόνα 61:Netbeans-Περιβαλλον.....	100
Εικόνα 62: SWI εγκατάσταση 1.....	101
Εικόνα 63: SWI εγκατάσταση 2.....	102
Εικόνα 64: Εισαγωγή βιβλιοθηκών.....	103
Εικόνα 65: Επιλογή και άνοιγμα Βιβλιοθηκών.....	103

Λίστα Πινάκων

Πίνακας 1:Πλεονεκτήματα-Μειονεκτήματα.....	11
Πίνακας 2:Αποσφαλμάτωση	25
Πίνακας 3: Τροφές και σίδηρος	35
Πίνακας 4:Πηγές βιταμίνης B12	37
Πίνακας 5: Πηγές φυλλικού οξέος.....	38
Πίνακας 6:Πίνακας Ορισμών	104
Πίνακας 7:Κώδικας Prolog	105

1. Εισαγωγή

Τα έμπειρα συστήματα ανήκουν στο κλάδο της Τεχνητής Νοημοσύνης(TN). Η TN επικεντρώνεται στην ικανότητα του υπολογιστή να χειρίζεται μη-αριθμητικά σύμβολα, να εξάγει συμπεράσματα και να εισάγει γεγονότα. Οι ικανότητες αυτές έχουν τουλάχιστον ένα κοινό χαρακτηριστικό, αποκτώνται εύκολα από ανθρώπους και βασίζονται συνήθως σε ένα σύνολο απόψεων/γνώσεων που κατέχει οποιοσδήποτε άνθρωπος, την λεγόμενη **κοινή λογική**.

Πριν ορίσουμε την Τεχνητή νοημοσύνη πρέπει πρώτα να ορίσουμε την Νοημοσύνη. Ο Howards Gardner στο βιβλίο του Frames of Mind : The theory of multiple intelligences (1983), διακρίνει σε κάθε άνθρωπο 8 τύπους νοημοσύνης (Γλωσσική, Λογική/Μαθηματική, Μουσική, Χωρική, Σωματική, Διαπροσωπική, Ενδοπροσωπική, Φυσιοκρατική)οι οποίοι είναι ευδιάκριτοι μέσα στον εγκέφαλο αλλά στην πράξη δημιουργείται ένα μίγμα από αυτούς. Στο ερμηνευτικό λεξικό του [Cambridge](#) λέει πως η νοημοσύνη είναι η ικανότητα για μάθηση, κατανόηση και κρίση ή αιτιολογημένη έκφραση γνώμης.

Ένας γενικός ορισμός της TN θα μπορούσε να είναι ο εξής:

“Η Τεχνητή Νοημοσύνη είναι ο τομέας της Επιστήμης των Υπολογιστών που ασχολείται με τη σχεδίαση και την υλοποίηση προγραμμάτων τα οποία είναι ικανά να μιμηθούν τις ανθρώπινες γνωστικές ικανότητες, εμφανίζοντας έτσι χαρακτηριστικά που αποδίδουμε συνήθως σε ανθρώπινη συμπεριφορά, όπως η επίλυση προβλημάτων, η αντίληψη μέσω της όρασης, η μάθηση, η εξαγωγή συμπερασμάτων, η κατανόηση φυσικής γλώσσας, κτλ.”[1].

Κάποιες από τις περιοχές που χρησιμοποιείτε η TN είναι:

- Απόδειξη Θεωρημάτων.
- Επεξεργασία Φυσικής Γλώσσας.
- Τεχνητή Όραση.
- Μηχανική Μάθηση.
- Αυτόνομα Robot.
- Έμπειρα συστήματα.

2. Συστήματα Γνώσης και Εργαλεία ανάπτυξης τους

2.1. Εισαγωγικές Έννοιες για τα Συστήματα Γνώσης.

Στην τεχνητή νοημοσύνη, ένα **έμπειρο σύστημα(ΕΣ)** είναι ένα υπολογιστικό σύστημα το οποίο μιμείται την ικανότητα ενός εμπειρογνώμονα στη λήψη αποφάσεων. Ο όρος γενικεύθηκε σε **Συστήματα Γνώσης**. Τα έμπειρα συστήματα σχεδιάστηκαν για να λύνουν πολύπλοκα προβλήματα συλλογισόμενα με βάση τη διαθέσιμη γνώση σε ένα πεδίο, (όπως κάνει ένας εμπειρογνώμονας και όχι εκτελώντας μία αλγοριθμική διαδικασία επίλυσης όπως στην περίπτωση του συμβατικού προγραμματισμού υπολογιστών). Τα πρώτα έμπειρα συστήματα δημιουργήθηκαν τη δεκαετία του 1970 και έπειτα αναπτύχθηκαν ραγδαία τη δεκαετία του 1980. Τα ΕΣ ήταν ανάμεσα στις πρώτες πραγματικά επιτυχημένες μορφές του λογισμικού της Τεχνητής Νοημοσύνης (ΤΝ).

Ένα σύστημα γνώσης έχει δομή διαφορετική από τα παραδοσιακά προγράμματα. Διαιρείται σε τρία μέρη, ένα σταθερό, το οποίο είναι η μηχανή εξαγωγής συμπερασμάτων, ένα άλλο μεταβλητό η βάση γνώσης. Για να τρέξει ένα σύστημα γνώσης, η μηχανή συλλογισμών παράγει νέα γνώση από τη βάση γνώσης, (όπως κάνει ο άνθρωπος που παράγει νέα γνώση από τη γνώση που διαθέτει). Το τρίτο μέρος είναι το περιβάλλον διεπαφής (interface), για να συνδιαλέγεται με τους χρήστες. Τα ΣΓ έχουν πολλά πλεονεκτήματα διότι λειτουργούν όπως ο άνθρωπος ειδικός/εμπειρογνώμονας. Ένα πλεονέκτημα είναι ο διάλογος μεταξύ χρήστη και συστήματος γνώσης γίνεται όπως ακριβώς θα γινόταν και μεταξύ δυο ανθρώπων. Για να γίνει αυτό πρέπει το σύστημα γνώσης να υποστηρίζει διεπαφή φιλική και κατανοητή ως προς το χρήστη.

Τα ΣΓ έχουν καλύτερη αξιοπιστία από ένα κλασσικό σύστημα, όπως ακριβώς και μία βάση δεδομένων. Με την απουσία ενός εμπειρογνώμονα πολύτιμες γνώσεις μπορούν να χαθούν. Εάν αυτές οι γνώσεις καταχωρηθούν σε ένα ΣΓ γίνονται «αιώνιες». Για να αναπτυχθεί ένα ΣΓ χρειάζεται η συνέντευξη ενός εμπειρογνώμονα και έπειτα να εισαχθεί αυτή η γνώση στο σύστημα, έτσι εμπλουτίζεται και ενισχύεται περαιτέρω.

Παρόλα τα πλεονεκτήματα και την χρησιμότητα των ΣΓ υπάρχουν σαφώς και μειονεκτήματα. Η σωστή μεθοδολογία ανάπτυξης μπορεί να μετριάσει τα προβλήματα. Ο παρακάτω πίνακας, (Πίνακας 1), δείχνει τα πλεονεκτήματα και μειονεκτήματα των ΣΓ.

Πλεονεκτήματα	Μειονεκτήματα
Άνθρωπος Ειδικός	Άνθρωπος Ειδικός
Δημιουργικότητα	Γνώση διαθέσιμη όταν ο ίδιος είναι παρών
Κοινή λογική	Δυσκολία μεταφοράς-αποτύπωσης γνώσης
Γνώση ορίων και δυνατοτήτων(μετα-γνώση)	Συναισθηματικές παρορμήσεις
Εκφραστική και λειτουργική επεξήγηση του τρόπου σκέψης τους	Απόδοση επηρεασμένη από εξωγενείς παράγοντες
Έλεγχος γνώσης που γίνεται υποσυνείδητα	Υψηλό κόστος
Αυτονομία στη μάθηση	Υποκειμενικότητα
Σύστημα Γνώσης	Σύστημα Γνώσης
Πρέπει η γνώση να ελέγχεται για ορθότητα, πληρότητα και συνέπεια	Απουσία έμπνευσης, περιορισμένο πεδίο σκέψης
Γνώση πάντα διαθέσιμη	Δυσχέρεια στη μεταφύτευση κοινής λογικής

Ευκολία μεταφοράς γνώσης	Μηχανική επεξήγηση του τρόπου λήψης αποφάσεων
Συνέπεια	Πρέπει να προγραμματιστούν για να μαθαίνουν αυτόματα
Εργάζεται οπουδήποτε	--
Χαμηλό κόστος λειτουργίας/ υψηλό κόστος ανάπτυξης	--
Αντικειμενικότητα αν η γνώση προέρχεται από πολλούς ειδικούς	--

Πίνακας 1: Πλεονεκτήματα-Μειονεκτήματα

2.2. Συστήματα Γνώσης στην Ιατρική

Όπως είναι γνωστό ο ρόλος της Ιατρικής είναι τετραπλός: *Έρευνα*, *Διάγνωση*, *Θεραπεία*, και *Εκπαίδευση* νέων ιατρών. Οι δυσκολίες κάθε ένα τομέα από αυτούς τους τέσσερεις είναι γνωστές. Ο Τομέας της *Έρευνας*, όσο και αν αυτή έχει κάνει ραγδαίες εξελίξεις, δυστυχώς έχει αφήσει το 80% των ασθενειών χωρίς σαφή αιτιολογία και χωρίς ριζική θεραπεία. Ακόμα και το απλό συνάχι, ουσιαστικά δεν θεραπεύεται από τη σύγχρονη ιατρική, απλώς αυτοϊάται μόνο του. Γνωστές ανίατες ασθένειες μαστίζουν τον πλανήτη. Σακχαρώδης Διαβήτης, Υπέρταση, Καρκίνος, Βρογχικό Ασθμα, Πολλαπλή Σκλήρυνση, Στεφανιαία Νόσος, Σχιζοφρένεια κλπ [5,8]. Ο τομέας της *Διάγνωσης* επίσης έχει κάνει θεαματικά άλματα χάρη στην τεχνολογία και τη συμβολή του ιατρικού λογισμικού. Ωστόσο το λογισμικό πολλών μηχανημάτων που χρησιμοποιούνται στην ιατρική (διαγνώσεις και θεραπείες) είναι κλειστό και ο κλάδος παρουσιάζει φαινόμενα μονοπωλίου και ολιγοπωλίου από μεγάλες εταιρείες τεχνολογίας (Siemens, General Electric, Philips, Abbott, κ.ά). Σε ότι αφορά την διαφορική διάγνωση, προβληματικό σημείο για την ανάπτυξη λογισμικού παραμένει η παρόμοια συμπτωματολογία για πολλές ασθένειες [7]. Ο τομέας της *Θεραπείας* πρακτικά έχει κάνει μεγάλα βήματα στην Ιατρική, χάρη στις προόδους της Βιοχημείας, της Χημείας και γενικότερα της Τεχνολογίας Φαρμάκων. Η παρουσία του Ιατρικού Λογισμικού είναι σαφέστατη και στον τομέα αυτό [5,8]. Τέλος, η *Εκπαίδευση* των νέων γιατρών είναι αυτή που έκανε πολύ σημαντικά βήματα και διευκόλυνε τους νέους φοιτητές και ειδικευόμενους ιατρούς στην απόκτηση νέων γνώσεων. Η συμβολή της Πληροφορικής εδώ ήταν καθοριστική.

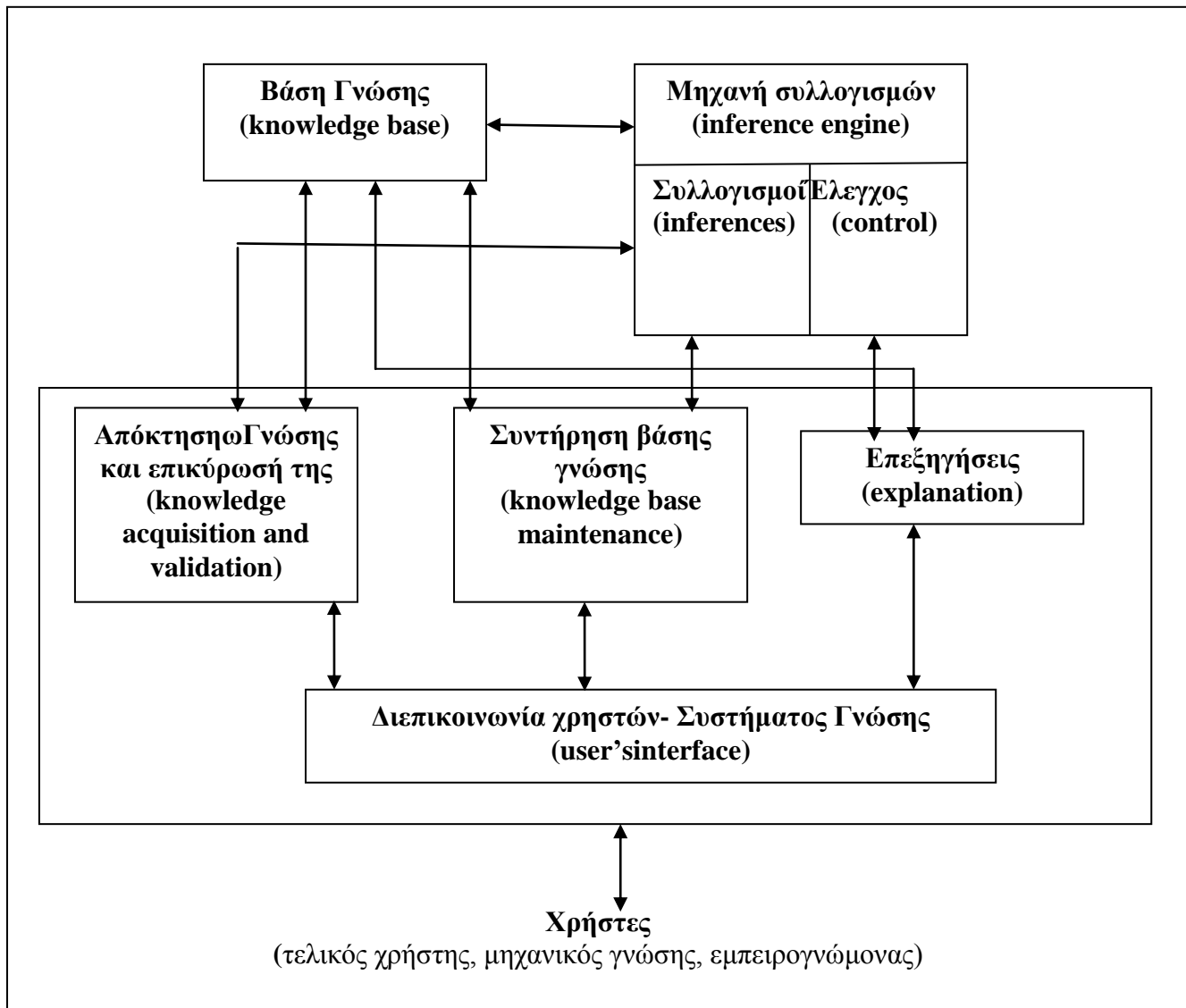
Το εφαρμοσμένο Ιατρικό Λογισμικό έχει επικεντρωθεί στους εξής τομείς [5,8].

- **Ενσωματωμένο Ιατρικό Λογισμικό:** Ο πρώτος τομέας είναι το *ενσωματωμένο ιατρικό λογισμικό, σε ιατρικές συσκευές* κάθε είδους. Βέβαια υπάρχουν και εταιρείες που κατασκευάζουν και εμπορεύονται ιατρικό Λογισμικό για απεικονιστικές και άλλες Ιατρικές συσκευές.
- **Προγράμματα Διαχείρισης Ιατρείου:** Ο δεύτερος τομέας είναι τα *απλά προγράμματα διαχείρισης ιατρείου και καταχώρησης ασθενών*. Δεν υπάρχει ενιαίος τρόπος σχεδίασης αυτών των προγραμμάτων, και αν κάποτε ο χρήστης αντιμετωπίσει πρόβλημα, ψάχνοντας την εταιρεία που το σχεδίασε, ανακαλύπτει ότι δεν υφίσταται πλέον. Κάποια παραδείγματα είναι: Το πρόγραμμα *ORTHO* αφορά τη διαχείριση

Ορθοπαιδικού Ιατρείου. Το πρόγραμμα *Cardio Data* αφορά τη διαχείριση καρδιολογικού Ιατρείου. Το πρόγραμμα *Medical Software* αφορά τη διαχείριση ιατρείων, κλινικών, κέντρων υγείας ενώ εύκολα μπορεί να προσαρμοστεί για κάθε επιμέρους ιατρική ειδικότητα. Βασίζεται πάνω σε open source κώδικα από το *OpenEMR*, κλπ

- **Λογισμικό Συγκεκριμένης Εφαρμογής:** Ο τρίτος τομέας είναι συνήθως On Line δικτυακά προγράμματα, *λογισμικό συγκεκριμένης εφαρμογής*:. Πχ. Υπάρχει δωρεάν On Line δικτυακό λογισμικό που ερευνά τις αλληλεπιδράσεις μεταξύ των φαρμάκων, και ενημερώνει τον φαρμακοποιό ή τον ιατρό ποια φάρμακα δεν ταιριάζουν στη λήψη. Σχεδιάστηκε από την εταιρεία Medscape και λέγεται *Drug Interaction Checker*, ([link](#)) Αξιόλογο είναι επίσης το λογισμικό του Εθνικού Συνταγολογίου της Ελλάδας, κάτι ανάλογο με τα αντίστοιχα άλλων χωρών, ([link](#)). Η Αμερικανική Εταιρεία ιατρικού Λογισμικού *Unbound Medicine* διαθέτει δωρεάν ([link](#)) μια σειρά από χρήσιμα ιατρικά προγράμματα προς χρήση ιατρών και νοσηλευτών όπως τα *Nursing Central*, το *Family Drug Guide*, το *Nurse's Clinical Pocket Guide*, κλπ. Αρκετά από αυτά λειτουργούν και στα νέα κινητά τηλέφωνα i-phone. Η Ρωσική εταιρεία Λογισμικού *SoftwareGeek* διαθέτει επίσης σειρά από Ιατρικό Λογισμικό για διάφορες χρήσεις.
- **Λογισμικά Διαφορικής Διάγνωσης (Computer Aided Diagnosis):** Ο Τέταρτος τομέας είναι τα δικτυακά (On Line) *λογισμικά διαφορικής διάγνωσης*. Τα υπάρχοντα συστήματα διάγνωσης με τη βοήθεια υπολογιστή *Computer Aided Diagnosis (CAD)* διαιρούνται σε δύο βασικές κατηγορίες, με βάση τις τεχνικές κατασκευής τους (development):
 - a) **Συστήματα Πιθανοτήτων (Propability Systems):** Αυτά είναι μοντέλα πιθανοθεωρητικού τύπου και δεν παρουσιάζουν ιδιαίτερο ενδιαφέρον.
 - b) **Συστήματα Βάσης Γνώσης (Knowledge Based Systems):** Αυτά είναι εφαρμογές της Τεχνητής Νοημοσύνης (Artificial Intelligence) αποκαλούνται και **Έμπειρα Συστήματα (Expert Systems)**. Ένα τέτοιο δωρεάν Λογισμικό στο διαδίκτυο, είναι το Αμερικανικό *DiagnosisPro* ([link](#)). Ένα επίσης ανάλογο πρόγραμμα σχεδιάστηκε από τη Microsoft για εξάσκηση Τηλεϊατρικής σε απομονωμένες και φτωχές περιοχές του πλανήτη. Λέγεται *NxOpinion*.
- **Ιατρικό Λογισμικό Τηλεϊατρικής:** Τηλεϊατρική (Telemedicine) είναι η χρήση του συνδυασμού Πληροφορικής και επικοινωνιακής τεχνολογίας. Ορίζει μια νέα, ασύλληπτη πριν μερικές δεκαετίες, δυνατότητα που καταργεί την απόσταση και μηδενίζει τον χρόνο στην ανταλλαγή πληροφοριών και στην χορήγηση βοήθειας ανάμεσα στους ιατρούς και τους αρρώστους. Η δυνατότητα αυτή οριοθετεί νέα επίπεδα σε όλους τους τομείς που συνθέτουν τη λειτουργία της Ιατρικής Επιστήμης: Εκπαίδευση, πρόσβαση στη γνώση, έρευνα, κλινική εφαρμογή, έλεγχο των ιατρικών πράξεων, βοήθεια σε δυσπρόσιτα σημεία, πρόληψη και γενικότερα ελαχιστοποιεί τους περιορισμούς χώρου και χρόνου κατά την άσκησή της. Στην εκπαίδευση και την ιατρική πράξη, οι νέες προοπτικές είναι τεράστιες διότι δίνονται οι εξής δυνατότητες:
 1. Ανταλλαγή εκπαιδευτικών ιατρικών προγραμμάτων σε πραγματικό χρόνο.
 2. Πρόσβαση των ιατρών σε όλες τις ιατρικές βιβλιοθήκες του κόσμου.
 3. Ιατρικές Τηλεδιασκέψεις, on line Ιατρικά συνέδρια, on line εξέταση ασθενούς, και μεταφορά Ιατρικού φακέλου με email.
 4. Παρακολούθηση ασθενούς εξ αποστάσεως (πχ. καρδιοπαθής με φορητή συσκευή, που αποστέλλει το ηλεκτροκαρδιογράφημα με το κινητό τηλέφωνο σε ειδικό κέντρο).
 5. Γενικός έλεγχος και παρακολούθηση πληθυσμών που κατοικούν σε απομακρυσμένα σημεία (νησιά, δυσπρόσιτες περιοχές). Η προσφορά ιατρικών

φροντιδών σε σημεία που δεν είναι δυνατή η άμεση πρόσβαση ιατρικού δυναμικού π.χ. ποντοπόρα πλοία, ορειβατικοί σταθμοί ή αποκλεισμένες περιοχές κ.λπ. Χαρακτηριστικό παράδειγμα η ύπαρξη κέντρου ελέγχου ορειβατών στο Έβερεστ το οποίο είναι συνδεδεμένο με το Πανεπιστήμιο του YALE και με αποστολή να παρακολουθεί ανά πάσα στιγμή τους ορειβάτες που πραγματοποιούν αναρριχήσεις πάνω από τα 7.000 μέτρα και που φέρουν αισθητήρες, οι οποίοι καταγράφουν συνεχώς την καρδιοαναπνευστική λειτουργία και προσδιορίζουν τα όρια αντοχής των ορειβατών.



Εικόνα 1:Σύστημα γνώσης

Πιο αναλυτικά τα πιο σημαντικά συστατικά είναι:

- **Η διεπαφή με το χρήστη.** Η διεπαφή με το χρήστη (interface) είναι ο τρόπος επικοινωνίας μεταξύ του χρήστη και των διαδικασιών επίλυσης προβλήματος των ΣΓ. Εάν δεν υπάρχει εύχρηστη διεπαφή το ΣΓ σύστημα δεν θεωρείται και πολύ χρήσιμο. Για αυτό λοιπόν πρέπει να δοθεί ιδιαίτερη προσοχή στο σχεδιασμό εύχρηστης διεπαφής προκειμένου το ΣΓ να γίνει φιλικό προς το χρήστη.
- **Η βάση γνώσης.** Η βάση γνώσης περιέχει όλα τα γεγονότα και τους κανόνες του προβλήματος. Τα γεγονότα μπορεί να είναι σε μορφή βασικών πληροφοριών ή να

εισάγονται από το χρήστη κατά τη διάρκεια της συλλογιστικής μέσω της διεπαφής. Οι κανόνες και τα γεγονότα αναπαρίστανται με τρόπο αντιληπτό από τους ανθρώπους.

- **Η μηχανή συλλογισμών.** Η μηχανή συλλογισμών αποτελείται από το τμήμα συλλογισμών και το τμήμα ελέγχου. Το **τμήμα συλλογισμών** αποτελείται από την υλοποίηση κάποιων μεθόδων εξαγωγής συμπερασμάτων, οι οποίοι εκτελούν τους συλλογισμούς χρησιμοποιώντας τη ΒΓ. Το **τμήμα ελέγχου** αποφασίζει τη σειρά με την οποία θα γίνουν οι συλλογισμοί.

Υπάρχουν πολλοί διαφορετικοί τύποι ΣΓ ακόμα και στον ίδιο τομέα εφαρμογής τους. Στον τομέα της ιατρικής θα μπορούσαμε να κάνουμε την παρακάτω ταξινόμηση των ΣΓ. Αυτή η ταξινόμηση στηρίζεται στη διαφορετική υπηρεσία που παρέχουν στο χρήστη. Κατά συνέπεια έχουν και διαφορετική λειτουργία.

- *Παραγωγή προειδοποιήσεων ή υπενθυμίσεων.* Υπάρχουν συστήματα τα οποία είναι σχεδιασμένα ώστε να προειδοποιεί τους χρήστες-ασθενείς τη για πιθανή μεταβολή της κατάστασης τους, είτε να υπενθυμίζει σε αυτούς τη λήψη φαρμακευτικής αγωγής ή τον καθορισμό ραντεβού.
- *Διαγνωστική Βοήθεια.* Σε δύσκολες περιπτώσεις ή σε περιπτώσεις που ο ειδικός είναι άπειρος ένα ΕΣ χρησιμοποιείται ως σύμβουλος του γιατρού δίνοντας πιθανές διαγνώσεις βασισμένες στα στοιχεία του ασθενούς.
- *Σχεδιασμός θεραπείας.* Τα ΕΣ χρησιμοποιούνται στο σχεδιασμό μια θεραπείας βασισμένης στη συγκεκριμένη κατάσταση του ασθενούς και σε υιοθετημένες οδηγίες θεραπείας.
- *Εκπαίδευση.* Τα ΕΣ χρησιμοποιούνται στην εκπαίδευση και εξάσκηση νοσοκομειακών ιατρών και σπουδαστών σε διάφορες ιατρικές εργασίες

2.3. Συστήματα Γνώσης στη διάγνωση της Αναιμίας

Το συγκεκριμένο σύστημα γνώσης ως κύριο στόχο έχει τη διάγνωση των διάφορων τύπων της Αναιμίας. Αυτή η ασθένεια σε πολλούς μπορεί να μοιάζει κάτι απλό, παρόλα αυτά πρέπει να σημειωθεί πως όπως κάθε ασθένεια έτσι και η Αναιμία έχει την επικινδυνότητα της. Οι αναιμίες διακρίνονται σε ποικίλες κατηγορίες οι οποίες βασίζονται σε διαφορετικές ταξινομήσεις. Η αντιμετώπιση περιλαμβάνει θεραπευτική παρέμβαση για το νόσημα που προκαλεί την αναιμία. Δεν υπάρχει γενικά παραδεκτό σχήμα ταξινόμησης της αναιμίας. Περαιτέρω ανάλυση της ασθένειας θα γίνει στο Κεφάλαιο «Εισαγωγικές Έννοιες της Αναιμίας.»

Ένα παράδειγμα τέτοιου συστήματος είναι το σύστημα το οποίο έγινε από το Πανεπιστήμιο της Ιταλίας. Το ANEMIA όπως το αποκαλούν χρησιμοποιεί το πλαίσιο του Rutgersπανεπιστημίου για την αναπαράσταση της γνώσης. Αυτό το πλαίσιο έχει χρησιμοποιηθεί στο χτίσιμο υπεράριθμων ιατρικών διαβουλευτικών μοντέλων, κυρίως στην Ρευματολογία, Οφθαλμολογία, Δερματολογία και Ενδοκρινολογία. Ένα έμπειρο διαβουλευτικό σύστημα αποτελείται κυρίως από ευρήματα, υποθέσεις και κανόνες απόφασης. Τα ευρήματα χρησιμοποιούνται για να συμπεράνουν υποθέσεις μέσω των κανόνων, με τη μορφή παραγωγικών κανόνων if-then. Ένας παραγωγικός κανόνας είναι ένας κανόνας υπό όρους που υποδηλώνει κάτω από ποιες προϋποθέσεις κατέληξε στο συμπέρασμα. Η δύναμη του συμπεράσματος υποδηλώνεται από αριθμητικά βάρη, με κλίμακα από -1 έως +1. Το -1 δηλώνει απόλυτη διάψευση μιας υπόθεσης, το +1 είναι η απόδειξη της υπόθεσης και το 0 δηλώνει την απουσία υπέρ ή ενάντια της υπόθεσης.

Το σύστημα είναι γραμμένο σε FORTRAN, τρέχει με VMS σε VAX 11/780 υπολογιστή του Πανεπιστημιακού Κέντρου Υπολογιστών της Ραβία. Μέχρι στιγμής υπάρχουν 64 υποθέσεις ή διαγνωστικά συμπεράσματα στη γνωστική βάση του προγράμματος, 61 μέσες υποθέσεις και 125 ευρήματα. Σε αντίθεση υπάρχουν 45 συστάσεις διαχείρισης. Αυτά τα στοιχεία είναι συνδεδεμένα λογικά με πάνω από 1200 κανόνες παραγωγής. Τα πιθανά διαγνωστικά αποτελέσματα είναι δομημένα σε ένα ταξινομημένο σχήμα. Περιλαμβάνει 64 τελικά διαγνωστικά συμπεράσματα και 16 διαγνωστικές κατηγορίες [12].

Ένα άλλο σύστημα είναι αυτό το οποίο έχει αναπτύξει το Τεχνικό Πανεπιστήμιο του Μονάχου. Το συγκεκριμένο σύστημα είναι ένα πρωτότυπο που περιλαμβάνει περίπου 100 ευρήματα και 15 διαγνώσεις. Η διαγνωστική διαδικασία μπορεί να χαρακτηριστεί από την παρακάτω συμπεριφορά, εισαγωγή των ευρημάτων που επηρεάζουν τα αποτελέσματα με τη σχετική διάγνωση, πρόωρη διάγνωση σε πρώιμο στάδιο, τα αποτελέσματα για τη διάγνωση σχετικά με τη συχνότητα και το επειγόν της κατάστασης, ο αποκλεισμός της διάγνωσης, σημασιολογικές σχέσεις μεταξύ διάγνωσης και τα ευρήματα, μη μονότονη συλλογιστική. Χρησιμοποιεί προς τα εμπρός και προς τα πίσω αλυσίδωση, γράφτηκε σε Pascal και τρέχει σε ATARIST 1040 και σε SIEMENSMX 2 [13].

2.4. Ένα Σύστημα Γνώσης για τη Διάγνωση Αναιμίας

Στην παρούσα πτυχιακή εργασία έχουμε αναπτύξει ένα Διαγνωστικό Σύστημα Αναιμίας (ΔιΣΑ). Το σύστημα εκτός από τη διάγνωση προτείνει και φαρμακευτική/θεραπευτική αγωγή για τον εξεταζόμενο. Η βάση γνώσης και η μηχανή συλλογισμών του ΔιΣΑ έχουν υλοποιηθεί σε Prolog και η διεπαφή του σε Java. Το σύστημα μπορεί να το χρησιμοποιήσει ένασειδικευμένος γιατρός όπως ο μικροβιολόγος/αιματολόγος ή παθολόγος ως σύμβουλο τόσο για την διάγνωση όσο και για τη φαρμακευτική αγωγή. Το σύστημα μπορεί και διαχειρίζεται πολλαπλές επισκέψεις από τον ίδιο χρήστη/εξεταζόμενο.

Το σύστημα πέραν της διάγνωσης παρέχει και δυνατότητες αναζήτησης παλαιού ιστορικού του χρήστη. Καθώς επίσης και δυνατότητα ενημέρωσης της βάσης κανόνων (Εικόνα 2) που χρησιμοποιείται για τη διάγνωση και θεραπεία. Η διάγνωση γίνεται βασισμένη στη γνώση που παράχθηκε έπειτα από συνεντεύξεις με ειδικούς καθώς επίσης και έπειτα από ανάγνωση επιστημονικών άρθρων και βιβλίων. Η θεραπεία που δίδεται είναι αντίστοιχη της διάγνωσης που έχει προηγηθεί. Η δυνατότητα ενημέρωσης βάσης κανόνων δίνεται όχι στους απλούς χρήστες αλλά σε κάποιο εμπειρογνώμονα ή στο μηχανικό γνώσης.

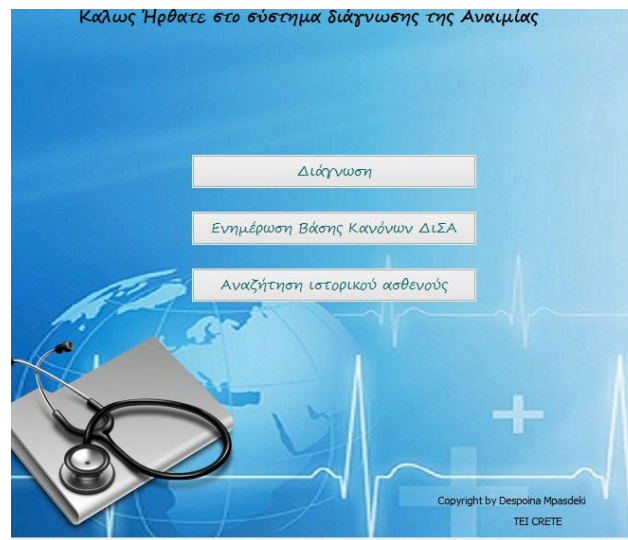
Η διεπικοινωνία της ενημέρωσης της βάσης κανόνων αποτελείται από τρία μέρη. Το πρώτο αφορά την ενημέρωση ενός κανόνα. Ο εμπειρογνώμονας εισάγει τον κωδικό τον οποίο θέλει να αλλάξει/ανανεώσει και έπειτα το σώμα του κανόνα και πατώντας ανανέωση ο κανόνας αλλάζει. Με τον ίδιο τρόπο μπορεί να διαγράψει έναν κανόνα, δίνοντας απλά τον κωδικό του κανόνα. Εάν θέλει να δημιουργήσει ένα νέο κανόνα δίνει μόνο το σώμα του, τον κωδικό τον εισάγει αυτόματα το σύστημα. Η κεφαλή όλων των κανόνων είναι ίδια. Έχει οριστεί από τον μηχανικό γνώσης, δεν αλλάζει και η μορφή της είναι πολύ γενική. Η κεφαλή όλων των κανόνων έχει την εξής μορφή ;

rule(Rule_id, Patient_data, Diagnosis)

όπου Rule_id είναι η ταυτότητα του κανόνα, Patient_data είναι τα δεδομένα του ασθενή (προσωπικά και ιατρικά δεδομένα) και Diagnosis είναι η διάγνωση του συστήματος.

Το σύστημα ΔιΣΑ σε σύγκριση με τα συστήματα που αναφέρθηκαν στην ενότητα 2.3 έχει αρκετές ομοιότητες, όπως το σύστημα Anemia έτσι και αυτό χρησιμοποιεί if-then κανόνες. Το σύστημα χρησιμοποιεί προς τα εμπρός αλυσίδωση, χρησιμοποιώντας το δέντρο αποφάσεων που παρουσιάζεται στην Ενότητα 6.1, σε αντίθεση με το σύστημα Τεχνικό

Πανεπιστημίου του Μονάχου που χρησιμοποιεί προς τα πίσω αλλά και προς τα εμπρός αλυσίδωση,



Εικόνα 2: Πάνελ επιλογών ΔιΣΑ

3. Η Prolog ως Εργαλείο Ανάπτυξης Συστημάτων Γνώσης

Η **Prolog** είναι μια δηλωτική γλώσσα προγραμματισμού με απλή σύνταξη, ισχυρή για ανάπτυξη εφαρμογών ιδίως της Τεχνητής Νοημοσύνης. Είναι μια γλώσσα λογικού προγραμματισμού η οποία χρησιμοποιείται στον τομέα της τεχνητής νοημοσύνης. Το όνομα Prolog το έβγαλε ο συνεργάτης του Kowalski ο Philippe Roussel και είναι συντομογραφία του γαλλικού «PROgramation et LOGique» (Προγραμματισμός και Λογική). Για να βρεθεί λύση του προβλήματος η Prolog κάνει πλήρη αναζήτηση. Αυτό επιτυγχάνεται με τη βοήθεια μιας ενσωματωμένης μηχανής αναζήτησης τύπου βάθος-πρώτα (depth-first) η οποία ενεργοποιείται αυτόματα κάθε φορά που κάνουμε μια ερώτηση στο πρόγραμμα μας.

Για τις εκδόσεις Prolog υπάρχουν δυο πρότυπα (Standard). Υπάρχει το παλιότερο Edinburgh-standard και το πιο σύγχρονο ISOstandard. Τα δυο standards διαφέρουν κυρίως στα ονόματα των built-in συναρτήσεων που αναγνωρίζει η γλώσσα. Οι νέοι compilers ακολουθούν το ISO-standard, όμως για λόγους συμβατότητας πολλοί αναγνωρίζουν και το Edinburgh [9].

Για να μάθει κάποιος να χρησιμοποιεί σωστά αυτή τη γλώσσα χρειάζεται αρκετός χρόνος και αυτό διότι οι περισσότεροι προγραμματιστές έχουν συνηθίσει άλλες "διαδικαστικές" γλώσσες. Είναι χαρακτηριστικό της Prolog ότι ο πηγαίος κώδικας που γράφουμε μπορεί να είναι πολύ πιο και περιεκτικός από οποιαδήποτε άλλη γλώσσα. Όσο περισσότερο εμβαθύνουμε στη καλή γνώση της γλώσσας αυτής τόσο λιγότερο χρόνο και χώρο θα χρειάζεται στην ανάπτυξη συστημάτων [9].

Η δομή της Prolog ευνοεί την top-down ανάπτυξη κώδικα, μπορούμε να εκφράσουμε στο top-level κατηγορήματα όλη τη λογική του προγράμματος με τέτοιο τρόπο ώστε να χειρίζεται όλα τα υπόλοιπα κατηγορήματα ως δεδομένα (data) παρά ως κώδικα.

3.1. Προτάσεις και Γεγονότα σε Prolog

Η Prolog βασίζεται σε ένα σύνολο μηχανισμών, οι οποίοι την κάνουν αρκετά δυνατή και ευέλικτη γλώσσα προγραμματισμού. Οι μηχανισμοί στους οποίους βασίζεται είναι οι εξής:

1. Ενοποίηση
2. Επαγωγικές δομές
3. Οπισθοδρόμηση

Για να γραφτεί ένα πρόγραμμα το οποίο επιτυχώς επιδεικνύει τεχνητή νοημοσύνη, θα πρέπει να γίνει *αναπαράσταση* της σχετικής γνώσης ως προτάσεις εκφρασμένες στη λογική πρώτης τάξης και να προσαρτηθεί σε αυτή τη *βάση γνώσης* μία αποτελεσματική *διαδικασία απόδειξης*. Ένα τέτοιο υπολογιστικό σύστημα μπορεί να λύνει προβλήματα εξάγοντας με τυπικό τρόπο θεωρήματα από τις προτάσεις της βάσης γνώσης. Η Prolog είναι ένα τέτοιο ευφύες σύστημα, δηλαδή ένα σύστημα απόδειξης θεωρημάτων. Η *θεωρία* είναι το πρόγραμμα και το *θεώρημα* είναι η ερώτηση ή ο στόχος τον οποίο έχει να αποδείξει.

Οι κυριότερες κατηγορίες εργαλείων για ανάπτυξη ΣΓ είναι τα κελύφη ΣΓ και οι γλώσσες προγραμματισμού υψηλού επιπέδου και ιδιαίτερα οι γλώσσες προγραμματισμού. Τα κελύφη συστημάτων γνώσης είναι έξυπνα εργαλεία για ανάπτυξη ΣΓ. Αποτελούνται από το μηχανισμό συλλογισμών και από το περιβάλλον υποστήριξης. Εάν από το σύστημα γνώσης αφαιρέσουμε τη βάση γνώσης, αυτό που μένει είναι η μηχανή συλλογισμών και το περιβάλλον υποστήριξης.

Οι γλώσσες προγραμματισμού TN είναι καλά εργαλεία για γρήγορη κατασκευή πρωτοτύπων, έτσι περισσότερα σχέδια του συστήματος μπορούν να ερευνηθούν και να

αξιολογηθούν με μικρό σχετικά κόστος χρόνου και προσπάθειας. Η υλοποίηση ΣΓ σε γλώσσες TN δίνει ευελιξία στον προγραμματιστή, δηλαδή ο προγραμματιστής μπορεί να δημιουργήσει δικούς του τρόπους αναπαράστασης γνώσης, να χειρίζεται αβεβαιότητα, να κατασκευάσει μηχανή συλλογισμών και ελέγχου των συλλογισμών καθώς και άλλων εργαλείων υποστήριξης του ΣΓ. Το μειονέκτημα είναι ότι κάποια εργαλεία τα οποία δεν είναι σημαντικά για την απόδοση του συστήματος γνώσης πρέπει να κατασκευαστούν από τον προγραμματιστή. Αντιθέτως, αυτά είναι διαθέσιμα στα κελύφη ΣΓ, επιπλέον οι γλώσσες TN δεν διαθέτουν φιλική επικοινωνία με το χρήστη όπως τα κελύφη ΣΓ. Τέτοιες γλώσσες είναι οι γλώσσες προγραμματισμού που έχουν σχεδιαστεί για ανάπτυξη εφαρμογών TN οι οποίες στηρίζονται στη λογική. Οι κυριότερες γλώσσες προγραμματισμού που σχεδιάστηκαν για ανάπτυξη τέτοιων εφαρμογών είναι η Prolog για το σχεσιακό λογικό προγραμματισμό και η Lisp για την οικογένεια του συναρτησιακού λογικού προγραμματισμού. Η υλοποίηση ΣΓ σε Prolog έχει τα εξής πλεονεκτήματα:

1. Η αναπαράσταση γνώσης μπορεί να γίνει σε Prolog με απλό και αποτελεσματικό τρόπο οποιαδήποτε και να είναι η μέθοδος αναπαράστασης γνώσης, πχ λογική, κανόνες παραγωγής, σημασιολογικά δίκτυα, κ.α. Ιδιαίτερα, όταν η μέθοδος αναπαράστασης γνώσης είναι «*if-then κανόνες*» τότε η αναπαράστασή τους σε γίνεται με ευθύ και άμεσο τρόπο. Αυτό γίνεται διότι οι προτάσεις γεγονότα και οι προτάσεις κανόνες της Prolog αντιστοιχούν ευθέως στα γεγονότα και στους *if-then* κανόνες ενός ΣΓ.
2. Ο προγραμματιστής μπορεί να χρησιμοποιήσει το μηχανισμό συλλογισμών για το ΣΓ που κατασκευάζει. Ο μηχανισμός αυτός ακολουθεί «*ανάστροφη/ προς τα πίσω αλυσίδωση*» κανόνων.

Η αναπαράσταση σε Prolog των κανόνων και των γεγονότων ενός συστήματος γνώσης είναι απλή, υπάρχει άμεση αντιστοίχιση μεταξύ κανόνων και γεγονότων. Η πιο δημοφιλής αναπαράσταση γνώσης είναι οι *κανόνες παραγωγής* ή *if-then κανόνες* ή *κανόνες*. Η αναπαράσταση γνώσης με **if-then κανόνες** είναι η πλέον κατάλληλη όταν η γνώση του χώρου έχει προέλθει από πολύχρονη εμπειρία και περιέχει εμπειρικές αλληλοσυνδέσεις. Όπου αναφέρεται η λέξη «*κανόνες*» θα σημαίνει «*if-then κανόνες*». Εάν πρόκειται για άλλους κανόνες, π.χ. Prolog κανόνες θα αναφέρονται με σαφήνεια. Ένα ευφρές σύστημα στο οποίο η αναπαράσταση γνώσης βασίζεται σε *if-then κανόνες* αποτελείται από τα εξής μέρη:

1. Ένα σύνολο **if-then κανόνων**.
2. Ένα σύνολο **γεγονότων**, τα οποία παριστούν γνώση η οποία είναι αληθής. Δηλαδή τα δεδομένα του προβλήματος.
3. Ένα *μεταφραστική κανόνων* ο οποίος ελέγχει την εφαρμογή των κανόνων.

Ένας **if-then** κανόνας μπορεί να παρασταθεί ευθέως και με απλό τρόπο σε Prolog. Για παράδειγμα ο παρακάτω ψευδοκώδικας

If Προϋπόθεση then Συμπέρασμα
ή
Προϋπόθεση → Συμπέρασμα

Αυτό σε Prolog αναπαρίσταται ως εξής:

Συμπέρασμα :- Προϋπόθεση
Δηλαδή
Συμπέρασμα **if** Προϋπόθεση

Η *προϋπόθεση* ενός κανόνα είναι μία έκφραση (παράσταση) η τιμή της οποίας μπορεί να είναι είτε αληθής είτε ψευδής. Για να είναι εφαρμόσιμος ένας κανόνας θα πρέπει η προϋπόθεση του να είναι αληθής, να ικανοποιείται από τη βάση γνώσης. Η *προϋπόθεση* του κανόνα αποτελείται από όρους, απλές λογικές εκφράσεις. Οι όροι της προϋπόθεσης μπορούν να συνδέονται με τελεστές and (,) or (;) not (\+). Τα γεγονότα εκφράζουν σχέσεις μεταξύ αντικειμένων του πεδίου του προβλήματος και αναπαρίστανται σε γεγονότα Prolog.

Το *συμπέρασμα* ενός κανόνα είναι η ενέργεια του κανόνα. Η ενέργεια ενός κανόνα μπορεί να έχει ως αποτέλεσμα είτε την προσθήκη είτε πολύ σπάνια την αφαίρεση ενός γεγονότος από τη βάση γεγονότων. Κάθε κανόνας θεωρείται ως ένα ανεξάρτητο κομμάτι γνώσης.

Έστω ότι έχουμε τον παρακάτω κανόνα, ο από το σύστημα ΔιΣΑ.

```
if(member('ΠΟΝΟΚΕΦΑΛΟΣ',Symptoms) or
    member('ΚΟΥΡΑΣΗ',Symptoms) or
    member('ΠΡΟΒΛΗΜΑΤΑ ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ',Symptoms) or Προϋπόθεση
    member('ΑΔΥΝΑΜΙΑ',Symptoms) or
    member('ΤΑΧΥΚΑΡΔΙΑ',Symptoms) or
    member('ΔΥΣΠΝΟΙΑ',Symptoms))
then
check_WBC_PLT(FullBloodTest,noPancytopenia) Συμπέρασμα/Ενέργεια
```

Η μετάφραση του κανόνα αυτού του κανόνα είναι η εξής: «**Εάν** κάποιο από τα συμπτώματα που επέλεξε ο ασθενής, δηλαδή ΠΟΝΟΚΕΦΑΛΟΣ, ΚΟΥΡΑΣΗ, ΠΡΟΒΛΗΜΑΤΑ ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ, ΑΔΥΝΑΜΙΑ, ΤΑΧΥΚΑΡΔΙΑ ή ΔΥΣΠΝΟΙΑ, *βρίσκεται στη λίστα των δεδομένων τότε* οι τιμές WBC_PLT¹ δηλώνουν όχι πανκυταροπενία ». Παρατηρούμε πως κάποια ονόματα με αγγλικούς χαρακτήρες είναι με κεφαλαία γράμματα και άλλα με πεζά. Όσα αρχίζουν με κεφαλαία γράμματα ονομάζονται *μεταβλητές* και όσα αρχίζουν με πεζά γράμματα ονομάζονται *σταθερές*. Σε αυτό τον κανόνα, «Symptoms» και «FullBloodTest» είναι μεταβλητές ενώ το «noPancytopenia» είναι σταθερά. Τα ελληνικά ονόματα είτε αρχίζουν με κεφαλαία ή πεζά πρέπει να βρίσκονται σε μονά εισαγωγικά και θεωρούνται σταθερές. Μεταβλητές σε Prolog με ελληνικά ονόματα δεν μπορούμε να έχουμε. Τα κατηγορήματα παριστάνουν σχέσεις μεταξύ αντικειμένων του πεδίου του προβλήματος, δηλαδή των δεδομένων. Για παράδειγμα, *check_WBC_PLT(FullBloodTest,noPancytopenia)*, το αντικείμενο είναι το «FullBloodTest» και συνδέεται με το αντικείμενο «noPancytopenia» με τη σχέση «check_WBC_PLT». Να επισημανθεί πως το FullBloodTest στο συγκεκριμένο παράδειγμα είναι λίστα.

Ο *μεταφραστής* κανόνων βρίσκει τους κανόνες των οποίων η προϋπόθεση ικανοποιείται. Δηλαδή βρίσκει αυτούς που είναι εφαρμόσιμοι. Εάν υπάρχουν περισσότεροι από έναν εφαρμόσιμοι κανόνες ο μεταφραστής καθορίζει τη σειρά εκτέλεσής τους. Το μεταφραστή κανόνων μπορούμε να τον δούμε σαν μία εξειδικευμένη μηχανή συλλογισμών για *if-then* κανόνες. Ο μεταφραστής κανόνων για να βρει εάν η προϋπόθεση ενός κανόνα ικανοποιείται, ταιριάζει την προϋπόθεση του κανόνα με τα γεγονότα που υπάρχουν στη βάση των γεγονότων. Εάν μια προϋπόθεση είναι αληθής, δηλαδή ταιριάζει με τα γεγονότα, τότε ο κανόνας είναι εφαρμόσιμος. Το *συμπέρασμα* του κανόνα μπορεί να έχει ως αποτέλεσμα την ενημέρωση της βάσης γεγονότων, είτε με πρόσθεση ενός νέου γεγονότος είτε με αφαίρεση ενός γεγονότος [2].

Ο παρακάτω στόχος είναι τμήμα από ένα κανόνα του ΔιΣΑ. Αμέσως μετά ακολουθεί η υλοποίηση του κατηγορήματος *check_WBC_PLT/2*. Το κατηγορήμα *check_WBC_PLT/2*

¹ Όλες οι συντομογραφίες βρίσκονται στο «Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.»

είναι αληθές εάν οι τιμές του WBC και του PLT βρίσκονται σε κάποια καθορισμένα διαστήματα, τότε ο ασθενής δεν έχει πανκυτταροπενία.

« ?- *check_WBC_PLT*([4700,1,75,4.3,45,18,225,32,12,125], *noPancytopenia*). »

check_WBC_PLT([WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT], *noPancytopenia*) :-
number(WBC), *number*(PLT),(WBC > 4500, PLT>150).

check_WBC_PLT([WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT], *noPancytopenia*) :-
number(WBC), *number*(PLT),(WBC > 4500, PLT<150).

check_WBC_PLT([WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT], *noPancytopenia*) :-
number(WBC), *number*(PLT),(WBC < 4500, PLT>150).

Περισσότερες πληροφορίες μπορείτε να βρείτε στη βιβλιογραφία ή στο διαδίκτυο στο οποίο υπάρχουν πολλά άρθρα και πληροφορίες σχετικά με την Prolog. Υπάρχουν αρκετές υλοποιήσεις της Prolog, κάποιες από αυτές είναι:

- XSB
- SWI-Prolog
- SISctus Prolog
- ECLiPSe Prolog
- Quintus Prolog
- YAP Prolog
- etc...

3.2. SWI-Prolog

Μία μοντέρνα έκδοση της Prolog είναι η SWI-Prolog, η οποία περιλαμβάνει πολλά νέα χαρακτηριστικά όπως τμηματικό προγραμματισμό (modularprogramming), πολυνηματικό (multi-threaded) εκτελέσιμο κώδικα, προγραμματισμό με περιορισμούς (constraintprogramming), γραφικό περιβάλλον επικοινωνίας με το χρήστη, καθώς και πληθώρα βιβλιοθηκών με έτοιμες συναρτήσεις. Η SWI-Prolog είναι μια πολύ αξιόλογη έκδοση της Prolog που έχει αναπτυχθεί στο Πανεπιστήμιο του Amsterdam και διατίθεται δωρεάν στην ιστοσελίδα <http://www.swi-prolog.org/>. Χρησιμοποιείται γιατί είναι μοντέρνα και διαθέτει πολλά νέα χαρακτηριστικά για ανάπτυξη εφαρμογών Τεχνητής Νοημοσύνης στο διαδίκτυο και στο σημασιολογικό ιστό. Διαθέτει όπως όλες σχεδόν οι σύγχρονες υλοποιήσεις της Prolog βιβλιοθήκη με πολλές έτοιμες συναρτήσεις. Η SWI-Prolog διαθέτει ένα γρήγορο και δυνατό, ISO/Edinburgh-style, μεταγλωττιστής (compiler). Προσφέρει ένα γρήγορο και ισχυρό περιβάλλον που παρέχει τα αναγκαία εργαλεία για υλοποίηση εφαρμογών Τεχνητής Νοημοσύνης και ιδιαίτερα διαδικτυακών εφαρμογών. Παρέχει δηλαδή βιβλιοθήκες με κατηγορήματα, τα οποία ο προγραμματιστής μπορεί να ενσωματώσει στο πρόγραμμά του. Αυτά υποστηρίζουν, μεταξύ άλλων, προγραμματισμό με περιορισμούς, ανάπτυξη εφαρμογών

επεξεργασίας φυσικής γλώσσας, εφαρμογών διαδικτύου και σημασιολογικού ιστού. Διαθέτει τα εξής χαρακτηριστικά:

Μηχανή της Prolog

1. Μικρή σε μέγεθος. Το πλήρες περιβάλλον ανάπτυξης (γραφικά, βιβλιοθήκες και πακέτα διεπικοινωνίας) χρειάζεται περίπου 40MB του σκληρού δίσκου.
2. Γρήγορη επεξεργασία.
3. Συλλογή απορριμμάτων (garbage collection).
4. Πολυνηματική υποστήριξη (multithreading).
5. Διαχείριση Unicode χαρακτήρων, ιδανική για διαδικτυακές και διεθνείς εφαρμογές.

Εργαλεία ανάπτυξης

1. Αποσφαλμάτωση πηγαίου κώδικα (Source-level debugger).
2. Δυνατότητα εκτέλεσης σε όλες τις κύριες πλατφόρμες (Windows, Linux, MacOSX).

Συμβατότητα

1. Συμβατή με την standardProlog και με όλες τις εκδόσεις της Prolog.(π.χ. Quintus SICStus Prolog, Ciao, YAP και GNU-Prolog κτλ).

Βιβλιοθήκες

1. Παροχή βιβλιοθηκών για εφαρμογές διαδικτύου, για προγραμματισμό με περιορισμούς (ConstraintHandlingRules) και διάφορες άλλες.

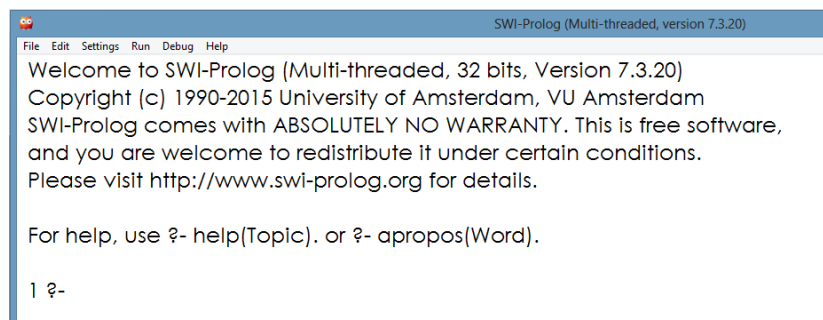
Συνδεσιμότητα

1. Παρέχει client και server βιβλιοθήκες για HTTP.
2. Ισχυρή διεπαφή C/C++/Java/GNU Readline.
3. Συνδεσιμότητα με τη βάση δεδομένων γίνεται μέσω της διεπικοινωνίας Open Data Base Connectivity (ODBC).

Φορητότητα

1. Φορητή. Υποστηρίζει διάφορα λογισμικά, σχεδόν όλες τις Unix/Linux πλατφόρμες, Windows (XP, Vista, both 32 and 64-bit versions), Mac OS X και άλλα. Υποστηρίζει 32-bit και 64-bit λογισμικό.

Η εγκατάσταση της SWI - Prolog αναφέρεται με το όνομα «swipl». Είναι ερμηνευτής, όπως όλες οι Prolog, και γι' αυτό μπορούμε να εκτελούμε διαδραστικά στόχους:



Εικόνα 3:SWI μήνυμα καλωσορίσματος

Κατά την έναρξη του προγράμματος, μπορούμε να φορτώσουμε το πρόγραμμα σας στον ερμηνευτή της Prolog βάζοντας το όνομα του αρχείου του προγράμματος μέσα σε μονά

εισαγωγικά (') και σε αγκύλες. Για παράδειγμα, ['check_WBC_PLT.pl'] φορτώνει το αρχείο check_WBC_PLT.pl που περιέχει προτάσεις που αφορούν τη βάση γνώσης του συστήματος ΔιΣΑ:

```
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 7.2.3)
Copyright (c) 1990-2015 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?- ['check_WBC_PLT'].
   true.
2 ?-|
```

Εικόνα 4:φόρτωση αρχείου anemia_diagnosis.pl

Το φόρτωμα ενός προγράμματος μπορεί να γίνει και με άλλους τρόπους είτε πληκτρολογώντας στον Interpreter της Prolog την εντολή «**consult('<path>/check_WBC_PLT.pl)**»,όπου path είναι το μονοπάτι του φακέλου που αποθηκεύσατε το πρόγραμμα.

Μετά την εγκατάσταση της SWI - Prolog σε σύστημα Windows, τα ακόλουθα χαρακτηριστικά της γλώσσας είναι διαθέσιμα στο χρήστη :

- Ένας φάκελος που ονομάζεται **swipl** και περιέχει τα εκτελέσιμα, βιβλιοθήκες, κλπ. του συστήματος. Τα αρχεία δεν εγκαθίστανται έξω από αυτόν τον κατάλογο.
- Ένα πρόγραμμα **swipl - win.exe**, παρέχοντας ένα παράθυρο για την αλληλεπίδραση με Prolog. Το **swipl.exe** πρόγραμμα είναι μια έκδοση της SWI - Prolog που τρέχει σε ένα παράθυρο κονσόλας.
- **H.pl** επέκταση αρχείου συσχετίζεται με το πρόγραμμα **swipl - win.exe**. Ανοίγοντας ένα αρχείο.pl η swipl - win.exe θα ξεκινήσει, θα αλλάξει τον παρόν κατάλογο στον κατάλογο που βρίσκεται το αρχείο και θα φορτώσει το αρχείο αυτό.

Μετά τη φόρτωση του αρχείουμε το κατηγορημα check_WBC_PLT/2, μπορούν να εκτελεστούν ερωτήσεις στο πρόγραμμα. Ένας στόχος είναι αληθής όταν οι τιμές των δεδομένων WBC και PLT βρίσκονται σε κάποια καθορισμένα διαστήματα. (εικόνα 5 δειχνει την εκτέλεση 2 στόχων). Ο πρώτος στόχος (όπως φαίνεται και στο κεφάλαιο 3.1) είναι αληθής και το σύστημα ανταποκρίνεται με true.Ενώ ο δεύτερος στόχος είναι ψευδής διότι οι τιμές οι τιμές των δεδομένων WBC και PLT δεν βρίσκονται στα καθορισμένα διαστήματα. Ο χρήστης μπορεί να πληκτρολογήσει ; ή το κενό, εάν θέλει να βρεθεί και άλλη απάντηση (εφόσον υπάρχει).

```

File Edit Settings Run Debug Help
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 7.2.3)
Copyright (c) 1990-2015 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic) . or ?- apropos(Word) .

1 ?- check_WBC_PLT([4700,1,75,4.3,45,18,225,32,12,125], noPancytopenia) .
true .
2 ?-

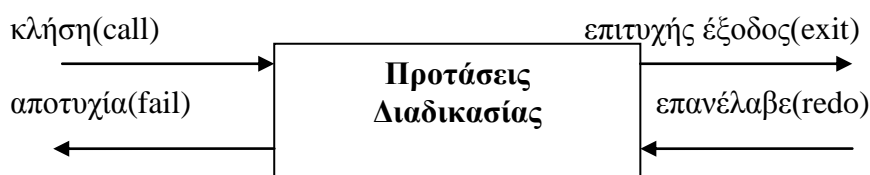
```

Εικόνα 5: Παράδειγμα ερώτησης σε Prolog

Μπορούμε να τρέξουμε το πρόγραμμα και να δούμε βήμα - βήμα την εκτέλεση του προγράμματος, ποιο κατηγορήμα καλείται, αν πέτυχε η ενοποίηση, τι τιμές έχουν οι μεταβλητές, πότε αρχίζει η οπισθοδρόμηση κλπ., χρησιμοποιώντας την εντολή trace. Ένα ισχυρό εργαλείο αποσφαλμάτωσης (debugging) χρειάζεται σε κάθε γλώσσα προγραμματισμού, στην Prolog η αποσφαλμάτωση είναι συνυφασμένη με τη διαδικασία ανίχνευσης (tracing). Για να μπορέσουμε όμως να ερμηνεύσουμε σωστά τα μηνύματα της ανίχνευσης πρέπει να καταλάβουμε με ποιο τρόπο "βλέπει" κάθε διαδικασία. Η Prolog είναι δηλωτική γλώσσα και έχει πολύ απλή διαδικαστική σημασιολογία. Άτυπα η διαδικαστική σημασιολογία της εκτέλεσης ενός προγράμματος Prolog μπορεί να εκφραστεί ως εξής: Για να εκτελεστεί ένας στόχος, το σύστημα ψάχνει την πρώτη πρόταση της οποίας η κεφαλή ενοποιείται με το στόχο. Η διαδικασία ενοποίησης βρίσκει τον πλέον γενικό ενοποιητή των δύο όρων, ο οποίος είναι μοναδικός εφόσον υπάρχει.

1. Εάν η ενοποίηση είναι επιτυχής, η αντικατάσταση της ενοποίησης εφαρμόζεται στο σώμα της πρότασης και οι στοιχειώδης τύποι του σώματος της πρότασης γίνονται νέοι στόχοι, οι οποίοι εκτελούνται με σειρά από αριστερά προς τα δεξιά.
2. Εάν το σύστημα αποτύχει να ενοποιήσει το στόχο, οπισθοδρομεί. Δηλαδή, απορρίπτει τον πλέον πρόσφατα ενεργοποιημένο στόχο, ακυρώνοντας τις αντικαταστάσεις οι οποίες έγιναν με την κεφαλή της πρότασης που ταίριαξε. Κατόπιν, επαναλαμβάνει τον αρχικό στόχο ο οποίος ενεργοποίησε τον απορριφθέντα στόχο και προσπαθεί να βρει μία επόμενη πρόταση η οποία ενοποιείται με το στόχο [2].

Το μοντέλο στο οποίο στηρίζεται η ανίχνευση (trace) φαίνεται στο παρακάτω σχήμα: [2]



Φανταστείτε κάθε συνάρτηση ως ένα κουτί με δυο εισόδους και δυο εξόδους. Οι εισοδοί έχουν τα ονόματα **call** και **redo** και οι έξοδοι τα **exit** και **fail**. Με ένα όνομα οι τέσσερις αυτές εισοδοί/έξοδοι λέγονται **ports**.

- **exit**:Αυτή η θύρα, παριστά την επιτυχή επιστροφή από τη διαδικασία.
- **fail**: Η έξοδος αυτή ενεργοποιείται όταν ο στόχος αποτύχει. Φυσικά δεν είναι ποτέ δυνατόν η fail και η exit να ενεργοποιούνται συγχρόνως. Αποτυχία μπορεί να συμβεί για έναν από τους εξής λόγους:
 1. Δεν υπάρχουν προτάσεις για να ικανοποιηθεί ο τρέχον (αρχικός) στόχος.
 2. Ο τρέχον στόχος, δεν ενοποιήθηκε με κάποια πρόταση.
 3. Η λύση που παράχθηκε για τον αρχικό στόχο αυτής της διαδικασίας, απορρίπτεται από άλλη διαδικασία του προγράμματος.

Μετά από κάποια αποτυχία, το σύστημα αρχίζει να οπισθοδρομεί.

- **call**: Η είσοδος αντιστοιχεί στην συνηθισμένη κλήση ενός κατηγορήματος. Κάτι όμως που πρέπει να τονίσουμε (για να δούμε τη διαφορά από την redo) είναι ότι η κλήση αυτή γίνεται "από την αρχή". Δηλαδή οι μεταβλητές θα πάρουν τις αρχικές τους τιμές, οι κανόνες και τα γεγονότα που θα χρησιμοποιηθούν θα είναι τα πρώτα στη σειρά και γενικά, οποιοδήποτε δείκτης (pointers) για εναλλακτικές κλήσεις δεν έχουν πάρει ακόμα τιμή.
- **redo**: Η είσοδος αυτή αντιστοιχεί στην αναζήτηση εναλλακτικών λύσεων για το στόχο του κατηγορήματος. Φυσικά έχει νόημα μόνο για μη ντετερμινιστικά κατηγορήματα. Το redo σε ένα ντετερμινιστικό κατηγορήμα θα προκαλέσει την ενεργοποίηση της εξόδου fail.

Η λειτουργία της ανίχνευσης λαθών ενεργοποιείται με την εντολή trace/0, ακολουθούμενη από τη συνάρτηση που θέλουμε να εξετάσουμε. Έστω ότι τρέχουμε το παραπάνω στόχο.

Τώρα η Prolog έχει μπει σε αυτό που λέμε trace-mode. Βλέπουμε ότι έχει ξεκινήσει την κλήση της συνάρτησης αλλά την έχει σταματήσει και περιμένει από εμάς να δώσουμε κάποια εντολή για το πώς να συνεχίσει. Σε αυτό το mode λειτουργίας η Prolog αναγνωρίζει ένα σύνολο πλήκτρων καθένα από τα οποία αντιστοιχεί σε μια εντολή. Το πλήρες σύνολο πλήκτρων και τη σημασία του καθενός μπορούμε να τα δούμε αν πατήσουμε το «?». Όμως το συνηθέστερο είναι να πατήσουμε το «Enter» ή το «c» που αντιστοιχούν στην εντολή creep, που σημαίνει να προχωρήσει μόλις ένα βήμα στην εκτέλεση του προγράμματος και να περιμένει επόμενη εντολή από εμάς. Έτσι, πατώντας συνεχώς «Enter» βλέπουμε τις διαδοχικές κλήσεις μέχρι το τελικό αποτέλεσμα:


```

File Edit Settings Run Debug Help
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 7.2.3)
Copyright (c) 1990-2015 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?- check_WBC_PLT([4700,1,75,4.3,45,18,225,32,12,125], noPancytopenia).
true .

2 ?- trace.
true.

[trace] 2 ?- check_WBC_PLT([4700,1,75,4.3,45,18,225,32,12,125], noPancytopenia).
Call: (7) check_WBC_PLT([4700, 1, 75, 4.3, 45, 18, 225, 32|...], noPancytopenia) ? creep
Call: (8) number(4700) ? creep
Exit: (8) number(4700) ? creep
Call: (8) number(125) ? creep
Exit: (8) number(125) ? creep
Call: (8) 4700>4500 ? creep
Exit: (8) 4700>4500 ? creep
Call: (8) 125>150 ? creep
Fail: (8) 125>150 ? creep
Redo: (7) check_WBC_PLT([4700, 1, 75, 4.3, 45, 18, 225, 32|...], noPancytopenia) ? creep
Call: (8) number(4700) ? creep
Exit: (8) number(4700) ? creep
Call: (8) number(125) ? creep
Exit: (8) number(125) ? creep
Call: (8) 4700>4500 ? creep
Exit: (8) 4700>4500 ? creep
Call: (8) 125<150 ? creep
Exit: (8) 125<150 ? creep
Exit: (7) check_WBC_PLT([4700, 1, 75, 4.3, 45, 18, 225, 32|...], noPancytopenia) ? creep
true .

```

Πίνακας 2: Αποσφαλμάτωση

Στην αρχή κάθε γραμμής μας δείχνει το όνομα του port που ενεργοποιείται. Σε περίπτωση που δίπλα από κάποιο port υπάρχει το σύμβολο '^' (call(8)) δηλώνει ότι το κατηγορημα το οποίο καλεί είναι ενσωματωμένο κατηγορημα (built-in) της Prolog και όχι κάποιο που έχει φτιάξει ο προγραμματιστής. Η παρένθεση που ακολουθεί δείχνει το βάθος της εκτέλεσης. Στο παράδειγμά μας η πρώτη κλήση ξεκινάει από βάθος 7 και καλεί κάποιες άλλες συναρτήσεις με βάθος 8, κ.ο.κ. Το βάθος αυξάνεται κατά 1 κάθε φορά που καλεί ένα νέο κατηγορημα. Το βάθος μειώνεται κατά 1 κάθε φορά που συναντάμε την τελεία κάποιου κανόνα.

Έτσι μπορούμε να ακολουθήσουμε βήμα-βήμα την εκτέλεση του προγράμματός μας και να διαπιστώσουμε σε ποιο σημείο δεν λειτουργεί σωστά. Αλλά ακόμα κι όταν τελειώσει η ανίχνευση, η Prolog παραμένει σε trace mode. Αυτό φαίνεται από τη λέξη trace που έχει προστεθεί στην αρχή του prompt:

[trace] 1 ?-

Για να επαναφέρουμε την Prolog στο mode φυσιολογικής λειτουργίας δίνουμε την εντολή notrace/0:

[trace] 1 ?- notrace.

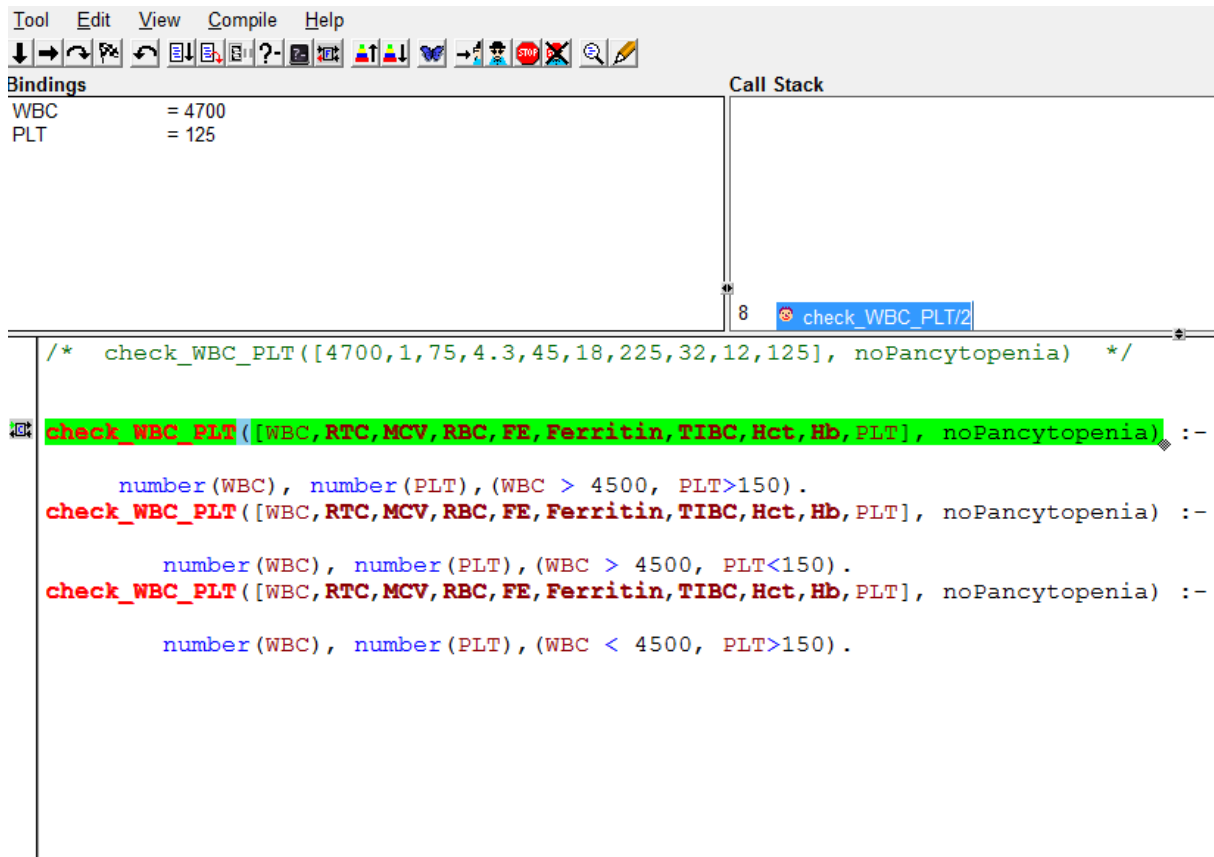
true.

Εναλλακτικά, η SWI-Prolog διαθέτει και ένα γραφικό περιβάλλον για debugging. Για να το ενεργοποιήσουμε δίνουμε την εντολή gtrace/0 αντί της trace/0. Π.χ.:

?- gtrace, check_WBC_PLT([4700,1,75,4.3,45,18,225,32,12,125], noPancytopenia).

% The graphical front-end will be used for subsequent tracing

Η γλώσσα μας ενημερώνει ότι από εδώ και πέρα το γραφικό περιβάλλον θα χρησιμοποιείται για τη διαδικασία ανίχνευσης και παράλληλα ανοίγει ένα καινούργιο παράθυρο στο οποίο μπορούμε να δούμε με παραστατικό τρόπο την εκτέλεση του προγράμματος πατώντας διαδοχικά το πλήκτρο «Space»:



Εικόνα 7:Γραφικό ιχνηλάτισης(tracing)

Η διαδικασία που μόλις είδαμε μπορεί να είναι εφικτή για μικρά προγράμματα, όμως για μεγαλύτερα δεν είναι καθόλου εύκολο να ακολουθούμε βήμα-βήμα την κάθε συνάρτηση που συναντάμε. Γι' αυτό το λόγο η Prolog μας επιτρέπει να δηλώσουμε ποιες από τις συναρτήσεις μας είναι πιθανόν να μην λειτουργούν σωστά, ώστε να χρειαστεί ανίχνευση μόνο σε αυτές. Αυτό το πετυχαίνουμε με τη βοήθεια των συναρτήσεων spy/1 και nospy/1. Συγκεκριμένα, αν δώσουμε:

?- spy(+Predicate_Name).

το πρόγραμμα θα εκτελείται κανονικά μέχρι να συναντήσει κλήση του κατηγορήματος με όνομα Predicate_Name, οπότε μπαίνει σε διαδικασία ανίχνευσης. Στην ορολογία της Prolog λέμε ότι έχουμε βάλει ένα spy-point στη συνάρτηση Predicate_Name. Μπορούν να υφίστανται πολλά spy-points ταυτόχρονα. Το spy-point αναιρεί η:

?- nospy(+Predicate_Name).

Για να σβήσουμε όλα τα spy-points δίνουμε:

**?- nospyall.
Yes**

4. Η Java ως Εργαλείο Κατασκευής Διεπικοινωνίας ενός Συστήματος Γνώσης.

Η διεπαφή υλοποιήθηκε με τέτοιο τρόπο ώστε να είναι φιλική προς το χρήστη. Δημιουργήθηκαν παράθυρα με τις απαραίτητες πληροφορίες ώστε να είναι κατανοητό προς το χρήστη τι στοιχεία πρέπει να συμπληρώσει. Δίπλα από κάθε πεδίο έχει τοποθετηθεί η κατάλληλη ετικέτα με το όνομα του πεδίου που πρέπει να συμπληρωθεί. Για παράδειγμα, για να συμπληρώσει το ονοματεπώνυμο του, δίπλα από το πεδίο συμπλήρωσης τοποθετήθηκε η ετικέτα **Ονοματεπώνυμο**.

- Η ετικέτα δηλώθηκε ως εξής **nameLabel = new javax.swing.JLabel();**
- Οορισμός του ονόματος της ετικέτας έγινε ως εξής:
nameLabel.setText("ΟΝΟΜΑΤΕΠΩΝΥΜΟ ");
- Το πεδίο για το ονοματεπώνυμο δηλώθηκε ως εξής:
nameField = new javax.swing.JTextField();

Το τελικό αποτέλεσμα φαίνεται στην Εικόνα 8. Με παρόμοιο τρόπο δημιουργήθηκαν και τα υπόλοιπα πεδία του συστήματος.



Εικόνα 8: Πεδίο Ονοματεπώνυμο

Με αυτό τον τρόπο δημιουργήθηκε η διεπαφή της Java. Για να γίνει η διεπικοινωνία της Java με την Prolog, χρησιμοποιήθηκε η βιβλιοθήκη `jpl` σε μορφή `jar` αρχείου, η οποία εισήχθη στο σύστημα με την εντολή **import org.jpl7.***, το `*` δηλώνει πως πρέπει να εισαχθούν όλα τα αντικείμενα που υπάρχουν στη βιβλιοθήκη ώστε να μπορέσουν να χρησιμοποιηθούν. Μετά από την εισαγωγή της βιβλιοθήκης για να γίνει η διεπικοινωνία πρέπει να φορτωθεί η βάση γνώση του ΔιΣΑ, δηλαδή το αρχείο **anemia_diagnosis.pl**. Αυτό γίνεται ως εξής:

- Δημιουργούμε ένα πίνακα `java` τύπου `term` στον οποίο δίνουμε το μονοπάτι (`path`) στο οποίο βρίσκεται το αρχείο. Αυτό έγινε με την παρακάτω εντολή:

```
Termconsult_arg[] = {  
    new Atom(Diagnosis.class.getResource("anemia_diagnosis.pl").getPath())  
};
```

- Έπειτα με την εντολή **query** περνάει ως όρισμα ο παραπάνω πίνακας, με αυτόν τον τρόπο καλείται το αρχείο. Αυτό έγινε με την παρακάτω εντολή:

```
Query consult_query  
    = new Query(  
    "consult",  
    consult_arg);
```

Εφόσον φορτωθεί το αρχείο καλούμε κάθε φορά τα κατηγορήματα του ΔιΣΑ που μας ενδιαφέρουν. (περαιτέρω ανάλυση σε παρακάτω ενότητα)

4.1. Εισαγωγή και Βασικά Χαρακτηριστικά της Java

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems (πιο συγκεκριμένα από τον James Gosling ο οποίος δούλεψε στη Sun) και τον Απρίλιο του 2010 εξαγοράστηκε από την Oracle. Η Java προκάλεσε ίσως το μεγαλύτερο ενδιαφέρον σε σύγκριση με οποιαδήποτε άλλη εξέλιξη στον κόσμο του Internet. Όλοι μιλούν γι' αυτήν και έχουν ενθουσιαστεί με τις δυνατότητες που προσφέρει. Είναι η πρώτη που κατάφερε να συμπεριλάβει ήχο και κίνηση σε μια ιστοσελίδα. Η Java επιπλέον επιτρέπει στους χρήστες να αλληλοεπιδρούν (interact) με την ιστοσελίδα. Εκτός από το να διαβάζει απλά και ίσως να συμπληρώνει μία φόρμα, ο χρήστης μπορεί τώρα να παίζει παιχνίδια, να συνομιλήσει, να λαμβάνει συνεχώς τις πιο πρόσφατες πληροφορίες και πολλά άλλα.

Η Java σχεδιάστηκε με σκοπό την ανάπτυξη εφαρμογών που θα τρέχουν σε ετερογενή δικτυακά περιβάλλοντα. Εκτός από το διαδίκτυο όμως μπορεί να χρησιμοποιηθεί και σε άλλες εφαρμογές δημιουργώντας μια διεπαφή φιλική προς το χρήστη και με πολύ απλό και εύκολο τρόπο. Αποτελεί βασική γλώσσα εφαρμογών για οποιοδήποτε λειτουργικό σύστημα αφού ένα από τα βασικότερα πλεονεκτήματα της είναι ότι η Java έναντι των περισσότερων άλλων γλωσσών έχει ανεξαρτησία του λειτουργικού συστήματος. Δηλαδή, τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix, Macintosh, Android κ.α. χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα.

Για τον παραπάνω λόγο είναι κατάλληλη για ολοκληρωμένες ασφαλείς εφαρμογές που τρέχουν στο διαδίκτυο, σε κινητές συσκευές, σε PC, Servers, Tablets κ.α. Όλες οι βιβλιοθήκες της JAVA είναι ελεύθερες όπως και η ίδια η JAVA, οπότε καθένας μπορεί να ξεκινήσει να δημιουργεί εφαρμογές χωρίς να έχει κόστος αγοράς λογισμικού. Το πρόγραμμα εκπαίδευσης JAVA στόχο έχει την εκμάθηση του αντικειμενοστραφή προγραμματισμού και την εφαρμογή αυτού με την JAVA, την πλήρη εξοικείωση με τις αρχές σχεδίασης και υλοποίησης λογισμικού, την δυνατότητα εφαρμογών με γραφικό περιβάλλον, την σύνδεση με βάσεις δεδομένων κ.α.

Η Java έχει τα ακόλουθα χαρακτηριστικά:

- *Αντικειμενοστραφής γλώσσα* (ομοιότητες εντολών με τη C++), δηλαδή τα πάντα στη Java είναι είτε κλάση, είτε μέθοδος ή αντικείμενο.
- Δημιουργία ανεξάρτητων εφαρμογών και applets (applet = προγράμματα που περιλαμβάνονται σε HTML σελίδες και εκτελούνται από τον Web Browser).
- *Interpreted γλώσσα*. Αυτό σημαίνει ότι ο java compiler δεν παράγει εκτελέσιμο κώδικα αλλά μια άλλη μορφή κώδικα σε πιο χαμηλό επίπεδο (bytecode) το οποίο από μόνο του δεν τρέχει σε καμία μηχανή. Προκειμένου λοιπόν να εκτελεστεί απαιτείται η χρήση ενός interpreter (=διερμηνέα) για να μετατρέψει το bytecode σε πραγματικό εκτελέσιμο κώδικα. Αυτό το χαρακτηριστικό δίνει τη δυνατότητα στα java bytecodes να μπορούν να τρέξουν σε οποιοδήποτε μηχανήμα, κάτω από οποιοδήποτε λειτουργικό, αρκεί να έχει εγκατασταθεί ένας java interpreter. Επίσης ένα άλλο χαρακτηριστικό του java bytecode είναι το μικρό του μέγεθος, (μόλις λίγα Kilobytes). Αυτό το κάνει ιδανικό για μετάδοση μέσω του δικτύου.
- *Κατανεμημένη (distributed) γλώσσα*. Δηλαδή ένα πρόγραμμα σε Java είναι δυνατό να το φέρουμε από το δίκτυο και να το τρέξουμε. Επίσης είναι δυνατό διαφορετικά κομμάτια του προγράμματος να έρθουν από διαφορετικά sites.
- *Ασφαλής (secure)*. Στο δίκτυο όμως υπάρχουν πολλοί κίνδυνοι για τον χρήστη μιας δικτυακής εφαρμογής, γι' αυτό η Java έχει σχεδιαστεί έτσι ώστε να ελαχιστοποιείται η πιθανότητα προσβολής του συστήματος του χρήστη από κάποιο applet γραμμένο για τέτοιο σκοπό.

- *Πολυνηματική (multithreaded) γλώσσα.* Η Java υποστηρίζει εγγενώς την χρήση πολλών νημάτων(threads).Δηλαδή ένα απλό πρόγραμμα σε Java μπορεί να κάνει πολλά, διαφορετικά προγράμματα ανεξάρτητα και αλληλοεπιδρώντα. Προκειμένου να το επιτύχει αυτό σε συστήματα με έναν επεξεργαστή, το Java runtime system (interpreter) υλοποιεί ένα δικό χρονοδρομολογητή (scheduler), ενώ σε συστήματα που υποστηρίζουν πολυεπεξεργασία η δημιουργία των threads ανατίθεται στο λειτουργικό σύστημα. Φυσικά όλα αυτά είναι αόρατα τόσο στον προγραμματιστή όσο και στον χρήστη.
- *Υποστηρίζει πολυμεσικές (multimedia) εφαρμογές.* Με αυτό εννοούμε ότι η Java παρέχει ευκολίες στη δημιουργία πολυμεσικών εφαρμογών. Αυτό επιτυγχάνεται τόσο με την ευελιξία της σαν γλώσσα όσο και με τις πλούσιες και συνεχώς εμπλουτιζόμενες βιβλιοθήκες της.
- *Ανεξάρτητη από το σύστημα,* δηλαδή τα προγράμματα σε Java μπορούν να διαβαστούν και να τρέξουν από μεταγλωττιστές σε διάφορες πλατφόρμες.

4.2. Η Java ως εργαλείο διεπαφής για διαδικτυακές εφαρμογές σε Prolog

Όπως αναφέραμε στην προηγούμενη ενότητα η Java έχει αρκετά χαρακτηριστικά και ένα από αυτά είναι πως μπορεί να δημιουργηθεί μια διεπαφή φιλική προς το χρήστη και με πολύ απλό και εύκολο τρόπο. Αποτελεί βασική γλώσσα εφαρμογών για οποιοδήποτε λειτουργικό σύστημα αφού ένα από τα βασικότερα πλεονεκτήματα της είναι ότι η Java έναντι των περισσότερων άλλων γλωσσών έχει ανεξαρτησία του λειτουργικού συστήματος. Επίσης είναι αντικειμενοστραφής καθώς και παράλληλη γλώσσα, δεδομένου ότι υποστηρίζει την οργάνωση των δεδομένων και υπολογισμών σε αντικείμενα, επιτρέπει επίσης την δημιουργία προγραμμάτων με ταυτόχρονα νήματα (threads) που εκτελούνται παράλληλα. Είναι μια γλώσσα υψηλού επιπέδου, είναι πιο δυναμική γλώσσα από την C ή C++/ Οι βιβλιοθήκες της αναπτύσσονται ελεύθερα με την πρόσθεση νέων μεταβλητών και κανόνων χωρίς να επηρεάζονται οι ήδη υπάρχουσες. Για αυτό το λόγο χρησιμοποιείται και για την υλοποίηση τους συγκεκριμένου συστήματος.

Το γραφικό περιβάλλον ενός συστήματος, δηλαδή η διεπαφή του, μπορεί να υλοποιηθεί σε C ή σε Java ή σε Visual Basic ή σε άλλη γλώσσα κατάλληλη για κατασκευή διεπαφών. Η SWI-Prolog χρησιμοποιεί μια αμφίδρομη διεπαφή μεταξύ Java και Prolog, την JPL. Η JPL είναι μια βιβλιοθήκη που χρησιμοποιεί το περιβάλλον της SWI-Prolog και τη JNI διεπαφή της Java ώστε να ενσωματωθεί Prolog σε Java καθώς και για την ενσωμάτωση Java σε Prolog. Κάποια σημαντικά χαρακτηριστικά της βιβλιοθήκης JPL παρατίθενται παρακάτω [11].

1. Είναι δυναμική. Δεν χρειάζεται να γίνει προεπεξεργασία για τη διαχείριση των κλάσεων της Java.
2. Είναι διαλειτουργική με το JavaAPI του JPL.
3. Υλοποιείται με δίκαιο ποσό Prolog κώδικα σε ένα μοντέλο (jpl.pl) και μια συγκεκριμένη SWI-Prolog ξένη βιβλιοθήκη (jpl.dll -Windows). Υλοποιείται σε ANSI C αλλά κάνει πολύ χρήση της SWI-Prolog Foreign Language interface.
4. Δεδομένα Java και αναφορές σε αντικείμενα αναπαρίσταντε μέσα στην Prolog χωρίς καμία απώλεια πληροφοριών (κάποιες εξαιρέσεις, Java float και double τιμές μετατρέπονται σε Prolog float. Java byte, char, short, int μετατρέπονται σε Prolog

integer τιμές. Αλλά οι διακρίσεις των τύπων οι οποίες χάνονται είναι άνευ ουσιαστικής σημασίας).

5. Όταν καλούνται από Prolog void μέθοδοι επιστρέφουν void τιμές.

6. Οι κλήσεις Prolog-Java και Java-Prolog είναι ανεξάρτητες σε μεγάλο βαθμό. Η πρώτη επικεντρώνεται στην αναπαράσταση όλων των Java τιμών δεδομένων και αντικειμένων σε Prolog και υποστηρίζει την διαχείριση αντικειμένων. Και η δεύτερη επικεντρώνεται στην αναπαράσταση κάθε Prolog όρου(term) σε Java, υποστηρίζει την κλήση στόχων μέσα στην Prolog και στην ανάκτηση των αποτελεσμάτων πίσω στη Java.

Όλες οι τιμές και οι αναφορές σε αντικείμενα που περνάνε μεταξύ της Prolog και της JavaVM (εικονική μηχανή) μέσω του *JPL Prolog API* θεωρούνται στιγμιότυπα τύπων μέσω ενός απλοποιημένου JPL συστήματος.

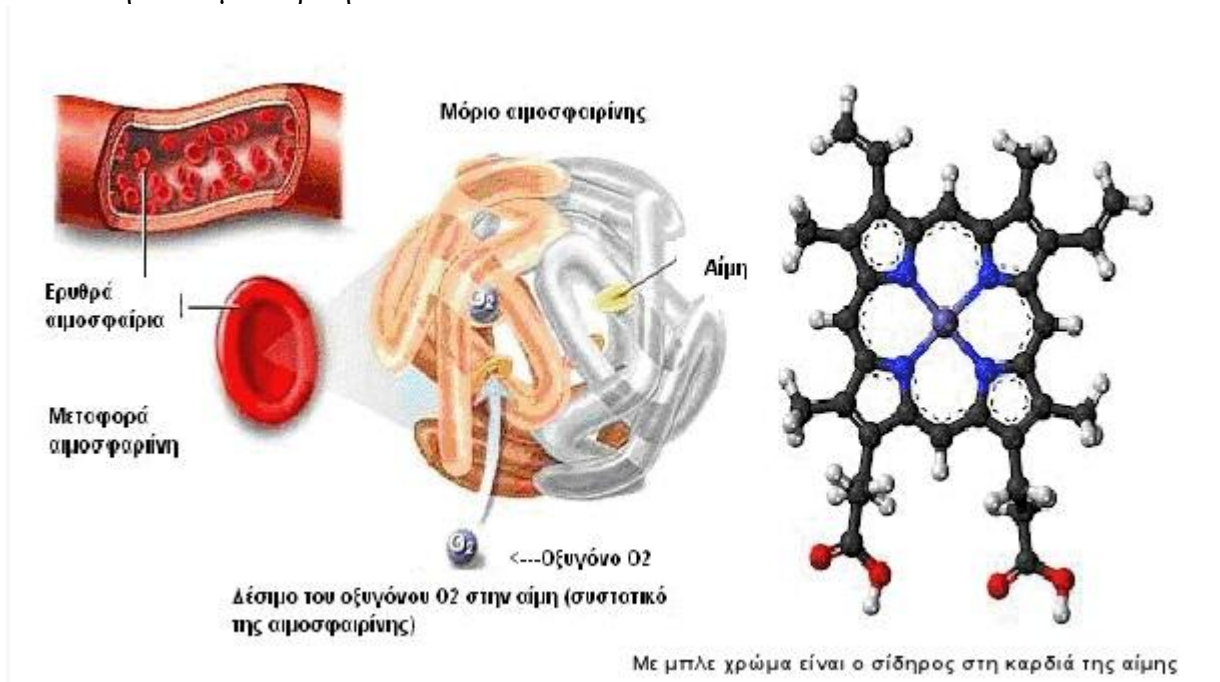
5. Εισαγωγικές Έννοιες της Αναιμίας.

Για να γίνει πιο κατανοητό το θέμα της αναιμίας πρέπει να εξηγηθούν κάποιοι όροι που πιθανώς όλοι ακούμε. Στα περισσότερα έμβια όντα υπάρχει αίμα. Τι είναι το αίμα; Είναι ένα σύνολο κυττάρων (ερυθρά, λευκά, αιμοπετάλια) με διαφορετικούς ρόλους που κυκλοφορούν μέσα σε ένα υγρό τον ορό (πλάσμα) που περιέχει εκτός από τα κύτταρα, νερό και διάφορες ουσίες διαλυμένες σε αυτό, ουσίες θρεπτικές όπως τα σάκχαρα, λιπίδια, πρωτεΐνες, μέταλλα (νάτριο, κάλιο, ασβέστιο), ορμόνες ελεύθερες ή συνδεδεμένες με πρωτεΐνες και φυσικά και άχρηστες ουσίες που αποβάλλονται από τα κύτταρα.

Ο βασικός ρόλος του αίματος είναι η μεταφορά-τροφοδοσία όλων των ιστών του σώματος με θρεπτικά στοιχεία και οξυγόνο και η αντίστοιχη μεταφορά πίσω των σκουπιδιών για απόρριψη. Είναι μια συνεχής και ατελείωτη θαυμαστή ζωντανή κυκλοφορία.

Ως αιματοκρίτη ορίζουμε το ποσοστό % των ερυθρών αιμοσφαιρίων μέσα στο σύνολο του αίματος, νούμερο που έχει ένα φυσιολογικό εύρος που κυμαίνεται κατά μέσο όρο γύρω στο 45%, και πιο συγκεκριμένα σε 40-52% για τους άνδρες και σε 36-48% για τις γυναίκες.

Τα ερυθρά αιμοσφαίρια είναι τα κύτταρα που κουβαλούν τα αέρια ανταλλαγής για την ενέργεια που χρειάζονται οι ιστοί, δηλαδή το οξυγόνο και το διοξείδιο του άνθρακα (απόβλητο), από τους πνεύμονες στους ιστούς. Για να το πετύχουν αυτό έχουν μια πρωτεΐνη την αιμοσφαιρίνη που έχει στενή συγγένεια με τα αέρια αυτά. Απαραίτητος στη δομή της αιμοσφαιρίνης είναι ο σίδηρος για τον οποίο γίνεται τόσος λόγος. Έλλειψη σιδήρου προκαλεί αλλαγή στο ποσό της αιμοσφαιρίνης, το μέγεθος των ερυθρών και τον αριθμό τους και τελικά σε πτώση του αιματοκρίτη.



Εικόνα 9: Μόριο αιμοσφαιρίνης

Ως αναιμία ορίζουμε τη μειωμένη ποσότητα ή ανεπάρκεια κάποιου στοιχείου του αίματος. Για παράδειγμα της αιμοσφαιρίνης η οποία προκαλεί σιδηροπενική αναιμία. Όπως γνωρίζουμε, η αιμοσφαιρίνη περιέχεται στα ερυθρά αιμοσφαίρια και είναι υπεύθυνη για τη

μεταφορά του οξυγόνου στους διάφορους ιστούς του σώματος. Τα αίτια που οδηγούν σε αναιμία ποικίλλουν και γι' αυτό η ταξινόμησή τους γίνεται συνήθως ανάλογα με το μηχανισμό που τις προκαλεί. Σε αυτούς συμπεριλαμβάνονται η μειωμένη παραγωγή ερυθρών αιμοσφαιρίων, η αύξηση του ρυθμού καταστροφής τους ή η απώλεια αίματος (π.χ. σε χειρουργείο, τραυματισμούς, έλκη).

Η αναιμία δεν αποτελεί στην ουσία κλινική εκδήλωση (σύμπτωμα), αλλά εργαστηριακό εύρημα. Δεν αποτελεί νόσο, αφού είναι αποτέλεσμα ποικίλων ασθενειών και, κατά συνέπεια, η διάγνωση "αναιμία" δεν έχει ουσιαστικό νόημα. Για να είναι η διάγνωση σαφής, θα πρέπει να προσδιορίζεται η συγκεκριμένη νόσος που προκαλεί την αναιμία ή να αναφέρεται σε συγκεκριμένο τύπο αναιμίας (Πηγή Πληροφοριών [3],[7]).

A. Αναιμίες που οφείλονται σε μειωμένη παραγωγή ερυθρών αιμοσφαιρίων

Στην κατηγορία αυτή συμπεριλαμβάνονται η απλαστική αναιμία, που οφείλεται σε μειωμένη παραγωγή ερυθρών αιμοσφαιρίων από τον μυελό των οστών και αποτελεί σπάνια διαταραχή.

Οι μεγαλοβλαστικές αναιμίες οφείλονται σε ανεπάρκεια φυλλικού οξέος και/ή της βιταμίνης B12 και χαρακτηρίζονται από την παρουσία μεγάλων ερυθροβλαστών. Έλλειψη βιταμίνης B12 παρατηρείται συνήθως σε αποκλειστικά χορτοφάγους και σε άτομα με παθήσεις του ειλεού, ενώ έλλειψη φυλλικού οξέος εμφανίζεται σε άτομα που δεν καταναλώνουν φρούτα και λαχανικά, σε αλκοολικούς και σε πάσχοντες από παθήσεις του λεπτού εντέρου.

Η σιδηροπενική αναιμία οφείλεται στα μειωμένα επίπεδα σιδήρου, που δεν επαρκούν για να καλύψουν τις ανάγκες σύνθεσης της αιμοσφαιρίνης. Είναι η πιο συχνή μορφή αναιμίας γι' αυτό και είναι αυτή που θα αναλύσουμε περισσότερο. Τέλος, στην κατηγορία αυτή κατατάσσουμε και τις αναιμίες που συνοδεύουν χρόνιες παθήσεις όπως τη χρόνια νεφρική ανεπάρκεια, τις ηπατοπάθειες και τα κακοήγη νεοπλάσματα [3],[7].

B. Αναιμίες που οφείλονται σε αυξημένη καταστροφή των ερυθρών αιμοσφαιρίων

Εδώ συγκαταλέγονται οι αναιμίες που τις περισσότερες φορές οφείλονται σε κληρονομικά αίτια και περιλαμβάνουν μεταξύ άλλων την κληρονομική σφαιροκυττάρωση και τις αιμοσφαιρινοπάθειες (α- και β-μεσογειακή αναιμία, δρεπανοκυτταρική αναιμία). Επειδή τα συμπτώματα αυτού του είδους αναιμίας είναι συνήθως σοβαρά, απαιτούνται πολλές φορές μεταγγίσεις αίματος στους ασθενείς.

Η αναιμία μπορεί να οφείλεται σε πολλούς παράγοντες, παρακάτω φαίνονται ορισμένοι από αυτούς.

- Κακή διατροφή, που δεν περιλαμβάνει αρκετές τροφές πλούσιες σε σίδηρο.
- Εγκυμοσύνη.
- Αδυναμία του μυελού των οστών να παράγει ερυθρά αιμοσφαίρια (απλαστική αναιμία).
- Κληρονομικές ή επίκτητες παθήσεις, όπως η δρεπανοκυτταρική και η μεσογειακή αναιμία, που αποτελούν αναιμίες αιμολυτικού τύπου.
- Χρόνιες παθήσεις (ηπατική νόσος, ερυθματώδης λύκος, κακοήθειες, λοιμώξεις όπως η πνευμονία).
- Νεφρική ανεπάρκεια.

Το βασικότερο σύμπτωμα είναι η καταβολή δυνάμεων. Νιώθετε εύκολα κόπωση και εξάντληση. Σας λείπει ενέργεια και ενώ καταβάλλετε την ίδια προσπάθεια για να ανταποκριθείτε στις καθημερινές σας δραστηριότητες, κουράζεστε περισσότερο από πριν. Μπορεί να δυσκολεύεστε να κάνετε τις δουλειές του σπιτιού, να συγκεντρωθείτε στο διάβασμα ή να παρακολουθήσετε τηλεόραση. Εάν αυξηθεί το βάρος σας, εξαντλούνται ακόμη περισσότερο τα επίπεδα ενέργειας που διαθέτετε. Εάν η αναιμία προχωρήσει, μπορεί να προκαλέσει χλωμάδα στο δέρμα, δύσπνοια, λαχάνιασμα, πονοκέφαλο και ζαλάδες

(ιδιαίτερα όταν στέκεστε όρθιοι), ταχυκαρδία, βοή στα αυτιά, δίψα, απώλεια βάρους, προβλήματα μνήμης, σύγχυση, υπνηλία.

Μόνο ο γιατρός μπορεί να κρίνει εάν έχετε αναιμία ή όχι, τον τύπο και την αντιμετώπισή της. Η αυθαίρετη λήψη συμπληρωμάτων σιδήρου, βιταμίνης B12, φυλλικού οξέος και άλλων βιταμινών χωρίς τη σύμφωνη γνώμη του ειδικού μπορεί όχι απλώς να μη λύσει το πρόβλημα, αλλά ακόμα και να το επιδεινώσει. Ο ειδικός θα αξιολογήσει τα συμπτώματα που θα του περιγράψετε και τα αποτελέσματα των αιματολογικών εξετάσεων, ελέγχοντας ειδικότερα τα επίπεδα της αιμοσφαιρίνης στο αίμα, που δεν θα πρέπει να πέφτουν κάτω από τα 12 g/dl. (1 dl=100 ml) .

Εάν υπάρχει έλλειψη σιδήρου, βιταμίνης B12 ή φυλλικού οξέος, μπορεί να σας συστήσει την κατανάλωση συμπληρωμάτων διατροφής και εμπλουτισμό της διατροφής σας με τροφές πλούσιες σε αυτά τα συστατικά. Θα σας παραπέμψει στον κατάλληλο γιατρό για την αντιμετώπιση μιας αιμορραγίας που τυχόν υπάρχει, π.χ. στο έντερο, ή ενός προβλήματος στους νεφρούς που προέκυψε λόγω λοιμώξεων που οδηγούν σε απώλεια σιδήρου. Η συνήθης αντιμετώπιση της αναιμίας στη χρόνια νεφρική ανεπάρκεια και στον καρκίνο είναι η χορήγηση ερυθροποιητίνης, της ορμόνης που βοηθά στην παραγωγή νέων ερυθρών αιμοσφαιρίων.

Υπάρχουν τρόποι να προφυλαχτεί κάποιος από την αναιμία, ειδικά εάν αυτή είναι ήπιας μορφής, όπως για παράδειγμα η Σιδηροπενική αναιμία.:

- Εμπλουτίστε τη διατροφή σας με τροφές πλούσιες σε σίδηρο, όπως συκώτι, μοσχαρίσιο κρέας, δημητριακά, αυγά, ξηρούς καρπούς και φρούτα. Τρώτε ωμά λαχανικά και εσπεριδοειδή που περιέχουν βιταμίνη C, η οποία βοηθά στην απορρόφηση του σιδήρου. Φυλλικό οξύ υπάρχει μεταξύ άλλων και στα φασολάκια, στα δημητριακά, στα μανιτάρια και στο μπρόκολο.
- Με τη σύμφωνη γνώμη του γιατρού σας, μπορείτε να χρησιμοποιήσετε σκευάσματα που περιέχουν βιταμίνες, φυλλικό οξύ και σίδηρο, για να ενισχύσετε τον οργανισμό σας.
- Προτιμήστε τους φυσικούς χυμούς και αποφύγετε τα αλκοολούχα ποτά.

Γ. Σιδηροπενική αναιμία

Μέχρι στιγμής μιλάμε γενικά περί αναιμίας, ας δούμε λοιπόν πιο αναλυτικά το κάθε είδος της. Στο σώμα υπάρχουν 4-5 γραμμάρια σιδήρου, εκ των οποίων το 1 γραμμάριο λειτουργεί ως απόθεμα με τη μορφή φερριτίνης. Κάθε μέρα, ο οργανισμός χάνει 1-3 mg σιδήρου: αυτά πρέπει να τα αναπληρώσουμε μέσω της διατροφής μας, διαφορετικά η έλλειψη σιδήρου αυξάνει μέχρι που φτάνει κάποια στιγμή να επηρεάσει και τα αποθέματα σιδήρου και αρχίζει η **σιδηροπενική αναιμία**. Η σιδηροπενική αναιμία είναι η πιο συχνή μορφή αναιμίας. Πλήττει κυρίως νεογνά, παιδιά, εφήβους και γυναίκες αναπαραγωγικής ηλικίας, κυφορούσες ή μη. Επιδημιολογικά στοιχεία δείχνουν ότι το 3% των γυναικών έχουν σιδηροπενική αναιμία, ενώ το 20% των γυναικών έχουν μειωμένα αποθέματα σιδήρου χωρίς όμως να παρουσιάζουν αναιμία. Όπως προαναφέρθηκε ο βασικός παράγοντας που καθορίζει τα συμπτώματα της σιδηροπενικής αναιμίας, είναι ο βαθμός έλλειψης σιδήρου. Αν πάσχετε από μικρή ανεπάρκεια σιδήρου μπορεί να μην έχετε κανένα σύμπτωμα. Αν όμως έχετε σοβαρή ανεπάρκεια σιδήρου, θα έχετε σοβαρά συμπτώματα και κυρίως κόπωση.

Τα βασικά συμπτώματα που πρέπει να σας ανησυχήσουν είναι η εύκολη κόπωση, η δύσπνοια, η ταχυκαρδία και το αίσθημα παλμών, ο πονοκέφαλος, ιδιαίτερα στο μέτωπο, εμβοές στα αυτιά, ανορεξία και εντερικές διαταραχές, ωχρό δέρμα και ζαλάδα που μπορεί να φτάσει στην λιποθυμία. Άλλα συμπτώματα είναι η ξηρότητα στο στόμα και το λαιμό, οι πληγές στη γλώσσα, η τριχόπτωση και τα νύχια που σπάνε εύκολα. Η σιδηροπενική αναιμία

μπορεί επίσης να εμφανίσει το σύνδρομο των ανήσυχων ποδιών. Τα συμπτώματα της αναιμίας εμφανίζονται όταν το σώμα δεν έχει αρκετό οξυγόνο.

Εκτός από τα παραπάνω συμπτώματα, αν δεν έχετε επαρκή επίπεδα σιδήρου στο αίμα σας, η καρδιά σας πρέπει να εργάζεται σκληρότερα για να μεταφέρει οξυγόνο στα όργανα του σώματος σας. Η σιδηροπενική αναιμία, μπορεί να οδηγήσει σε καρδιακά προβλήματα, όπως είναι ταχυπαλμία, θωρακικό άλγος και καρδιακή ανεπάρκεια και εγκεφαλικό. Σε μια έγκυο γυναίκα, οι συνέπειες της σιδηροπενικής αναιμίας μπορεί να οδηγήσουν σε πρόωρο τοκετό ή σε γέννηση λιποβαρούς μωρού.

Κάποιοι άνθρωποι βρίσκονται στις ομάδες υψηλού κινδύνου απόκτησης σιδηροπενικής αναιμίας. Επειδή οι γυναίκες χάνουν αίμα κατά την έμμηνο ρύση γενικά διατρέχουν μεγαλύτερο κίνδυνο από σιδηροπενική αναιμία. Τα βρέφη, ειδικά εκείνα που είχαν χαμηλό βάρος κατά τη γέννηση τους ή γεννήθηκαν πρόωρα, όποια δεν παίρνουν αρκετό σίδηρο από το μητρικό γάλα ενδέχεται να διατρέχουν κίνδυνο, από έλλειψη σιδήρου. Τα παιδιά χρειάζονται επιπλέον σίδηρο κατά τη διάρκεια της ανάπτυξης. Εάν το παιδί δεν τρώει μια υγιεινή και ποικίλη διατροφή, μπορεί να διατρέχει κίνδυνο αναιμίας. Οι άνθρωποι που δεν τρώνε κρέας μπορεί να έχουν μεγαλύτερο κίνδυνο σιδηροπενικής αναιμίας και αν δεν τρώνε άλλα πλούσια σε σίδηρο τρόφιμα. Οι άνθρωποι που δίνουν τακτικά αίμα μπορεί να έχουν αυξημένο κίνδυνο να πάθουν σιδηροπενική αναιμία, δεδομένου ότι με την αιμοδοσία μπορεί να καταστρέφουν αποθήκες σιδήρου. Η χαμηλή αιμοσφαιρίνη που σχετίζεται με τη δωρεά αίματος μπορεί να είναι ένα προσωρινό πρόβλημα που μπορεί να αποκατασταθεί με την κατανάλωση περισσότερων τροφών πλούσιες σε σίδηρο. Αν σας πουν ότι δεν μπορείτε να δώσετε αίμα λόγω της χαμηλής αιμοσφαιρίνης, ρωτήστε το γιατρό σας αν πρέπει να ανησυχείτε.

Εάν έχετε κάποιο από αυτά τα συμπτώματα και ανησυχείτε μπορείτε να επισκεφθείτε το γιατρό σας. Η διάγνωση γίνεται με αιματολογικές εξετάσεις και, σε περίπτωση ανεπαρκούς εντοπισμού του αίτιου, συμπληρώνονται από έλεγχο κοπράνων, προκειμένου να διαπιστωθεί αν η αιτία είναι η εντερική αιμορραγία.

Αν έχετε έλλειψη σιδήρου σε σημείο που να εμφανίσετε αναιμία, η κατανάλωση τροφών με πολύ σίδηρο είναι ευεργετική, αλλά δεν είναι αρκετή, για να αντιμετωπιστεί η νόσος σας. Η ανεπάρκεια σιδήρου δεν μπορεί να θεραπευτεί σε μία νύχτα. Μπορεί να χρειαστεί να παίρνετε συμπληρώματα σιδήρου για πολλούς μήνες ή και περισσότερο, για να αναπληρώσετε τα αποθέματα του σώματος. Στις εγκυμονούσες γυναίκες χορηγείται συνήθως σίδηρος για όλη τη διάρκεια της εγκυμοσύνης τους για πρόληψη ή και θεραπεία της σιδηροπενικής αναιμίας.

Στους ενήλικες, αν τα συμπληρώματα σιδήρου δεν αυξήσουν τα επίπεδα του σιδήρου του αίματος, η νόσος μπορεί να οφείλεται σε μια πηγή αιμορραγίας ή διαταραχή της απορρόφησης του σιδήρου. Και στις δυο περιπτώσεις πρέπει να γίνει γνωστό το αίτιο και να αντιμετωπιστεί. Τα αντισυλληπτικά χάπια μπορεί να βοηθήσουν στη θεραπεία της σιδηροπενικής αναιμίας μειώνοντας τις βαριές περιόδους. Στα άτομα με απώλεια αίματος λόγω έλκους χορηγούνται συνήθως αντιβιοτικά για τη θεραπεία του έλκους. Τα προβλήματα της κακής απορρόφησης σιδήρου μπορούν επίσης να αντιμετωπιστούν με φάρμακα.

Αν η αιτία της σιδηροπενικής αναιμίας είναι η απώλεια αίματος λόγω όγκου ή πολύποδα που αιμορραγεί, μπορεί να είναι απαραίτητη η εγχείρηση. Αν έχετε βαριά αναιμία και χρειάζεται άμεση διόρθωση, ο γιατρός σας μπορεί να συστήσει μετάγγιση συμπυκνωμένων ερυθρών αιμοσφαιρίων. Όλα αυτά δεν πρέπει να σας πανικοβάλλουν, για αυτό σε περίπτωση που διαπιστώσετε ότι έχετε κάποια από τα συμπτώματα καλό είναι να

επισκεφθείτε αμέσως το γιατρό σας [3],[7].

Είδος τροφής	Περιεκτικότητα σε σίδηρο mg/100gr	Είδος τροφής	Περιεκτικότητα σε σίδηρο mg/100gr
Λαχανικά: σπανάκι φασόλια	3.0 2.3	Κρέας: μόσχου χοίρου	2.9 2.2
Φρούτα: βερίκοκα	4.9	Όσπρια: φασόλια φακή μπιζέλια	10.3 8.3 6.0
Ξηροί καρποί: δαμάσκηνα σύκα σταφίδες αμύγδαλα	3.9 3.1 3.3 4.4	Διάφορα: αυγό σοκολατα αλεύρι πλήρες	2.5 4.4 4.0

Πίνακας 3: Τροφές και σίδηρος²

Δ. Μεγαλοβλαστική αναιμία

Η **Μεγαλοβλαστική αναιμία** είναι μια μορφή αναιμίας που οφείλεται στην ανεπάρκεια βιταμίνης B12 και/ή φυλλικού οξέος. Και τα δύο είναι ουσίες που είναι απαραίτητες για την αιμοποίηση – ανεπαρκής πρόσληψή τους, ή υπερβολικά γρήγορη κατανάλωσή τους, ή ανικανότητα του οργανισμού να τις προσλάβει ή να τις αξιοποιήσει σημαίνει πως μειώνεται η ικανότητά μας να παράγουμε «καινούργιο αίμα». Προκαλείται όταν τα ερυθρά αιμοσφαίρια δεν παράγονται κανονικά. Επειδή τα κύτταρα είναι πολύ μεγάλα, μπορεί να μην είναι σε θέση να βγούνε από το μυελό των οστών για να εισέλθουν στην κυκλοφορία του αίματος και να παραδώσουν το οξυγόνο.

Η **βιταμίνη B12** είναι μια βασική πρωτεΐνη, που παίζει πολλούς ρόλους στον οργανισμό, καθώς είναι απαραίτητη για την ανάπτυξη και τη διατήρηση υγιούς νευρικού συστήματος, για την παραγωγή του DNA και για τη δημιουργία των ερυθρών αιμοσφαιρίων. Είναι ένα θρεπτικό συστατικό που βρίσκεται σε ορισμένα τρόφιμα, όπως το κρέας, τα ψάρια, τα αυγά και το γάλα. Μερικοί άνθρωποι δεν μπορούν να απορροφήσουν αρκετή βιταμίνη B12 από την τροφή τους, που οδηγεί σε μεγαλοβλαστική αναιμία. Μεγαλοβλαστική αναιμία που προκαλείται από ανεπάρκεια της βιταμίνης B-12 αναφέρεται ως κακοήθης αναιμία.

Η ανεπάρκεια βιταμίνης B12 συχνά προκαλείται από την έλλειψη μιας πρωτεΐνης στο στομάχι που ονομάζεται "ενδογενής παράγοντας". Χωρίς τον ενδογενή παράγοντα, η βιταμίνη B - 12 δεν μπορεί να απορροφηθεί, ανεξάρτητα από τη διατροφή. Είναι επίσης

²Πίνακας από το google

δυνατό να αναπτυχθεί κακοήθης αναιμία, επειδή δεν υπάρχει αρκετή βιταμίνη Β - 12 στη διατροφή σας.

Το **φυλλικό οξύ** είναι ένα άλλο θρεπτικό συστατικό που είναι σημαντικό για την ανάπτυξη των υγιών ερυθρών αιμοσφαιρίων. Φυλλικό οξύ βρίσκεται σε τρόφιμα όπως το βοδινό συκώτι, σπανάκι, και τα λαχανάκια Βρυξελλών. Μπορείτε επίσης να το βρείτε σε εμπλουτισμένα δημητριακά.

Η διατροφή σας είναι ένας σημαντικός παράγοντας για να βεβαιωθείτε ότι έχετε αρκετό φυλλικό οξύ. Ανεπάρκεια φυλλικού οξέος μπορεί επίσης να προκληθεί από χρόνια κατάχρηση αλκοόλ, αφού το αλκοόλ παρεμβαίνει με την ικανότητα του σώματος να απορροφήσει το φυλλικό οξύ. Οι έγκυες γυναίκες είναι πιο πιθανό να έχουν έλλειψη φυλλικού οξέος, λόγω των υψηλών ποσοτήτων φυλλικού οξέος που απαιτείται από το αναπτυσσόμενο έμβρυο.

Το πιο κοινά συμπτώματα της μεγαλοβλαστικής αναιμίας είναι η κόπωση. Τα συμπτώματα μπορεί να διαφέρουν από άτομο σε άτομο. Τα συνήθως περιλαμβάνουν, δυσκολία στην αναπνοή, μυϊκή αδυναμία, ωχρότητα του δέρματος, γλωσσίτιδα, απώλεια όρεξης/βάρους, ταχυκαρδία, λεία γλώσσα, μούδιασμα άκρων.

Με μια γενική εξέταση αίματος ο γιατρός σας μπορεί να ελέγξει τον αριθμό και την εμφάνιση των ερυθρών αιμοσφαιρίων σας. Αν έχετε μεγαλοβλαστική αναιμία θα εμφανιστούν μεγαλύτερα και υπανάπτυκτα. Θα συγκεντρώσει επίσης το ιατρικό ιστορικό σας και θα εκτελέσει μια φυσική εξέταση για να αποκλειστούν άλλες αιτίες των συμπτωμάτων σας.

Ένα άλλο τεστ που ο γιατρός σας μπορεί να χρησιμοποιήσει για να βοηθήσει στη διάγνωση είναι η δοκιμασία Schilling. Η δοκιμασία Schilling είναι ένα τεστ αίματος που αξιολογεί την ικανότητά απορρόφησης βιταμίνης Β12. Αφού πάρει ένα μικρό συμπλήρωμα της ραδιενεργούς βιταμίνης Β12, θα συλλέξει ένα δείγμα ούρων προς ανάλυση. Θα λάβει στη συνέχεια το ίδιο ραδιενεργό συμπλήρωμα σε συνδυασμό με την πρωτεΐνη «ενδογενής παράγοντας» που χρειάζεται το σώμα σας για να είναι σε θέση να απορροφήσει τη βιταμίνη Β12. Στη συνέχεια, θα παρέχει άλλο ένα δείγμα ούρων έτσι ώστε να μπορεί να συγκριθεί με την πρώτη.

Η θεραπεία σχετίζεται και με την αιτία που προκλήθηκε η αναιμία καθώς επίσης και με την ηλικία και την συνολική υγεία σας. Στην περίπτωση μεγαλοβλαστικής αναιμίας λόγω έλλειψης βιταμίνης Β12 ίσως χρειαστεί η χορήγηση της βιταμίνης σε ενέσιμη μορφή ή υπό την μορφή συμπληρωμάτων δια του στόματος και ο εμπλουτισμός της διατροφής με τροφές πλούσιες σε βιταμίνη Β12. Τέτοιες τροφές φαίνονται στον παρακάτω πίνακα (Πίνακας 4).

Πηγές βιταμίνης B₁₂	
Ποσότητα τροχίμου	Βιταμίνη B₁₂ (μg)
Δημητριακά	
1 φλιτζάνι All-Bran	0.5
1 φλιτζάνι Bran-Flakes	0.8
1 φλιτζάνι Corn-Flakes	0.5
Γάλα & γαλακτοκομικά προϊόντα	
γάλα (280ml) πλήρες, ημίπαχο, άπαχο	1.0
Γάλα σόγιας (ανάλογα την εταιρεία)	
1 κεσεδάκι γιαούρτι	0.3
Τυρί (50g)	
1 αυγό (60g)	0.4
Λίπη & έλαια	
Βούτυρο, μαργαρίνη, λάδι	Ίχνη
Ενισχυμένη μαργαρίνη (10g)	0.5
Κρέας & ψάρι	
Κρέας ψητό (85g)	1.6
Συκώτι αρνιού μαγειρεμένο (90g)	70.0
Νεφρό αρνιού μαγειρεμένο (75g)	55.0
Άσπρο ψάρι μαγειρεμένο (150g)	1.5-4.0
2 φιλέτα ρέγγας, μαγειρεμένα (110g)	9.0
2 φιλέτα παστίης ρέγγας μαγειρεμένα (130g)	8.0
2 φιλέτα σκουμπρί μαγειρεμένα (110g)	15.0
Σαρδέλες σε κονσέρβα (70g)	10.0
Τόνος σε κονσέρβα	4.0

Εξαιρετικές πηγές: **(έντονα γράμματα)**

Σημείωση: τα φυτικά τρόφιμα είναι ελλιπή σε B₁₂ (εκτός αν είναι εμπλουτισμένα), εκτός από τις περιπτώσεις όπου περιλαμβάνεται τυχαία σχηματισμένη B₁₂ στο νερό ή στο χόμα, από μικροοργανισμούς.

Πίνακας 4: Πηγές βιταμίνης B₁₂³

Στην περίπτωση μεγαλοβλαστικής αναιμίας λόγω έλλειψης φυλλικού οξέος ίσως χρειαστεί η χορήγηση του σε ενέσιμη μορφή ή υπό την μορφή συμπληρωμάτων δια του στόματος και ο εμπλουτισμός της διατροφής με τροφές πλούσιες σε φυλλικό οξύ. Τέτοιες τροφές φαίνονται στον παρακάτω πίνακα (Πίνακας 5) [3],[7].

³ Πίνακας από google

Φυλλικό οξύ στις τροφές

Τροφή	gη φυλλικού οξέος/ 100gη τροφής
Συκώτι	145
Βοδινό κρέας	4
Χοιρινό κρέας	5
Αρνί	3
Σαρδέλες κονσέρβας	16
Τόνος	15
Σολομός	20
Αυγά	49
Σόγια	-
Ξηροί καρποί	106
Μπιζέλια	133
Σπανάκι	91
Μπροκόλο	56
Αποξηραμένα βερικόκα	14

Πίνακας 5: Πηγές φυλλικού οξέος⁴

Ε. Μεσογειακή αναιμία

Η **μεσογειακή αναιμία(θαλασσαιμία)** είναι μια κληρονομική διαταραχή του αίματος, πράγμα που σημαίνει ότι τουλάχιστον ένας από τους γονείς σας πρέπει να είναι φορέας της νόσου. Η θαλασσαιμία αποτελεί μία ποσοτική βλάβη και συγκεκριμένα στη μειωμένη σύνθεση των σφαιρινών που αποτελούν την αιμοσφαιρίνη. Η θαλασσαιμία είναι αποτέλεσμα της μειωμένης παραγωγής των φυσιολογικών σφαιρινών της αιμοσφαιρίνης, που οφείλεται σε μεταλλάξεις των φυσιολογικών γονιδίων που τις κωδικοποιούν. Οι δύο κύριες μορφές θαλασσαιμίας είναι **α - θαλασσαιμία** και **β - θαλασσαιμία**. Στην α- θαλασσαιμία, τουλάχιστον ένα από τα γονίδια άλφα σφαιρίνης έχει μια μετάλλαξη ή ανωμαλία. Στην β-θαλασσαιμία, τα γονίδια βήτα σφαιρίνης επηρεάζονται.

Εάν μόνο ένας από τους γονείς είναι ένας φορέας θαλασσαιμίας, ο ασθενής μπορεί να αναπτύξει μια μορφή της νόσου η οποία είναι γνωστή ως **στίγμα** θαλασσαιμίας. Εάν συμβεί αυτό, τότε μάλλον δεν θα υπάρξουν συμπτώματα, αλλά θα είναι φορέας της νόσου. Μερικοί άνθρωποι με στίγμα θαλασσαιμίας αναπτύσσουν ήπια συμπτώματα.

Τα συμπτώματα ποικίλουν ανάλογα με τη μορφή της νόσου που υπάρχει. Το **στίγμα** συνήθως δεν έχει συμπτώματα εκτός από ορισμένες περιπτώσεις που τα συμπτώματα είναι αυτά της σιδηροπενικής αναιμίας. Η **β- θαλασσαιμία** έρχεται σε δύο σοβαρές μορφές, οι οποίες είναι η **μείζονα θαλασσαιμία ή αναιμία του Cooley** και η **ενδιάμεση θαλασσαιμία**. Τα συμπτώματα της μείζονος θαλασσαιμίας εμφανίζονται γενικά πριν από τα δεύτερα γενέθλια ενός παιδιού. Η σοβαρή αναιμία που σχετίζεται με αυτή την κατάσταση μπορεί να είναι απειλητική για τη ζωή. Άλλα σημεία και συμπτώματα περιλαμβάνουν σχολαστικότητα, ωχρότητα, συχνές λοιμώξεις, απώλεια όρεξης, ίκτερο, το οποίο είναι ένα κιτρίνισμα του δέρματος ή του λευκού των ματιών, διευρυμένα όργανα. Αυτή η μορφή θαλασσαιμίας είναι συνήθως τόσο σοβαρή που απαιτεί τακτικές μεταγγίσεις αίματος. Η ενδιάμεση θαλασσαιμία

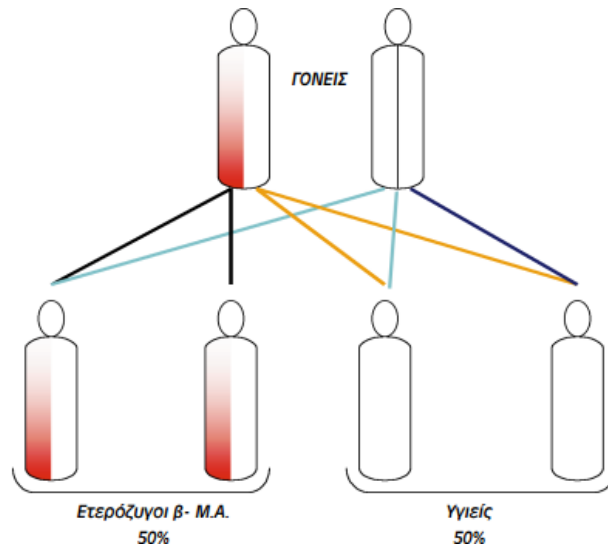
⁴ Πίνακας από google

είναι μια λιγότερο σοβαρή μορφή της β - θαλασσαιμίας. Τα άτομα με ενδιάμεση θαλασσαιμία δεν χρειάζονται μεταγγίσεις αίματος.

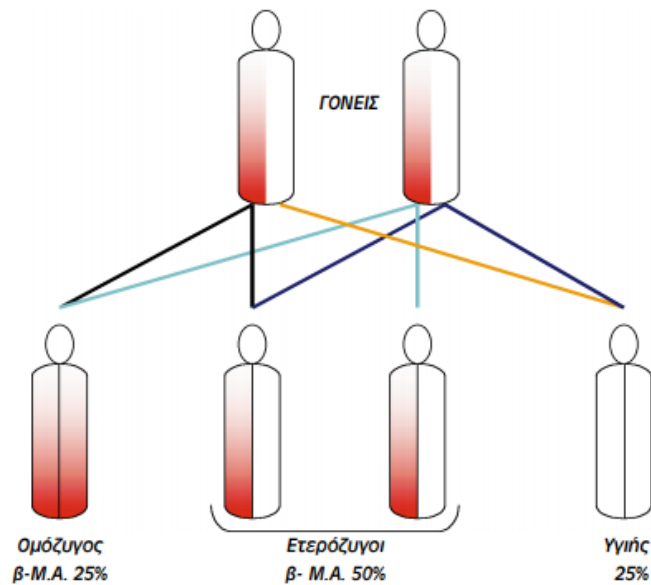
Η **α - θαλασσαιμία** έχει επίσης δύο σοβαρές μορφές, οι οποίες είναι η νόσος **αιμοσφαιρίνη Η (HbH)** και ο **εμβρυϊκός ύδρωπας**. Η νόσος HbH μπορεί να προκαλέσει προβλήματα οστών, τα μάγουλα, το μέτωπο και το σαγόνι μπορεί να είναι διογκωμένα. (δείτε «*Εικόνα 12: Ασθενής με μεσογειαική αναιμία*»). Επιπλέον, η νόσος αυτή πολλές φορές προκαλεί ίκτερο, ο οποίος είναι κιτρίνισμα του δέρματος ή του λευκού των ματιών, μια εξαιρετικά διευρυμένη σπλήνα καθώς και υποσιτισμό. Ο εμβρυϊκός ύδρωπας είναι μια αρκετά σοβαρή μορφή θαλασσαιμίας. Συνήθως συμβαίνει πριν από τη γέννηση, τα περισσότερα άτομα με αυτή την κατάσταση είναι είτε νεκρά ή πεθαίνουν σύντομα μετά τη γέννησή τους.

Για τη διάγνωση της θαλασσαιμίας, ο γιατρός σας θα πάρει ένα δείγμα αίματος και θα το στείλει σε ένα εργαστήριο ώστε να ελεγχθεί για αναιμία και να δει αν υπάρχει μη φυσιολογική αιμοσφαιρίνη. Το δείγμα θα εξετασθεί κάτω από ένα μικροσκόπιο για να διαπιστωθεί αν τα ερυθρά αιμοσφαίρια έχουν περίεργο σχήμα. Το ασυνήθιστο σχήμα στα ερυθρά αιμοσφαίρια είναι ένα σημάδι της θαλασσαιμίας. Συνήθως εκτελείται και μια άλλη εξέταση η οποία είναι γνωστή ως **ηλεκτροφόρηση αιμοσφαιρίνης**. Αυτή η εξέταση διαχωρίζει τα διάφορα μόρια των ερυθρών αιμοσφαιρίων και επιτρέπει να προσδιοριστούν μόρια ανώμαλου τύπου. Ανάλογα με τον τύπο και τη σοβαρότητα της θαλασσαιμίας, μια φυσική εξέταση μπορεί επίσης να βοηθήσει το γιατρό σας να κάνει τη διάγνωση. Για παράδειγμα, μια εξαιρετικά διευρυμένη σπλήνα μπορεί να είναι σημάδι της νόσου HbH.

Η θεραπεία για τη θαλασσαιμία εξαρτάται από τον τύπο και τη σοβαρότητα της νόσου, ο γιατρός σας θα κρίνει ποια θεραπεία θα λειτουργήσει καλύτερα για την περίπτωσή σας. Ορισμένες θεραπείες περιλαμβάνουν, συχνές μεταγγίσεις αίματος, μεταμόσχευση μυελού των οστών (BMT), φάρμακα και συμπληρώματα είτε κάποια πιθανή χειρουργική επέμβαση για την αφαίρεση του σπλήνα ή της χοληδόχου κύστης. Ίσως ο γιατρός κρίνει πως δεν πρέπει να λαμβάνεται βιταμίνες ή συμπληρώματα που περιέχουν σίδηρο, αυτό ισχύει στην περίπτωση που χρειάζεστε μετάγγιση αίματος. Οι άνθρωποι που κάνουν μεταγγίσεις αίματος λαμβάνουν επιπλέον σίδηρο και το σώμα δεν μπορεί εύκολα να απαλλαγεί από αυτό. Ο σίδηρος μπορεί να συσσωρεύεται σε ιστούς κάτι το οποίο μπορεί να είναι δυνητικά θανατηφόρο. Μπορεί επίσης να χρειαστείτε θεραπεία αποσιδήρωσης αν κάνετε μεταγγίσεις αίματος. Αυτό γίνεται με την χορήγηση ενός χημικού σε ενέσιμη μορφή το οποίο δεσμεύει το σίδηρο και άλλα βαρέα μέταλλα και βοηθά στην αφαίρεση του επιπλέον σιδήρου από το σώμα.



Εικόνα 10:πιθανότητα πάθησης τέκνων όταν ο ένας γονέας είναι υγιής και ο άλλος φορέας β μεσογειακής αναιμίας



Εικόνα 11: πιθανότητα πάθησης τέκνων όταν και οι δύο γονείς είναι φορείς β μεσογειακής αναιμίας

Οι άνθρωποι που έχουν ήπιες μορφές θαλασσαιμίας μπορεί τυπικά να ζήσουν μια φυσιολογική ζωή. Διαφορετικές μορφές θαλασσαιμίας μπορεί να είναι ήπιες έως σοβαρές. Σε σοβαρές περιπτώσεις, η καρδιακή ανεπάρκεια είναι ένας κίνδυνος. Ο γιατρός σας μπορεί να σας δώσει περισσότερες πληροφορίες σχετικά με τις προοπτικές σας και να εξηγήσει πώς οι θεραπείες μπορεί να βοηθήσουν στη βελτίωση τη ζωή σας ή στην αύξηση της αναμενόμενης διάρκειας ζωής σας [3],[7].



Εικόνα 12: Ασθενής με μεσογειακή αναιμία⁵

ΣΤ. Δρεπανοκυτταρική αναιμία

Η **δρεπανοκυτταρική αναιμία**, ή δρεπανοκυτταρική νόσος (**SCD**), είναι μια γενετική ασθένεια των ερυθρών αιμοσφαιρίων (RBCs). Κανονικά τα ερυθρά αιμοσφαίρια έχουν σχήμα δίσκου έτσι τους δίνει την ευελιξία να ταξιδέψουν μέσα ακόμα και από τα μικρότερα αιμοφόρα αγγεία. Ωστόσο, με αυτή τη νόσο, τα ερυθρά αιμοσφαίρια έχουν μια ανώμαλη μορφή ημισελήνου (" δρεπάνι "). Αυτό τα καθιστά κολλώδη και άκαμπτα. Έτσι παιδεύονται σε μικρά αιμοφόρα αγγεία και δεν επιτρέπουν στο αίμα να φτάσει στα διάφορα μέρη του σώματος και μπορεί να προκαλέσει πόνο και βλάβη των ιστών. Το SCD είναι μια αυτοσωματική υπολειπόμενη κατάσταση. Χρειάζεται δύο αντίγραφα του γονιδίου να έχουν την ασθένεια, αλλιώς αν υπάρχει μόνο ένα αντίγραφο του γονιδίου, λέγεται ότι υπάρχει **δρεπανοκυτταρικό γνώρισμα**.

Τα συμπτώματα της δρεπανοκυτταρικής αναιμίας συνήθως εμφανίζονται σε νεαρή ηλικία, μερικές φορές παρατηρούνται σε μωρά ήδη από 4 μηνών, αλλά γενικά συμβαίνουν γύρω στους 6 μήνες.

Ενώ υπάρχουν πολλοί τύποι SCD, όλοι έχουν παρόμοια συμπτώματα σε διάφορα επίπεδα σοβαρότητας. Κάποια από τα συμπτώματα και τις επιπλοκές που παρουσιάζονται περιλαμβάνουν

- Υπερβολική κόπωση ή ευερεθιστότητα (από αναιμία).
- Σχολαστικότητα (στα μωρά).
- Ενούρηση (από συναφή προβλήματα στα νεφρά).
- Ίκτερο (κιτρίνισμα των ματιών και του δέρματος).
- Οίδημα και πόνος στα χέρια και τα πόδια.
- Συχνές λοιμώξεις.
- Πόνος στο στήθος.

⁵ Πίνακας από google

- Επώδυνος πόνος στα κόκκαλα.
- Επώδυνη και μεγάλη σπλήνα.
- Προβλήματα ανάπτυξης.
- Πληγές στα πόδια.
- Καρδιακά προβλήματα.

Τα παιδιά που κινδυνεύουν από δρεπανοκυτταρική αναιμία είναι αυτά που και οι δύο γονείς φέρουν δρεπανοκυτταρικό γνώρισμα. Η εξέταση της ηλεκτροφόρησης αιμοσφαιρίνης μπορεί να καθορίσει ποιος τύπος δρεπανοκυτταρικής αναιμίας υπάρχει. Οι άνθρωποι από τις περιοχές που έχουν ενδημική ελονοσία είναι πιο πιθανό να είναι φορείς. Δηλαδή κάτοικοι από Αφρική, Ινδία, Μεσόγειο, Σαουδική Αραβία.

Η SCD μπορεί να προκαλέσει σοβαρές επιπλοκές. Αυτές οι επιπλοκές εμφανίζονται όταν τα δρεπανοκυτταρικά κύτταρα μπλοκάρουν τα αγγεία σε διάφορες περιοχές του σώματος. Οι επώδυνες ή κατεστραμμένες εμφράξεις ονομάζονται δρεπανοκυτταρικές κρίσεις.

Η **αναιμία** είναι η έλλειψη ερυθρών αιμοσφαιρίων. Τα δρεπανοειδή κύτταρα εύκολα σπάνε. Αυτό το σπάσιμο εκτός των ερυθροκυττάρων ονομάζεται χρόνια αιμόλυση. Τα ερυθρά αιμοσφαίρια γενικά ζουν για περίπου 120 ημέρες. Τα δρεπανοειδή κύτταρα ζουν για μέγιστο χρονικό διάστημα 10 έως 20 ημερών.

Το **Σύνδρομο Χεριών-Ποδιών** εμφανίζεται όταν το δρεπανοειδές σχήμα των ερυθρών αιμοσφαιρίων μπλοκάρει τα αιμοφόρα αγγεία στα χέρια ή τα πόδια, αυτό προκαλεί διόγκωση στα χέρια και τα πόδια. Μπορεί επίσης να προκαλέσει έλκη. Τα πρησμένα χέρια/πόδια είναι συχνά το πρώτο σημάδι της δρεπανοκυτταρικής αναιμίας σε μωρά. Η **Παγίδευση Σπλήνα** είναι μια απόφραξη των αγγείων του σπλήνα από δρεπανοειδή κύτταρα και προκαλεί μια ξαφνική, επώδυνη διόγκωση του σπλήνα.

Καθυστερημένη ανάπτυξη συχνά συμβαίνει σε άτομα με SCD. Τα παιδιά είναι γενικά πιο κοντά, αλλά γενικά ανακτούν το ύψος τους κατά την ενηλικίωση. Η σεξουαλική ωρίμανση μπορεί επίσης να καθυστερήσει. Αυτό συμβαίνει επειδή τα δρεπανοειδή ερυθρά αιμοσφαίρια δεν δύναται να παρέχουν αρκετό οξυγόνο και θρεπτικά συστατικά. **Νευρολογικές επιπλοκές** περιλαμβάνουν επιληπτικές κρίσεις, αιμορραγία στον εγκέφαλο, ή ακόμη και κόμα. Αυτές προκαλούνται από μπλοκαρίσματα του εγκεφάλου, θα πρέπει να αναζητηθεί άμεση θεραπεία. Η **τύφλωση** που προκαλείται από μπλοκαρίσματα στα αιμοφόρα αγγεία των ματιών. Αυτό μπορεί να βλάψει τον αμφιβληστροειδή. **Δερματικά έλκη** στα πόδια μπορεί να προκληθούν αν τα αγγεία έχουν μπλοκαριστεί σε αυτά τα σημεία.

Από τη στιγμή που το SCD παρεμποδίζει την παροχή οξυγόνου στο αίμα, υπάρχει επίδραση στα αιμοφόρα αγγεία. Με τον καιρό, αυτό μπορεί να οδηγήσει σε μια **διευρυμένη καρδιά** και επακόλουθη **καρδιακή νόσο**. Αυτό μπορεί επίσης να οδηγήσει σε πόνο γνωστό και ως σύνδρομο στήθους. Υψηλή αρτηριακή πίεση (υπέρταση) και **εγκεφαλικό επεισόδιο** μπορεί επίσης να αναπτυχθεί.

Ο **Πριαπισμός** είναι μια παρατεταμένη, επώδυνη στύση που μπορεί να παρατηρηθεί σε ορισμένους άνδρες με δρεπανοκυτταρική αναιμία. Αυτό συμβαίνει όταν τα αιμοφόρα αγγεία του πέους μπλοκάρονται, κάτι που μπορεί να οδηγήσει σε ανικανότητα, αν αφεθεί χωρίς θεραπεία.

Οι **χολόλιθοι** είναι μια επιπλοκή που δεν προκαλείται από απόφραξη αγγείου. Αντ'αυτού, έχει προκληθεί από τη διάσπαση των ερυθροκυττάρων. Ένα υποπροϊόν αυτής της κατηγορίας είναι η χολερυθρίνη. Υψηλά επίπεδα χολερυθρίνης μπορεί να οδηγήσουν σε πέτρες στη χολή. Αυτό καλείται επίσης **χρωστικές πέτρες**.

Υπάρχουν διάφορες θεραπείες για τη δρεπανοκυτταρική αναιμία, ανάλογα με το είδος της. Αντιβιοτικά όπως η πενικιλίνη μπορούν να δοθούν από την ηλικία των 2 μηνών έως 5

ετών (θα βοηθήσει στην πρόληψη της πνευμονίας). Επίσης παρέχεται συμπληρωματικό οξυγόνο μέσω μιας μάσκας, η οποία κάνει ευκολότερη την αναπνοή και βελτιώνει τα επίπεδα οξυγόνου στο αίμα.

Φαρμακευτική αγωγή χρησιμοποιείται για την ανακούφιση του πόνου κατά τη διάρκεια μιας δρεπανοκυτταρικής κρίσης. Μπορεί να χρειαστεί ναρκωτικά φάρμακα ή ισχυρά παυσίπονα όπως η μορφίνη. Η υδροξουρία (Droxia, Hydrea) βοηθά να αυξηθεί η παραγωγή της εμβρυϊκής αιμοσφαιρίνης και μπορεί να μειώσει τον αριθμό των μεταγγίσεων αίματος. Ορισμένες φορές και εμβολιασμοί, οι οποίοι μπορούν να βοηθήσουν στην πρόληψη των λοιμώξεων, διότι αυτοί οι ασθενείς τείνουν να έχουν χαμηλότερη ανοσία.

Μια χειρουργική επέμβαση μπορεί να γίνει για τη θεραπεία της επίμονης και επώδυνης στύσης. Επίσης, η μεταμόσχευση μυελού των οστών έχει χρησιμοποιηθεί για τη θεραπεία αυτού του είδους της αναιμίας. Οι καλύτεροι υποψήφιοι είναι παιδιά ηλικίας κάτω των 16 ετών, οι οποίοι έχουν σοβαρές επιπλοκές και έχουν βρει ένα συμβατό δότη.

Οι μεταγγίσεις αίματος μπορούν επίσης να είναι ένας τρόπος αντιμετώπισης της δρεπανοκυτταρικής αναιμίας. Τα συσκευασμένα ερυθρά κύτταρα απομακρύνονται από το αίμα του δότη και χορηγείται σε ασθενείς. Αυτό βελτιώνει τη μεταφορά οξυγόνου και θρεπτικών συστατικών.

Η δρεπανοκυτταρική αναιμία είναι μια κληρονομική νόσος. Μιλήστε με έναν σύμβουλο γενετικής αν ανησυχείτε ότι μπορεί να είστε φορέας. Αυτό μπορεί να σας βοηθήσει να καταλάβετε τις πιθανές θεραπείες, τα προληπτικά μέτρα, και τις αναπαραγωγικές επιλογές [3],[7].

6. Σχεδιαστική περιγραφή του Διαγνωστικού Συστήματος Αναιμίας (ΔιΣΑ)

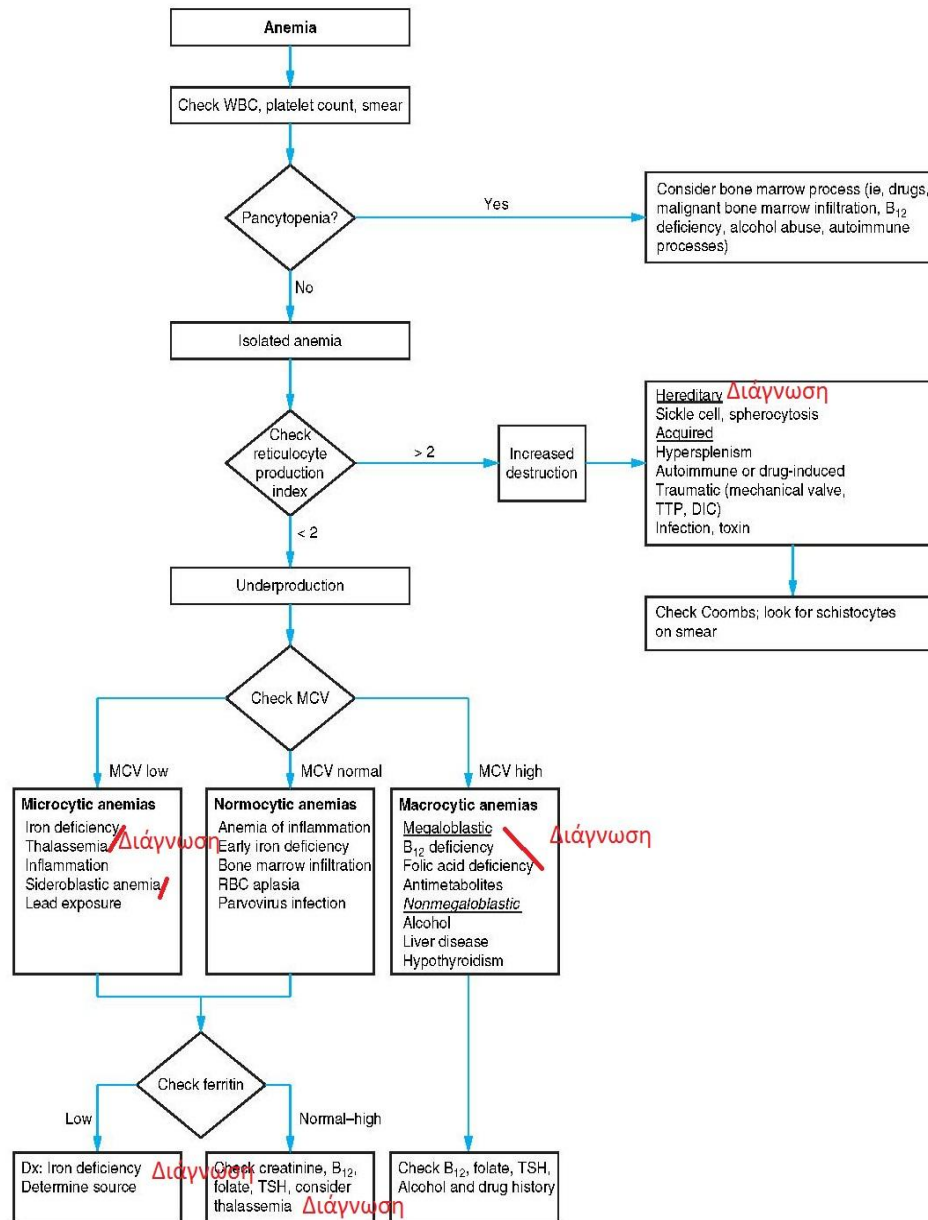
6.1. Αναπαράσταση του μηχανισμού διάγνωσης ως δέντρο αποφάσεων.

Υπήρξαν φορές που αισθανθήκατε κάποια αδυναμία, μήπως έντονο πονοκέφαλο και υπνηλία και σκεφτήκατε πως ίσως πάσχετε από κάποια ασθένεια; Σκεφτήκατε ποτέ πως ο γιατρός προβαίνει στη διάγνωση της; Με τη διάγνωση ο γιατρός αναγνωρίζει την ασθένεια, βλέποντας την εικόνα που σχηματίζουν τα συμπτώματα και τα ευρήματα που δίνει η σωματική εξέταση. Οι εργαστηριακές εξετάσεις μπορούν να επιβεβαιώσουν αυτή του τη γνώμη. Έτσι, με παρόμοιο τρόπο και το σύστημα κάνει τη διάγνωση. Όπως όταν επισκέπτεσθε κάποιο ιατρείο και ο γιατρός σας κάνει κάποιες ερωτήσεις για τα συμπτώματα που εμφανίζεται και ψάχνει να βρει **ίχνη-κλειδιά**, ή εάν του προσκομίσετε εξετάσεις που έχετε, τις ελέγχει και βγάζει συμπέρασμα έτσι και σ' ένα διαγνωστικό σύστημα ο κάθε χρήστης πρέπει να συμπληρώσει και να επιλέξει κάποια στοιχεία, αυτά τα στοιχεία περνάνε ως δεδομένα στο σύστημα της Prolog, η οποία με τη σειρά της, με τους κατάλληλους αλγορίθμους τα επεξεργάζεται, τα αναλύει και καταλήγει στο αποτέλεσμα της διάγνωσης αυτής.

Η διάγνωση που προηγείται της θεραπευτικής αγωγής διαφέρει πολύ από τις διάφορες ιατρικές συμβουλές που μπορεί να μας δώσουν ακόμα και ειδικοί. Στη διάγνωση, αφού πρώτα εντοπισθεί το είδος της ασθένειας θα μπορεί εάν είναι ήπιας μορφής να αντιμετωπισθεί με την βελτίωση της διατροφής και πολύ πιθανό βέβαια να χορηγηθούν και κάποια συμπληρώματα διατροφής ή χορήγηση βιταμινών και μετάλλων, για την ενίσχυση του οργανισμού, εάν είναι πιο βαριάς μορφής θα χρειαστεί πιο σοβαρή αντιμετώπιση. Ενώ, οι ιατρικές συμβουλές δίνονται ως χρήσιμη γνώση για πιθανή αξιοποίηση της.

Σε αυτό το σημείο αξίζει να σημειωθεί πως αυτό το εργαλείο δεν παρέχει ιατρικές συμβουλές και πως δεν αντικαθιστά σε καμία περίπτωση την επαγγελματική ιατρική συμβουλή, είναι απλώς ένα βοηθητικό πρόγραμμα διάγνωσης, το οποίο έχει ως στόχο να διευκολύνει και να ελαφρύνει τα 'χέρια' του εμπειρογνώμονα-ειδικού-γιατρού.

Η διάγνωση του συστήματος ΔιΣΑ στηρίχθηκε σε γνώση την οποία απέσπασα μέσω συνεντεύξεων με γιατρούς και από βιβλιογραφία. Η κύρια βιβλιογραφία που χρησιμοποιήθηκε είναι το βιβλίο του [4]. Συγκεκριμένα από αυτό το βιβλίο χρησιμοποιήθηκε το παρακάτω διαγνωστικό διάγραμμα.



DIC, disseminated intravascular coagulation; MCV, mean corpuscular volume; TTP, thrombotic thrombocytopenic purpura.

Εικόνα 13: Διάγραμμα

DIC: Disseminated Intravascular Coagulation (Διάχυτη ενδοαγγειακή πήξη)

MCV: Mean corpuscular volume (μέσος όγκος ερυθρών)

TTP: Thrombotic Thrombocytopenic Purpura (Θρομβωτική θρομβοπενική πορφύρα)

Το διάγραμμα από το βιβλίο [4] το αλλάξαμε προσθέτοντας λεπτομέρειες και μετατρέποντας το σε δέντρο αποφάσεων.

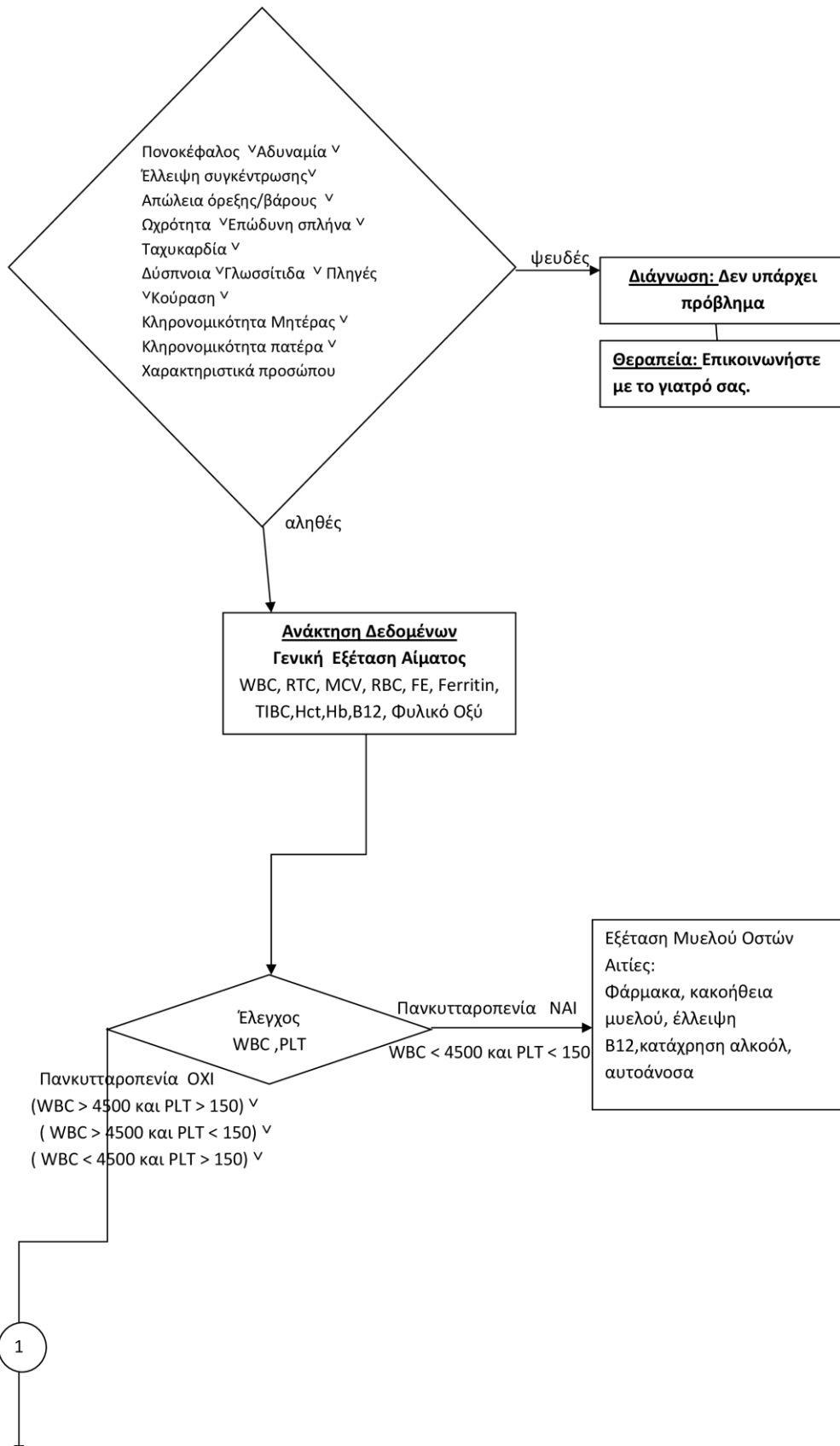
Ο πιο απλός τρόπος για να παρουσιάσουμε τη διαγνωστική συλλογιστική ενός συστήματος γνώσης ώστε να είναι κατανοητή από ένα μη ειδικό σε συστήματα γνώσης είναι το δέντρο αποφάσεων.

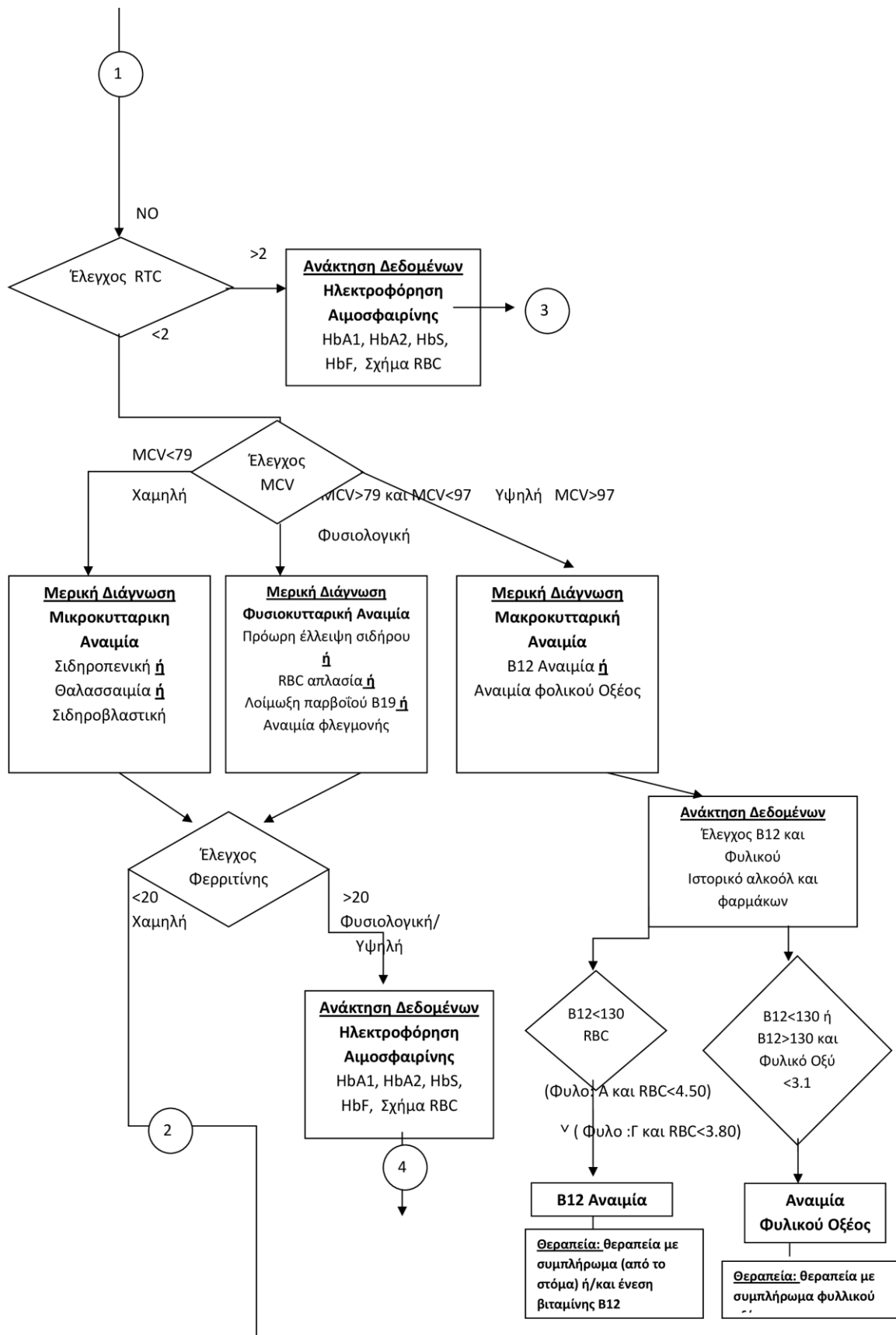
Ένα δένδρο απόφασης είναι ένα εργαλείο υποστήριξης αποφάσεων το οποίο χρησιμοποιεί ένα δενδροειδές μοντέλο αποφάσεων και τις πιθανές συνέπειές τους, συμπεριλαμβανομένων και των αποτελεσμάτων. Ο δενδροειδής γράφος περιγράφει παραστατικά τα γεγονότα, δεδομένα του προβλήματος, που μπορεί να συμβούν ή τις αποφάσεις που πρέπει να γίνουν καθώς και τα αποτελέσματα που σχετίζονται με τους συνδυασμούς των γεγονότων ή των αποφάσεων. Τα γεγονότα και οι τιμές καθορίζονται στο δενδροειδή γράφο για κάθε αποτέλεσμα.

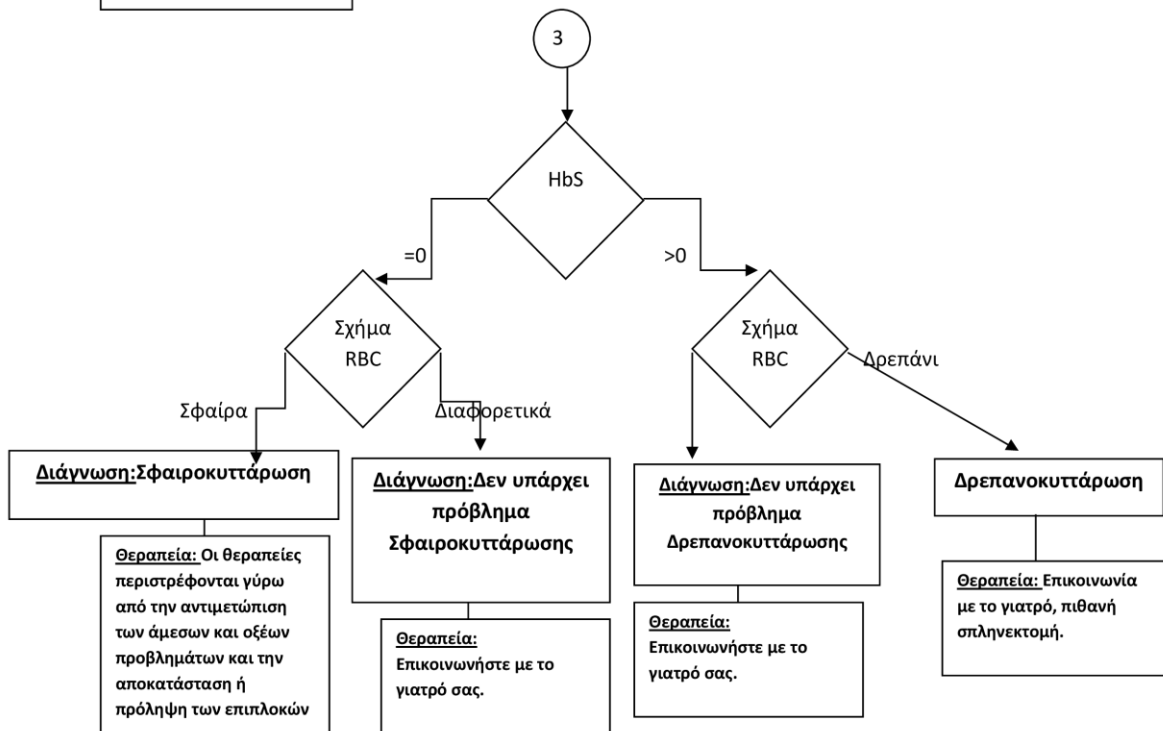
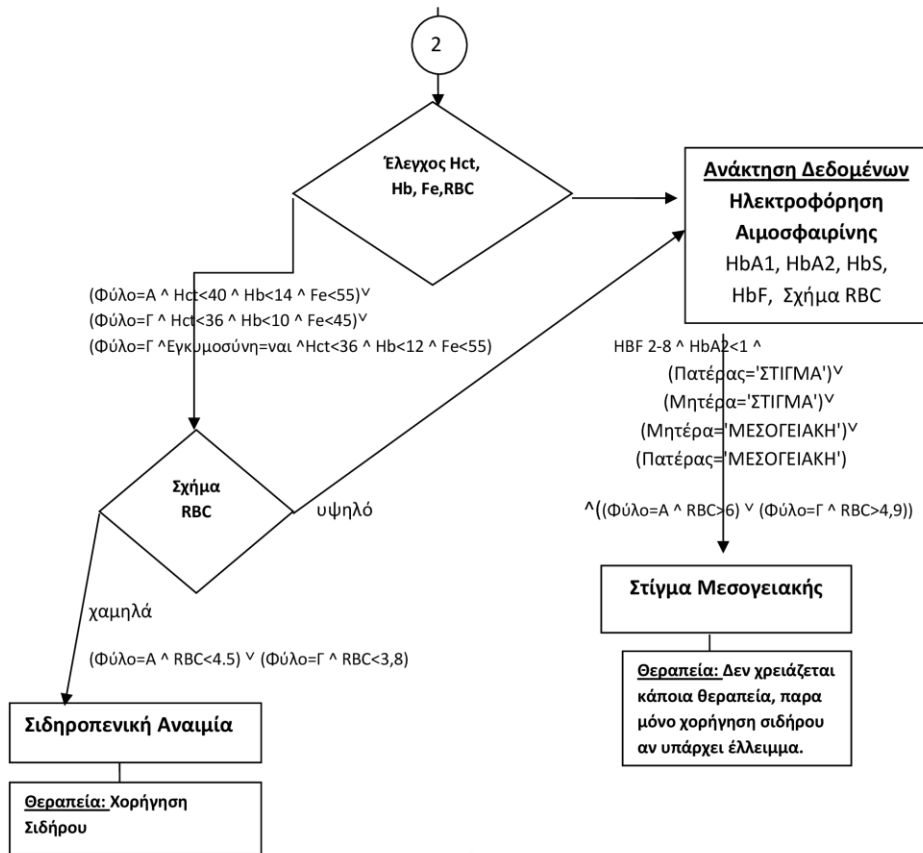
Τα μοντέλα δέντρων απόφασης περιλαμβάνουν έννοιες όπως οι *κόμβοι*, *κλαδιά*, και *φύλλα* τα οποία αναφέρονται και ως *τελικές τιμές* ή *τελικοί κόμβοι*. Οι *εσωτερικοί κόμβοι* είναι *ιδιότητες* ή *καταστάσεις* ή *γεγονότα* που παριστάνονται με μεταβλητές. Τα *κλαδιά* ενός εσωτερικού κόμβου αντιστοιχούν στις διαφορετικές τιμές που υπάρχουν στο πεδίο της ιδιότητας ή της κατάστασης ή του γεγονότος του εσωτερικού κόμβου. Τα *φύλλα* του δένδρου αντιστοιχούν στο αποτέλεσμα που προκύπτει όταν οι μεταβλητές των ιδιοτήτων ή της κατάστασης ή των γεγονότων στο μονοπάτι, από τη ρίζα προς το φύλο, πάρουν αντίστοιχες τιμές [2].

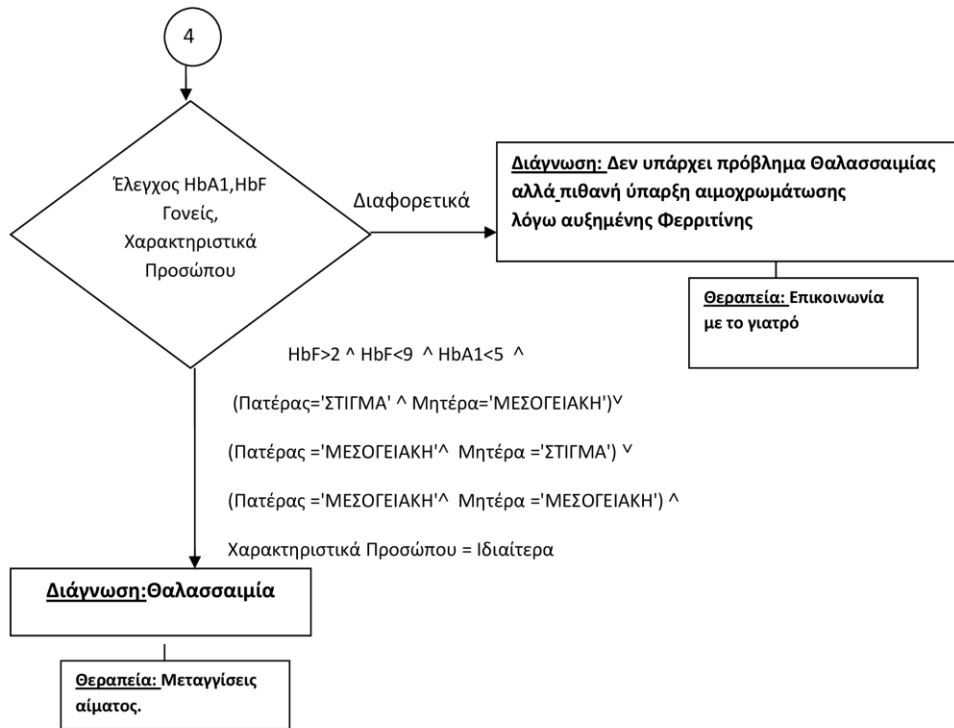
Στο παρακάτω δέντρο αποφάσεων [4] φαίνεται η λογική η οποία ακολουθήθηκε και στην οποία βασίστηκε η δημιουργία των κανόνων για τη διάγνωση της αναιμίας. Με βάση αυτό το διάγραμμα δημιουργήθηκαν 16 κανόνες (rules) οι οποίοι φαίνονται στην ενότητα 14. Η επεξήγηση των συμβόλων του διαγράμματος είναι η εξής:

1. \vee σημαίνει λογικό OR
2. \wedge σημαίνει λογικό AND
3. Τα ορθογώνια σημαίνουν εκτέλεση κάποιας διαδικασίας, επεξεργασία των δεδομένων.
4. Οι ρόμβοι σημαίνουν ότι εκτελείται κάποια απόφαση (εκτέλεση λογικής παράστασης) η οποία ανάλογα οδηγεί σε διαφορετικά μονοπάτια.



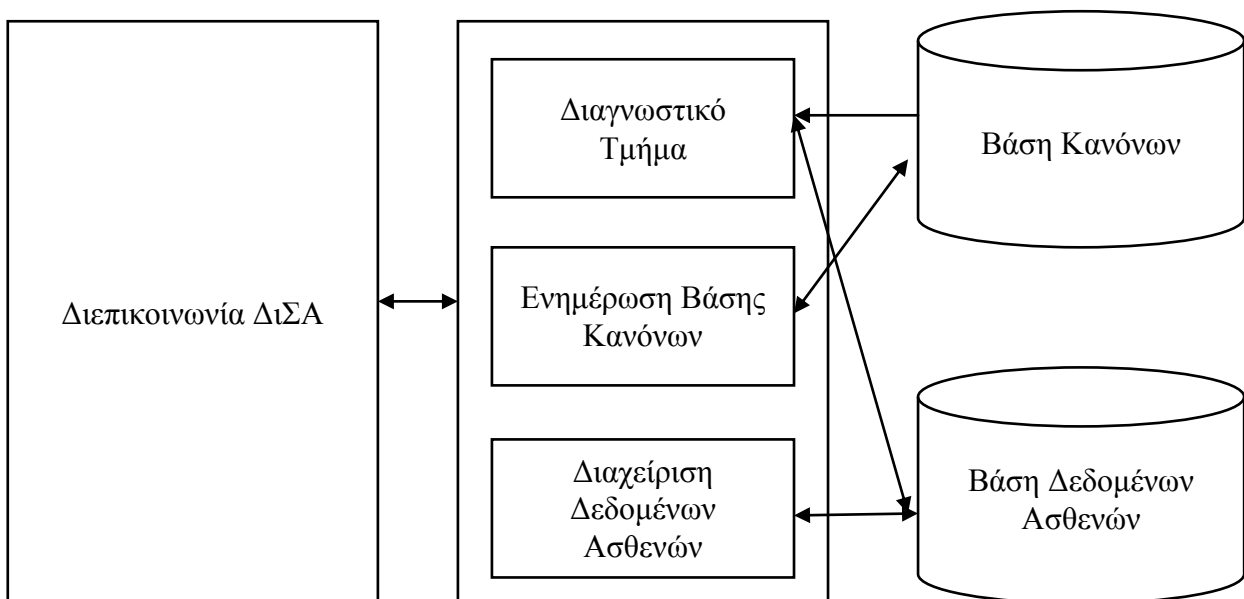






6.2. Αρχιτεκτονική περιγραφή του ΔιΣΑ

Για την παραγωγή της αρχιτεκτονικής του συστήματος λαμβάνονται υπόψη κάποιες δεσμεύσεις, δηλαδή, οι απαιτήσεις των χρηστών οι οποίοι θέλουν να βλέπουν ένα περιβάλλον φιλικό προς αυτούς, κατανοητό και εύχρηστο, η τεχνολογία που θα χρησιμοποιηθεί και το μοντέλο γνώσης. Το μοντέλο γνώσης κωδικοποιείται χρησιμοποιώντας εργαλεία ανάπτυξης έμπειρων συστημάτων, η αναπαράσταση γνώσης και το διαγνωστικό μέρος του συστήματος ΔιΣΑ έχει υλοποιηθεί στη γλώσσα προγραμματισμού Prolog και η διεπαφή έχει υλοποιηθεί σε Java. Το σύστημα δεν χρησιμοποιεί βάρη ούτε πιθανότητες διότι η διάγνωση στηρίζεται σε εργαστηριακά δεδομένα τα οποία είναι έγκυρα και ασφαλή, αυτά με τη σειρά τους κατευθύνουν τη διάγνωση με ασφάλεια. Τα βάρη και οι παράγοντες βεβαιότητας χρησιμοποιούνται όταν η διάγνωση στηρίζεται σε συμπτώματα και οι εργαστηριακές εξετάσεις δεν μπορούν με ασφάλεια να οδηγήσουν σε συγκεκριμένη διάγνωση. Στο παρακάτω διάγραμμα φαίνεται η αρχιτεκτονική του συστήματος που υλοποιήθηκε.



Εικόνα 14: Αρχιτεκτονική Συστήματος ΔιΣΑ

Η περιγραφή των τμημάτων της αρχιτεκτονικής έχει ως εξής:

1. **Διεπικοινωνία ΔιΣΑ.** Αυτό το τμήμα διαχειρίζεται τη διεπαφή του χρήστη με το ΔιΣΑ. Ο χρήστης έχει τη δυνατότητα να εκτελέσει τρεις εργασίες.

➤ **Διάγνωση**

Η διάγνωση μπορεί να γίνει είτε σε νέο χρήστη είτε σε παλιό χρήστη. Συμπληρώνοντας τα απαραίτητα πεδία γίνεται η επικοινωνία με το διαγνωστικό σύστημα της Prolog και τη βάση κανόνων και προκύπτει η ανάλογη διάγνωση.

➤ **Ενημέρωση Βάσης Κανόνων ΔιΣΑ**

Χρήστες που είναι είτε εμπειρογνώμονες ή ο μηχανικός γνώσης έχουν τη δυνατότητα ενημέρωσης της βάσης κανόνων. Ένας τέτοιος χρήστης μπορεί να προσθέσει, να αφαιρέσει ή να αλλάξει ένα κανόνα του ΔιΣΑ.

➤ **Αναζήτηση Ιστορικού Ασθενούς**

Παρέχεται επίσης η δυνατότητα ανάκτησης παλαιών δεδομένων ενός ασθενή επικοινωνώντας με τη βάση δεδομένων των ασθενών.

2. Διαγνωστικό τμήμα.

Αυτό το τμήμα περιέχει τη διαγνωστική μηχανή του ΔιΣΑ που εξετάζει και πυροδοτεί τους κατάλληλους κανόνες για να γίνει η διάγνωση. Επικοινωνεί με τη βάση κανόνων και με βάση τα ιατρικά δεδομένα του ασθενή επιλέγει κανόνες και τους πυροδοτεί. Τα δεδομένα του ασθενή ρυθμίζουν την πυροδότηση των κανόνων. Δηλαδή ένας κανόνας εκτελείται εφόσον τα δεδομένα του ασθενή ικανοποιούν τους υποστόχους που υπάρχουν στο σώμα του κανόνα.

3. Ενημέρωση Βάσης Κανόνων.

Αυτό το τμήμα διαχειρίζεται την ενημέρωση της βάσης των κανόνων. Δηλαδή, υποστηρίζει τη δημιουργία νέων κανόνων, τη διαγραφή και την τροποποίηση των υπάρχοντων κανόνων του ΔιΣΑ.

4. Διαχείριση δεδομένων ασθενών.

Αυτό το τμήμα διαχειρίζεται τα δεδομένα των χρηστών/ασθενών. Ο χρήστης αυτού του τμήματος μπορεί να είναι ο εμπειρογνώμονας, δηλαδή ο γιατρός. Του δίνεται η δυνατότητα να προσθέσει ένα νέο πελάτη/ασθενή, να ανακτήσει τα δεδομένα ενός πελάτη/ασθενή από τις προηγούμενες επισκέψεις του.

5. Βάση κανόνων.

Περιέχει όλους τους κανόνες που χρησιμοποιεί το ΔιΣΑ για τη διάγνωση και θεραπεία των ασθενών.

Βάση δεδομένων ασθενών.

Περιέχει όλα τα απαραίτητα δεδομένα κάθε πελάτη/ασθενή του γιατρού (προσωπικά στοιχεία, συμπτώματα, εξετάσεις, διάγνωση, θεραπεία).

6.3. Περιγραφή της διεπαφής του ΔιΣΑ

Όπως προαναφέρθηκε η τεχνολογία που θα χρησιμοποιηθεί είναι αυτή της Java. Το σύστημα αποτελείται από 12 διαφορετικά παράθυρα, το καθένα από τα οποία έχει το δικό του ρόλο. Τα παράθυρα αυτά είναι τα εξής:

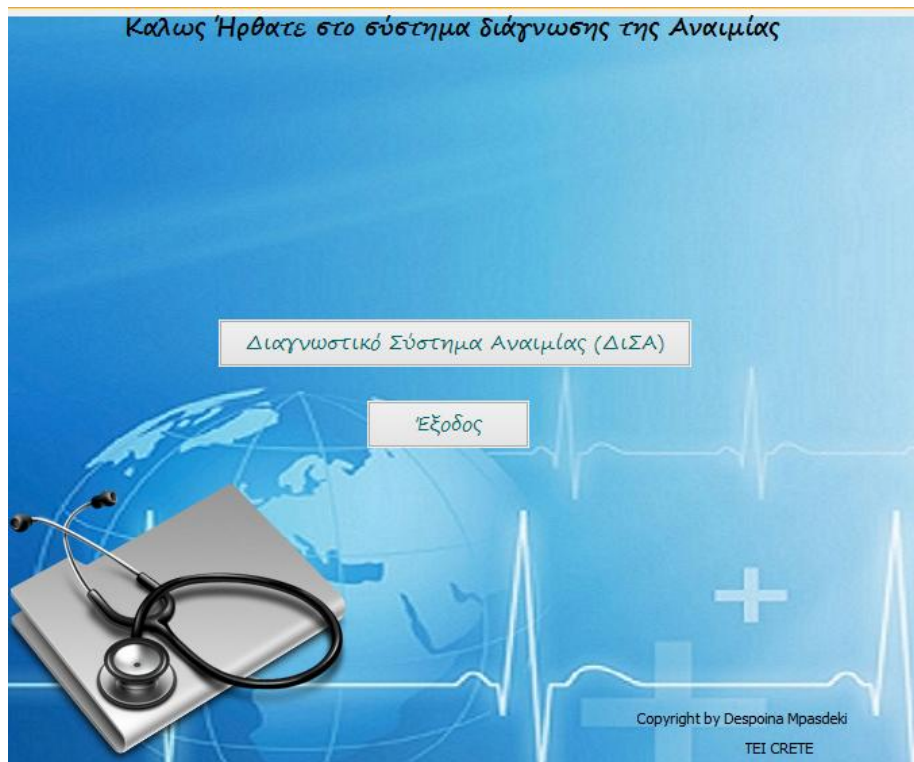
- Το πρώτο παράθυρο περιέχει την αρχική οθόνη υποδοχής του χρήστη και είναι υπεύθυνο για το έναυσμα του προγράμματος. Επιλέγοντας τη δραστηριότητα «Διαγνωστικό Σύστημα Αναιμίας» του ΔιΣΑ («*Εικόνα 15: Αρχική Οθόνη ΔιΣΑ*»), ο χρήστης μεταφέρεται στο επόμενο παράθυρο («*Εικόνα 2: Πάνελ επιλογών ΔιΣΑ*»). Σε αυτό το παράθυρο υπάρχουν οι 3 βασικές λειτουργίες του ΔιΣΑ που αναφέρθηκαν παραπάνω (Ενότητα 2.4).
- Το δεύτερο παράθυρο («*Εικόνα 2: Πάνελ επιλογών ΔιΣΑ*») το οποίο δίνει τη δυνατότητα στους χρήστες (εμπειρογνώμονας ή μηχανικός γνώσης), να επιλέξουν μια από τις εξής τρεις λειτουργίες: 1) Εάν ο εμπειρογνώμονας θέλει να προχωρήσει στη διάγνωση. 2) Εάν ο εμπειρογνώμονας θέλει να κάνει αναζήτηση δεδομένων δια πιθανή προηγούμενη διάγνωσης ώστε να δει τις προηγούμενες διαγνώσεις και τα αντίστοιχα δεδομένα του ασθενή. 3) Ο εμπειρογνώμονας ή ο μηχανικός γνώσης να κάνει ενημέρωση της βάσης γνώσης, δηλαδή του κανόνες, του συστήματος. Στην τελευταία περίπτωση, η αναζήτηση

γίνεται με βάση το ονοματεπώνυμο του ασθενή, εμφανίζονται στην οθόνη όλα τα παλιά δεδομένα που σχετίζονται με αυτό το όνομα. Μετά την αναζήτηση, ο γιατρός έχει τη δυνατότητα να προβεί σε νέα διάγνωση.

- Τα επόμενα παράθυρα (πχ «*Εικόνα 16: Παράθυρο Συμπτωμάτων*») αφορούν τη διάγνωση, περιέχουν πεδία συμπλήρωσης των στοιχείων του πελάτη ασθενή, όπως για παράδειγμα ονοματεπώνυμο, ηλικία, διεύθυνση και τα λοιπά, αποτελέσματα καθώς και κάποια συμπτώματα τα οποία πιθανόν να έχει.
- Στο τελευταίο παράθυρο («*Εικόνα 17: Παράθυρο Διάγνωσης*»), γίνεται η διάγνωση επιλέγοντας τη διαραστηριότητα «Διάγνωση». Το αποτέλεσμα της διάγνωσης εμφανίζεται στο πεδίο με τίτλο Διάγνωση, αφού πρώτα φορτωθεί το αρχείο που είναι υπεύθυνο για τη διάγνωση (επεξήγηση στο Κεφάλαιο 7).

Μετά την διάγνωση υπάρχει η δυνατότητα είτε του τερματισμού του προγράμματος, είτε - εάν ο χρήστης το επιθυμεί- της επανεκτέλεσης του. Το πρόγραμμα δίνει και άλλη μία επιπρόσθετη λειτουργία, αυτή της θεραπείας. Η δυνατότητα αυτή μπορεί να εκτελεσθεί μετά τη διαδικασία της διάγνωσης, διότι η θεραπεία είναι αντίστοιχη της διάγνωσης.

Στην περίπτωση που ο χρήστης θέλει να κάνει ενημέρωση της βάσης κανόνων καλείτε να επιλέξει ανάμεσα σε τρεις επιλογές, α) της ανανέωσης β) της διαγραφής ενός υπάρχοντος κανόνα γ) της δημιουργίας ενός νέου κανόνα («*Εικόνα 19: Παράθυρο ενημέρωσης βάσης κανόνων*»). Για να μπορέσει να κάνει ενημέρωση κανόνων χρειάζεται να συμπληρώσει πεδία που του ζητούνται (για παράδειγμα όπως φαίνεται στην «*Εικόνα 31: Δεδομένα για αλλαγή κανόνα*» χρειάζεται ο κωδικός του και το σώμα του κανόνα). Παρακάτω παρατίθενται ορισμένα από αυτά τα παράθυρα. Πρέπει να σημειωθεί επίσης πως σε κάθε παράθυρο, υπάρχει η δυνατότητα οπισθοδρόμησης σε προηγούμενο παράθυρο.



Εικόνα 15: Αρχική Οθόνη ΔιΣΑ

ΕΠΙΛΕΞΤΕ ΤΑ ΣΥΜΠΤΩΜΑΤΑ:

<input type="checkbox"/> ΠΟΝΟΚΕΦΑΛΟΣ	<input type="checkbox"/> ΑΙΣΘΗΜΑ ΚΡΥΟΥ	<input type="checkbox"/> ΩΧΡΟΤΗΤΑ ΔΕΡΜΑΤΟΣ	<input type="checkbox"/> ΑΠΩΛΕΙΑ ΒΑΡΟΥΣ
<input type="checkbox"/> ΖΑΛΑΔΑ	<input type="checkbox"/> ΕΠΩΔΥΝΗ ΣΠΛΗΝΑ	<input type="checkbox"/> ΔΥΣΠΝΟΙΑ	<input type="checkbox"/> ΠΛΗΓΕΣ
<input type="checkbox"/> ΤΑΧΥΚΑΡΔΙΑ	<input type="checkbox"/> ΑΔΥΝΑΜΙΑ	<input type="checkbox"/> ΑΠΩΛΕΙΑ ΟΡΕΞΗΣ	<input type="checkbox"/> ΠΟΝΟΣ ΣΤΑ ΑΚΡΑ
<input type="checkbox"/> ΚΟΥΡΑΣΗ	<input type="checkbox"/> ΠΛΗΓΕΣ ΓΛΩΣΣΑΣ	<input type="checkbox"/> ΠΡΟΒΛΗΜΑΤΑ ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ	<input type="checkbox"/> ΕΛΛΕΙΨΗΣ ΔΙΑΤΡΟΦΗΣ
<input type="checkbox"/> ΦΑΡΜΑΚΑ	<input type="checkbox"/> ΑΛΚΟΟΛ	ΕΞΩΤΕΡΙΚΗ ΟΨΗ ΠΡΟΣΩΠΟΥ	
		<input type="radio"/> ΚΑΝΟΝΙΚΗ	
		<input type="radio"/> ΙΔΙΑΙΤΕΡΗ	
ΓΟΝΕΑΣ 1		<input type="radio"/> ΟΧΙ	ΓΟΝΕΑΣ 2
		<input type="radio"/> ΣΤΙΓΜΑ	<input type="radio"/> ΣΤΙΓΜΑ
		<input type="radio"/> ΜΕΣΟΓΕΙΑΚΗ	<input type="radio"/> ΜΕΣΟΓΕΙΑΚΗ

Εικόνα 16: Παράθυρο Συμπτωμάτων

Καλως Ήρθατε στο σύστημα διάγνωσης της Αναιμίας

ΑΠΟΤΕΛΕΣΜΑ

0%

ΦΟΡΤΩΣΗ...

Εικόνα 17: Παράθυρο Διάγνωσης



Εικόνα 18:Παράθυρο Αναζήτησης



Εικόνα 19:Παράθυρο ενημέρωσης βάσης κανόνων

Εάν ο ασθενής κάνει την πρώτη του επίσκεψη στο γιατρό, ο γιατρός θα πρέπει να καταχωρήσει το νέο πελάτη στο διαγνωστικό σύστημα, η επιλογή που πρέπει να επιλέξει τη δραστηριότητα «Διάγνωση». Επιλέγοντας τη λειτουργία της «Διάγνωσης», ο εμπειρογνώμονας (χρήστης), μεταφέρεται σε μια αλληλουχία παραθύρων, που περιλαμβάνει ένα περιβάλλον στο οποίο αρχικά ο χρήστης εισάγει τα προσωπικά δεδομένα του πελάτη/ασθενή, όπως ονοματεπώνυμο, ηλικία, φύλο διεύθυνση και τα λοιπά. Έπειτα εισάγει τα συμπτώματα, επιλέγοντας αυτά που έχει από τη λίστα που εμφανίζεται. Στη συνέχεια καλείται να συμπληρώσει κάποια πεδία τα οποία είναι τα αποτελέσματα από τις ιατρικές εξετάσεις τις οποίες πρέπει ήδη να έχει κάνει ο ασθενής. Το πρώτο παράθυρο («Εικόνα 20: Παράθυρο εξετάσεων») που θα εμφανιστεί περιέχει τα πεδία τα οποία πρέπει να συμπληρωθούν με δεδομένα από απλό αιματολογικό έλεγχο (checkup) με επιπλέον την εξέταση Σιδήρου και Φερριτίνης. Αυτά τα εργαστηριακά δεδομένα είναι αναγκαία για τη διάγνωση απλών μορφών της Αναιμίας, όπως η Σιδηροπενική Αναιμία.

Εικόνα 20: Παράθυρο εξετάσεων

Σε περίπτωση που ο ασθενής έχει πραγματοποιήσει επιπλέον αιματολογικές εξετάσεις οι οποίες περιλαμβάνουν τη βιταμίνη B12 ή έξτρα τιμές της αιμοσφαιρίνης που σχετίζονται με κατηγοριοποίηση της όπως HbA1, HbA2, κτλ, τότε επιλέγοντας τη δραστηριότητα «Λοιπές εξετάσεις», στη νέα οθόνη («Εικόνα 24: Παράθυρο έξτρα εξετάσεων») υπάρχουν αντίστοιχα πεδία στα οποία μπορούν να καταχωρηθούν οι τιμές. Τέλος επιλέγοντας τη δραστηριότητα «Διάγνωση» εκτελείται η διαγνωστική διαδικασία του ΔιΣΑ.

ΕΙΣΑΓΕΤΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΞΕΤΑΣΕΩΝ:

Β12 ΒΙΤΑΜΙΝΗ	<input type="text"/>	HbA1	<input type="text"/>
ΦΥΛΙΚΟ ΟΞΥ	<input type="text"/>	HbA2	<input type="text"/>
ΣΧΗΜΑ ΕΡΥΘΡΩΝ	<input type="radio"/> ΦΥΣΙΟΛΟΓΙΚΟ <input type="radio"/> ΔΡΕΠΑΝΟΕΙΔΕΣ <input type="radio"/> ΣΦΑΙΡΙΚΟ	HbF	<input type="text"/>
		HbS	<input type="text"/>



Εικόνα 21: Παράθυρο έξτρα εξετάσεων

Το σύστημα παρέχει τη δυνατότητα να προτείνει θεραπεία, επιλέγοντας την ενέργεια «Θεραπεία». Εάν ο χρήστης/εμπειρογνώμονας έχει ολοκληρώσει τη διάγνωση και θέλει να προχωρήσει σε νέα αρχική δραστηριότητα του ΔιΣΑ πρέπει να επιλέξει τη λειτουργία της Έναρξης που βρίσκεται κάτω δεξιά της τελευταίας οθόνης («Εικόνα 17: Παράθυρο Διάγνωσης»). Διαφορετικά, μπορεί να επιλέξει τη δραστηριότητα «Έξοδος», για τερματισμό του προγράμματος.

Αξίζει να σημειωθεί πως κάθε πεδίο πρέπει να συμπληρώνεται με προσοχή. Δίπλα από κάθε πεδίο εισαγωγής ιατρικών αποτελεσμάτων, υπάρχει ένα μικρό εικονίδια με ετικέτα “i”, το οποίο παρέχει τη δυνατότητα προβολής των φυσιολογικών τιμών κάθε αντίστοιχου πεδίου.

Έστω ότι από το παράθυρο, «Εικόνα 17: Παράθυρο Διάγνωσης», επιλέγουμε τη λειτουργία «Διάγνωση». Σε αυτή την περίπτωση η java τρέχει τον παρακάτω κώδικα. Η πρώτη εντολή αποθηκεύει σε μια συμβολοσειρά (string) το κατηγορημα kb_test με 8 ορίσματα. Τα 6 πρώτα ορίσματα (condata, persdata, symptoms, fbcTest, specBloodTest, electrHbTest) είναι οι λίστες που περιέχουν όλα τα στοιχεία των ασθενών και τα άλλα 2 ορίσματα είναι οι μεταβλητές Res και Ther. Η πρώτη επιστρέφει το αποτέλεσμα της διάγνωσης και η άλλη επιστρέφει το αποτέλεσμα της θεραπείας. Η δεύτερη εντολή, η οποία ανήκει στη βιβλιοθήκη jpl, περνά μέσα από τη συνάρτηση Query(tt) της java, το στόχο που πρέπει να εκτελέσει η Prolog. Η τρίτη εντολή, παίρνει το αποτέλεσμα της διάγνωσης και το εμφανίζει στο αντίστοιχο πεδίο του αποτελέσματος. Το αποτέλεσμα το παίρνει μέσω του qt.oneSolution().get("Res") το μετατρέπει σε συμβολοσειρά, αφαιρεί όλα τα περιττά σύμβολα και στη συνέχεια το εμφανίζει στο πεδίο.

1^η εντολή: String tt = "kb_test(" + condata + "," + persdata + "," + symptoms + "," + fbcTest + "," + specBloodTest + "," + electrHbTest + ",Res,Ther)." + "\n";

2^η εντολή: Query qt = new Query(tt);

3^η εντολή: Diagnosis_txt.append("Μετιςδοθείσες πληροφορίες πάσχετε από: " + "\n")

```
+ "\n" + qt.oneSolution().get("Res").toString().replaceAll("\\[\\]\\\\(\\\\)\\\\\\\\'+", " ").trim());
```

Εάν ο πελάτης είναι ήδη καταχωρημένος στο παρελθόν από το γιατρό, τότε ο γιατρός μπορεί να ανατρέξει σε παλιές διαγνώσεις, πατώντας «Αναζήτηση Ιστορικού Ασθενούς» και εισάγοντας το ονοματεπώνυμό του. Μετά την αναζήτηση θα εμφανιστούν όλα τα δεδομένα που έχουν αποθηκευτεί στο όνομα του ασθενούς, μπορεί δηλαδή να ανατρέξει και να δει τα συμπτώματα που είχε δηλώσει αλλά και τη διάγνωση που του είχε γίνει. Μπορεί επίσης να κάνει νέα διάγνωση για αυτό τον ασθενή επιλέγοντας τη λειτουργία της διάγνωσης.

Για να γίνει η αναζήτηση το interface επικοινωνεί με την Prolog μέσω του κατηγορήματος **search/3**. Σ' αυτό το κατηγορήμα δίνονται ως όρισμα το ονοματεπώνυμο του ασθενή (patient_name) και επιστρέφει τον κωδικό του ασθενή (M) και το ιστορικό του (L). Το παρακάτω τμήμα κώδικα δείχνει τον τρόπο κλήσης του κατηγορήματος **search/3** μέσω της διεπικοινωνίας της Java. Η πρώτη εντολή αποθηκεύει σε μια συμβολοσειρά το όνομα του ασθενή που θέλουμε να αναζητήσουμε. Η δεύτερη εντολή αποθηκεύει σε μια συμβολοσειρά το κατηγορήμα search με 3 ορίσματα. Το πρώτο όρισμα είναι το όνομα του ασθενή (patient_name), το δεύτερο είναι η λίστα με το ιστορικό του ασθενή (L) και το τρίτο είναι ο κωδικός του ασθενή (M). Η τρίτη εντολή περνά μέσα από τη συνάρτηση Query(q7) της Java, το στόχο που πρέπει να εκτελέσει η Prolog. Η τέταρτη και πέμπτη εντολή αποθηκεύει σε μια συμβολοσειρά τη λίστα του ιστορικού και τον κωδικό του ασθενή, αντίστοιχα. Η έκτη εντολή, παίρνει τη συμβολοσειρά στην οποία αποθηκεύσαμε το ιστορικό του ασθενή (search_txt) και το εμφανίζει στο αντίστοιχο πεδίο του αποτελέσματος («Εικόνα 18:Παράθυρο Αναζήτησης»).

1^η εντολή: patient_name = "" + search_pat_txt.getText() + "";

2^η εντολή: String t7 = "search(" + patient_name + ",L,M)." + "\n";

3^η εντολή: Query q7 = new Query(t7);

4^η εντολή: search_txt = q7.oneSolution().get("L").toString();

5^η εντολή: user_id = q7.oneSolution().get("M").toString();

6^η εντολή: search_res.append("Ιστορικό ασθενή: " + "\n"
+ "\n" + search_txt.replaceAll("\\[\\]\\\\(\\\\)\\\\\\\\'+", "").trim());

Η διάγνωση ενός παλιού ασθενή γίνεται με τον ίδιο τρόπο όπως δείξαμε και στην παραπάνω διαδικασία της διάγνωσης. Η πρώτη εντολή αποθηκεύει σε μια συμβολοσειρά (string) το κατηγορήμα kb_search με 9 ορίσματα. Το πρώτο όρισμα είναι η ταυτότητα ενός συγκεκριμένου ασθενή (user_id). Τα υπόλοιπα 6 ορίσματα (condata, persdata, symptoms, fbcTest, specBloodTest, electrHbTest) είναι οι λίστες που περιέχουν όλα τα στοιχεία των ασθενών και τα άλλα 2 ορίσματα είναι οι μεταβλητές Res και Ther. Η πρώτη επιστρέφει το αποτέλεσμα της διάγνωσης και η άλλη επιστρέφει το αποτέλεσμα της θεραπείας. Η δεύτερη εντολή, η οποία ανήκει στη βιβλιοθήκη jpl, περνά μέσα από τη συνάρτηση Query(qs) της java, το στόχο που πρέπει να εκτελέσει η Prolog. Η τρίτη εντολή αποθηκεύει στη συμβολοσειρά diagnosis το αποτέλεσμα της διάγνωσης, μέσω της εντολής qs.oneSolution().get("Res"). Η τέταρτη εντολή, παίρνει τη συμβολοσειρά (δηλαδή το αποτέλεσμα της διάγνωσης) και το εμφανίζει στο αντίστοιχο πεδίο του αποτελέσματος. Το αποτέλεσμα το μετατρέπει σε συμβολοσειρά, αφαιρεί όλα τα περιττά σύμβολα και στη συνέχεια το εμφανίζει στο πεδίο του αποτελέσματος («Εικόνα 17:Παράθυρο Διάγνωσης»).

1^η εντολή: `String ts = kb_search(" + user_id + "," + " + condata + "," + persdata + "," + symptoms + "," + fbcTest + "," + specBloodTest + "," + electrHbTest + ",Res,Ther)." + "\n";`

2^η εντολή: `Query qs = new Query(ts);`

3^η εντολή: `diagnosis = qs.oneSolution().get("Res").toString();`

4^η εντολή: `Diagnosis_txt.append("Με τις δοθείσες πληροφορίες πάσχετε από: " + "\n" + "\n" + diagnosis.replaceAll("\\\\|\\\\\\\\|\\\\\\\\\\\\|\\\\\\\\\\\\\\\\|\\\\\\\\\\\\\\\\\\\\", "").trim());`

Για να γίνει όμως η επικοινωνία πρέπει να χρησιμοποιηθούν οι κατάλληλες βιβλιοθήκες από το αρχείο «*jpl.jar*». Το αρχείο «*anemia_diagnosis.pl*» περιέχει όλους τους κανόνες που είναι υλοποιημένοι σε γλώσσα Prolog οι οποίοι είναι υπεύθυνοι για το αποτέλεσμα της διάγνωσης. (Εγχειρίδιο Χρήσης Κεφάλαιο 12 για περαιτέρω λεπτομέρειες + Κεφάλαιο 7.5 για επεξήγηση της σύνδεσης).

6.4. Περιγραφή της Αναπαράστασης της Βάσης Γνώσης του ΔιΣΑ

Το εργαλείο που χρησιμοποιείται για την ανάπτυξη αυτού του έμπειρου συστήματος είναι η γλώσσα Prolog. Οι κανόνες του συστήματος από τη μεριά της Prolog περιλαμβάνουν τη βάση γνώσης του συστήματος, δηλαδή την εμπειρογνωμοσύνη του συστήματος όπως εκμαιεύθηκε από τον ειδικό κατά την ανάπτυξη του έμπειρου συστήματος. Η βάση γνώσης αποτελείται εκτός από τους κανόνες και από τα γεγονότα ή δεδομένα των ασθενών, τα οποία έχουν καταχωρηθεί. Τους κανόνες και τα γεγονότα της βάσης γνώσης θα τα παρουσιάσουμε στη συνέχεια αναλυτικά.

Κανόνες Βάσης Γνώσης

Για την υλοποίηση του συστήματος έχουν γραφτεί οι κανόνες της βάσης γνώσης οι οποίοι κατευθύνουν τη διαδικασία διάγνωσης. Οι κανόνες έχουν την εξής γενική μορφή:

Εάν (Συμπτώματα \subseteq Έγκυρα_Συμπτώματα) \wedge
(Δεδομένα_Ασθενή = $\langle \Delta 1, \Delta 2, \dots, \Delta k \rangle$) \wedge
 $\Delta 1 \in [\text{Εύρος_τιμών } \Delta 1]$ \wedge
 $\Delta 2 \in [\text{Εύρος_τιμών } \Delta 2]$ $\wedge \dots \wedge$
 $\Delta k \in [\text{Εύρος_τιμών } \Delta k]$

Τότε

Διάγνωση \in Είδος_Αναιμίας \wedge
Θεραπεία \in Τρόπος_Αντιμετώπισης_Ασθένειας

Όπου *Συμπτώματα* είναι το σύνολο των συμπτωμάτων του ασθενή, *Έγκυρα_Συμπτώματα* είναι το σύνολο των συμπτωμάτων που εμφανίζονται σε κάποιον ασθενή που έχει αναιμία. Δηλαδή, *Έγκυρα_Συμπτώματα* = {Πονοκέφαλος, Αδυναμία, Έλλειψη_συγκέντρωσης, Απώλεια_όρεξης, Απώλεια_βάρους, Ωχρότητα, Επώδυνη_σπλήνα, Ταχυκαρδία, Δύσπνοια, Γλωσσίτιδα, Πληγές, Κούραση, Κληρονομικότητα_Μητέρας, Κληρονομικότητα_πατέρα, Χαρακτηριστικά_προσώπου} και τα *Δεδομένα_Ασθενή* περιλαμβάνουν το σύνολο των ιατρικών εξετάσεων και τα προσωπικά δεδομένα του ασθενή.

Τα *Ιατρικά_δεδομένα* περιλαμβάνουν τις εξής εξετάσεις: α) τη γενική αίματος, β) την εξέταση ηλεκτροφόρισης αιμοσφαιρίνης και γ) εξέταση B12 βιταμίνης και Φυλικού οξέος. Τα *Προσωπικά_δεδομένα* περιλαμβάνουν τα εξής δεδομένα του ασθενή: α) φύλο, β) ηλικία και γ) εγκυμοσύνη.

Η αναπαράσταση των κανόνων έγινε με Prolog κανόνες της παρακάτω μορφής:

rule(Rule_id, Visit_id, User_id, Δεδομένα_Ασθενή, Διάγνωση,Θεραπεία) :-

Έλεγχος_δεδομένων_ασθενή.

Όπου *Rule_id* είναι ο κωδικός του κανόνα, το *Visit_id* είναι ο κωδικός επίσκεψης του ασθενή, το *User_id* είναι ο κωδικός του ασθενή, τα *Δεδομένα_Ασθενή*, *Διάγνωση*, *Θεραπεία* είναι τα ίδια όπως την προηγούμενη παράγραφο. Ο *Έλεγχος_δεδομένων_ασθενή* είναι μία σύζευξη στόχων της Prolog που εξετάζουν τα *Ιατρικά_δεδομένα* του ασθενή εαν εμπίπτουν σε παθολογικές τιμές και αν ικανοποιούνται τα αντιστοιχα *Προσωπικά_δεδομένα*.

Για παράδειγμα ο παρακάτω κανόνας είναι από το σύστημα ΔιΣΑ. Η ταυτότητα του είναι το r8, το *Visit_id* και το *UserId* είναι ο κωδικός επίσκεψης και ο κωδικός του ασθενή. Το *DataCom* είναι τα στοιχεία του ασθενή τα οποία περιλαμβάνουν το Ονοματεπώνυμο, τη Διεύθυνση και το Τηλέφωνο του ασθενή, *PersData* είναι τα προσωπικά στοιχεία του ασθενή όπως εξηγήθηκε στην παραπάνω παράγραφο. Ακολουθεί μια λίστα η οποία περιλαμβάνει τα Συμπτώματα (*Symptoms*) και τα Ιατρικά_δεδομένα (*FullBloodTest*, *SpecBloodTest*, *ElectricalHb*). Το *Diagnosis* είναι η διάγνωση και το *Therapy* είναι η Θεραπεία.

rule(r8, kb_da(Visit_id, UserId, DataCom, PersData,
[Symptoms, FullBloodTest, SpecBloodTest, ElectricalHb], D), Diagnosis, Therapy) :-

check_WBC_PLT(FullBloodTest, noPancytopenia),

check_RTC(FullBloodTest, rtcLess2),

check_MCV(FullBloodTest, mcvHigh),

check_Folic(SpecBloodTest, folic),

Diagnosis = 'Αναιμία Φυλικού Οξέος',

Therapy = 'Θεραπεία με συμπλήρωμα φυλλικού οξέος'.

Ο παραπάνω κανόνας μπορεί να ενεργοποιηθεί (η προϋπόθεσή του ικανοποιείται) με τον παρακάτω στόχο.

“ ?- *rule(R, kb_da(13, 13, ['Κίμωνας Ιωαννίδης', 'Βρυλήσια Αττικής', '6984575625'], [male, no, 65], [['ΑΠΩΛΕΙΑ ΒΑΡΟΥΣ', 'ΑΠΩΛΕΙΑ ΟΡΕΞΗΣ'], [4800, 1, 100, 5.5, 45, 22, 320, 39, 13, 170], [125, 2], [], D, T), Res, Ther).* ”

Γεγονότα/Δεδομένα Βάσης Γνώσης

Τα γεγονότα της βάσης γνώσης αναπαριστούνται με το κατηγορημα *kb_da/7*. Στο κάθε όρισμα του *kb_da/7* έχει δοθεί όνομα σχετικό με το ρόλο του.

kb_da(Visit_id, User_id, DataCommunication, PersonalData,

[Symptoms, FullBloodTest, SpecialBloodTest, ElectricalHbTest], Diagnosis).

- Το πρώτο όρισμα, δηλαδή το *Visit_id*, είναι ο κωδικός επίσκεψης του ασθενή.
- Το δεύτερο όρισμα, *User_id*, είναι η ταυτότητα του κάθε ασθενή. Η οποία παραμένει ίδια σε κάθε καταχώρηση του ασθενή.

Τα υπόλοιπα 3 ορίσματα είναι κάποιες λίστες

- Το τρίτο όρισμα, *DataCommunication*, είναι μια λίστα που περιέχει τα δεδομένα επικοινωνίας (όνομα, διεύθυνση, τηλέφωνο) του ασθενή.

- Το τέταρτο όρισμα, *PersonalData*, περιέχειτα προσωπικά δεδομένα (ηλικία, φύλο, εγκυμοσύνη) του ασθενή,
- Το πέμπτο όρισμα είναι μια λίστα με τέσσερα στοιχεία. Το πρώτο στοιχείο, το *Symptoms*, έχει τα συμπτώματα και τα υπόλοιπα τρία, *FullBloodTest*, *SpecialBloodTest* και *ElectricalHbTest*, έχουν εξετάσεις.
- Το όρισμα, *Diagnosis* είναι μια συμβολοσειρά που έχει τη διάγνωση του ασθενή.
- Το όρισμα, *Therapy* είναι μια συμβολοσειρά που έχει τη θεραπεία που έχει προτείνει ο γιατρός στον ασθενή.

Τα παρακάτω είναι 2 στιγμιότυπα του κατηγορήματος kb_da/7.

kb_da (4, 4, ['Δήμητρα Περπερίδου', 'Καραμανώλη 5 55142 Θεσσαλονίκη', '6977458521'], [female,yes,32], [['ΠΟΝΟΚΕΦΑΛΟΣ', 'ΠΡΟΒΛΗΜΑΤΑ ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ', 'ΕΛΛΕΙΨΗΣ ΔΙΑΤΡΟΦΗΣ', 'ΔΥΣΠΝΟΙΑ', 'ΤΑΧΥΚΑΡΔΙΑ'], [4800, 1, 72, 3.7, 50, 15, 280, 32, 10, 180],[,[]], 'Σιδηροπενική Αναιμία', 'Χορήγηση Σιδήρου').

kb_da (17, 17, ['Μαρία Παπαμιχαήλ', 'Κονίτσης 54 59100 Βέροια', '6984525154'], [female,no,42], [['ΑΔΥΝΑΜΙΑ', 'ΩΧΡΟΤΗΤΑ ΔΕΡΜΑΤΟΣ', 'ΑΠΩΛΕΙΑ ΟΡΕΞΗΣ'], [4900, 3, 99, 3.2, 45, 28, 320, 45, 13, 170],SpecialBloodTest,[HbA1,HbA2,2,HbF,Schema_RBC]], 'Δρεπανοκυττάρωση', 'Επικοινωνία με το γιατρό, πιθανή σπληνεκτομή.').

Για τον ασθενή με κωδικό 17, τα ιατρικά δεδομένα *SpecialBloodTest*, *HbA1*, *HbA2*, *HbF*, *Schema_RBC* δεν έχουν τιμές., γι' αυτό παρουσιάζονται στο παράδειγμα ως μεταβλητές.

7. Υλοποίηση τους Συστήματος

Σε αυτό το κεφαλαίο θα περιγράψουμε την υλοποίηση του συστήματος ΔιΣΑ. Η περιγραφή χωρίζεται στο τμήμα που έχει υλοποιηθεί σε Prolog και στο τμήμα που έχει υλοποιηθεί σε Java. Η περιγραφή του κώδικα σε Prolog θα περιλαμβάνει τα κυριότερα κατηγορήματα. Δηλαδή αυτά που σχετίζονται με την αναπαράσταση της βάσης γνώσης και αυτά που αφορούν τη διαδικασία της διάγνωσης. Για κάθε κατηγορήμα θα υπάρχει η γενική περιγραφή της λειτουργίας του, η περιγραφή των ορισμάτων του καθώς και στιγμιότυπα κλήσης του. Η περιγραφή του κώδικα σε Java θα περιλαμβάνει τις κυριότερες κλάσεις και πως γίνεται η αλληλεπίδραση με τον κώδικα σε Prolog ώστε να έχουμε το επιθυμητό αποτέλεσμα της διεπαφής.

7.1. Περιγραφή της Υλοποίησης της Βάσης Γνώσης του ΔιΣΑ σε Prolog.

Παρουσιάσαμε στην παράγραφο 6.4 την αναπαράσταση της βάσης γνώσης. Σε αυτή την ενότητα θα παρουσιάσουμε τον τρόπο υλοποίησης της σε Prolog.

Κανονες Βάσης Γνώσης

*rule(Rule_id, Visit_id, User_id, Δεδομένα_Ασθενή, Διάγνωση,Θεραπεία) :-
Έλεγχος_δεδομένων_ασθενή.*

Το κατηγορήμα *rule(Rule_id, Visit_id, User_id, Δεδομένα_Ασθενή, Διάγνωση, Θεραπεία)* δέχεται τον κωδικό του κανόνα, τον κωδικό της επίσκεψης, τον κωδικό του ασθενή και τα δεδομένα του ασθενή (ιατρικά, προσωπικά). Επιστρέφει τη διάγνωση και τη προτεινόμενη θεραπεία εφόσον ικανοποιείται ο έλεγχος των δεδομένων του ασθενή (*Έλεγχος_δεδομένων_ασθενή*). Ο έλεγχος των δεδομένων του ασθενή περιλαμβάνει τους εξής ελέγχους:

- Το κατηγορήμα *check_WBC_PLT(FullBloodTest,noPancytopenia)* είναι αληθές αν οι τιμές του PLT και του WBC βρίσκονται σε κάποια διαστήματα τα οποία εξασφαλίζουν ότι δεν υπάρχει πανκυτταροπενία.
- Το κατηγορήμα *check_RTC(FullBloodTest,rtcLess2)*, είναι αληθές εαν η τιμή του RTC βρίσκεται σε κάποιο διάστημα το οποίο χρησιμοποιείται στη διαδικασία της διάγνωσης.
- Το κατηγορήμα *check_MCV(FullBloodTest,mcvHigh)*, είναι αληθές εαν η τιμή του MCV βρίσκεται σε κάποιο διάστημα το οποίο χρησιμοποιείται στη διαδικασία της διάγνωσης.
- Το κατηγορήμα *check_Folic(SpecBloodTest,folic)*, είναι αληθές εαν η τιμή του Φυλικού Οξέος βρίσκεται σε κάποιο διάστημα το οποίο χρησιμοποιείται στη διαδικασία της διάγνωσης.

*rule(r8,kb_da(Visit_id,UserId,DataCom,PersData,
[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D), Diagnosis, Therapy) :-*

*check_WBC_PLT(FullBloodTest,noPancytopenia),
check_RTC(FullBloodTest,rtcLess2),
check_MCV(FullBloodTest,mcvHigh),*

check_Folic(SpecBloodTest,folic),
Diagnosis = 'Αναιμία Φυλικού Οξέος',
Therapy= 'Λήψη συμπληρώματος φυλικού οξέος'.

Ο παραπάνω κανόνας επιστρέφει τη διάγνωση 'Αναιμία Φυλικού Οξέος' και θεραπεία 'Λήψη συμπληρώματος φυλικού οξέος', εφόσον τα δεδομένα του ασθενή ικανοποιούν τις συνθήκες όπως εκφράζονται από το σώμα του κανόνα r8.

Γεγονότα Βάσης Γνώσης

Η αναπαράσταση των δεδομένων των ασθενών περιλαμβάνει εκτός από το κατηγορήμα *kb_da/7* και τα εξής κατηγορήματα:

- Το κατηγορήμα *max_visit_id(N)* είναι αληθές εαν N είναι το μέγιστο πλήθος των επισκέψεων που έχει δεχτεί μέχρι τώρα ο γιατρός.
- Το κατηγορήμα *max_patient_id(N)* είναι αληθές εαν N είναι το μέγιστο πλήθος των ασθενών που έχουν επισκεφθεί το γιατρό.
- Το κατηγορήμα *allVisits(List_visits_ids)* είναι αληθές εαν το *List_visits_ids* είναι μια λίστα που έχει τον κωδικό όλων των επισκέψεων που έγιναν στο γιατρό.
- Το κατηγορήμα *allPatients(List_patients_ids)* είναι αληθές εαν το *List_patients_ids* είναι μια λίστα που έχει τον κωδικό όλων των ασθενών που επισκέφθηκαν το γιατρό.

Ακολουθούν πιθανά στιγμιότυπα της βάσης των γεγονότων για τα κατηγορήματα *max_visit_id/1*, *max_patient_id/1*, *allVisits/1* και *allPatients/1*.

max_visit_id(28).

max_patient_id(24).

allVisits([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28]).

allPatients([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24]).

Η χρήση αυτών των κατηγορημάτων είναι σημαντική για την ενημέρωση της βάσης των δεδομένων και για την επεξεργασία τους κατά τη διαδικασία της διάγνωσης. Θεωρείστε το παρακάτω παράδειγμα:

*kb_da (25, 22, ['Παναγιωτης' , '71410 Ηράκλειο' , '2810123456'], [male, no, 50],
[['ΠΟΝΟΚΕΦΑΛΟΣ' , 'ΑΔΥΝΑΜΙΑ' , 'ΠΡΟΒΛΗΜΑΤΑ ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ'], [4300, 1,
75, 3.7, 41, 18, 250, 30, 9, 180], [], []], 'Σιδηροπενική Αναιμία', 'Χορήγηση Σιδήρου').*

Σε αυτό το παράδειγμα οι κωδικοί **25** και **22** αντιστοιχούν στον κωδικό επίσκεψης και στον κωδικό ασθενή. Η σύνδεση των στοιχείων του ασθενούς όπως φαίνεται στο παραπάνω γεγονός τη Prolog με τα προαναφερθέντα κατηγορήματα, (*max_visit_id/1*, *max_patient_id/1*, *allVisits/1* και *allPatients/1* τα οποία χρησιμοποιούντε για την επεξεργασία των δεδομένων του ασθενή), γίνεται μέσω αυτών των δύο κωδικών.

7.2. Περιγραφή της υλοποίησης της διαδικασίας Διάγνωσης του ΔιΣΑ σε Prolog.

Το κατηγορήμα *diagnose/3* εκτελεί τη διαδικασία της διάγνωσης. Το κατηγορήμα *diagnose(Visit_id,Diagnosis,Therapy)* είναι αληθές εαν *visit_id* είναι ο κωδικός μια επίσκεψης ενός ασθενή, *Diagnosis* είναι η διάγνωση που έγινε και *Therapy* είναι η θεραπεία

που προτάθηκε. Αυτό το κατηγορήμα καλεί το κατηγορήμα `clause/2` στο οποίο δίνοντας τον κωδικό της επίσκεψης ο οποίος είναι μοναδικός για όλες τις εγγραφές της βάσης των γεγονότων, επιστρέφει τα δεδομένα της συγκεκριμένης επίσκεψης του ασθενή. Το ΔιΣΑ προχωράει στη διαδικασία της διάγνωσης μόνο εφόσον πρόκειται για μια νέα επίσκεψη ενός ασθενή (είτε νέου είτε παλιού ασθενή). Στη συνέχεια, καλείται το κατηγορήμα `rule/4` από το `diagnose/3`, στο οποίο περνάει τα δεδομένα του ασθενή (ιατρικά και προσωπικά) και επιστρέφει το αποτέλεσμα της διάγνωσης και της θεραπείας, εφόσον κάποιος κανόνας ικανοποιείται.

```
diagnose(Visit_id,Diagnosis,Therapy) :-
  clause(kb_da(Visit_id,Patient_Id,DataCom,PersData,
    [Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D),Body),
  rule(R_Id,kb_da(Visit_id,Patient_Id,DataCom,PersData,
    [Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D),Diagnosis,Therapy),nl.
```

Για παράδειγμα εαν τρέξουμε το σύστημα με τον εξής στόχο:

«?- diagnose(25,Diagnosis,Therapy)»

Θα παρει τα δεδομένα του ασθενή με κωδικό επίσκεψης **25** καλώντας το `clause/2`, δηλαδή θα πάρει τα δεδομένα από την παρακάτω εγγραφή:

```
kb_da (25, 22, ['Παναγιωτης' , '71410 Ηράκλειο' , '2810123456'], [male, no, 50],
[['ΠΙΟΝΟΚΕΦΑΛΟΣ' , 'ΑΔΥΝΑΜΙΑ' , 'ΠΡΟΒΛΗΜΑΤΑ ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ'], [4300, 1,
75, 3.7, 41, 18, 250, 30, 9, 180], [], []], 'Σιδηροπενική Αναιμία', 'Χορήγηση Σιδήρου').
```

Στη συνέχεια θα καλέσει το στόχο:

```
rule(R_Id,kb_da(Visit_id,Patient_Id,DataCom,PersData,
  [Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D),Diagnosis,Therapy)
```

Ο οποίος με τα δεδομένα που έχουν δεσμευτεί οι μεταβλητές θα έχει την παρακάτω μορφή:

```
rule(R,kb_da(25, 22, ['Παναγιωτης' , '71410 Ηράκλειο' , '2810123456'], [male, no, 50],
[['ΠΙΟΝΟΚΕΦΑΛΟΣ' , 'ΑΔΥΝΑΜΙΑ' , 'ΠΡΟΒΛΗΜΑΤΑ ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ'], [4300, 1,
75, 3.7, 41, 18, 250, 30, 9, 180 ] , [], []], D),Res,Ther) .
```

Το παραπάνω στιγμιότυπο του στόχου, ικανοποιείται και θα επιστρέψει ως αποτέλεσμα διάγνωσης 'Σιδηροπενική Αναιμία' και θεραπεία 'Χορήγηση Σιδήρου'.

7.3. Περιγραφή της υλοποίησης της διαδικασίας ενημέρωσης της βάσης γνώσης.

Για να έχει τη δυνατότητα το σύστημα να κάνει ανανέωση της βάσης κανόνων έχουν δημιουργηθεί 3 κατηγορήματα, το `change_rec/2`, το `create_rec/2` και το `delete_rec/2`.

1. Το κατηγορήμα **change_rec(R_id,Body)** έχει ως πρώτο όρισμα, **R_id**, την ταυτότητα του κανόνα τον οποίο θέλουμε να αλλάξουμε και ως δεύτερο όρισμα, **Body**, το νέο σώμα του κανόνα το οποίο θα αντικαταστήσει το υπάρχον. Στο σώμα του

κατηγορήματος `change_rec/2` υπάρχουν οι κλήσεις σε κατηγορήματα τα οποία κάνουν την ενημέρωση του κανόνα. Αυτό γίνεται με τα παρακάτω 3 βήματα.

- a. Πρέπει το πρόγραμμα να βρει τον κανόνα με το συγκεκριμένο κωδικό καλώντας το ενσωματωμένο κατηγορήμα `clause/2`.
 - b. Έπειτα να σβήσει τον κανόνα με τον ίδιο κωδικό, με την εντολή `retractall/1`.
 - c. Στο τέλος για να ξανα εισάγει τον κανόνα στη βάση γνώσης καλεί το `assert/1`.
Στην ουσία αυτό το κατηγορήμα βρίσκει τον κανόνα με βάση τη ταυτότητα του, όπως ορίστηκε, τον διαγράφει και το ξανά εισάγει με το νέο σώμα (το `Body`). Εάν η ταυτότητα που έδωσε ο χρήστη δεν αντιστοιχεί με κάποιο κανόνα, το κατηγορήμα δεν ικανοποιείται και επιστρέφει `false`.
2. Το κατηγορήμα **`create_rec(Body)`**, έχει ως όρισμα το σώμα του κανόνα τον οποίο θέλουμε να εισάγουμε στη βάση γνώσης. Όλοι οι κανόνες έχουν την ίδια κεφαλή αλλά αλλάζει ο κωδικός τους. Για αυτό το `create_rec/1` για να δημιουργήσει ένα κανόνα χρειάζεται μόνο το νέο σώμα του κανόνα, την κεφαλή την δημιουργεί αυτόματα μόν του. Το κατηγορήμα υλοποιείται ως εξής,
- a. Δημιουργεί ένα νέο κωδικό για τον κανόνα.
 - b. Ενημερώνει τη σχετική λίστα στη βάση των γεγονότων (`rules/1`)
 - c. Στη συνέχεια, δημιουργεί την κεφαλή του κανόνα και καταχωρεί στη βάση κανόνων το νέο κανόνα.
3. Το κατηγορήμα **`delete_rec(R_id)`**, έχει ως όρισμα τον κωδικό του κανόνα τον οποίο θέλουμε να σβήσουμε. Ο κωδικός κάθε κανόνα είναι μοναδικός και είναι αρκετός για να γίνει η διαγραφή ενός κανόνα. Επειδή όλοι οι ενεργοί κανόνες υπάρχουν σε μια λίστα στο γεγονός `rules(R_id)`, θα πρέπει να ενημερωθεί και αυτή η λίστα. Τα βήματα διαγραφής ενός κανόνα είναι τα εξής:
- a. Διαγράφει το συγκεκριμένο κανόνα με το `retractall(R_id)`.
 - b. Ενημέρωση της λίστας των ενεργών κανόνων με κλήση στο κατηγορήμα `del_rule_id(R_id)`.
- Το κατηγορήμα **`delete_rec(R_id)`**, επιστρέφει `true` μόνο όταν ο κωδικός (`R_id`) που του δόθηκε αντιστοιχεί σε κωδικό ενεργού κανόνα, αλλιώς επιστρέφει `false`.

7.4. Περιγραφή της υλοποίησης της διεπαφής του ΔιΣΑ σε Java

Το βασικό κομμάτι σε κάθε πρόγραμμα που δημιουργείτε είναι η εισαγωγή των απαραίτητων βιβλιοθηκών. Οι βιβλιοθήκες υπάρχουν ώστε να μπορεί να λειτουργεί σωστά το ΔιΣΑ. Η γλώσσα Java προσφέρει μια πλούσια συλλογή βιβλιοθηκών για διάφορες χρήσεις. Οι παρακάτω 4 βιβλιοθήκες είναι οι κυριότερες που χρησιμοποιούνται για την υλοποίηση της διεπαφής του ΔιΣΑ.

- Η **`java.awt`** περιέχει όλες τις κλάσεις για τη δημιουργία διεπαφής χρήστη και για δημιουργία γραφικών και εικόνων.

```
import java.awt.Toolkit;  
import java.awt.event.WindowEvent;  
import java.awt.Toolkit;  
import java.awt.event.WindowEvent;
```

- Η **`java.util`** προσφέρει δομές δεδομένων, αλγορίθμους, κανονικές εκφράσεις, κ.ά.
`import java.util.ArrayList;` //βιβλιοθήκη για δημιουργία των `arraylist` που χρησιμοποιούνται
`import java.util.logging.Level;`
`import java.util.logging.Logger;`

- Η **java.swing** είναι για τη δημιουργία γραφικών περιβαλλόντων επικοινωνίας.
`import javax.swing.JFileChooser;`
`import javax.swing.JOptionPane;`
`import javax.swing.table.DefaultTableModel;`

- Η **org.jpl7** είναι βιβλιοθήκη για τη διεπικοινωνία Java και Prolog
`import org.jpl7.Atom;`
`import org.jpl7.Query; // Βιβλιοθήκες της jpl (εξωτερική βιβλιοθήκη) ώστε`
`// να γίνει η σύνδεση`
`import org.jpl7.Term; //prolog-javaka και να μπορέσουν να χρησιμοποιηθούν`
`import org.jpl7.*; //queries- terms- atoms και ότι άλλο χρειαζόμαστε`

Το σύστημα χρησιμοποιεί 12 παράθυρα. Για κάθε παράθυρο έχει οριστεί ένα συγκεκριμένο μέγεθος και έχει τοποθετηθεί σε συγκεκριμένο σημείο. Ο επόμενος κώδικας καθορίζει τις συντεταγμένες τοποθέτησης του παράθυρου.

```
intx= 315; // καθορίζει τις συντεταγμένες του παραθύρου
inty= 215;
this.setLocation(x, y); // μετακινεί το παράθυρο
```

Το πρώτο παράθυρο του ΔιΣΑ («*Εικόνα 1: Σύστημα γνώσης*») είναι η είσοδος στο σύστημα ΔιΣΑ. Σε αυτό το παράθυρο υπάρχουν δύο λειτουργίες, η μία είναι το “Διαγνωστικό Σύστημα Αναιμίας (ΔιΣΑ)” και η άλλη είναι η “Έξοδος”. Επιλέγοντας την δραστηριότητα “Διαγνωστικό Σύστημα Αναιμίας (ΔιΣΑ)”, το πρόγραμμα ανοίγει το παράθυρο «*Εικόνα 2: Πάνελ επιλογών ΔιΣΑ*», όπου δίνονται οι επιλογές:

1. Για διάγνωση.
2. Για ενημέρωση της βάσης κανόνων.
3. Για αναζήτησης ιστορικού ασθενούς .
4. Για έξοδο από το σύστημα. Η λειτουργία του τερματισμού καλεί μία μέθοδο για να τερματίσει το πρόγραμμα, η μέθοδος αυτή ονομάζεται **close** και θα αναλυθεί παρακάτω.

Σε κάθε σωστό και έγκυρο ιατρικό σύστημα πρέπει να υπάρχουν τα στοιχεία του ασθενή. Έτσι και αυτή η διεπαφή περιέχει μια φόρμα συμπλήρωσης των στοιχείων του ασθενή τα οποία αποθηκεύονται σε λίστες. Ο χρήστης γιατρός του συστήματος πρέπει να δώσει ιδιαίτερη προσοχή σε αυτό το θέμα ώστε να αποθηκευτούν σωστά τα στοιχεία και να αποφευχθεί οποιοδήποτε πρόβλημα. Για τη δημιουργία αυτής της φόρμας χρησιμοποιήθηκαν κάποια πεδία για συμπλήρωση από το χρήστη (textfields) και ειδικά κουμπιά με εναλλακτική επιλογή (radiobuttons). Ο κώδικας που υπάρχει για αυτά τα πεδία και κουμπιά έχει δημιουργηθεί αυτόματα από το Netbeans IDE μέσω του drag&drop που χρησιμοποιήθηκε. Παρακάτω είναι ένα παράδειγμα κώδικα διεπικοινωνίας ο οποίος δημιούργησε το παράθυρο «*Εικόνα 22: Παράθυρο στοιχείων ΔιΣΑ*».

Στο παρακάτω τμήμα κώδικα δηλώνονται όλα τα πεδία (textfields), labels & buttons

```
prgnYesRbtn = new javax.swing.JRadioButton(); //button
poliLabel= new javax.swing.JLabel(); //label
AddrField = new javax.swing.JTextField(); //textfield
GinRadBtn = new javax.swing.JRadioButton(); //button
andrasRadBtn= new javax.swing.JRadioButton(); //button
```

```

egkiosLabel= new javax.swing.JLabel(); //label
telField= new javax.swing.JTextField(); //textfield
prgnNoRadbtn= new javax.swing.JRadioButton(); //button
telLabel= new javax.swing.JLabel(); //label
nameField= new javax.swing.JTextField(); //textfield
ageField= new javax.swing.JTextField(); //textfield
ContinueBtn1 = new javax.swing.JButton(); //button

```

Το παρακάτω τμήμα κώδικα δείχνει τους τίτλους των ετικετών, πού και πώς έχουν τοποθετηθεί και τους listeners για να αποκτήσουν λειτουργία.

```

poliLabel.setText("ΠΟΛΗ"); // το κείμενο της ετικέτας
stoixeia.add(poliLabel); //πώς θα τοποθετηθεί στο πλαίσιο
poliLabel.setBounds(20, 250, 50, 14); //πού θα τοποθετηθεί
stoixeia.add(AddrField); // πώς θα τοποθετηθεί στο πλαίσιο
AddrField.setBounds(200, 300, 200, 30); // πού θα τοποθετηθεί

```

Η επόμενη εντολή συνδέει 2 radiobuttons ώστε να έχουν εναλλακτική επιλογή. Για παράδειγμα, Φύλο: Γυναίκα ή Αντρας.

```
pregGroup.add(GinRadBtn);
```

Το παρακάτω τμήμα κώδικα ορίζει το κείμενο του radiobutton, του ορίζει τη διαφάνεια (ορατό, μη ορατό) και listeners για να αποκτήσουν λειτουργία.

```

GinRadBtn.setText("ΓΥΝΑΙΚΑ"); //ορίζεται το κείμενο του radiobutton
GinRadBtn.setOpaque(false); //η διαφάνεια γίνεται false (ορατό)
//Οι επόμενες 5 γραμμές είναι ο listener
GinRadBtn.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) { //προστίθεται Listener
GinRadBtnActionPerformed(evt);
}
});

```

```

stoixeia.add(GinRadBtn); //πώς έχει τοποθετηθεί στο πλαίσιο
GinRadBtn.setBounds(200, 200, 90, 23); //πού έχει τοποθετηθεί

```

Εικόνα 22: Παράθυρο στοιχείων ΔιΣΑ

Με τον ίδιο τρόπο δημιουργούνται και τα υπόλοιπα πεδία που υπάρχουν στο παράθυρο. Στο κάτω μέρος του παραθύρου υπάρχει η δραστηριότητα «Συνέχεια». Αυτή η δραστηριότητα δημιουργήθηκε όπως και οι υπόλοιπες αλλά την ώρα που γίνεται εισάγουμε το κουμπί στο παράθυρο δεν έχει καμία απολύτως λειτουργικότητα, απλά υπάρχει! Για να αποκτήσει λειτουργικότητα οποιοδήποτε στοιχείο του παραθύρου που πάμε να κατασκευάσουμε, πρέπει να του οριστεί ένα Java event (Java γεγονός). Ένα java γεγονός αντιστοιχεί στον ορισμό μιας δραστηριότητας. Το Java event που έχει οριστεί στη δραστηριότητα του πλήκτρου «Συνέχεια» είναι το **mouseClicked**, το οποίο ορίζει στο σύστημα τη λειτουργία της δραστηριότητας όταν αυτή επιλεγθεί. Αυτό που πρέπει να γίνει τώρα είναι με το πάτημα αυτής δραστηριότητας να μεταφερθούμε σε επόμενο παράθυρο και εάν θέλουμε να κάνει και κάποιες άλλες λειτουργίες, όπως για παράδειγμα να αποθηκευτούν κάπου τα στοιχεία που έχει εισάγει ο χρήστης στο παράθυρο. Το κάθε παράθυρο αλλά και κάθε στοιχείο (κουμπιά, ετικέτες κτλ.) έχει κάποιο `variablename`, αυτό το `variablename` χρησιμοποιείται για να οριστεί ποτε θα εμφανίζεται και ποτε θα χρησιμοποιηθεί.

```

stoixeia.setVisible(false); //απενεργοποιεί το παράθυρο «stoixeia»
exetaseis.setVisible(true); //και ενεργοποιεί το επόμενο παράθυρο «exetaseis» που αφορά
τις εξετάσεις

```

Για να αποθηκευτούν τα στοιχεία που έχουν εισαχθεί από το χρήστη σε ένα παράθυρο, έχουν δηλωθεί 6 λίστες `arraylist` από `string` με το όνομα **condata**, **persdata**, **symptoms**, **fbctest**, **specBloodTest**, **electrHbTest**. Με το πάτημα ενός κουμπιού αποθηκεύονται οι τιμές των στοιχείων σε αντίστοιχες 6 λίστες που αναφέρθηκαν παραπάνω. Για παράδειγμα έστω ότι πατάμε το κουμπί «Συνέχεια» από το παράθυρο «Εικόνα 22: Παράθυρο στοιχείων ΔιΣΑ», τότε αποθηκεύονται οι τιμές των στοιχείων του ασθενή στις λίστες (`condata`, `persdata`). Οι λίστες είναι δημόσιες για να τις βλέπει όλο το πρόγραμμα και να τις χρησιμοποιεί.

```
//δηλώσεις λιστών
```

```
public ArrayList<String>condata = new ArrayList<>();  
public ArrayList<String>persdata = new ArrayList<>();  
public ArrayList<String>symptoms = new ArrayList<>();  
public ArrayList<String>fbcTest = new ArrayList<>();  
public ArrayList<String>specBloodTest = new ArrayList<>();  
public ArrayList<String>electrHbTest = new ArrayList<>();
```

```
// Η επόμενη εντολή παίρνει το κείμενο που πληκτρολογήθηκε και το αποθηκεύει  
//σε μεταβλητή (age) σε μορφή συμβολοσειράς  
age= ageField.getText();  
persdata.add(age); // προσθέτει τη συμβολοσειρά της μεταβλητής age στη λίστα persdata.
```

Τα επόμενα παράθυρα της διαγνωστικής διαδικασίας του ΔιΣΑ, είναι αυτά των εξετάσεων («Εικόνα 21: Παράθυρο έξτρα εξετάσεων», «Εικόνα 22: Παράθυρο στοιχείων ΔιΣΑ»). Το παράθυρο («Εικόνα 23: Παράθυρο Εξετάσεων») έχει ως variable name **e3etaseis**, περιέχει πεδία με τις βασικές εξετάσεις που γίνονται κατά την γενική ανάλυση αίματος συν τις εξετάσεις για σίδηρο και Φερριτίνη. Αυτό το παράθυρο αποτελείται από 10 ετικέτες - labels- με το όνομα της κάθε εξέτασης και το αντίστοιχο πεδίο και δίπλα κάθε label ωστέ να συμπληρώσει ο χρήστης την τιμή της εξέτασης. Δίπλα από κάθε πεδίο textfield υπάρχει ένα εικονίδιο το οποίο εάν πατηθεί εμφανίζεται ένα αναδυόμενο μήνυμα που περιέχει τις πληροφορίες για τις φυσιολογικές τιμές της κάθε εξέτασης.

Σε αυτό το παράθυρο, υπάρχει και ένα checkbox με το όνομα «Λοιπές Εξετάσεις». Εάν επιλεγθεί αυτό το πλήκτρο, το πρόγραμμα ανακατευθύνεται στο επόμενο παράθυρο («Εικόνα 24: Παράθυρο έξτρα εξετάσεων»), το οποίο περιέχει –ακριβώς με τον ίδιο τρόπο, όπως αναφέρθηκε παραπάνω- κάποιες πιο εξειδικευμένες εξετάσεις, οι οποίες έχουν επισημανθεί από το γιατρό πως πρέπει να γίνουν. Ταυτόχρονα αποθηκεύει στις λίστες ό,τι πληκτρολογήθηκε, με τρόπο που περιγράφεται παρακάτω. Σε καθένα από αυτά τα δύο παράθυρα υπάρχουν οι λειτουργίες, “Συνέχεια” και “Πίσω”. Η λειτουργία της οπισθοδρόμησης (“Πίσω”) καλεί την μέθοδο *close* η οποία κλείνει το παράθυρο. Η λειτουργία “Συνέχεια”, μας πηγαίνει σε επόμενο παράθυρο αλλά παράλληλα έχει κάνει και κάποιες άλλες λειτουργίες, όπως για παράδειγμα αποθηκεύει στις λίστες τα στοιχεία του παρόντος παραθύρου.

ΕΙΣΑΓΕΤΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΞΕΤΑΣΕΩΝ:

ΛΕΥΚΑ ΑΙΜΟΣΦΑΙΡΙΑ	<input type="text"/>	ΑΙΜΟΣΦΑΙΡΙΝΗ	<input type="text"/>
ΜΕΣΟΣ ΟΓΚΟΣ ΕΡΥΘΡΩΝ	<input type="text"/>	ΑΙΜΑΤΟΚΡΙΤΗΣ	<input type="text"/>
ΑΙΜΟΠΕΤΑΛΙΑ	<input type="text"/>	ΦΕΡΙΤΙΝΗ	<input type="text"/>
ΔΕΙΚΤΗΣ ΕΡΥΘΡΟΚΥΤΤΑΡΩΝ	<input type="text"/>	ΣΙΔΗΡΟΣ	<input type="text"/>
ΕΡΥΘΡΑ ΑΙΜΟΣΦΑΙΡΙΑ	<input type="text"/>	ΤΙΒC	<input type="text"/>

ΛΟΙΠΕΣ ΕΞΕΤΑΣΕΙΣ

All Fields must be completed!

Εικόνα 23: Παράθυρο Εξετάσεων

ΕΙΣΑΓΕΤΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΞΕΤΑΣΕΩΝ:

Β12 ΒΙΤΑΜΙΝΗ	<input type="text"/>	HbA1	<input type="text"/>
ΦΥΛΙΚΟ ΟΞΥ	<input type="text"/>	HbA2	<input type="text"/>
ΣΧΗΜΑ ΕΡΥΘΡΩΝ	<input checked="" type="radio"/> ΦΥΣΙΟΛΟΓΙΚΟ	HbF	<input type="text"/>
	<input type="radio"/> ΔΡΕΠΑΝΟΕΙΔΕΣ	HbS	<input type="text"/>
	<input type="radio"/> ΣΦΑΙΡΙΚΟ		

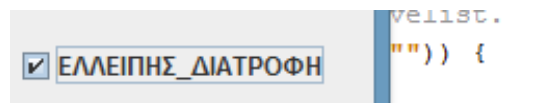
Εικόνα 24: Παράθυρο έξτρα εξετάσεων

Το παράθυρο «Εικόνα 16: Παράθυρο Συμπτωμάτων», περιέχει κάποια checkboxes και radiobuttons με συμπτώματα ή κάποια χαρακτηριστικά που είναι σημαντικά. Για την ακρίβεια, περιέχει 16 checkboxes με συμπτώματα, 2 με χαρακτηριστικά και 3 γκρουπ από radiobuttons με χαρακτηριστικά και το ιστορικό (ιστορικό πατέρα, ιστορικό μητέρας). Καθώς επίσης και δύο λειτουργίες, τη “ΔιΣΑ” και την “ΠΙΣΩ”. Η λειτουργία ΔιΣΑ ανακατευθύνει

το σύστημα στο επόμενο και τελικό παράθυρο της διάγνωσης («Εικόνα 17:Παράθυρο Διάγνωσης»). Όπως τα υπόλοιπα έτσι και αυτή λειτουργία του «ΔιΣΑ» πριν την εμφάνιση του επόμενου παραθύρου αποθηκεύει τα στοιχεία του σε λίστες. Αυτό το παράθυρο έχει στοιχεία τα οποία μπορούν μόνο να επιλεγθούν. Όταν επιλέξουμε τη λειτουργία «ΔιΣΑ» τα επιλεγμένα στοιχεία τα περνάει στις λίστες ως συμβολοσειρές. Για τα radiobuttons χρησιμοποιείται το Java event *eventItemStateChanged* και για τα checkboxes χρησιμοποιούνται εντολές *if.. else..*, οι οποίες λειτουργούν ακριβώς με τον ίδιο τρόπο όπως τα Java events. Για παράδειγμα ο επόμενος κώδικας μας δείχνει την υλοποίηση *if..else.. event* και ενός Java event που εκτελούν ακριβώς την ίδια λειτουργία. Εάν το στοιχείο (radiobutton ή checkbox) επιλεγθεί τότε αποθηκεύεται η τιμή του σε αντίστοιχη λίστα. Στην πρώτη περίπτωση έχουμε checkbox το οποίο αποθηκεύεται στη λίστα *symptom*. Στην δεύτερη περίπτωση είναι radiobutton και αποθηκεύεται στη λίστα *electrHbTest*.

1. If.. else.. event

```
if(diet.isSelected()) {
    symptoms.add(diet.getText()); //προσθέτει το επακριβές κείμενο του κουμπιού
                                στη λίστα symptoms
} else{
    diett= "NO";                // εάν δεν επιλέχθηκε κάνει το string ίσο με NO και δεν το προσθέτει σε
    καμία λίστα
}
}
```



Εικόνα 25: κουτί επιλογής

2. Java event

```
if(evt.getStateChange() == java.awt.event.ItemEvent.SELECTED) {
    electrHbTest.add(Shape_Sickle.getText());
} else if (evt.getStateChange() == java.awt.event.ItemEvent.DESELECTED) {
    Sxima_erithron = null;
}
}
```

Το παράθυρο («Εικόνα 17:Παράθυρο Διάγνωσης»), με *variablename* *diagnwsi*, έχει ένα *textarea*, 5 κουμπιά καθώς και ένα *progressbar*. Τα κουμπιά “Πίσω” και “Εναρξη” έχουν ως λειτουργία την οπισθοδρόμηση και την συνέχιση του συστήματος ΔιΣΑ στην αρχή. Το κουμπί “Εξοδος”, τερματίζει το σύστημα ΔιΣΑ χρησιμοποιώντας τη μέθοδο *close*. Η μέθοδος *close* ορίζει ως Java event του παραθύρου (*window*) το *WINDOW_CLOSING*. Ο παρακάτω κώδικας περιγράφει τις εντολές για τον τερματισμό του συστήματος ΔιΣΑ.

```
//μέθοδος close για τερματισμό προγράμματος
private void close() {
    WindowEvent winClosingEvent= new
    WindowEvent(this,WindowEvent.WINDOW_CLOSING);
    Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent);
}
}
```

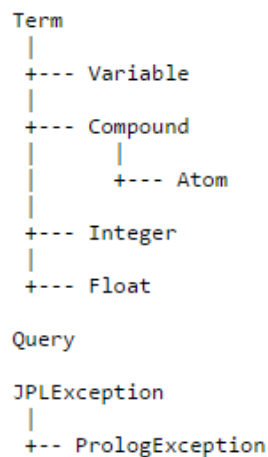
7.5. Περιγραφή Σύνδεσης Java-Prolog

Για τη σύνδεση Java-Prolog, χρησιμοποιείται η βιβλιοθήκη `jpl`. Η βιβλιοθήκη JPL (A Java Interface to Prolog) αποτελείται από ένα σύνολο κλάσεων Java και συναρτήσεων C, η οποία παρέχει διεπικοινωνία μεταξύ Java και Prolog. Χρησιμοποιεί το Java Native Interface (JNI) για να συνδεθεί με μια μηχανή Prolog μέσω της Prolog Foreign Language Interface (FLI). Η JPL δεν είναι μια καθαρή υλοποίηση Java της Prolog. Η JPL κάνει εκτεταμένη χρήση των εφαρμογών της Prolog για υποστηριζόμενες πλατφόρμες. Η τρέχουσα έκδοση του JPL λειτουργεί μόνο με SWI-Prolog.

Προς το παρόν, η JPL υποστηρίζει μόνο την ενσωμάτωση μιας μηχανής Prolog εντός της Java VM. Μελλοντικές εκδόσεις μπορεί να υποστηρίζουν την ενσωμάτωση της Java VM μέσα στη Prolog. Η JPL είναι σχεδιασμένη σε δύο επίπεδα, μια χαμηλού επιπέδου διεπαφή με την Prolog FLI και μια υψηλού επιπέδου διεπαφή Java για τον προγραμματιστή Java που δεν ενδιαφέρεται για τις λεπτομέρειες της Prolog FLI. Η διεπαφή χαμηλού επιπέδου προορίζεται για C προγραμματιστές.

Το **JPL 3.0.1 Java-calls-Prolog API** παρέχει ένα σύνολο από κλάσεις, οι οποίες κρύουν σχεδόν όλες τις κουραστικές λεπτομερείς που υπάρχουν στη διεπαφή χαμηλού επιπέδου. Είναι λιγότερο ευέλικτη από τη διεπαφή χαμηλού επιπέδου, αλλά είναι πιο φυσική και κοντά στην Prolog από τη διεπαφή χαμηλού επιπέδου.

Το πακέτο JPL περιέχει όλες τις κλάσεις για το περιβάλλον της υψηλού επιπέδου διεπαφής. Καμία από τις κλάσεις όμως δεν αντιστοιχεί σε κάποιον από τους τύπους δεδομένων στο Prolog Foreign Language Interface (FLI). Το API αποτελείται από την παρακάτω ιεραρχία κλάσεων: [11]



- **Term** είναι μια αφηρημένη κλάση της οποίας μόνο οι υποκλάσεις της μπορούν να έχουν στιγμιότυπα. Το **Term** είναι μια αφηρημένη κλάση, οπότε δεν μπορεί να αρχικοποιηθεί κατευθείαν, για να δημιουργηθεί ένα **Term**, δημιουργείτε ένα στιγμιότυπο μιας από τις 5 υποκλάσεις του.
- Κάθε στιγμιότυπο ενός **Query** περιέχει ένα **Term** (που υποδηλώνει το στόχο που πρέπει να αποδειχθεί) και πολλά άλλα.
- Κάθε στιγμιότυπο του **Compound** έχει ένα (`java.lang.String`) όνομα και μια σειρά από ορίσματα τύπου **Term** (πρέπει να υπάρχει τουλάχιστον ένα).
- **Atom** είναι μια εξειδίκευση του **Compound** με μηδέν ορίσματα. Για να δημιουργηθεί ένα **Atom**, περνάμε ένα όνομα τύπου **String** στον constructor του (`new Atom(...)`). Το

όνομα του Atom δεν είναι απαραίτητο να είναι με πεζά γράμματα, μπορεί να είναι οποιοδήποτε UTF-8 συμβολοσειρά.

Στην προκειμένη περίπτωση το `stringname` είναι το μονοπάτι (`Diagnosis.class.getResource("anemia_diagnosis.pl").getPath()`) του αρχείου που θέλουμε να φορτώσουμε.

```
Term consult_arg[] = {  
new Atom(Diagnosis.class.getResource("anemia_diagnosis.pl").getPath())  
};
```

Το **Query** περιλαμβάνει ένα **Term**, που αναπαριστά το στόχο της Prolog. Το παρακάτω query αναπαριστά το στόχο `consult("μονοπατι_arχειου\anemia_diagnosis.pl")`, ο οποίος με την προηγούμενη εντολή έχει καταχωρηθεί στη μεταβλητή `consult_arg[]`. Το παρακάτω παράδειγμα είναι ένα στιγμιότυπο του Query.

```
Query consult_query  
= new Query(  
"consult",  
consult_arg);
```

Ο παρακάτω κώδικας μας δείχνει πως γίνονται ερωτήσεις σε Prolog μέσω της Java. Αποθηκεύουμε σε μια συμβολοσειρά (String tt) το στόχο που θέλουμε να τρέξουμε σε Prolog. Περνάμε σε ένα στιγμιότυπο του Query (Query qt), τη συμβολοσειρά tt. Η Prolog τρέχει το στόχο (tt) και στη συνέχεια παίρνουμε τις τιμές των μεταβλητές που μας ενδιαφέρουν.

```
String tt = "kb_test(" + condata + "," + persdata + "," + symptoms + "," + fbcTest + "," +  
specBloodTest + "," + electrHbTest + ",Res,Ther)." + "\n";  
Query qt = new Query(tt);
```

Η παρακάτω εντολή γράφει το μήνυμα «Με τις δοθείσες πληροφορίες πάσχετε από» και δίπλα την απάντηση από την ερώτηση. Η εντολή «qt.oneSolution()» επιστρέφει όλη την απάντηση της Prolog και η εντολή «qt.oneSolution().get("Res")» επιστρέφει το αποτέλεσμα της διάγνωσης, δηλαδή την τιμή της μεταβλητής Res. Η εντολή «qt.oneSolution().get("Res").toString().replaceAll("\\[[\\]]\\(\\|\\)\\|\\|'+", " ").trim()» μετατρέπει την απάντηση σε συμβολοσειρά έχοντας αφαιρέσει τους περιττούς χαρακτήρες.

```
Diagnosis_txt.append("Με τις δοθείσες πληροφορίες πάσχετε από: " + "\n"  
+ "\n" + qt.oneSolution().get("Res").toString().replaceAll("\\[[\\]]\\(\\|\\)\\|\\|'+", " ").trim());
```

8. Υποδείγματα με τις κύριες λειτουργίες του συστήματος ΔιΣΑ.

Σε αυτό το κεφάλαιο επιδεικνύουμε το σύστημα ΔιΣΑ μέσα από κάποια επιλεγμένα παραδείγματα τα οποία καλύπτουν τις κυριότερες λειτουργίες του συστήματος.

I. ΔΙΑΓΝΩΣΗ

Τα επόμενα 4 παράθυρα μας κάνουν επίδειξη της διαδικασίας διάγνωσης.

- Στο παράθυρο «*Εικόνα 26: Στοιχεία ασθενή*» γίνεται εισαγωγή των στοιχείων του προς εξέταση ασθενή.
- Στο παράθυρο «*Εικόνα 27: Δεδομένα εξετάσεων*» εισάγονται τα αποτελέσματα των αιματολογικών εξετάσεων.
- Στο παράθυρο «*Εικόνα 28: Συμπτώματα ασθενούς*» Εικόνα 28 εισάγονται τα συμπτώματα του ασθενή.
- Στα παράθυρα «*Εικόνα 30: Θεραπεία ασθενούς*» επιστρέφει το ΔιΣΑ τη διάγνωση μαζί με τη φαρμακευτική αγωγή. Εάν θέλαμε μόνο τη διάγνωση, δεν επιλέγουμε την δραστηριότητα «*Θεραπεία*» («*Εικόνα 29: Διάγνωση ασθενούς*»).

The screenshot shows a web form titled "ΣΤΟΙΧΕΙΑ ΑΣΘΕΝΗ:" (Patient Information). The form contains the following fields and options:

- ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΔΕΣΠΟΙΝΑ ΜΠΑΣΔΕΚΗ
- ΗΛΙΚΙΑ: 25
- ΦΥΛΟ: ΑΝΔΡΑΣ ΓΥΝΑΙΚΑ
- ΕΓΚΥΟΣ: ΝΑΙ ΟΧΙ
- ΠΟΛΗ: ΗΡΑΚΛΕΙΟ
- ΔΙΕΥΘΥΝΣΗ: ΠΑΝΟΣ 19
- ΤΗΛΕΦΩΝΟ: 6955312546

At the bottom of the form is a button labeled "ΣΥΝΕΧΕΙΑ" (Continue). A red error message at the bottom right states "All Fields must be completed!". The background features a blue globe and a white ECG line.

Εικόνα 26: Στοιχεία ασθενή

ΕΙΣΑΓΕΤΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΞΕΤΑΣΕΩΝ:

ΛΕΥΚΑ ΑΙΜΟΣΦΑΙΡΙΑ	<input type="text" value="4300"/>	ΑΙΜΟΣΦΑΙΡΙΝΗ	<input type="text" value="9"/>
ΜΕΣΟΣ ΟΓΚΟΣ ΕΡΥΘΡΩΝ	<input type="text" value="75"/>	ΑΙΜΑΤΟΚΡΙΤΗΣ	<input type="text" value="30"/>
ΑΙΜΟΠΕΤΑΛΙΑ	<input type="text" value="180"/>	ΦΕΡΙΤΙΝΗ	<input type="text" value="18"/>
ΔΕΙΚΤΗΣ ΕΡΥΘΡΟΚΥΤΤΑΡΩΝ	<input type="text" value="1"/>	ΣΙΔΗΡΟΣ	<input type="text" value="41"/>
ΕΡΥΘΡΑ ΑΙΜΟΣΦΑΙΡΙΑ	<input type="text" value="3.7"/>	ΤΙΒC	<input type="text" value="250"/>
<input type="checkbox"/> ΛΟΙΠΕΣ ΕΞΕΤΑΣΕΙΣ			

All Fields must be completed!

Εικόνα 27: Δεδομένα εξετάσεων

ΕΠΙΛΕΞΤΕ ΤΑ ΣΥΜΠΤΩΜΑΤΑ:

<input type="checkbox"/> ΠΟΝΟΚΕΦΑΛΟΣ	<input type="checkbox"/> ΔΙΣΘΗΜΑ_ΚΡΥΟΥ	<input type="checkbox"/> ΩΧΡΟΤΗΤΑ_ΔΕΡΜΑΤΟΣ	<input type="checkbox"/> ΑΠΩΛΕΙΑ_ΒΑΡΟΥΣ
<input type="checkbox"/> ΖΑΛΑΔΑ	<input type="checkbox"/> ΕΠΩΔΥΝΗ_ΣΠΛΗΝΑ	<input checked="" type="checkbox"/> ΔΥΣΠΝΟΙΑ	<input type="checkbox"/> ΠΛΗΓΕΣ
<input checked="" type="checkbox"/> ΤΑΧΥΚΑΡΔΙΑ	<input type="checkbox"/> ΑΔΥΝΑΜΙΑ	<input type="checkbox"/> ΑΠΩΛΕΙΑ_ΟΡΕΞΗΣ	<input type="checkbox"/> ΠΟΝΟΣ_ΣΤΑ_ΑΚΡΑ
<input type="checkbox"/> ΚΟΥΡΑΣΗ	<input type="checkbox"/> ΓΛΩΣΣΙΤΙΔΑ	<input checked="" type="checkbox"/> ΠΡΟΒΛΗΜΑΤΑ_ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ	<input checked="" type="checkbox"/> ΕΛΛΕΙΠΗΣ_ΔΙΑΤΡΟΦΗ
<input type="checkbox"/> ΦΑΡΜΑΚΑ	<input type="checkbox"/> ΑΛΚΟΟΛ	ΕΞΩΤΕΡΙΚΗ ΟΨΗ ΠΡΟΣΩΠΟΥ	<input type="radio"/> ΚΑΝΟΝΙΚΗ
		ΓΟΝΕΑΣ 1	<input type="radio"/> ΙΔΙΑΙΤΕΡΗ
		<input type="radio"/> ΟΧΙ	
		<input type="radio"/> ΣΤΙΓΜΑ	
		<input type="radio"/> ΜΕΣΟΓΕΙΑΚΗ	
		ΓΟΝΕΑΣ 2	<input type="radio"/> ΟΧΙ
		<input type="radio"/> ΣΤΙΓΜΑ	
		<input type="radio"/> ΜΕΣΟΓΕΙΑΚΗ	

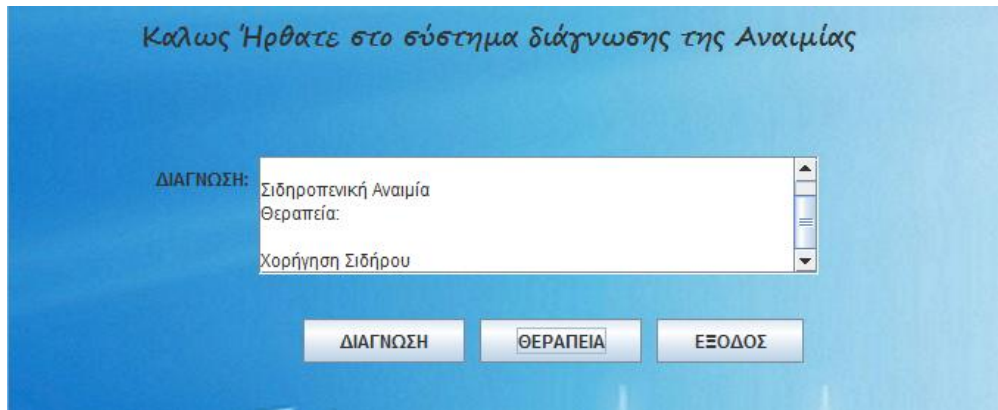
Εικόνα 28: Συμπτώματα ασθενούς

Καλώς Ήρθατε στο σύστημα διάγνωσης της Αναιμίας

ΔΙΑΓΝΩΣΗ: Με τις δοθείσες πληροφορίες πιάσχετε από:

Σιδηροπενική Αναιμία

Εικόνα 29: Διάγνωση ασθενούς

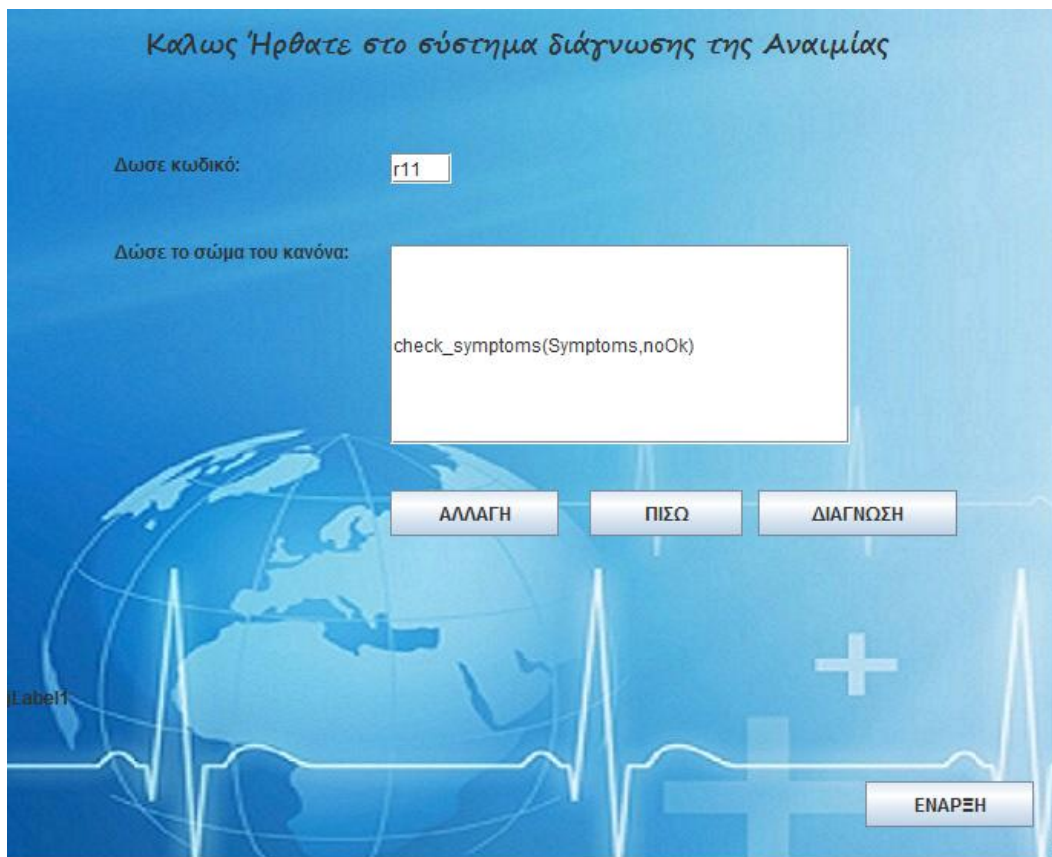


Εικόνα 30: Θεραπεία ασθενούς

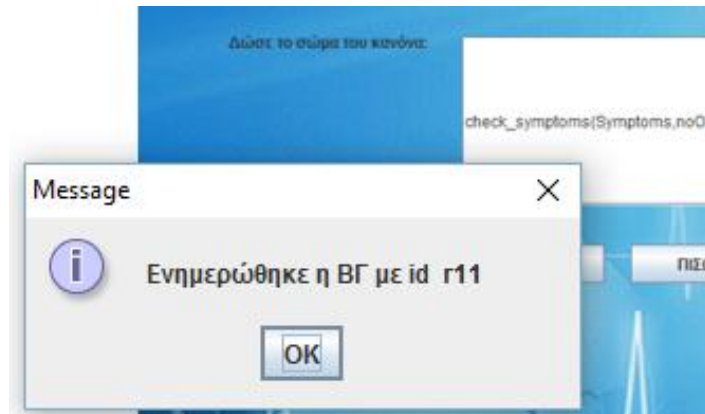
II. ΕΝΗΜΕΡΩΣΗ ΒΑΣΗΣ ΚΑΝΟΝΩΝ

Τα επόμενα 2 παράθυρα μας κάνουν επίδειξη της διαδικασίας ενημέρωσης της βάσης κανόνων. Ας πάρουμε ως παράδειγμα την αλλαγή ενός κανόνα.

- Στο παράθυρο «Εικόνα 31: Δεδομένα για αλλαγή κανόνα» δίνεται ο κωδικός του κανόνα που προορίζεται για αλλαγή καθώς επίσης και το νέο σώμα του κανόνα.
- Το παράθυρο «Εικόνα 32: Επιτυχής αλλαγή κανόνα» εμφανίζεται όταν η αλλαγή του κανόνα γίνει με επιτυχία.



Εικόνα 31: Δεδομένα για αλλαγή κανόνα



Εικόνα 32: Επιτυχής αλλαγή κανόνα

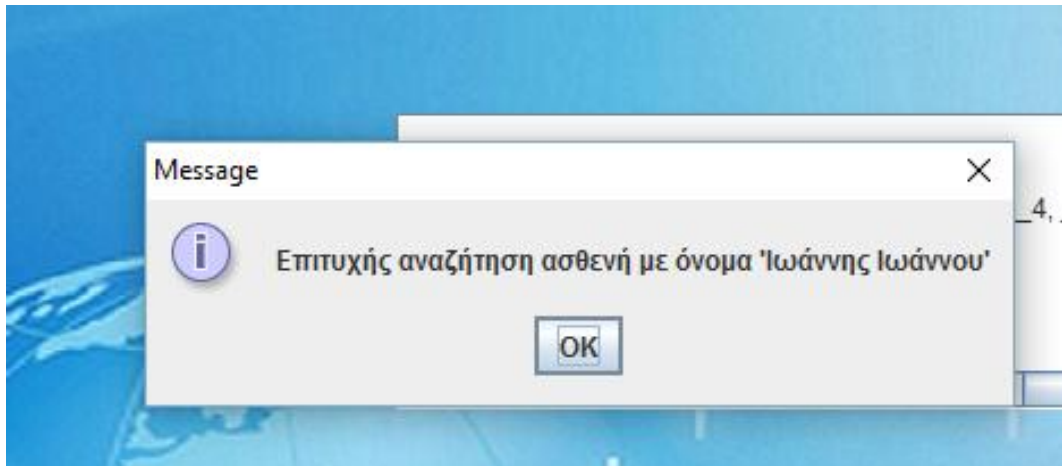
III. ΑΝΑΖΗΤΗΣΗ ΙΣΤΟΡΙΚΟΥ

Τα επόμενα 3 παράθυρα μας κάνουν επίδειξη της αναζήτησης ιστορικού ενός ασθενή.

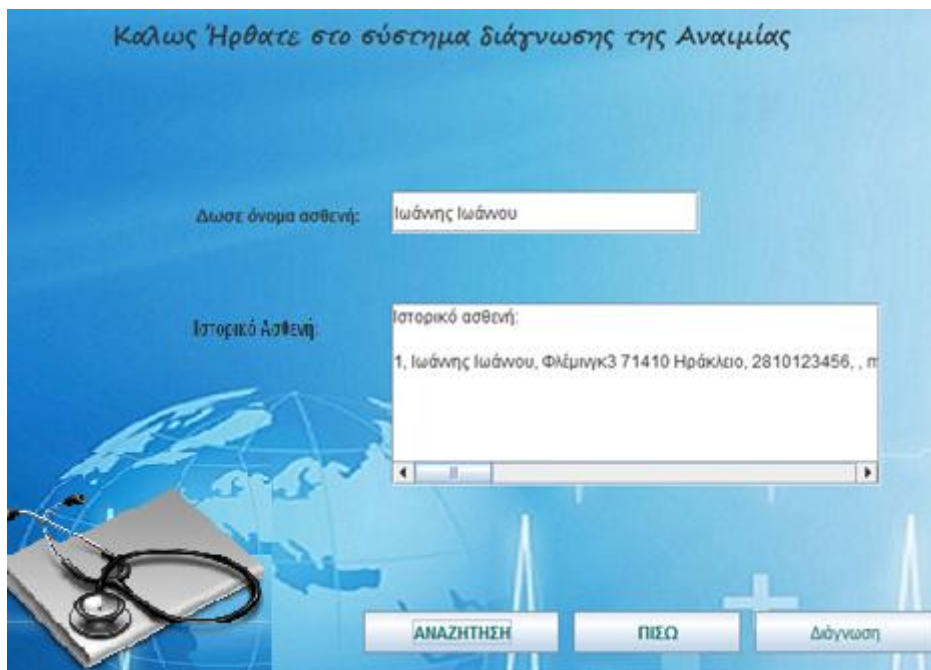
- Στο παράθυρο «Εικόνα 33: Αναζήτηση Ασθενούς» γίνεται η εισαγωγή του ονόματος του ασθενή, στο αντίστοιχο πεδίο.
- Το παράθυρο «Εικόνα 34: Επιτυχής αναζήτηση ασθενούς» εμφανίζεται όταν η αναζήτηση είναι επιτυχής.
- Στο παράθυρο «Εικόνα 35: Εμφάνιση δεδομένων ασθενή» εμφανίζεται όλο το ιστορικό του ασθενή στο πεδίο «Ιστορικό Ασθενή».



Εικόνα 33: Αναζήτηση Ασθενούς



Εικόνα 34: Επιτυχής αναζήτηση ασθενούς



Εικόνα 35: Εμφάνιση δεδομένων ασθενή

IV Παραδείγματα επίδειξης επιτυχούς και ενεπιτυχούς διάγνωσης

Ακολουθεί η επίδειξη 4 παραδειγμάτων. Και τα 4 αφορούν επιτυχή διάγνωση για α) Σιδηροπενική αναιμία, β) Στίγμα μεσογειακής αναιμίας, γ) Μεσογειακή αναιμία και δ) B12 αναιμία.

➤ ΣΙΔΗΡΟΠΕΝΙΚΗ ΑΝΑΙΜΙΑ

ΣΤΟΙΧΕΙΑ ΑΣΘΕΝΗ:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΔΗΜΗΤΡΗΣ ΠΑΠΑΔΗΜΗΤΡΙΟΥ

ΗΛΙΚΙΑ 45

ΦΥΛΟ ΑΝΔΡΑΣ ΓΥΝΑΙΚΑ ΝΑΙ ΟΧΙ

ΠΟΛΗ ΗΡΑΚΛΕΙΟ

ΔΙΕΥΘΥΝΣΗ ΦΛΕΜΙΓΚ 3

ΤΗΛΕΦΩΝΟ 6977145125

ΣΥΝΕΧΕΙΑ

All Fields must be completed!

Εικόνα 36: Σιδηροπενική_Αναιμία:Στοιχεία Ασθενή

ΕΙΣΑΓΕΤΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΞΕΤΑΣΕΩΝ:

ΛΕΥΚΑ ΑΙΜΟΣΦΑΙΡΙΑ	4800	ΑΙΜΟΣΦΑΙΡΙΝΗ	12
ΜΕΣΟΣ ΟΓΚΟΣ ΕΡΥΘΡΩΝ	75	ΑΙΜΑΤΟΚΡΙΤΗΣ	33
ΑΙΜΟΠΕΤΑΛΙΑ	125	ΦΕΡΙΤΙΝΗ	18
ΔΕΙΚΤΗΣ ΕΡΥΘΡΟΚΥΤΤΑΡΩΝ	1	ΣΙΔΗΡΟΣ	45
ΕΡΥΘΡΑ ΑΙΜΟΣΦΑΙΡΙΑ	4.3	ΤΙΒC	280

ΛΟΙΠΕΣ ΕΞΕΤΑΣΕΙΣ

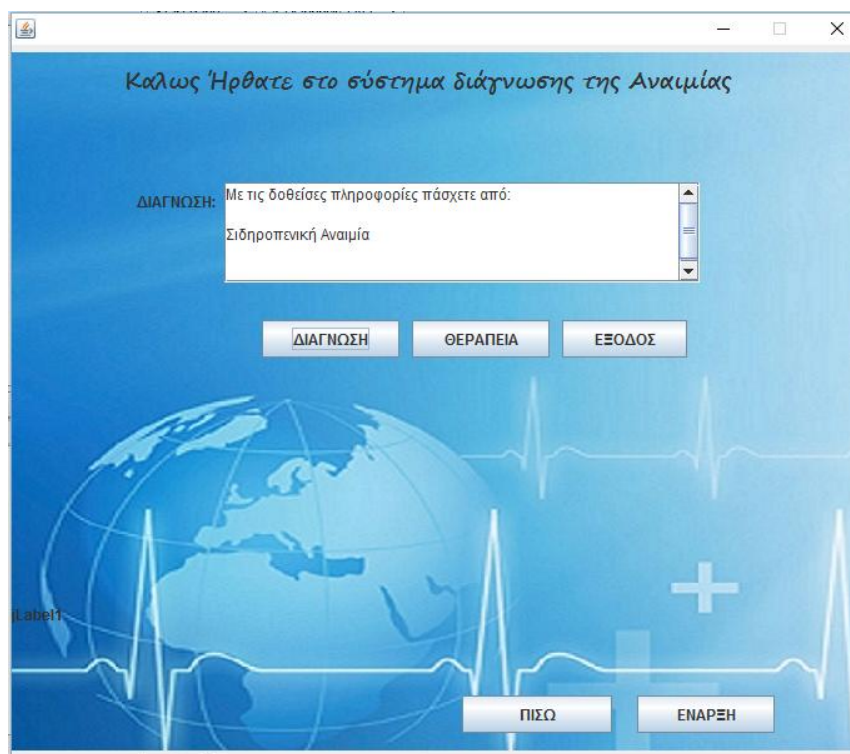
ΔΙΣΑ ΠΙΣΩ

All Fields must be completed!

Εικόνα 37: Σιδηροπενική_Αναιμία:Εξετάσεις ΓΕΑ



Εικόνα 38: Σιδηροπενική_Αναιμία:Συμπτώματα



Εικόνα 39: Σιδηροπενική_Αναιμία:Διάγνωση

➤ ΣΤΙΓΜΑ ΜΕΣΟΓΕΙΑΚΗΣ ΑΝΑΙΜΙΑΣ

ΣΤΟΙΧΕΙΑ ΑΣΘΕΝΗ:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΜΑΡΙΑ ΠΑΠΑΝΙΚΟΛΑΟΥ

ΗΛΙΚΙΑ: 23

ΦΥΛΟ:

 ΑΝΔΡΑΣ

 ΓΥΝΑΙΚΑ

ΕΓΚΥΟΣ:

 ΝΑΙ

 ΟΧΙ

ΠΟΛΗ: ΗΡΑΚΛΕΙΟ

ΔΙΕΥΘΥΝΣΗ: ΚΑΡΑΜΑΝΛΗ 15

ΤΗΛΕΦΩΝΟ: 6977145124

ΣΥΝΕΧΕΙΑ

All Fields must be completed!

Εικόνα 40: Στίγμα_Μεσογειακης:Στοιχεία Ασθενή

ΕΙΣΑΓΕΤΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΞΕΤΑΣΕΩΝ:

ΛΕΥΚΑ ΑΙΜΟΣΦΑΙΡΙΑ	4800	ΑΙΜΟΣΦΑΙΡΙΝΗ	13
ΜΕΣΟΣ ΟΓΚΟΣ ΕΡΥΘΡΩΝ	78	ΑΙΜΑΤΟΚΡΙΤΗΣ	39
ΑΙΜΟΠΕΤΑΛΙΑ	155	ΦΕΡΙΤΙΝΗ	15
ΔΕΙΚΤΗΣ ΕΡΥΘΡΟΚΥΤΤΑΡΩΝ	1	ΣΙΔΗΡΟΣ	45
ΕΡΥΘΡΑ ΑΙΜΟΣΦΑΙΡΙΑ	7.8	ΤΙΒC	320

ΛΟΙΠΕΣ ΕΞΕΤΑΣΕΙΣ

ΔΙΣΑ ΠΙΣΩ

All Fields must be completed!

Εικόνα 41: Στίγμα_Μεσογειακης:Εξετάσεις

ΕΠΙΛΕΞΤΕ ΤΑ ΣΥΜΠΤΩΜΑΤΑ:

ΠΟΝΟΚΕΦΑΛΟΣ ΔΙΣΘΗΜΑ_ΚΡΥΟΥ ΩΧΡΟΤΗΤΑ_ΔΕΡΜΑΤΟΣ ΑΠΩΛΕΙΑ_ΒΑΡΟΥΣ

ΖΑΛΑΔΑ ΕΠΩΔΥΝΗ_ΣΠΛΗΝΑ ΔΥΣΠΝΟΙΑ ΠΛΗΓΕΣ

ΤΑΧΥΚΑΡΔΙΑ ΑΔΥΝΑΜΙΑ ΑΠΩΛΕΙΑ_ΟΡΕΞΗΣ ΠΟΝΟΣ_ΣΤΑ_ΑΚΡΑ

ΚΟΥΡΑΣΗ ΓΛΩΣΣΙΤΙΔΑ ΠΡΟΒΛΗΜΑΤΑ_ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ ΕΛΛΕΙΠΗΣ_ΔΙΑΤΡΟΦΗ

ΦΑΡΜΑΚΑ ΑΛΚΟΟΛ

ΕΞΩΤΕΡΙΚΗ ΟΨΗ ΠΡΟΣΩΠΟΥ

ΚΑΝΟΝΙΚΗ

ΙΔΙΑΙΤΕΡΗ

ΓΟΝΕΑΣ 1 ΟΧΙ ΓΟΝΕΑΣ 2 ΟΧΙ

ΣΤΙΓΜΑ ΣΤΙΓΜΑ

ΜΕΣΟΓΕΙΑΚΗ ΜΕΣΟΓΕΙΑΚΗ

Εικόνα 42: Στίγμα_Μεσογειακής:Συμπτώματα

Καλώς Ήρθατε στο σύστημα διάγνωσης της Αναιμίας

ΔΙΑΓΝΩΣΗ: Με τις δοθείσες πληροφορίες πάσχετε από:

Στίγμα Θαλασσαιμίας

Εικόνα 43: Στίγμα_Μεσογειακής:Διάγνωση

➤ ΜΕΣΟΓΕΙΑΚΗ ΑΝΑΙΜΙΑ

ΣΤΟΙΧΕΙΑ ΑΣΘΕΝΗ:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΓΙΩΡΓΟΣ ΓΕΩΡΓΙΟΥ

ΗΛΙΚΙΑ: 34

ΦΥΛΟ: ΑΝΔΡΑΣ (ΓΥΝΩΣ) ΝΑΙ
 ΓΥΝΑΙΚΑ ΟΧΙ

ΠΟΛΗ: ΗΡΑΚΛΕΙΟ

ΔΙΕΥΘΥΝΣΗ: ΑΘΗΤΑΚΗ 14

ΤΗΛΕΦΩΝΟ: 6988547545

ΣΥΝΕΧΕΙΑ

All Fields must be completed!

Εικόνα 44: Μεσογειακή_Αναιμία:Στοιχεία Ασθενή

ΕΙΣΑΓΕΤΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΞΕΤΑΣΕΩΝ:

ΛΕΥΚΑ ΑΙΜΟΣΦΑΙΡΙΑ	4400	ΔΕΙΚΤΗΣ ΕΡΥΘΡΟΚΥΤΤΑΡΩΝ	1
ΜΕΣΟΣ ΟΓΚΟΣ ΕΡΥΘΡΩΝ	75	ΕΡΥΘΡΑ ΑΙΜΟΣΦΑΙΡΙΑ	5,5
ΣΙΔΗΡΟΣ	45	ΦΕΡΙΤΙΝΗ	22
ΤΙΒC	320	ΑΙΜΑΤΟΚΡΙΤΗΣ	39
ΑΙΜΟΣΦΑΙΡΙΝΗ	13	ΑΙΜΟΠΕΤΑΛΙΑ	125

ΛΟΙΠΕΣ ΕΞΕΤΑΣΕΙΣ

ΔίΣΑ ΠΙΣΩ

All Fields must be completed!

Εικόνα 45: Μεσογειακή_Αναιμία:Εξετάσεις ΓΕΑ

ΕΙΣΑΓΕΤΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΞΕΤΑΣΕΩΝ:

B12 ΒΙΤΑΜΙΝΗ	<input type="text"/>	HbA1	<input type="text" value="4"/>
ΦΥΛΙΚΟ ΟΞΥ	<input type="text"/>	HbA2	<input type="text"/>
ΣΧΗΜΑ ΕΡΥΘΡΩΝ	<input checked="" type="radio"/> ΦΥΣΙΟΛΟΓΙΚΟ	HbF	<input type="text" value="2"/>
	<input type="radio"/> ΔΡΕΠΑΝΟΕΙΔΣ	HbS	<input type="text"/>
	<input type="radio"/> ΣΦΑΙΡΙΚΟ		

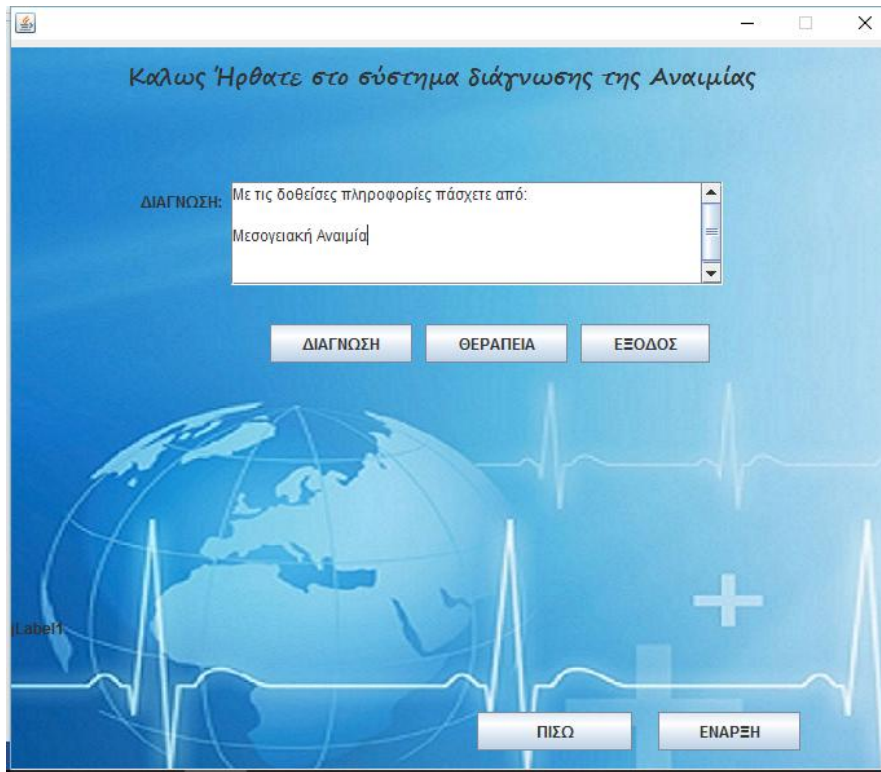
Label1

Εικόνα 46: Μεσογειακή_Αναιμία:Εξτρα εξετάσεις

ΕΠΙΛΕΞΤΕ ΤΑ ΣΥΜΠΤΩΜΑΤΑ:

<input checked="" type="checkbox"/> ΠΟΝΟΚΕΦΑΛΟΣ	<input type="checkbox"/> ΔΙΣΘΗΜΑ_ΚΡΥΟΥ	<input checked="" type="checkbox"/> ΩΧΡΟΤΗΤΑ_ΔΕΡΜΑΤΟΣ	<input type="checkbox"/> ΑΠΟΛΕΙΑ_ΒΑΡΟΥΣ
<input type="checkbox"/> ΖΑΛΑΔΑ	<input type="checkbox"/> ΕΠΟΔΥΝΗ_ΣΠΛΗΝΑ	<input type="checkbox"/> ΔΥΣΠΝΟΙΑ	<input type="checkbox"/> ΠΛΗΓΕΣ
<input type="checkbox"/> ΤΑΧΥΚΑΡΔΙΑ	<input type="checkbox"/> ΑΔΥΝΑΜΙΑ	<input type="checkbox"/> ΑΠΟΛΕΙΑ_ΟΡΕΞΗΣ	<input type="checkbox"/> ΠΟΝΟΣ_ΣΤΑ_ΑΚΡΑ
<input type="checkbox"/> ΚΟΥΡΑΣΗ	<input type="checkbox"/> ΓΛΩΣΣΙΠΙΔΑ	<input type="checkbox"/> ΠΡΟΒΛΗΜΑΤΑ_ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ	<input type="checkbox"/> ΕΛΛΕΙΠΗΣ_ΔΙΑΤΡΟΦΗ
<input type="checkbox"/> ΦΑΡΜΑΚΑ	<input type="checkbox"/> ΑΛΚΟΟΛ	ΕΞΩΤΕΡΙΚΗ ΟΨΗ ΠΡΟΣΩΠΟΥ	<input checked="" type="radio"/> ΚΑΝΟΝΙΚΗ
			<input type="radio"/> ΙΔΙΑΙΤΕΡΗ
ΓΟΝΕΑΣ 1	<input type="radio"/> ΟΧΙ	ΓΟΝΕΑΣ 2	<input type="radio"/> ΟΧΙ
	<input type="radio"/> ΣΤΙΓΜΑ		<input checked="" type="radio"/> ΣΤΙΓΜΑ
	<input checked="" type="radio"/> ΜΕΣΟΓΕΙΑΚΗ		<input type="radio"/> ΜΕΣΟΓΕΙΑΚΗ

Εικόνα 47: : Μεσογειακή_Αναιμία:Συμπτώματα



Εικόνα 48: Μεσογειακή_Αναιμία:Διάγνωση

➤ **B12 ΑΝΑΙΜΙΑ**

ΣΤΟΙΧΕΙΑ ΑΣΘΕΝΗ:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΚΩΝΣΤΑΝΤΙΝΑ ΠΟΘΑΚΟΥ

ΗΛΙΚΙΑ 18

ΦΥΛΟ ΑΝΔΡΑΣ ΓΥΝΑΙΚΑ ΕΓΚΥΟΣ ΝΑΙ ΟΧΙ

ΠΟΛΗ ΗΡΑΚΛΕΙΟ

ΔΙΕΥΘΥΝΣΗ ΗΡΑΚΛΕΙΟ

ΤΗΛΕΦΩΝΟ 6944587845

ΣΥΝΕΧΕΙΑ

All Fields must be completed!

Εικόνα 49: B12_Αναίμια:Στοιχεία Ασθενή

ΕΙΣΑΓΕΤΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΞΕΤΑΣΕΩΝ:

ΛΕΥΚΑ ΑΙΜΟΣΦΑΙΡΙΑ 4400 ΔΕΙΚΤΗΣ ΕΡΥΘΡΟΚΥΤΤΑΡΩΝ 1

ΜΕΣΟΣ ΟΓΚΟΣ ΕΡΥΘΡΩΝ 100 ΕΡΥΘΡΑ ΑΙΜΟΣΦΑΙΡΙΑ 4.2

ΣΙΔΗΡΟΣ 45 ΦΕΡΙΤΙΝΗ 22

ΤΙΒC 320 ΑΙΜΑΤΟΚΡΙΤΗΣ 39

ΑΙΜΟΣΦΑΙΡΙΝΗ 13 ΑΙΜΟΠΕΤΑΛΙΑ 170

ΛΟΙΠΕΣ ΕΞΕΤΑΣΕΙΣ

ΔΙΑ ΔΙΑ ΠΙΣΩ

All Fields must be completed!

Εικόνα 50: B12_Αναίμια:Εξετάσεις ΓΕΑ

ΕΙΣΑΓΕΤΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΞΕΤΑΣΕΩΝ:

B12 ΒΙΤΑΜΙΝΗ	<input type="text" value="125"/>	HbA1	<input type="text"/>
ΦΥΛΙΚΟ ΟΞΥ	<input type="text" value="4"/>	HbA2	<input type="text"/>
ΣΧΗΜΑ ΕΡΥΘΡΩΝ	<input type="radio"/> ΦΥΣΙΟΛΟΓΙΚΟ <input type="radio"/> ΔΡΕΠΑΝΟΕΙΔΕΣ <input type="radio"/> ΣΦΑΙΡΙΚΟ	HbF	<input type="text"/>
		HbS	<input type="text"/>

ΔΙΑ ΔΙΑ

ΠΙΣΩ

Εικόνα 51: B12_Αναίμια: Εξτρα εξετάσεις

ΕΠΙΛΕΞΤΕ ΤΑ ΣΥΜΠΤΩΜΑΤΑ:

<input type="checkbox"/> ΠΟΝΟΚΕΦΑΛΟΣ	<input type="checkbox"/> ΑΙΣΘΗΜΑ_ΚΡΥΟΥ	<input type="checkbox"/> ΟΧΡΟΤΗΤΑ_ΔΕΡΜΑΤΟΣ	<input checked="" type="checkbox"/> ΑΠΩΛΕΙΑ_ΒΑΡΟΥΣ
<input type="checkbox"/> ΖΑΛΑΔΑ	<input type="checkbox"/> ΕΠΩΔΥΝΗ_ΣΠΛΗΝΑ	<input type="checkbox"/> ΔΥΣΠΝΟΙΑ	<input type="checkbox"/> ΠΛΗΓΕΣ
<input type="checkbox"/> ΤΑΧΥΚΑΡΔΙΑ	<input type="checkbox"/> ΑΔΥΝΑΜΙΑ	<input checked="" type="checkbox"/> ΑΠΩΛΕΙΑ_ΟΡΕΞΗΣ	<input type="checkbox"/> ΠΟΝΟΣ_ΣΤΑ_ΑΚΡΑ
<input type="checkbox"/> ΚΟΥΡΑΣΗ	<input type="checkbox"/> ΓΛΩΣΣΙΠΙΔΑ	<input type="checkbox"/> ΠΡΟΒΛΗΜΑΤΑ_ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ	<input type="checkbox"/> ΕΛΛΕΙΠΗΣ_ΔΙΑΤΡΟΦΗ
<input type="checkbox"/> ΦΑΡΜΑΚΑ	<input type="checkbox"/> ΑΛΚΟΟΛ	ΕΞΩΤΕΡΙΚΗ ΟΨΗ ΠΡΟΣΩΠΟΥ	<input type="radio"/> ΚΑΝΟΝΙΚΗ
			<input type="radio"/> ΙΔΙΑΙΤΕΡΗ
ΓΟΝΕΑΣ 1	<input type="radio"/> ΟΧΙ	ΓΟΝΕΑΣ 2	<input type="radio"/> ΟΧΙ
	<input type="radio"/> ΣΤΙΓΜΑ		<input type="radio"/> ΣΤΙΓΜΑ
	<input type="radio"/> ΜΕΣΟΓΕΙΑΚΗ		<input type="radio"/> ΜΕΣΟΓΕΙΑΚΗ

ΣΥΝΕΧΕΙΑ

ΠΙΣΩ

Εικόνα 52: B12_Αναίμια: Συμπτώματα

Καλως Ήρθατε στο σύστημα διάγνωσης της Αναιμίας

ΔΙΑΓΝΩΣΗ:

Με τις δοθείσες πληροφορίες πάσχετε από:

B12 Αναιμία

ΔΙΑΓΝΩΣΗ

ΘΕΡΑΠΕΙΑ

ΕΞΟΔΟΣ



ΠΙΣΩ

ΕΝΑΡΞΗ

Εικόνα 53: B12_Αναιμία:Διάγνωση

9. Απόδοση του διαγνωστικού συστήματος ΔιΣΑ

Εξετάσαμε το σύστημα ΔιΣΑ σε 22 βεβαιωμένα δεδομένα τα οποία είχαμε στη διάθεση μας. Το ΔιΣΑ διάγνωσε σωστά όλες τις περιπτώσεις που εξετάσαμε, με βάση τους κανόνες και τα στοιχεία που εισήχθησαν στη βάση κανόνων, δεν διαγνώστηκε κάποιο λάθος αποτέλεσμα.

10. Συμπεράσματα και Μελλοντικές Επεκτάσεις

Αναιμία είναι η πτώση του αιματοκρίτη ή της αιμοσφαιρίνης ή Κκάποιων στοιχείων του αίματος σε τιμές κάτω των φυσιολογικών. Μπορεί να οφείλεται σε πλήθος αιτιών, οι οποίες ταξινομούνται σε τρεις ομάδες: α) σε μειωμένη παραγωγή ερυθρών αιμοσφαιρίων, β) σε αυξημένη καταστροφή ερυθρών αιμοσφαιρίων, γ) σε απώλεια αίματος. Στην πρώτη ομάδα βρίσκονται κυρίως οι σιδηροπενικές και οι μεγαλοβλαστικές αναιμίες. Στη δεύτερη, οι αιμολυτικές δηλαδή οι αιμοσφαιρινοπάθειες και οι θαλασσαιμίες. Στην Τρίτη ομάδα, υπάρχουν οι αναιμίες που οφείλονται σε αιμορραγίες.

Οι αναιμίες μπορεί να είναι κληρονομικές και επίκτητες. Οι κληρονομικές μεταβιβάζονται από γενιά σε γενιά και ακολουθούν τους νόμους της κληρονομικότητας (*«Εικόνα 10:πιθανότητα πάθησης τέκνων όταν ο ένας γονέας είναι υγιής και ο άλλος φορέας β μεσογειακής αναιμίας»*). Οι επίκτητες μπορεί να συμβούν σε κάθε άτομο είτε από παράγοντα είτε από απώλεια αίματος ή από τοξική επίδραση εξωγενών αιτιών στο αιμοποιητικό σύστημα.

Η κλινική εικόνα των αναιμιών περιλαμβάνει συμπτώματα για όλες τις αναιμίες γενικά, τα οποία προκύπτουν από την πτώση του αιματοκρίτη καθώς και συμπτώματα ειδικά για κάθε αναιμία. Τα συμπτώματα μπορεί να οφείλονται είτε στην επίδραση της έλλειψης του αιμοποιητικού παράγοντα και από άλλα συστήματα που τον χρησιμοποιούν (για παράδειγμα έλλειψη σιδήρου από το πεπτικό ή έλλειψη B12 από το νευρικό σύστημα), είτε στη βασική νόσο που προκάλεσε την αναιμία, ή σε επιδράσεις της θεραπείας στον οργανισμό (συσσώρευση σιδήρου από μεταγγίσεις).

Η θεραπεία της αναιμίας είναι ανάλογη με το αίτιο. Μπορεί να είναι απλή χορήγησης ενός αιμοποιητικού παράγοντα (π.χ. μασώμενα δισκία σιδήρου για ορισμένο διάστημα) ή και ιδιαίτερα πολύπλοκη όπως οι μεταγγίσεις δια βίου.

Τα αποτελέσματα της παρούσας εργασίας δείχνουν την επιτυχή εφαρμογή της τεχνολογίας της Τεχνητής Νοημοσύνης σ' ένα πεδίο στο οποίο η εξειδικευμένη γνώση περιορίζεται σ' ένα μικρό αριθμό ειδικών. Συνεπώς, αυτή η γνώση είναι σπάνια, δαπανηρή στην απόκτηση της από ανθρώπους ειδικούς και δαπανηρή στη χρήση της για τις ανάγκες των τελικών χρηστών. Στο σύστημα ΔιΣΑ έγινε μεταφορά της ιατρικής γνώσης από βιβλία καθώς και της εμπειρίας ειδικών γιατρών σε ένα σύγχρονο αυτοματοποιημένο σύστημα διαχείρισης της. Το σύστημα ΔιΣΑ κάνει διάγνωση για τον τύπο της αναιμίας ενός ασθενούς με βάση εργαστηριακές εξετάσεις, επιπλέον προτείνει την θεραπεία που πρέπει να ακολουθηθεί. Η απόδοση του συστήματος μας μετρήθηκε σε 22 βεβαιωμένες περιπτώσεις αναιμίας και η απόδοση του ήταν 100%.

Τα ιατρικά διαγνωστικά συστήματα τα οποία έχουν υψηλό βαθμό απόδοσης σε ορθή διάγνωση μπορούν να χρησιμοποιηθούν είτε από ένα έμπειρο γιατρό συμβουλευτικά ή για την εκπαίδευση ενός ειδικευόμενου ιατρού. Η άμεση αλληλεπίδραση τους με τον ασθενή θα μπορούσε να γίνει αλλά σε κάθε περίπτωση χρειάζεται η εξειδικευμένη γνώση ενός ειδικού είτε γιατρού ή νοσηλευτικού προσωπικού για να καταχωρήσει τα ιατρικά δεδομένα. Θα μπορούσε ένα ιατρικό διαγνωστικό σύστημα αυτόματα να δέχεται ως είσοδο τις ιατρικές εξετάσεις και να προχωρεί στη διάγνωση. Επιπλέον, εφόσον ο ασθενής αλληλεπιδρά χρησιμοποιώντας άμεσα το διαγνωστικό σύστημα θα πρέπει η διεπαφή του διαγνωστικού

συστήματος να είναι σε φυσική γλώσσα, φιλική και απλή ώστε να είναι κατανοητή από τον χρήστη-ασθενή.

Το σύστημα ΔιΣΑ είναι ένα πρότυπο διαγνωστικό σύστημα. Θα μπορούσε να χρησιμοποιηθεί πιλοτικά σε ένα νοσοκομείο ώστε να μετρηθεί ενδελεχώς η απόδοση του και να βελτιωθεί ακόμα περισσότερο η απόδοση του ώστε να φτάσει στο επίπεδο του εμπορεύσιμου προϊόντος. Μια μελλοντική επέκταση του θα ήταν να δέχεται αυτόματα τις αιματολογικές εξετάσεις ενός ασθενούς από ένα αρχείο και να προχωρούσε αυτόματα στη διάγνωση. Η αλληλεπίδραση με το χρήστη να περιοριζόταν σε θέματα διαχειριστικά του συστήματος.

11. Βιβλιογραφία

11.1. Βιβλία

1. Ι.Βλαχάβας, Π.Κεφαλάς, Ν.Βασιλειάδης, Φ.Κόκκορας, Η.Σακελλαρίου, Τεχνητή νοημοσύνη, Γ' Έκδοση, Εκδόσεις Πανεπιστημίου Μακεδονίας, 2011.
2. Μ. Μαρακάκης. Prolog: Προγραμματισμός σε Λογική για Τεχνητή Νοημοσύνη, Εκδόσεις Νέων Τεχνολογιών, 2014.
3. Φ. Γερανιωτάκη, Γ.Μπόλλας, Ν.Σοφούλης, Αιματολογία- Αιμοδοσία II, 2^{ος} Κύκλος, 2001.
4. S. D.C Stern, A. S. Cifu, D. Alckorn, Symptoms to Diagnosis, 2η Έκδοση, Εκδότικός Οίκος: McGraw-Hill Education - Europe, 2009.
5. O.Rienhoff, U.Piccolo, B.Schneider, Experts systems and decision support in medicine, Springer-Verlag, 1988.
6. J.Gosling, B.Joy. G Steele, G. Bracha, A. Buckley, The Java® Language Specification, 7^η έκδοση, 2013
7. Davidson's, *Παθολογία*, 19η έκδοση, Ιατρικές εκδόσεις Π.Χ.Πασχαλίδης , 2005.
8. J. Ignizio, Introduction to Expert Systems, Εκδόσεις McGraw-Hill, 1991
9. Κυριάκος Σγάρμπας. Η γλώσσα προγραμματισμού Prolog, Εκδόσεις Πανεπιστημίου Πατρών, Β Έκδοση, 2016 ([Link](#))

11.2. Τοποθεσίες και άρθρα Web

10. J. Wielemaker, SWI-Prolog Manual, Εκδόσεις University of Amsterdam, 5.6 Έκδοση, 2008 ([Link](#))
11. P. Singleton, JPL 3.x Prolog API in SWI-Prolog documentation, 2004 ([Link1](#), [Link2](#))
12. P. Christiani, S. Quaglini, M. Stefanelli, An Expert System for Anemia Diagnosis , Εκδόσεις [Univercity of Pavia](#).

13. M. Eckardt, U.Ruch, M. Schnabel, Knowledge Based System for Diagnosing Anemia, Εκδόσεις [University of Munich](#).

12. Παράρτημα Α: Εγκατάσταση Βασικών Προγραμμάτων

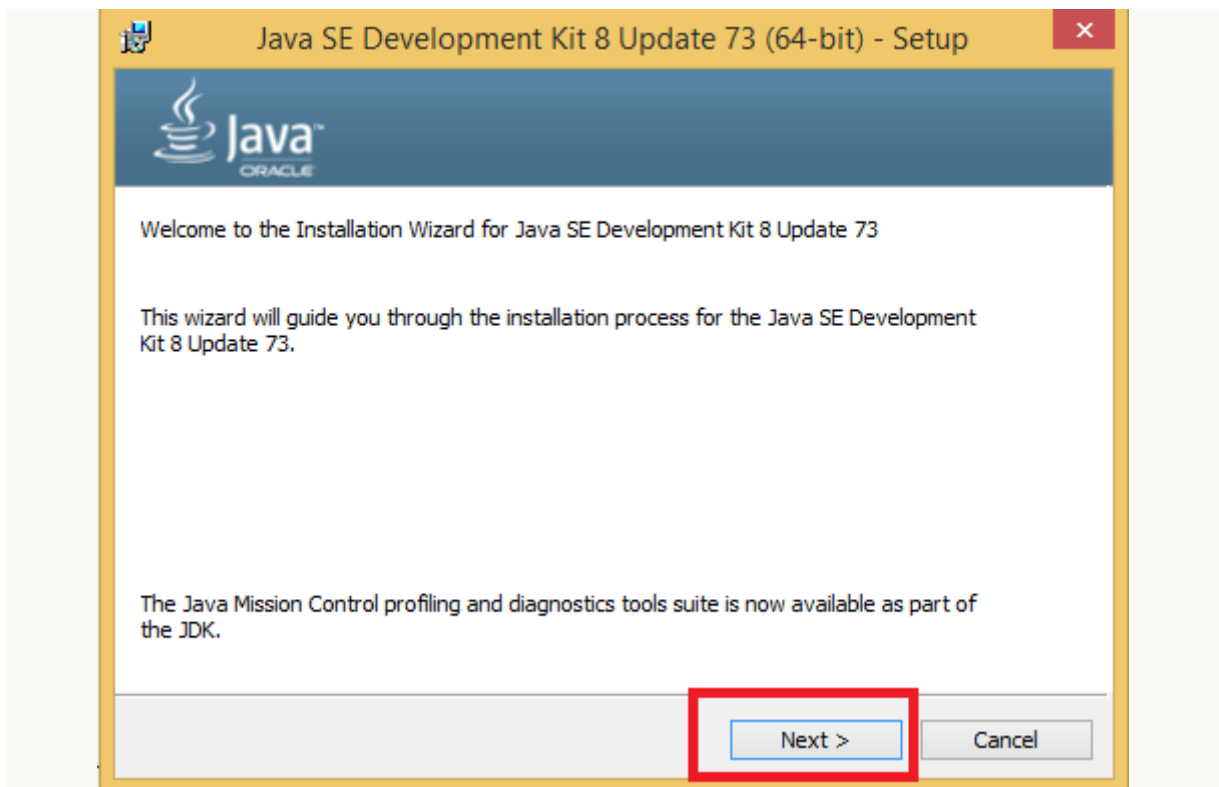
* Προσέχουμε να χρησιμοποιούμε την ίδια έκδοση σε όλα τα προγράμματα. Για παράδειγμα, εάν το λειτουργικό μας είναι 32-bit χρησιμοποιούμε σε όλα τα παρακάτω προγράμματα 32-bit και αντίστοιχα για 64-bit, σε διαφορετική περίπτωση δεν θα λειτουργεί σωστά το ΔΙΣΑ. *

12.1. Εγκατάσταση Netbeans

Κάνουμε εγκατάσταση του Netbeans και μετά το τέλος της εγκατάστασης προσθέτουμε το αρχείο Anemia στο φάκελο NetbeansProjects που δημιουργήθηκε στα έγγραφα σας. (αφού πρώτα το κάνουμε αποσυμπίεση). Μέσα σε αυτό το φάκελο υπάρχουν όλα τα αρχεία που χρειαζόμαστε. (το αρχείο "diagnosis.java" που είναι η διεπαφή, το αρχείο "anem.pl" που είναι το αρχείο της Prolog.) Μέσα στο φάκελο αυτό επίσης υπάρχουν και οι βιβλιοθήκες τις οποίες πρέπει να προσθέσουμε στο πρόγραμμα, εάν δεν υπάρχουν.

Η διαδικασία εγκατάστασης έχει ως εξής. Το πρώτο βήμα που έχουμε να κάνουμε, έτσι ώστε να έχουμε την δυνατότητα να δημιουργήσουμε εφαρμογές σε Java, είναι να κατεβάσουμε την τελευταία έκδοση του JDK από την Oracle: [DOWNLOAD JAVA](#)

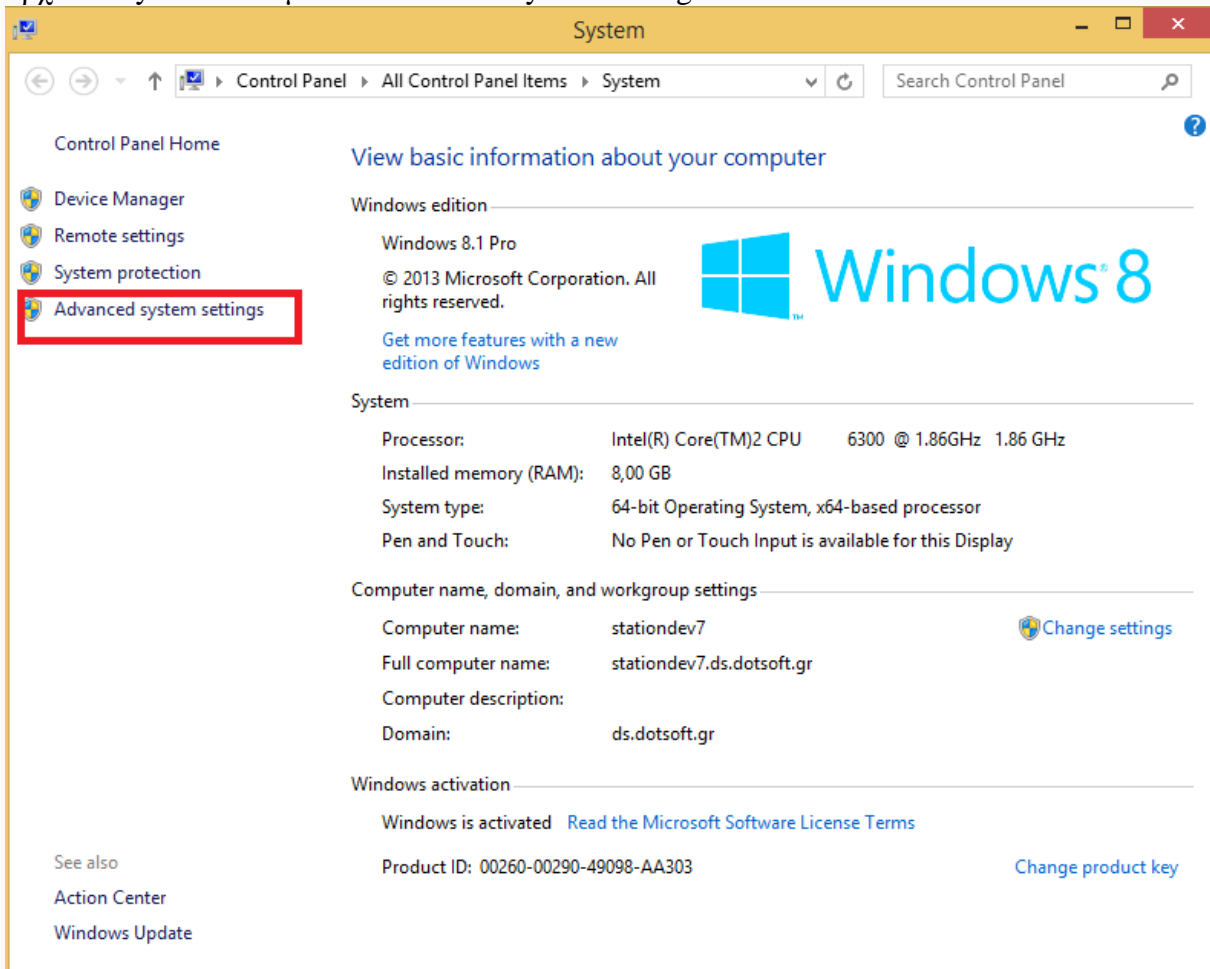
Θα χρησιμοποιήσουμε Java σε υπολογιστή που έχει Windows λειτουργικό σύστημα, οπότε θα κατεβάσω την Windows έκδοση του JDK. Όταν το κατέβασμα του αρχείου έχει ολοκληρωθεί, ξεκινήστε την εγκατάσταση του κάνοντας διπλό κλικ επάνω στο αρχείο JDK. Κατά την διαδικασία της εγκατάστασης απλά αποδεχτείτε όλες τις προεπιλεγμένες ρυθμίσεις:



Εικόνα 54:JDK εγκατάσταση

Αφού ολοκληρωθεί η εγκατάσταση του JDK, θα πρέπει να πραγματοποιήσουμε ένα ακόμα βήμα όσο αφορά το περιβάλλον των Windows: Να ενημερώσουμε το περιβάλλον των

Windows σε ποιο φάκελλο βρίσκεται το εκτελέσιμο αρχείο του JDK. Εάν έχετε Windows XP τότε θα χρειαστεί να πάτε στο «Start», μετά στο «Control Panel», αλλάζετε την μορφή σε «Classic View» και τέλος επιλέξτε «System». Εάν έχετε Windows 7 ή 8 τότε απλά επιλέξτε αρχικά «System» και μετά «Advanced SystemSettings»:

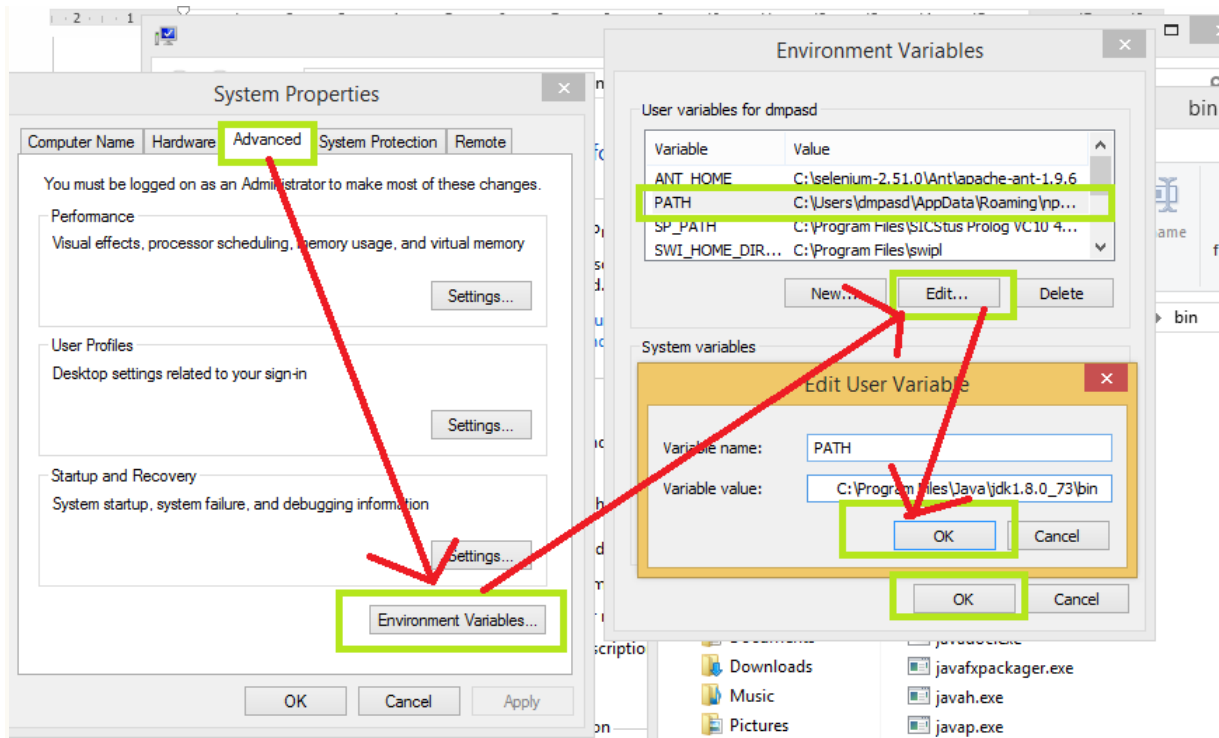


Εικόνα 55: Ρυθμίσεις για προχωρημένους

Στο παράθυρο «System Properties» κάνετε κλικ επάνω στην καρτέλα «Advanced» και μετά πατάτε το «Environment Variables» κουμπί. Βρείτε και επιλέξτε το όνομα Path. Κάντε κλικ στο «Edit»,» κουμπί. Προσθέστε το `jdk\bin` directory στην αρχή και τελειώστε την εντολή με semicolon (;) για να ξεχωρίσει από την επόμενη εντολή. Η εντολή σας θα πρέπει να μοιάζει κάπως έτσι:

```
C:\ProgramFiles\Java\jdk1.8.0_73\bin
```

Μετά την ολοκλήρωση αυτής της διαδικασίας χρειάζεται να κάνετε επανεκκίνηση του υπολογιστή σας για οριστικοποιηθούν οι αλλαγές που κάνατε.



Εικόνα 56:JavaPath(μονοπάτι)

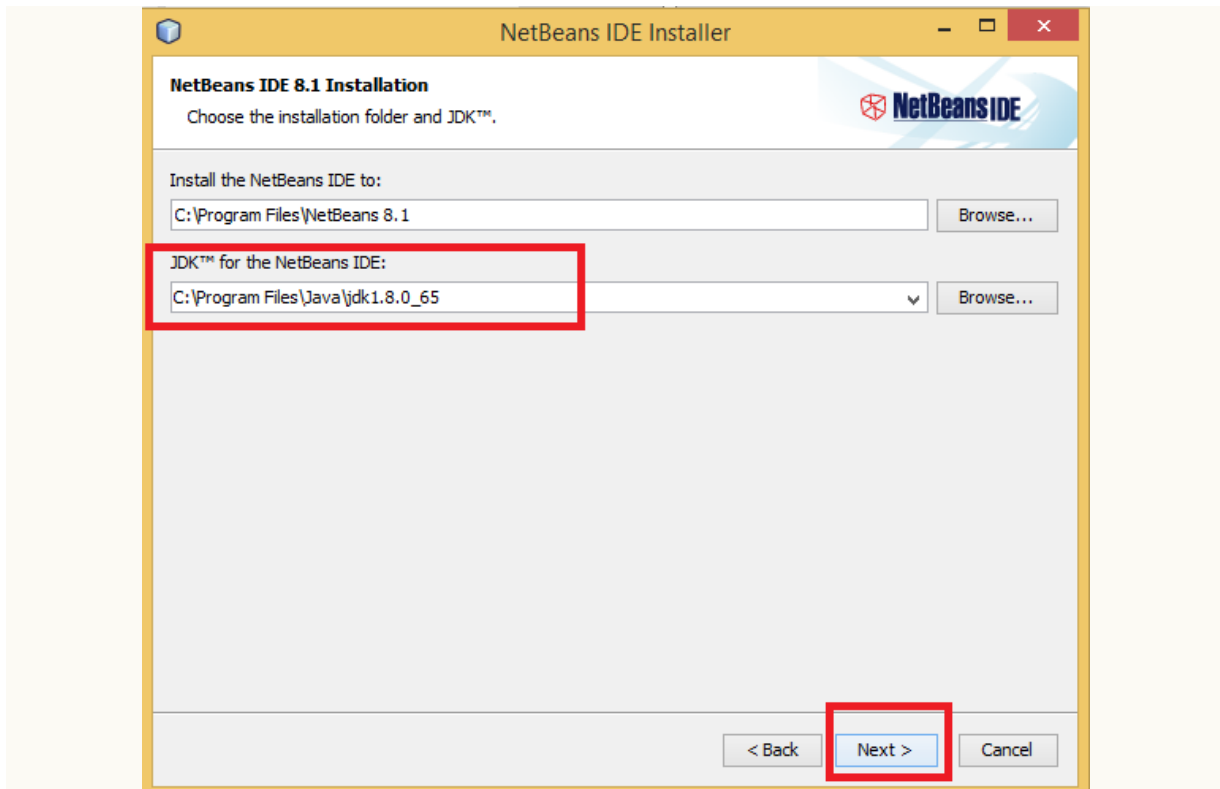
Στη σελίδα DOWNLOAD NETBEANS θα βρείτε επιλογές του προγράμματος που περιλαμβάνουν διαφορετικά επίπεδα λειτουργικότητας. Εμείς επιθυμούμε να εγκαταστήσουμε το NetBeans με όλες τις λειτουργίες του εγκαταστημένες και ενεργοποιημένες. Για αυτό το λόγο θα επιλέξουμε την στήλη που ονομάζεται ALL. Ενδιαφερόμαστε για την εγκατάσταση του NetBeans στην πλατφόρμα των Windows. Για αυτό βεβαιωνόμαστε ότι στην επιλογή Platform εμφανίζεται το όνομα Windows.

Στην εμφανιζόμενη φόρμα έρευνας που κάνει το NetBeans μπορείτε να απαντήσετε στις λίγες ερωτήσεις που εμφανίζονται ή απλά να πατήσετε το κουμπί με την επιλογή «No thanks, take me to download» για να προσπεράσετε αυτή την διαδικασία και να κατεβάσετε το προϊόν. Επιλέγουμε σε ποιο φάκελο θέλουμε να αποθηκεύσουμε το αρχείο που θα κατεβάσουμε και περιμένουμε να ολοκληρωθεί η διαδικασία λήψης του αρχείου.

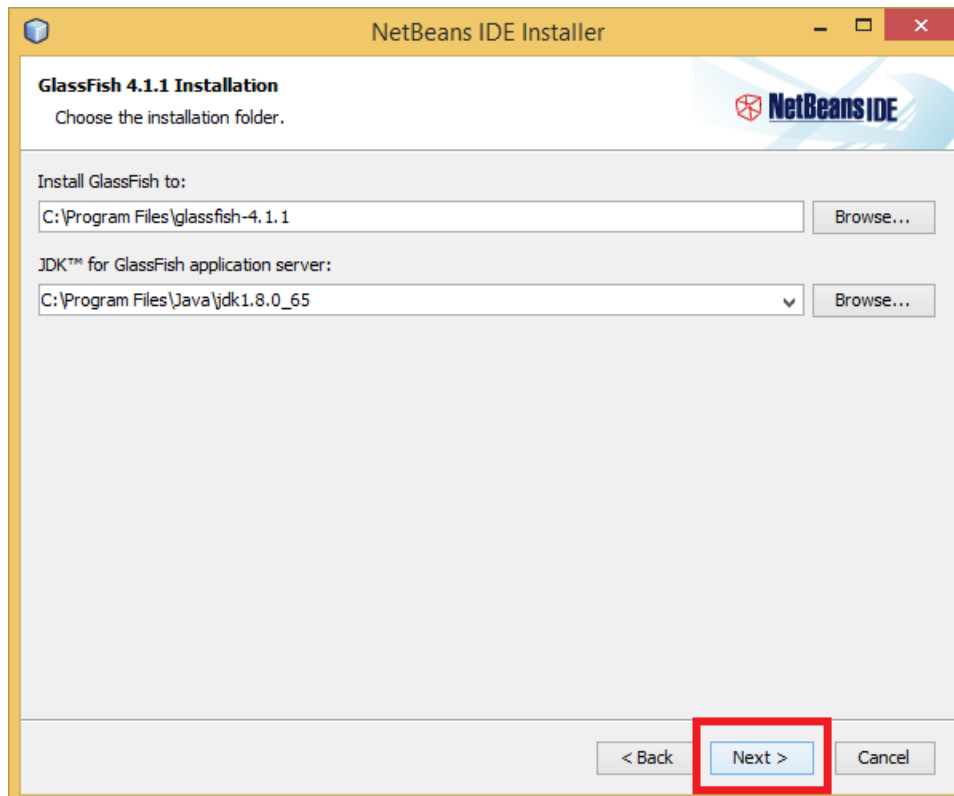
Για να εγκαταστήσετε το NetBeans απλά μεταβείτε στο φάκελο ή στην επιφάνεια εργασίας όπου έγινε η λήψη του αρχείου και κάντε διπλό κλικ επάνω στο εκτελέσιμο αρχείο. Στο «Welcome to the NetBeans IDE 8.1 Installer» παράθυρο πατήστε το κουμπί «NEXT». Αποδεχόμαστε τους όρους χρήσης, ώστε να συνεχίσουμε.

Στο παράθυρο «JUnit License Agreement» δεχόμαστε να εγκαταστήσουμε και το JUnit framework που θα μας βοηθήσει αργότερα στο να ελέγχουμε την λειτουργία των κλάσεων μας και των μεθόδων τους. Πατήστε και εδώ το κουμπί «NEXT» για να συνεχίσετε.

Η σωστή λειτουργία του NetBeans στηρίζεται επάνω στο JDK. Άλλωστε το NetBeans παρέχει μόνο το περιβάλλον ανάπτυξης των εφαρμογών σας. Την αληθινή δουλειά την κάνει το JDK. Οπότε, το NetBeans ψάχνει σε αυτό το σημείο να βρει το ήδη εγκαταστημένο JDK. Δεν χρειάζεται να κάνετε τίποτα σε αυτό το βήμα εκτός από το να επιβεβαιώσετε ότι βρήκε το JDK και απλά πατάτε το «NEXT» κουμπί:



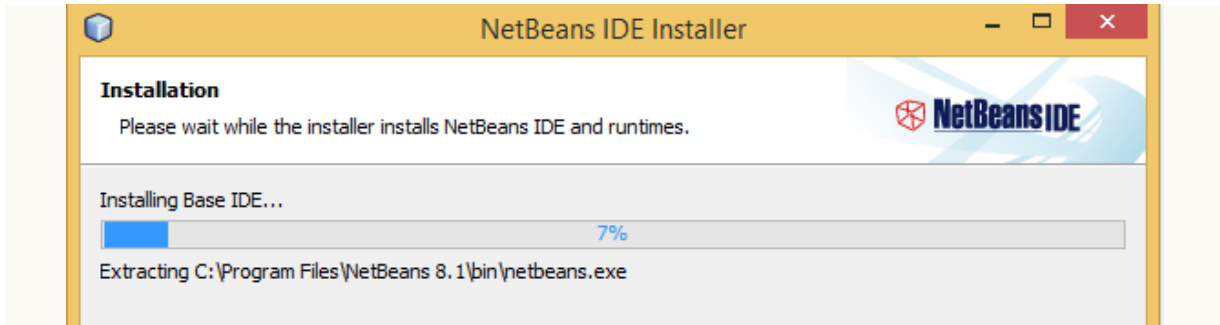
Εικόνα 57:JDK Netbeans εγκατάσταση



Εικόνα 58: Glassfish εγκατάσταση(συνέχεια Netbeans)

Ο Glassfish είναι ο δωρεάν και ανοιχτού κώδικα application server επάνω στον οποίο μπορούμε να στηριχτούμε αργότερα για να τρέξουμε enterprise και web εφαρμογές. Σε αυτό το σημείο δεχόμαστε το προεπιλεγμένο directory στο οποίο θα εγκατασταθεί και πατάμε το κουμπί «NEXT».

Σε αυτό το σημείο ξεκινάει η εγκατάσταση. Και απλά περιμένουμε να ολοκληρωθεί:



Εικόνα 59 Netbeans εγκατάσταση

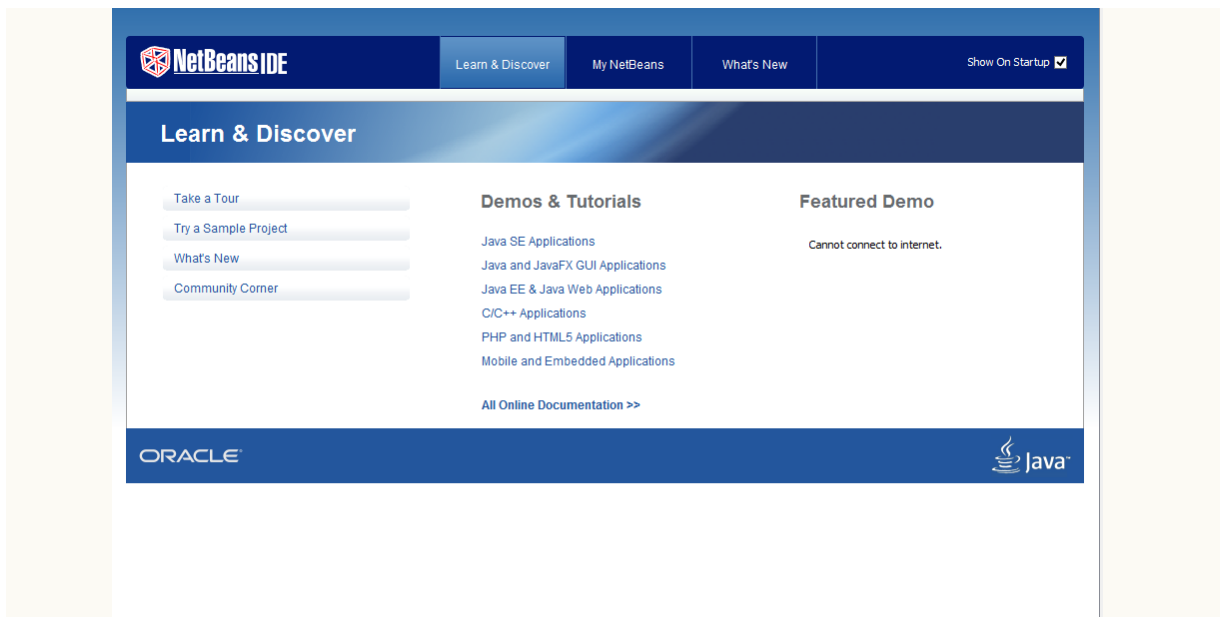
Στο τελευταίο παράθυρο εγκατάστασης του NetBeans, απλά ενημερωνόμαστε για το επιτυχές τέλος της διαδικασίας και κατά πόσο θέλουμε να συνεισφέρουμε στο NetBeans project αφήνοντας την εταιρεία να μαζεύει κάποια στατιστικά στοιχεία τα οποία θα αξιοποιήσει για την περαιτέρω βελτίωση του προγράμματος.

Αφού αποφασίσετε, απλά πατήστε το κουμπί «FINISH» για να επισημοποιήσετε την ολοκλήρωση της εγκατάστασης. Στην επιφάνεια εργασίας του υπολογιστή σας, θα πρέπει τώρα να εμφανίζεται ένα εικονίδιο σε σχήμα κύβου με το όνομα NetBeans IDE 8.1:



Εικόνα 60 Netbeans Συντόμευση

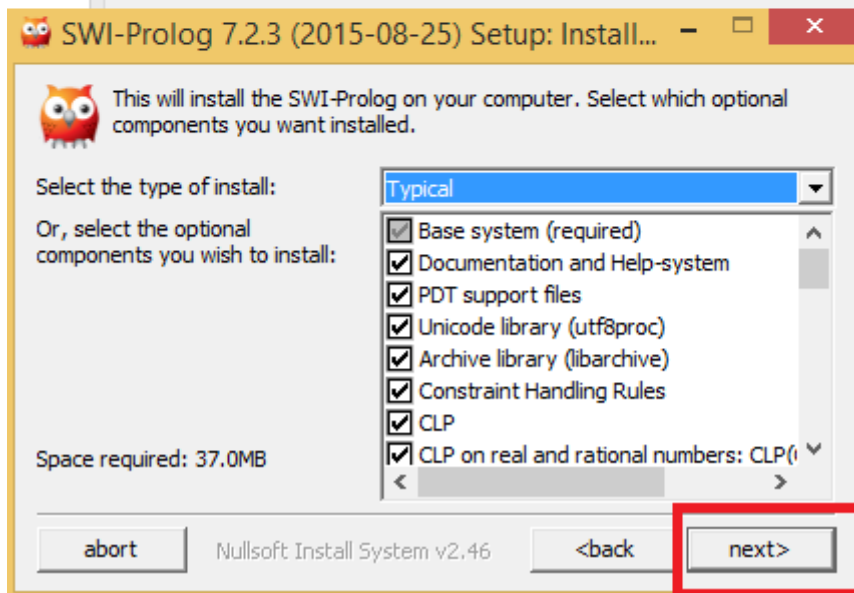
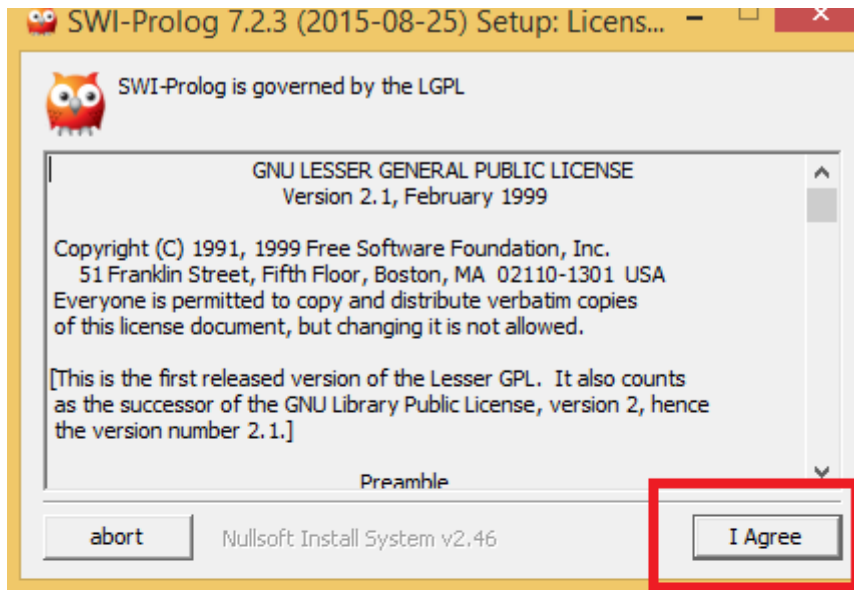
Κάνοντας διπλό κλικ επάνω στο εικονίδιο του «NetBeans», θα βρεθείτε στο περιβάλλον εργασίας του μέσα από το οποίο θα αναπτύξουμε το πρόγραμμα:



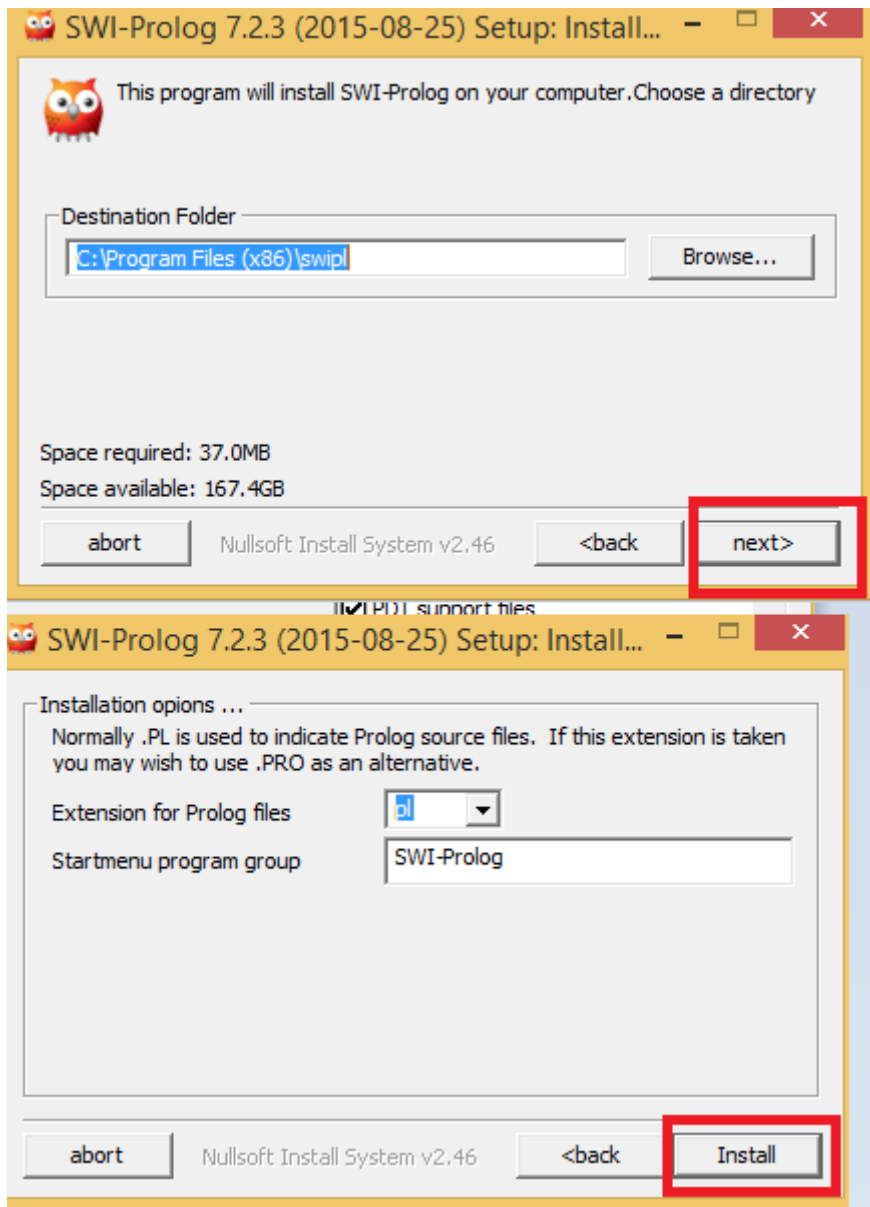
Εικόνα 61: Netbeans-Περιβαλλον

12.2. Εγκατάσταση SWI-Prolog.(*)

Πατώντας τον παρακάτω σύνδεσμο μπορείτε να κατεβάσετε το πρόγραμμα. [DOWNLOAD SWI- Prolog](#). Τα βήματα που πρέπει να ακολουθήσετε είναι πολύ απλά, στην ουσία δεν χρειάζεται καμία απολύτως ρύθμιση, απλά χρησιμοποιείτε τις προεπιλεγμένες ρυθμίσεις και πατάτε μόνο «NEXT». («Εικόνα 62: SWI εγκατάσταση 1» και «Εικόνα 63: SWI εγκατάσταση 2»)



Εικόνα 62: SWI εγκατάσταση 1

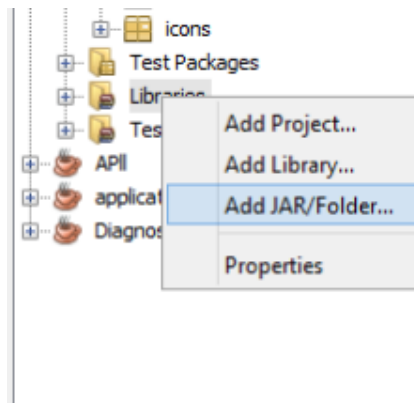


Εικόνα 63: SWI εγκατάσταση 2

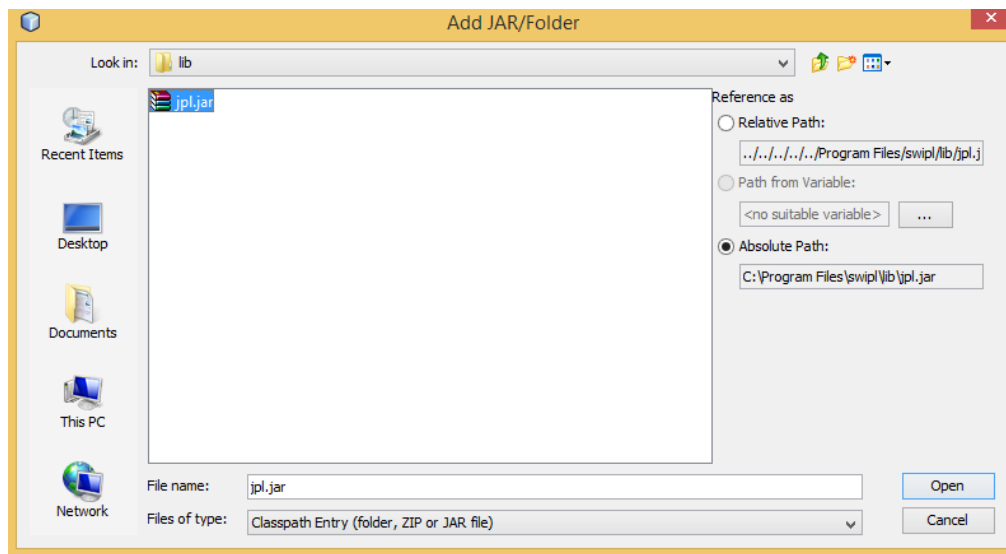
12.3. Βήματα

Εισαγωγή Βιβλιοθηκών Netbeans

Η βιβλιοθήκη που πρέπει να προσθέσουμε είναι η jpl.jar Αυτό που πρέπει να κάνετε είναι να πατήσετε δεξί κλικ στο Libraries (στο Netbeans) και έπειτα AddJAR/Folder(«Εικόνα 64: Εισαγωγή βιβλιοθηκών») και να επιλέξετε τις 3 βιβλιοθήκες από το φάκελο και τέλος να πατήσετε «Open» («Εικόνα 65: Επιλογή και άνοιγμα Βιβλιοθηκών»).



Εικόνα 64: Εισαγωγή βιβλιοθηκών



Εικόνα 65: Επιλογή και άνοιγμα Βιβλιοθηκών

Πρέπει επίσης να καθορίσετε και το path όπως ακριβώς και με το JDK. Τα path που χρειάζονται είναι τα παρακάτω και τα εισάγετε ακριβώς όπως την «Εικόνα 56:JavaPath(μονοπάτι)». Έχοντας ενδιάμεσα σε κάθε path semicolon (;) για να ξεχωρίζει η κάθε εντολή.Μετά το τέλος της εισαγωγής των pathπαιτείται επανεκκίνηση.

C:\ProgramFiles\swipl\bin
 C:\Program Files\swipl\bin\libswipl.dll
 C:\Program Files\swipl\lib\jpl.jar (βιβλιοθήκη Java)
 C:\Program Files\swipl\bin\jpl.dll (τοπικήβιβλιοθήκη -για Windows στηνπερίπτωσημας)
 C:\Program Files\swipl\library\jpl.pl (βιβλιοθήκη Prolog)
 C:\Program Files\Java\jdk1.8.0_31\lib\tools.jar
 C:\Program Files\Java\jdk1.8.0_31\jre\lib\rt.jar

13. Παράρτημα Β: Ορισμοί μεταβλητών του συστήματος

Στον παρακάτω πίνακα παρατίθεται ο ορισμός των μεταβλητών που χρησιμοποιούνται στο διαγνωστικό σύστημα ΔιΣΑ.

Πίνακας 6: Πίνακας Ορισμών

Όνομα	Συντομογραφία
Ερυθρα Αιμοσφαιρία	RBC
Μέσος όγκος Ερυθρών Αιμοσφαιρίων	MCV
Λευκά αιμοσφαίρια	WBC
Δείκτης ερυθροκυττάρων	RTC
Αιμοπετάλια	PLT
Φερριτίνη	Ferritin
Σίδηρος	Fe
Αιματοκρίτης	Hct
Αιμοσφαιρίνη	Hb
Βιταμίνη B12	B12
Φυλικό Οξύ	Folic_acid
Αιμοσφαιρίνη A1	HbA1
Αιμοσφαιρίνη A2	HbA2
Αιμοσφαιρίνη S	HbS
Αιμοσφαιρίνη F	HbF
Σχήμα ερυθρών αιμοσφαιρίων	Shape_RBC
Σιδηροδεσμευτική ικανότητα	TIBC

14. Παράρτημα Γ: Κώδικας Prolog

Πίνακας 7:Κώδικας Prolog

```
:- ['kb_anemia_V5b.pl'].
:- encoding(utf8).
:- style_check(-singleton).
:- dynamic rule/3.
:- dynamic rules/1.

/* κωδικας που καλειτε για να αποθηκευσει τα δεδομενα στη βαση
και μετα απο αυτο το κατηγορημα καλειτε το κυρίως κατηγορημα
diagnose που είναι υπεύθυνο για τη διάγνωση
*/

% prwti episkepsi
/* GOAL
kb_test(['Παναγιωτης','71410 Ηράκλειο','2810123456'],
[male,no,50],[ΠΟΝΟΚΕΦΑΛΟΣ','ΑΔΥΝΑΜΙΑ','ΠΡΟΒΛΗΜΑΤΑ
ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ],[4300,1,75,3.7,41,18,250,30,9,180],[[],[], Res,Ther) .

*/

kb_test(DataCom,PersData,Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb,Res,Ther):
-
    clause(max_visit_id(X),Body1), X1 is X+1,
    clause(max_patient_id(K),Body1), K1 is K+1,

    assert(kb_da(X1,K1,DataCom,PersData,[Symptoms,FullBloodTest,SpecBloodTest,El
ectricalHb],D,T)),
    clause(allVisits(L),Body2),
    clause(allPatients(M),Body3),
    retractall(allVisits(L)),
    retractall(allPatients(M)),
    append(L,[X1],L1),
    append(M,[K1],M1),
    asserta(allVisits(L1)),
    asserta(max_visit_id(X1)),
    asserta(allPatients(M1)),
    asserta(max_patient_id(K1)),
    save, diagnose(X1,D,Therapy) , Res = D, Ther=Therapy,

    clause(kb_da(X1,K1,DataCom,PersData,[Symptoms,FullBloodTest,SpecBloodTest,El
ectricalHb],D,T),Body3),
```

```
retractall(kb_da(X1,K1,DataCom,PersData,[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T)),
```

```
assert(kb_da(X1,K1,DataCom,PersData,[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T)),save,write(Ther).
```

```
    % 2i episkepsi  
/* GOAL
```

```
kb_search(3,['Δέσποινα Μπασδέκη','Πανός 19 71410  
Ηράκλειο',6955457845],[female,no,24],['ΠΟΝΟΚΕΦΑΛΟΣ','ΠΡΟΒΛΗΜΑΤΑ  
ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ','ΕΛΛΕΙΨΗΣ ΔΙΑΤΡΟΦΗ','ΔΥΣΠΝΟΙΑ','ΤΑΧΥΚΑΡΔΙΑ'],  
[4300,1,75,3.7,41,18,250,30,9,180],_._,Res,Ther) .
```

```
*/
```

```
kb_search(Uid,DataCom,PersData,Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb,Res,Ther):-
```

```
    clause(max_visit_id(X),Body1), X1 is X+1,  
    clause(max_patient_id(K),Body1),
```

```
    assert(kb_da(X1,Uid,DataCom,PersData,[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T)),
```

```
    clause(allVisits(L),Body2),  
    clause(allPatients(M),Body3),  
    retractall(allVisits(L)),  
    retractall(allPatients(M)),  
    append(L,[X1],L1),  
    append(M,[K],M1),  
    asserta(allVisits(L1)),  
    asserta(max_visit_id(X1)),  
    asserta(allPatients(M1)),  
    asserta(max_patient_id(K)),  
    save,  
    diagnose(X1,D,Therapy),  
    Res = D, Ther=Therapy,
```

```
    clause(kb_da(X1,Uid,DataCom,PersData,[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T),Body3),
```

```
    retractall(kb_da(X1,Uid,DataCom,PersData,[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T)),
```

```
    assert(kb_da(X1,Uid,DataCom,PersData,[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T)),save,
```

```
    write(Ther).
```

```
/* GOAL  
diagnose(26,Diagnosis,Therapy)
```

```
*/
```

```
diagnose(Visit_id,Diagnosis,Therapy) :-
```

```
  clause(kb_da(Visit_id,Patient_Id,DataCom,PersData,
```

```
    [Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T),Body),
```

```
  rule(R_Id,kb_da(Visit_id,Patient_Id,DataCom,PersData,
```

```
    [Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T),Diagnosis,Therapy),nl.
```

```
/* GOAL
```

```
diagnose(26,Diagnosis,Therapy)
```

```
*/
```

```
rules([r1,r2,r3,r4,r5,r6,r7,r8,r9,r10,r11,r12,r13,r14]).
```

```
/*
```

```
% RULE 1
```

```
rule(R,kb_da(25,22,['Παναγιωτης','71410 Ηράκλειο','2810123456'],[male,no,50],
```

```
  [['ΠΙΟΝΟΚΕΦΑΛΟΣ','ΑΔΥΝΑΜΙΑ','ΠΡΟΒΛΗΜΑΤΑ
```

```
  ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ'],[4300,1,75,3.7,41,18,250,30,9,180],[[],[],_._)
```

```
,Res,Ther) .
```

```
kb_da(23,1,['Ιωάννης Ιωάννου','Φλέμινγκ3 71410
```

```
  Ηράκλειο','2810123456'],[male,na,50],[['ΠΙΟΝΟΚΕΦΑΛΟΣ','ΑΔΥΝΑΜΙΑ','ΠΡΟΒΛΗΜΑΤΑ
```

```
  Α ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ'],_G4067,_G4070,[],_._).
```

```
*/
```

```
rule(r1,kb_da(Visit_id,UserId,DataCom,PersData,
```

```
  [Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T),Diagnosis,Therapy) :-
```

```
  (member('ΠΙΟΝΟΚΕΦΑΛΟΣ',Symptoms);
```

```
    member('ΚΟΥΡΑΣΗ',Symptoms);
```

```
    member('ΠΡΟΒΛΗΜΑΤΑ ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ',Symptoms);
```

```
    member('ΑΔΥΝΑΜΙΑ',Symptoms);
```

```
    member('ΤΑΧΥΚΑΡΔΙΑ',Symptoms);
```

```
    member('ΔΥΣΠΝΟΙΑ',Symptoms)),
```

```
  check_WBC_PLT(FullBloodTest,noPancytopenia) ,
```

```
  check_RTC(FullBloodTest,rtcLess2),
```

```
  check_MCV(FullBloodTest,mcvLow),
```

```
  checkFerritin(FullBloodTest,ferLow),
```

```
  check_HCT_Hb_FE([FullBloodTest,PersData],hctLow),
```

```
  checkRBC([FullBloodTest,PersData],rbcLow),
```

```
  Diagnosis = 'Σιδηροπενική Αναιμία',
```

```
  Therapy='Χορήγηση Σιδήρου'.
```

% RULE 2

```
rule(r2, kb_da(Visit_id, UserId, DataCom, PersData,  
[Symptoms, FullBloodTest, SpecBloodTest, ElectricalHb], D, T), Diagnosis, Therapy) :-
```

```
    check_WBC_PLT(FullBloodTest, noPancytopenia) ,  
    check_RTC(FullBloodTest, rtcLess2),  
    check_MCV(FullBloodTest, mcvLow),  
    checkFerritin(FullBloodTest, ferLow),  
    check_HCT_Hb_FE([FullBloodTest, PersData], hctLow),  
    checkRBC([FullBloodTest, PersData], rbcHigh),  
    checkElectrHb([Symptoms, ElectricalHb], electro),  
    Diagnosis = 'Στίγμα Θαλασσαιμίας',  
    Therapy='Δεν χρειάζεται κάποια θεραπεία, παρα μόνο χορήγηση σιδήρου αν
```

υπάρχει έλλειμμα.'

% RULE 3

```
rule(r3, kb_da(Visit_id, UserId, DataCom, PersData,  
[Symptoms, FullBloodTest, SpecBloodTest, ElectricalHb], D, T), Diagnosis, Therapy) :-
```

```
    check_WBC_PLT(FullBloodTest, noPancytopenia) ,  
    check_RTC(FullBloodTest, rtcLess2),  
    check_MCV(FullBloodTest, mcvLow),  
    checkFerritin(FullBloodTest, ferLow),  
    check_HCT_Hb_FE([FullBloodTest, PersData], hctHigh),  
    checkRBC([FullBloodTest, PersData], rbcHigh),  
    checkElectrHb([Symptoms, ElectricalHb], electro),  
    Diagnosis = 'Στίγμα Θαλασσαιμίας',  
    Therapy='Δεν χρειάζεται κάποια θεραπεία, παρα μόνο χορήγηση σιδήρου αν
```

υπάρχει έλλειμμα.'

% RULE 4

```
rule(r4, kb_da(Visit_id, UserId, DataCom, PersData,  
[Symptoms, FullBloodTest, SpecBloodTest, ElectricalHb], D, T), Diagnosis, Therapy) :-
```

```
    check_WBC_PLT(FullBloodTest, noPancytopenia) ,  
    check_RTC(FullBloodTest, rtcLess2),  
    check_MCV(FullBloodTest, mcvLow),  
    checkFerritin(FullBloodTest, ferLow),  
    check_HCT_Hb_FE([FullBloodTest, PersData], hctHigh),  
    checkRBC([FullBloodTest, PersData], rbcHigh),  
    checkElectrHb([Symptoms, ElectricalHb], electroNo),  
    Diagnosis = 'Στίγμα Θαλασσαιμίας',  
    Therapy='Δεν χρειάζεται κάποια θεραπεία, παρα μόνο χορήγηση σιδήρου αν
```

υπάρχει έλλειμμα.'

% RULE 5

```
rule(r5, kb_da(Visit_id, UserId, DataCom, PersData,  
[Symptoms, FullBloodTest, SpecBloodTest, ElectricalHb], D, T), Diagnosis, Therapy) :-
```

```
check_WBC_PLT(FullBloodTest,noPancytopenia) ,
check_RTC(FullBloodTest,rtcLess2),
check_MCV(FullBloodTest,mcvLow),
checkFerritin(FullBloodTest,ferHigh),
checkElectrofHb([Symptoms,ElectricalHb],yesElectr),
Diagnosis='Θαλασσαιμία',
Therapy='Μεταγγίσεις αίματος'.
```

% RULE 6

```
rule(r6,kb_da(Visit_id,UserId,DataCom,PersData,
[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T), Diagnosis,Therapy) :-
```

```
check_WBC_PLT(FullBloodTest,noPancytopenia) ,
check_RTC(FullBloodTest,rtcLess2),
check_MCV(FullBloodTest,mcvNormal),
checkFerritin(FullBloodTest,ferHigh),
checkElectrofHb([Symptoms,ElectricalHb],yesElectr),
Diagnosis='Θαλασσαιμία',
Therapy='Μεταγγίσεις αίματος'.
```

% RULE 7

```
rule(r7,kb_da(Visit_id,UserId,DataCom,PersData,
[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T), Diagnosis,Therapy) :-
```

```
check_WBC_PLT(FullBloodTest,noPancytopenia) ,
check_RTC(FullBloodTest,rtcLess2),
check_MCV(FullBloodTest,mcvHigh),
check_B12_RBC([PersData,FullBloodTest,SpecBloodTest],b12Low),
check_specBloodTest(SpecBloodTest, ok),
Diagnosis = 'B12 Αναιμία',
Therapy='θεραπεία με συμπλήρωμα (από το στόμα) ή/και ένεση βιταμίνης
```

B12'.

% RULE 8

```
rule(r8,kb_da(Visit_id,UserId,DataCom,PersData,
[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T), Diagnosis,Therapy) :-
```

```
check_WBC_PLT(FullBloodTest,noPancytopenia) ,
check_RTC(FullBloodTest,rtcLess2),
check_MCV(FullBloodTest,mcvHigh),
check_Folic(SpecBloodTest,folic),
Diagnosis = 'Αναιμία Φυλλικού Οξέος',
Therapy='θεραπεία με συμπλήρωμα φυλλικού οξέος'.
```

% RULE 9

```
rule(r9,kb_da(Visit_id,UserId,DataCom,PersData,
[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T), Diagnosis,Therapy) :-
```

```
check_WBC_PLT(FullBloodTest,noPancytopenia),
```

```
check_RTC(FullBloodTest,rtcUp2),
check_HBS(ElectricalHb,hbSZero),
Diagnosis = 'Σφαιροκυττάρωση.',
Therapy='Οι θεραπείες περιστρέφονται γύρω από την αντιμετώπιση των
άμεσων και οξέων προβλημάτων και την αποκατάσταση ή πρόληψη των επιπλοκών και των
κρίσεων'.
```

% RULE 10

```
rule(r10,kb_da(Visit_id,UserId,DataCom,PersData,
[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T), Diagnosis,Therapy) :-
```

```
check_WBC_PLT(FullBloodTest,noPancytopenia) ,
check_RTC(FullBloodTest,rtcUp2),
check_HBS(ElectricalHb,hbsUpZero),
Diagnosis = 'Δρεπανοκυττάρωση.',
Therapy='Επικοινωνία με το γιατρό, πιθανή σπληνεκτομή.'.
```

% RULE 11

```
rule(r11,kb_da(Visit_id,UserId,DataCom,PersData,
[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T), Diagnosis,Therapy) :-
```

```
check_WBC_PLT(FullBloodTest,yesPancytopenia),
Diagnosis = 'Έλγχος Μυελού των οστών',
Therapy='Επικοινωνηστε με το γιατρό σας'.
```

% RULE 12

```
rule(r12,kb_da(Visit_id,UserId,DataCom,PersData,
[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T), Diagnosis,Therapy) :-
```

```
check_WBC_PLT(FullBloodTest,noPancytopenia) ,
check_RTC(FullBloodTest,rtcUp2),
check_HBS(ElectricalHb,hbSZero) ;
check_HBS(ElectricalHb,hbsUpZero),
Diagnosis = 'Δεν υπάρχει πρόβλημα Δρεπανοκυττάρωσης ή
Σφαιροκυττάρωσης.',
Therapy='Επικοινωνηστε με το γιατρό σας'.
```

% RULE 13

```
rule(r13,kb_da(Visit_id,UserId,DataCom,PersData,
[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T), Diagnosis,Therapy) :-
```

```
check_WBC_PLT(FullBloodTest,noPancytopenia) ,
check_RTC(FullBloodTest,rtcLess2),
check_MCV(FullBloodTest,mcvLow),
checkFerritin(FullBloodTest,ferHigh),
checkElectrofHb([Symptoms,ElectricalHb],noElectr),
Diagnosis = 'Δεν υπάρχει πρόβλημα αναιμίας, πιθανή αιμοχρωμάτωση λόγω
αυξημένης φερριτίνης.',
Therapy='Επικοινωνηστε με το γιατρό σας'.
```

% RULE 14

```

rule(r14,kb_da(Visit_id,UserId,DataCom,PersData,
  [Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T), Diagnosis,Therapy) :-
  check_symptoms(Symptoms,noOk),
  Diagnosis = 'Δεν υπάρχει πρόβλημα αναιμίας.',

  Therapy='Επικοινωνηστε με το γιατρό σας'.

% RULE 15

rule(r15,kb_da(Visit_id,UserId,DataCom,PersData,
  [Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T), Diagnosis,Therapy) :-
  check_symptoms(Symptoms,Ok),
  Diagnosis = 'Ελλειπής στοιχεία.',
  Therapy='Επικοινωνηστε με το γιατρό σας'.

/* τα κατηγορηματα που καλούνται απο τους κανονες
και κάνουν τους απαραίτητους υπολογισμούς
*/

check_symptoms(Symptoms,ok) :-
  check_symptomList(Symptoms), !.
check_symptoms(Symptoms,noOk).

check_symptomList([H|T]) :-
  SymptomList = ['ΠΙΟΝΟΚΕΦΑΛΟΣ', 'ΑΔΥΝΑΜΙΑ',
  ΠΡΟΒΛΗΜΑΤΑ_ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ',
  'ΑΠΩΛΕΙΑ_ΒΑΡΟΥΣ', 'ΑΠΩΛΕΙΑ_ΟΡΕΞΗΣ', 'ΩΧΡΟΤΗΤΑ_ΔΕΡΜΑΤΟΣ',
  'ΕΠΩΔΥΝΗ_ΣΠΛΗΝΑ', 'ΤΑΧΥΚΑΡΔΙΑ',
  'ΔΥΣΠΝΟΙΑ',ΓΛΩΣΣΙΤΙΔΑ',ΠΛΗΓΕΣ',
  'ΕΛΛΕΙΠΗΣ_ΔΙΑΤΡΟΦΗ','ΖΑΛΑΔΑ','ΚΟΥΡΑΣΗ', 'πατερας_Στίγμα',
  'πατερας_Μεσογειακή',
  'μητερα_Στίγμα', 'μητερα_Μεσογειακή', 'ιδιαιτερο'],
  member(H,SymptomList).
check_symptomList([H|T]) :-
  check_symptomList(T).

check_WBC_PLT([WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT], noPancytopenia)
:-
  number(WBC), number(PLT),(WBC > 4500, PLT>150)
  ; number(WBC), number(PLT),(WBC > 4500, PLT<150)
  ; number(WBC), number(PLT),(WBC < 4500, PLT>150).

check_WBC_PLT([WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT],yesPancytopenia)
:-
  number(WBC), number(PLT),
  (WBC < 4500, PLT<150).

```

```

check_RTC([WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT],rtcLess2):-
    number(RTC),RTC<2.
check_RTC([WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT],rtcUp2):-
    number(RTC),RTC>2.

check_MCV([WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT],mcvLow):-
    number(MCV), MCV<79.

check_MCV([WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT],mcvNormal):-
    number(MCV), MCV>79,MCV<97.

check_MCV([WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT],mcvHigh):-
    number(MCV), MCV>97.

check_HBS([HbA1,HbA2,HbS,HbF,Schema_RBC],hbSZero):-
    number(HbS), HbS=0,
    Schema_RBC='ΣΦΑΙΡΙΚΟ'.

check_HBS([HbA1,HbA2,HbS,HbF,Schema_RBC],hbSZero):-
    number(HbS), HbS=0.

check_HBS([HbA1,HbA2,HbS,HbF,Schema_RBC],hbsUpZero):-
    number(HbS), HbS>0,
    Schema_RBC='ΔΡΕΠΑΝΟΕΙΔΕΣ'.
check_HBS([HbA1,HbA2,HbS,HbF,Schema_RBC],hbsUpZero):-
    number(HbS), HbS>0.

checkFerritin([WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT],ferLow):-
    number(Ferritin),Ferritin<20.
checkFerritin([WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT],ferHigh):-
    number(Ferritin),Ferritin>20.

checkElectroHb([Symptoms, [HbA1,HbA2,HbS,HbF,Schema_RBC]],yesElectr):-
    ((member('ιδιαίτερο', Symptoms));
     ((member('πατερας_Στίγμα', Symptoms),
          (member('μητερα_Μεσογειακή', Symptoms)))) ;
     ((member('πατερας_Μεσογειακή', Symptoms),
          (member('μητερα_Στίγμα', Symptoms)))) ;
     ((member('πατερας_Μεσογειακή', Symptoms),
          (member('μητερα_Μεσογειακή', Symptoms)))));
     (number(HbF),HbF>2,HbF<9);
     (number(HbA1), HbA1<5) .

checkElectroHb([Symptoms,ElectricalHb],noElectr).

```



```

check_HCT_Hb_FE([[WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT],
[Filo,Egkimosini,Hlikia] ],hctLow):-
    (Filo='male',number(Hct),number(Hb),number(FE),Hct<40,Hb<14,FE<55)
    ;( Filo='female',number(Hct),number(Hb),number(FE),Hct<36,Hb<10,FE<45)
    ;
(Filo='female',Egkimosini=yes,number(Hct),number(Hb),number(FE),Hct<36,Hb<12,FE<55)
.

check_HCT_Hb_FE([[WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT],
[Filo,Egkimosini,Hlikia] ],hctHigh).

checkRBC([[WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT], [Filo,Egkimosini,Hlikia]
],rbcLow):-
    (Filo='male',number(RBC),RBC<4.5);
    (Filo='female',number(RBC),RBC<3.8).

checkRBC([[WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT], [Filo,Egkimosini,Hlikia]
],rbcHigh):-
    (Filo='male',number(RBC),RBC>6);
    (Filo='female',number(RBC),RBC>4.9).

checkElectrHb([Symptoms, [HbA1,HbA2,HbS,HbF,Schema_RBC]],electro):-
    ( member('πατερας_Στίγμα', Symptoms));
    (member('μητερα_Στίγμα', Symptoms));
    ( member('μητερα_Μεσογειακή', Symptoms));
    ( member('πατερας_Μεσογειακή', Symptoms)) ;
    (number(HbF),HbF>2,HbF<9 );
    (number(HbA1), HbA1<1).

checkElectrHb([Symptoms, ElectricalHb],electroNo).

check_B12_RBC([ [Filo,Egkimosini,Hlikia]
,[WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT],[B12,Folic_acid]],b12Low):-
    ((Filo='male',number(RBC),RBC>6);
    (Filo='female',number(RBC),RBC>4.9)) ,
    number(B12), B12<130,
    number(Folic_acid), Folic_acid>3.1.

check_B12_RBC([ [Filo,Egkimosini,Hlikia]
,[WBC,RTC,MCV,RBC,FE,Ferritin,TIBC,Hct,Hb,PLT],[B12,Folic_acid]],b12Nores).

check_Folic([B12,Folic_acid],folic):-
    number(Folic_acid), Folic_acid<3.1.

```

```
check_Folic([B12,Folic_acid],folicNores).
```

```
/* κατηγορημα που αποθηκευει τα νεα δεδομενα μαζι  
με τα παλια στο αρχειο της βασης  
*/
```

```
save :-
```

```
    tell('kb_anemia.pl'),  
    write(':- dynamic kb_da/7, allPatients/1, max_patient_id/1, max_visit_id/1,  
allVisits/1.').  
    clause(max_patient_id(X),Body1), nl,write(max_patient_id(X)),write('.').nl,  
    clause(max_visit_id(K),Body2), nl,write(max_visit_id(K)),write('.').nl,  
    clause(allPatients(L),Body3), write(allPatients(L)),nl,  
    clause(allVisits(M),Body4), write(allVisits(M)),write('.').  
    nl,  
    get_kb_data(M),  
    told.
```

```
get_kb_data([]).
```

```
get_kb_data([Visit_id|Rest_visit_id]) :-
```

```
    clause(kb_da(Visit_id,Patient_Id,DataCom,PersData,  
[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T),Body3),  
    fix(DataCom, [], DataCom2),fix(Symptoms, [], Symptoms2),  
  
    fix(ElectricalHb, [], ElectricalHb2),fix([D],[],[D2]),  
    write(kb_da(Visit_id,Patient_Id, DataCom2,PersData,  
[Symptoms2,FullBloodTest,SpecBloodTest,ElectricalHb2],D2,T2)),write('.').  
nl,  
    get_kb_data(Rest_visit_id).
```

```
/* το κατηγορημα fix ειναι για να παιρνει τα δεδομενα  
και κατα τη μεταφορα τους στο αλλο αρχειο  
να αποθηκευει τις συμβολοσειρες με μονα εισαγωγικα  
αυτο γινεται γιατι εαν το ονομα  
ειναι με κενα βαζοντας το σε μονα εισαγωγικα  
το παιρνει ως συμβολοσειρα.  
Αν παραλειψουμε αυτο το κατηγορημα ολα  
τα δεδομενα αποθηκευονται χωρις εισαγωγικα  
(εκτος απο τους ακεραιους που δεν χρειάζονται ετσι και αλλιως)*/
```

```
fix([], X, X).
```

```
fix([H|T], S ,X):- var(H),
```

```
    append(S,[H], V),
```

```
    fix(T, V, X).
```

```
fix([H|T], S ,X):- nonvar(H),string_concat("\", H, C1), string_concat(C1, "\", C2),
```

```
append(S,[C2], V),
fix(T, V, X).
```

```
% ενημέρωση Βάσης Γνώσης
```

```
/* GOAL
```

```
change_rec(r12,check_symptoms(Symptoms,noOk)).
```

```
*/
```

```
change_rec(R_id,Body) :-
```

```
    clause(rule(R_id,A,B),Body1),
```

```
    retractall(rule(R_id,A,B)),
```

```
    asserta((rule(R_id,A,B) :- Body)),save_rule.
```

```
/* GOAL
```

```
create_rec(check_symptoms(Symptoms,noOk)).
```

```
*/
```

```
create_rec(Body) :-
```

```
    clause(rules(R_ids), B1),
```

```
        last(R_ids,LastRule_id),
```

```
        atom_codes(LastRule_id, Rule_codes),
```

```
        Rule_codes = [H|T],
```

```
        atom_codes(AtomNum,T),
```

```
        atom_number(AtomNum,Num),
```

```
        NewNum is Num+1,
```

```
        atom_codes(NewNum,NewNumCodes),
```

```
        atom_codes(NewRule_id,[114|NewNumCodes]),
```

```
        append(R_ids, [NewRule_id], New_R_ids),
```

```
        retract(rules(R_ids)),
```

```
        asserta((rules(New_R_ids))),
```

```
        asserta((rule(NewRule_id,A,B) :- Body)),save_rule.
```

```
/* GOAL
```

```
delete_rec(r3)
```

```
*/
```

```
delete_rec(R_id) :-
```

```
    % clause(rule(R_id,A,B),Body1),
```

```
    retractall((rule(R_id,A,B))),
```

```
    del_rule_id(R_id),save_rule.
```

```
/* GOAL
```

```
del_rule_id(r3).
```

```
*/
```

```
del_rule_id(R_id) :-
```

```
    rules(R_ids),
```

```
    make_new_rule_ids(R_ids, R_id, New_R_ids),
```

```
    retract(rules(R_ids)), asserta(rules(New_R_ids)).
```

```
/*
```

GOALS

```
1. make_new_rule_ids([r1,r2,r3,r5,r6], r3, RestRids). RestRids = [r1,r2,r5,r6]
*/
make_new_rule_ids([Rid|RestRids], Rid, RestRids).
make_new_rule_ids([Rid|RestRids], Rid1, [Rid|NewRestRids]) :-
    Rid \= Rid1,
    make_new_rule_ids(RestRids, Rid1, NewRestRids).

/* GOAL

% search_same('Ιωάννης Ιωάννου',L,Res,Ther)
*/
search(Onomateponimo,L,M):-
    bagof([Userid,[Onomateponimo,Dieu8insi,Tilefwno],PersData,[Symptoms,FullBlood
Test,SpecBloodTest,ElectricalHb],D),kb_da(Visit_id,Userid,[Onomateponimo,Dieu8insi,Tile
fwno],PersData,[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D,T)^kb_da(Visit_id
,Userid,[Onomateponimo,Dieu8insi,Tilefwno],PersData,[Symptoms,FullBloodTest,SpecBloo
dTest,ElectricalHb],D,T),L),
    write(L),nl,
    bagof(Userid,kb_da(Visit_id,Userid,[Onomateponimo,Dieu8insi,Tilefwno],PersData,[
Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb],D)^kb_da(Visit_id,Userid,[Onomatep
onimo,Dieu8insi,Tilefwno],PersData,[Symptoms,FullBloodTest,SpecBloodTest,ElectricalHb]
,D,T),M).

save_rule :-
    tell('rule_new.pl'),
    write(':- dynamic rule/3, rules/1.').nl,
    write(':- [\kb_anemia_V5b.pl\].').nl,
    write(':- encoding(utf8).').nl,
    write(':- style_check(-singleton).'),
    clause(rules(R_ids),B),nl,
    write((rules(R_ids) :- B)),
    write('.'),
    get_rule_data(R_ids),
    told.

get_rule_data([]).
get_rule_data([Rule_id|Rest_Rule_Id]) :-
    clause(rule(Rule_id,A,B),Body1),
    % fix(B,[],B2),
    write(rule(Rule_id,A,B2):-Body1),write('.'), nl,
    get_rule_data(Rest_Rule_Id).
```



15. Παράρτημα Δ: ΚώδικαςJava

Παρακάτω συμπεριλαμβάνεται το τμήμα από τον κώδικα της Java που προωθεί στην Prolog τις κλήσεις σε κατηγορήματα του ΔιΣΑ.

1. Κλήση κατηγορήματος που αφορά την αλλαγή του κανόνα στη βάση γνώσης, *«change_rec(r12,check_symptoms(Symptoms,noOk)).»*

```
private void changeActionPerformed(java.awt.event.ActionEvent evt) {
    Term consult_arg[] = {
        new Atom(Diagnosis.class.getResource("anemia_diagnosis.pl").getPath())
    };
    Query consult_query
        = new Query( // to kanei query gia na ginei
        "consult", //to consult
        consult_arg);

    boolean consulted = consult_query.hasSolution();

    if (!consulted) {
        System.err.println("Consult failed");
        System.exit(1);
    }

    idch = id_txt.getText();
    bodych = "" +body_txt.getText()+"";

    String t5 = "change_rec(" + idch + "," + bodych + ")." + "\n";

    System.out.println("FUNCTION IS " + t5);
    Query q5 = new Query(t5);
    diagnosis = q5.oneSolution().toString();
    System.out.println(diagnosis);
    JOptionPane.showMessageDialog(null, "Ενημερώθηκε η ΒΓ με id " + idch);
}
```

2. Κλήση κατηγορήματος που αφορά την διαγραφή του κανόνα από τη βάση γνώσης, *«delete_rec(r3)»*

```
private void delete_btnActionPerformed(java.awt.event.ActionEvent evt) {
    Term consult_arg[] = {
        new Atom(Diagnosis.class.getResource("anemia_diagnosis.pl").getPath())
    };
    Query consult_query
        = new Query( // to kanei query gia na ginei
        "consult", //to consult
        consult_arg);
```

```

boolean consulted = consult_query.hasSolution();

if (!consulted) {
    System.err.println("Consult failed");
    System.exit(1);
}

iddel = id_txt2.getText();

String t6 = "delete_rec(" + iddel + ")." + "\n";
Query q6 = new Query(t6);
result = q6.oneSolution().toString();

JOptionPane.showMessageDialog(null, "Διαγράφηκε ο κανόνας με id " + iddel);
}

```

3. Κλήση κατηγορήματος που αφορά τη δημιουργία ενός κανόνα βάσης κανόνων, «**create_rec(check_symptoms(Symptoms,noOk)).**»

```

private void createActionPerformed(java.awt.event.ActionEvent evt) {
    Term consult_arg[] = {
        new Atom(Diagnosis.class.getResource("anemia_diagnosis.pl").getPath())
    };
    Query consult_query
        = new Query( // to kanei query gia na ginei
            "consult", //to consult
            consult_arg);

    boolean consulted = consult_query.hasSolution();

    if (!consulted) {
        System.err.println("Consult failed");
        System.exit(1);
    }
//

    bodycr = body_txt1.getText();
//change_rec(Id,Body)
    String t9 = "create_rec(" + bodycr + ")." + "\n";
    Query q9 = new Query(t9);
    diagnosis = q9.oneSolution().toString();
    JOptionPane.showMessageDialog(null, "Ενημερώθηκε η ΒΓ με id ");
}

```

4. Κλήση κατηγορήματος που αφορά την αναζήτηση του ιστορικού ενός ασθενή, «*search_same('Ιωάννης Ιωάννου',L,Res,Ther)*»

```
private void sear_btnActionPerformed(java.awt.event.ActionEvent evt) {
    Term consult_arg[] = {
        new Atom(Diagnosis.class.getResource("anemia_diagnosis.pl").getPath())
    };
    Query consult_query
        = new Query( // to kanei query gia na ginei
            "consult", //to consult
            consult_arg);

    boolean consulted = consult_query.hasSolution();

    if (!consulted) {
        System.err.println("Consult failed");
        System.exit(1);
    }

    patient_name = "" + search_pat_txt.getText() + "";

    String t7 = "search(" + patient_name + ",L,M)." + "\n";

    Query q7 = new Query(t7);

    search_txt = q7.oneSolution().get("L").toString();
    user_id = q7.oneSolution().get("M").toString();
    search_res.append("Ιστορικό ασθενή: " + "\n"
        + "\n" + search_txt.replaceAll("\\[[\\]]\\(\\)\\|\\|'+", "").trim());

    JOptionPane.showMessageDialog(null, "Επιτυχής αναζήτηση ασθενή με όνομα " +
patient_name);
}
```

5. Κλήση κατηγορήματος που αφορά την διάγνωση ενός ασθενή, είτε νέου «*kb_test* (*'Παναγιωτης', '71410 Ηράκλειο', '2810123456'*), [*male, no, 50*], [*'ΠΟΝΟΚΕΦΑΛΟΣ', 'ΑΔΥΝΑΜΙΑ', 'ΠΡΟΒΛΗΜΑΤΑ ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ'*], [*4300, 1, 75, 3.7, 41, 18, 250, 30, 9, 180*], [], [], *Res,Ther*)», είτε παλιού «*kb_search*(*3, 'Δέσποινα Μπασδέκη', 'Πανός 19 71410 Ηράκλειο', 6955457845*), [*female, no, 24*], [*'ΠΟΝΟΚΕΦΑΛΟΣ', 'ΠΡΟΒΛΗΜΑΤΑ ΑΥΤΟΣΥΓΚΕΝΤΡΩΣΗΣ', 'ΕΛΛΕΙΠΗΣ ΔΙΑΤΡΟΦΗ', 'ΔΥΣΠΝΟΙΑ', 'ΤΑΧΥΚΑΡΔΙΑ'*], [*4300, 1, 75, 3.7, 41, 18, 250, 30, 9, 180*],[],[],*Res,Ther*)».

```
private void diagnosis() {
    if (search_int == 1) {
```



```

Term consult_arg[] = {
    new Atom(Diagnosis.class.getResource("anemia_diagnosis.pl").getPath())
};
Query consult_query
    = new Query( // to kanei query gia na ginei
        "consult", //to consult
        consult_arg);

boolean consulted = consult_query.hasSolution();

if (!consulted) {
    System.err.println("Consult failed");
    System.exit(1);
}

String ts = "kb_search(" + user_id + "," + condata + "," + persdata + "," + symptoms
+ "," + fbcTest + "," + specBloodTest + "," + electrHbTest + ",Res,Ther)." + "\n";

Query qs = new Query(ts);
diagnosis = qs.oneSolution().get("Res").toString();
Diagnosis_txt.append("Με τις δοθείσες πληροφορίες πάσχετε από: " + "\n"
    + "\n" + diagnosis.replaceAll("\\[[\\]]\\(\\|\\)\\|\\|'+", "").trim());
} else {
    Term consult_arg[] = {
        new Atom(Diagnosis.class.getResource("anemia_diagnosis.pl").getPath())
    };
    Query consult_query
        = new Query( // to kanei query gia na ginei
            "consult", //to consult
            consult_arg);

    boolean consulted = consult_query.hasSolution();
    if (!consulted) {
        System.err.println("Consult failed");
        System.exit(1);
    }

    if (fbcTest == null || fbcTest.size() < 10) {
        Diagnosis_txt.append("\n" + "Συμπληρώστε σωστά τα πεδία Γενικής αίματος" +
            "\n"
            + "για να λάβετε σωστά αποτελέσματα." + "\n");
    } else {
        System.out.println("Η λίστα είναι σωστά συμπληρωμένη");
    }
}

```

```

String tt = "kb_test(" + condata + "," + persdata + " ," + symptoms + "," + fbcTest +
"," + specBloodTest + "," + electrHbTest + ",Res,Ther)." + "\n";
System.out.println("FUNCTION IS " + tt);
Query qt = new Query(tt);
// qt.nextSolution().get("Ther").toString();
diagnosis = qt.oneSolution().get("Res").toString();
Diagnosis_txt.append("Με τις δοθείσες πληροφορίες πάσχετε από: " + "\n"
+ "\n" + diagnosis.replaceAll("\\[[\\]]\\(\\|\\)\\|\\|'+", "").trim());
}
}

```