ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Σχεδιασμός - Δημιουργία Διαγνωστικής Εφαρμογής με τη Γλώσσα Visual Prolog. Μελέτη Περίπτωσης: Ψυχολογία

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

του

ΚΑΓΙΟΓΛΙΔΗΣ ΙΩΑΝΝΗΣ

(ΑΕΜ: 1940)

Επιβλέπων : Μιχαήλ Δόσης

Καθηγητής

Καστοριά ……. - 2023

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Σχεδιασμός - Δημιουργία Διαγνωστικής Εφαρμογής με τη Γλώσσα Visual Prolog. Μελέτη Περίπτωσης: Ψυχολογίας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

του

ΚΑΓΙΟΓΛΙΔΗΣ ΙΩΑΝΝΗΣ

(ΑΕΜ: 1940)

Επιβλέπων : Μιχαήλ Δόσης

Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την ημερομηνία εξέτασης

.....	.....	.....
Ον/μο Μέλους	Ον/μο Μέλους	Ον/μο Μέλους
Ιδίοτητα Μέλους	Ιδίοτητα Μέλους	Ιδίοτητα Μέλους

Καστοριά ……. - 2023

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά την οικογένεια μου για όλη τη στήριξη που έδωσαν. Θέλω να εκφράσω και τις ευχαριστίες μου και στον κ. Μπάτο Παναγιώτη για τις συμβουλές που παρείχε κατά την εκπόνηση της εργασίας.

Περίληψη

Στην παρούσα πτυχιακή εργασία διερευνήθηκαν τα διαγνωστικά προγράμματα με στόχο τη δημιουργία μιας διαγνωστικής εφαρμογής. Η εν λόγω εφαρμογή σχεδιάστηκε και υλοποιήθηκε στο περιβάλλον Visual Prolog, χρησιμοποιώντας μια ενσωματωμένη βάση δεδομένων για τον υπολογισμό των ερωτήσεων και απαντήσεων.

Η βασική λειτουργία της εφαρμογής είναι να κάνει ερωτήσεις στον χρήστη και στη συνέχεια παρέχει διαγνωστικά αποτελέσματα βάσει των απαντήσεών του.

Ως παράδειγμα εφαρμογής διαγνωστικού ελέγχου, χρησιμοποιήθηκαν ερωτήσεις που σχετίζονται με τον έλεγχο ψυχικών ασθενειών. Αυτό σημαίνει ότι ο χρήστης θα απαντήσει σε ερωτήσεις που αφορούν την ψυχική του κατάσταση, και η εφαρμογή θα αξιολογήσει τις απαντήσεις του προκειμένου να προσφέρει διαγνωστικά αποτελέσματα ή προτάσεις για περαιτέρω δράση.

Λέξεις* Κλειδιά*: Διαγνωστική Εφαρμογή, Visual Prolog, Συμπτώματα, Βάση Δεδομένων

Abstract

In this thesis the diagnostic programs were investigated with the aim of creating a diagnostic application. The application in question was designed and implemented in the Visual Prolog environment, using an embedded database to compute the questions and answers.

The basic function of the app is to ask the user questions and then provide diagnostic results based on their answers.

As an example of a diagnostic application, questions related to screening for mental illness were used. This means that the user will answer questions about their mental state, and the app will evaluate their answers in order to offer diagnostic results or suggestions for further action.

Key* Words*: Diagnostic Application, Visual Prolog, Symptoms, Database

Πίνακας Περιεχομένων

Λίστα Εικόνων

Λίστα Πινάκων

Εισαγωγή

Σκοπός αυτής της εργασίας είναι η δημιουργία μιας διαγνωστικής εφαρμογής σε παραθυρικό περιβάλλον υλοποιημένη στη γλώσσα Visual Prolog, όπου υποβάλλονται ερωτήσεις στον χρήστη και η εφαρμογή παράγει αποτελέσματα βάσει των απαντήσεων που δόθηκαν.

Στο πρώτο κεφάλαιο εξηγείται το τι είναι ένα διαγνωστικό πρόγραμμα και ο αντικειμενοστραφής προγραμματισμός. Επίσης εξηγείται και η γλώσσα Visual Prolog και οι διαφορές της με την γλώσσα Prolog.

Στο δεύτερο κεφάλαιο αναλύεται το πώς είναι δομημένη η εφαρμογή που δημιουργήθηκε στα πλαίσια της πτυχιακής εργασίας και έπειτα η επεξήγηση μερών του κώδικα αυτής.

Τέλος στο τρίτο κεφάλαιο παρουσιάζονται παραδείγματα εκτέλεσης της εφαρμογής με επεξήγηση και ανάλυση των βημάτων και αποτελεσμάτων.

Κεφάλαιο 1. Πηγές και Εργαλεία

1.1 Διαγνωστικό Πρόγραμμα

Ένα διαγνωστικό πρόγραμμα (γνωστό και ως Test Mode) είναι ένα πρόγραμμα υπολογιστή αυτόματης ακολουθίας που καθορίζει τη λειτουργική κατάσταση εντός του λογισμικού, υλικού ή οποιουδήποτε συνδυασμού αυτών σε ένα στοιχείο, ένα σύστημα ή ένα δίκτυο συστημάτων. Ιδανικά τα διαγνωστικά προγράμματα παρέχουν στον χρήστη καθοδήγηση σχετικά με τυχόν ζητήματα ή προβλήματα που εντοπίζονται κατά τη λειτουργία του. [1]

1.1.1 Μέθοδοι λειτουργίας Διαγνωστικών Προγραμμάτων

Ένα διαγνωστικό πρόγραμμα για μια συσκευή ή σύστημα μπορεί να είναι ανεξάρτητο ή ενσωματωμένο. Οι Μέθοδοι Λειτουργίας που ακολουθούν είναι διατεταγμένες όσο γίνεται κατά σειρά αυξανόμενης πολυπλοκότητας και αυξανόμενης αξίας διαγνωστικών πληροφοριών.

- **Παρακολούθηση δεικτών συστήματος στο παρασκήνιο** για στατιστική ανάλυση τάσεων και καταγραφή μη φυσιολογικών γεγονότων.

- **Διαγνωστικά Βασιζόμενα σε λύσεις** τα οποία δοκιμάζουν για γνωστούς τρόπους αποτυχίας προσδιορίζοντας εάν ανιχνεύονται τα γνωστά τους συμπτώματα.

- **Μαύρο κουτί** το οποίο δοκιμάζει έναν μηχανισμό χωρίς να γνωρίζει πώς λειτουργεί, και απλώς εστιάζει στην ακρίβεια των δεδομένων εξόδου που βασίζονται σε μια γνωστή είσοδο.

- **Λευκό κουτί** το οποίο χρησιμοποιεί γνώση των εσωτερικών λειτουργιών ενός μηχανισμού για άμεση δοκιμή.

- **Προσανατολισμός στη λειτουργία**, ένας συνδυασμός μαύρου και λευκού κουτιού, με μία ή περισσότερες λειτουργίες μαύρου κουτιού που παρεμβάλλονται με μία ή περισσότερες λειτουργίες λευκού κουτιού. Αυτός ο τρόπος δοκιμής δεν προτιμάται, ωστόσο, ορισμένα πολύπλοκα συστήματα δεν διαθέτουν τις απαραίτητες διεπαφές για να εκτελέσουν τον ένα ή τον άλλο τύπο ανεξάρτητα.

- **Ενσωματωμένα διαγνωστικά παρασκηνίου** που εκτελούν δοκιμές στα στοιχεία ενός συστήματος κατά τη διάρκεια αδράνειας του.

- **Διαγνωστικά διαπλοκής λειτουργίας**, τα οποία ενσωματώνουν διαγνωστικά στην κανονική λειτουργία ενός στοιχείου συστήματος, επομένως οποιοσδήποτε

οριακός τρόπος λειτουργίας διαγιγνώσκεται αμέσως. Παραδείγματα στοιχείων υλικού με δυνατότητες που βοηθούν ένα διαγνωστικό πρόγραμμα είναι:

- Οι **σύγχρονοι σκληροί δίσκοι** διαθέτουν εντολές Self-Monitoring, Analysis and Reporting Technology (S.M.A.R.T.) που παρέχουν πληροφορίες σχετικά με τις συνθήκες εσωτερικών σφαλμάτων, π.χ. αριθμός επανάληψης περιστροφής, αριθμός κακών τομέων κ.τ.λ.
- Ορισμένα συστήματα ενδέχεται να χρησιμοποιούν μνήμη **κώδικα διόρθωσης σφαλμάτων (ECC)** η οποία καταγράφει συμβάντα αποτυχίας μνήμης που διορθώθηκαν αυτόματα.

1.1.2 Αρχιτεκτονικές

➢**Διαγνωστικό ενός σκοπού**, που αναφέρεται επίσης ως διαγνωστικό "καθορισμένου σκοπού", όπως ένα πρόγραμμα που επικυρώνει τη διαμόρφωση του Windows DirectX.

➢**Διαγνωστικό πολλαπλών χρήσεων**, ένα μονολιθικό πρόγραμμα που εκτελεί πολλαπλές εργασίες που μπορεί να είναι ή να μην είναι κατάλληλες για όλες τις χρήσεις.

➢**Αρθρωτό διαγνωστικό**, το οποίο συνδυάζει σετ διαγνωστικών μίας χρήσης, τύπου Lego, σε ένα περιβάλλον που προσαρμόζεται εύκολα στις ιδιαίτερες απαιτήσεις του κλάδου. Το κλειδί για τον σχεδιασμό του είναι το επαναχρησιμοποιήσιμο λειτουργικό σύστημα υλικού και λογισμικού που εκτελεί όλα τα διαγνωστικά του προγράμματα. Παραδείγματα εφαρμογών είναι:

- **Δοκιμές κατασκευής** με έμφαση στον έλεγχο θεμάτων που σχετίζονται με τη συναρμολόγηση και τη βελτιστοποίηση για το χρόνο
- **Διαγνωστικά στοχευμένα στον τελικό χρήστη**, με εύκολη και κατανοητή μη τεχνική παρουσίαση και έμφαση στις λύσεις
- **Δοκιμές Σέρβις/Εγγύησης**, με επίκεντρο τον εντοπισμό μιας αποτυχημένης ή οριακής μονάδας με δυνατότητα αντικατάστασης πεδίου (FRU)
- **Refurbishing Centric**, το οποίο προσπαθεί να προσδιορίσει εάν ένα σύστημα μπορεί να μεταπωληθεί ή να επαναχρησιμοποιηθεί, με έμφαση στο βάθος των δοκιμών, με κόστος για τον χρόνο που δαπανάται για δοκιμές

➢**Διαγνωστικό σύστημα με γνώμονα τη γνώση** (όπως ένας τεχνικός ή ένας διαγνωστικός) όπου η γνώση που αποκτήθηκε με την πάροδο του χρόνου χρησιμοποιείται ως «νοητικό μοντέλο» της λειτουργίας του συστήματος και ενημερώνει το διαγνωστικό σύστημα μέσω λογικής συλλογιστικής για μία ή περισσότερες πιθανές ή πιθανές αιτίες για μια κατάσταση να υπάρχει. [1]

1.2 Αντικειμενοστραφής Προγραμματισμός

Ο Αντικειμενοστραφής Προγραμματισμός (Object-Oriented Programming) είναι ένας τρόπος μοντελοποίησης λογισμικού στον οποίο τα πάντα αναπαρίστανται ως αντικείμενα. Τα αντικείμενα μπορεί να είναι οποιαδήποτε πραγματική ή λογική οντότητα, όπως μια μηχανή εκτύπωσης, ένα φλιτζάνι καφέ ή μια οικονομική εταιρεία. Κάθε αντικείμενο έχει τα δικά του δεδομένα και λειτουργίες που πρέπει να εκτελέσει σε δεδομένα. Κάθε αντικείμενο μπορεί να επικοινωνεί με το άλλο όπως και όταν απαιτείται. Αυτό είναι παρόμοιο με αυτό που βιώνουμε στον πραγματικό κόσμο. [2]

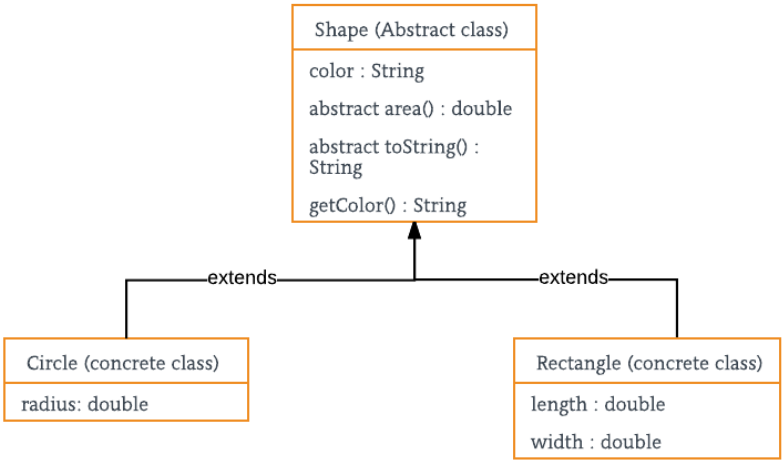
> *"Αντικειμενοστραφής Προγραμματισμός για μένα σημαίνει μόνο ανταλλαγή μηνυμάτων, τοπική διατήρηση και προστασία και απόκρυψη της διαδικασίας κατάστασης και ακραία καθυστερημένη δέσμευση όλων των πραγμάτων. Μπορεί να γίνει στο Smalltalk και στο LISP. Υπάρχουν πιθανώς άλλα συστήματα στα οποία αυτό είναι δυνατό, αλλά δεν τα γνωρίζω. [3]*

Ένα αντικείμενο είναι οποιαδήποτε οντότητα σε πραγματικό χρόνο ή λογική οντότητα που δημιουργείται στη μνήμη του υπολογιστή και αποτελείται από Δεδομένα και Λειτουργίες για εκτέλεση σε δεδομένα.

Ακολουθούν τα χαρακτηριστικά του αντικειμενοστραφούς προγραμματισμού:

- **Αφαίρεση**

Η αφαίρεση ορίζει λειτουργίες υψηλού επιπέδου που πρέπει να διαθέτει ένα αντικείμενο. Γενικά στην αφαίρεση, εστιάζουμε μόνο στις λειτουργίες που μπορεί να εκτελέσει ένα αντικείμενο παρά στον τρόπο υλοποίησης της προβλεπόμενης λειτουργίας. Η αφαίρεση βοηθά στο επίπεδο σχεδίασης ή κατά τη στιγμή της μοντελοποίησης του λογισμικού πριν από την υλοποίηση.



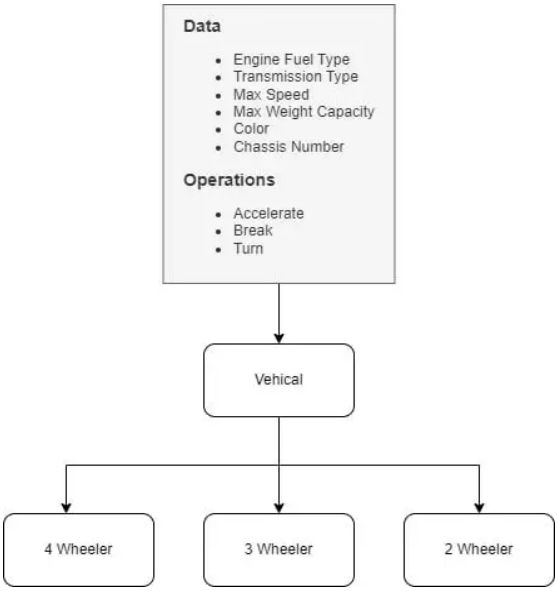
Εικόνα 1. Παράδειγμα αφαίρεσης για Java

Πηγή: https://medium.com/cloud-security/abstraction...

- **Κληρονομικότητα**

Κληρονομικότητα σημαίνει απόκτηση ιδιοτήτων και συμπεριφοράς από την Γονική κλάση.

Όπως φαίνεται στο παράδειγμα στην Εικόνα 2. Η κλάση όχημα έχει δεδομένα και μεθόδους τα οποία κληρονομούν οι κλάσεις δίτροχο, τρίτροχο και τετράτροχο.

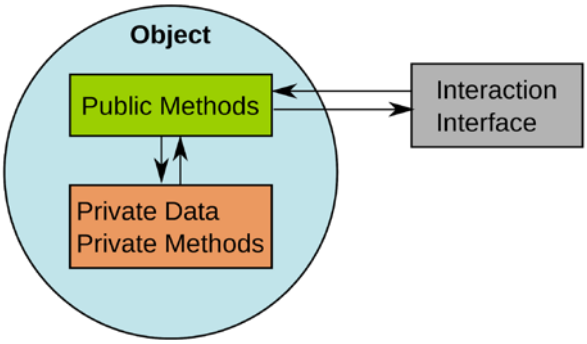


- **Ενθυλάκωση**

Η ενθυλάκωση είναι η διαδικασία συναρμολόγησης των αντικειμένων με ορισμένους περιορισμούς.

Στον Αντικειμενοστραφή Προγραμματισμό, ενσωματώνουμε τα δεδομένα και τη λειτουργία του αντικειμένου σε μια ενιαία μονάδα που ονομάζεται Κλάση με ορισμένους περιορισμούς. Οι τύποι περιορισμών ονομάζονται προσδιοριστές πρόσβασης και είναι : Δημόσιο, Ιδιωτικό, Προστατευμένο & Εσωτερικό κ.λπ..

Οι προσδιοριστές πρόσβασης μπορούν να εφαρμοστούν σε δεδομένα, λειτουργίες και κλάσεις.



Εικόνα 3. Παράδειγμα Ενθυλάκωσης

Τα δεδομένα και μέθοδοι της κλάσης είναι προσβάσιμα μόνο από τις δημόσιες μεθόδους.

Πηγή: https://www.dotnettutorial.co.in/2018/05/...:

- **Πολυμορφισμός**

Πολυμορφισμός είναι η ικανότητα να ενεργούμε διαφορετικά σύμφωνα με διαφορετικούς τύπους εισόδου. [2]

1.3 Βάση Δεδομένων

Μια βάση δεδομένων είναι μια οργανωμένη συλλογή δομημένων πληροφοριών ή δεδομένων, που συνήθως αποθηκεύονται ηλεκτρονικά σε ένα σύστημα υπολογιστή. Μια βάση δεδομένων ελέγχεται συνήθως από ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS). Τα δεδομένα και το DBMS, μαζί με τις εφαρμογές που σχετίζονται με αυτά, αναφέρονται ως σύστημα βάσης δεδομένων, συχνά συντομευμένο απλά ως βάση δεδομένων.

Τα δεδομένα εντός των πιο κοινών τύπων βάσεων δεδομένων που λειτουργούν σήμερα μοντελοποιούνται συνήθως σε γραμμές και στήλες σε μια σειρά πινάκων για να καταστήσουν αποτελεσματική την επεξεργασία και την αναζήτηση δεδομένων. Μπορεί στη συνέχεια να είναι εύκολη η πρόσβαση, διαχείριση, τροποποίηση, ενημέρωση, έλεγχος και οργάνωση των Δεδομένων. Οι περισσότερες βάσεις δεδομένων χρησιμοποιούν δομημένη γλώσσα ερωτημάτων (SQL) για τη σύνταξη και την αναζήτηση δεδομένων. [4]

1.3.1 Τύποι βάσεων δεδομένων

Υπάρχουν πολλοί τύποι βάσεων δεδομένων. Μπορούν να ταξινομηθούν ανάλογα με τον τύπο περιεχομένου: βιβλιογραφικό, πλήρες κείμενο, αριθμητικά και εικόνες. Στην πληροφορική, οι βάσεις δεδομένων ταξινομούνται συχνά με βάση την οργανωτική προσέγγιση που χρησιμοποιούν.

Μερικές από τις κύριες βάσεις δεδομένων του οργανισμού περιλαμβάνουν τα ακόλουθα:

- **Σχεσιακή**: Αυτή η προσέγγιση σε πίνακα ορίζει τα δεδομένα ώστε να μπορούν να αναδιοργανωθούν και να προσπελαστούν με πολλούς τρόπους. Οι σχεσιακές βάσεις δεδομένων αποτελούνται από πίνακες. Τα δεδομένα τοποθετούνται σε προκαθορισμένες κατηγορίες σε αυτούς τους πίνακες.

  Κάθε πίνακας έχει στήλες με τουλάχιστον μία κατηγορία δεδομένων και σειρές που έχουν μια συγκεκριμένη παρουσία δεδομένων για τις κατηγορίες που ορίζονται στις στήλες.

  Οι πληροφορίες σε μια σχεσιακή βάση δεδομένων σχετικά με έναν συγκεκριμένο πελάτη οργανώνονται σε γραμμές, στήλες και πίνακες. Αυτά είναι ευρετηριασμένα για να διευκολύνουν την αναζήτηση χρησιμοποιώντας ερωτήματα SQL ή NoSQL.

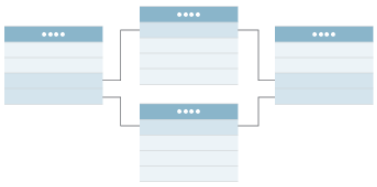
- **Διανεμημένες**: Αυτή η βάση δεδομένων αποθηκεύει εγγραφές ή αρχεία σε πολλές φυσικές τοποθεσίες. Η επεξεργασία δεδομένων κατανέμεται επίσης και αναπαράγεται σε διάφορα μέρη του δικτύου.

- **Cloud**: Αυτές οι βάσεις δεδομένων είναι χτισμένες σε δημόσιο, ιδιωτικό ή υβριδικό Cloud για ένα εικονικό περιβάλλον.
  Οι χρήστες χρεώνονται με βάση τον αποθηκευτικό χώρο και το εύρος ζώνης που χρησιμοποιούν. Έχουν επίσης επεκτασιμότητα κατόπιν ζήτησης και υψηλή διαθεσιμότητα. Αυτές οι βάσεις δεδομένων μπορούν να λειτουργήσουν με εφαρμογές που αναπτύσσονται ως λογισμικό ως υπηρεσία.

- **NoSQL**: Οι βάσεις δεδομένων NoSQL είναι καλές όταν ασχολούμαστε με μεγάλες συλλογές κατανεμημένων δεδομένων.
  Μπορούν να αντιμετωπίσουν ζητήματα απόδοσης μεγάλων δεδομένων καλύτερα από τις σχεσιακές βάσεις δεδομένων. Επίσης, αναλύουν καλά μεγάλα μη δομημένα σύνολα δεδομένων και δεδομένα σε εικονικούς διακομιστές στο cloud. Αυτές οι βάσεις δεδομένων μπορούν επίσης να ονομαστούν μη σχεσιακές βάσεις δεδομένων.

- **Αντικειμενοστραφής**: Αυτές οι βάσεις δεδομένων διατηρούν δεδομένα που δημιουργούνται χρησιμοποιώντας αντικειμενοστραφείς γλώσσες προγραμματισμού.
  Επικεντρώνονται στην οργάνωση αντικειμένων παρά σε ενέργειες και δεδομένα ή στη λογική. Για παράδειγμα, μια εγγραφή δεδομένων εικόνας θα ήταν ένα αντικείμενο δεδομένων και όχι μια αλφαριθμητική τιμή.

- **Γραφική παράσταση**: Αυτές οι βάσεις δεδομένων είναι ένας τύπος βάσης δεδομένων NoSQL. Αποθηκεύουν, χαρτογραφούν και αναζητούν σχέσεις χρησιμοποιώντας έννοιες από τη θεωρία γραφημάτων.
  Οι βάσεις δεδομένων γραφημάτων αποτελούνται από κόμβους και ακμές. Οι κόμβοι είναι οντότητες και συνδέουν τους κόμβους. [5]

	Graph database	Relational database
FORMAT	Nodes and edges	Tables with rows and columns
RELATIONSHIPS	Considered data, represented by edges between nodes	Related across tables, established using foreign keys between tables
COMPLEX QUERIES	Run quickly and do not require joins	Require complex joins between tables
TOP USE CASES	Relationship-heavy use cases, including fraud detection and recommendation engines	Transaction-focused use cases, including online transactions and accounting
EXAMPLE		

Εικόνα 4. Σύγκριση Βάσεων Δεδομένων Σχεσιακής με Γραφικής Παράστασης

Πηγή: https://www.techtarget.com/...

1.4 Visual Prolog

Η Visual Prolog είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που βασίζεται στη λογική γλώσσα προγραμματισμού Prolog. Ο στόχος του Visual Prolog είναι να διευκολύνει τις προγραμματικές λύσεις σύνθετων προβλημάτων με έμφαση στη γνώση.

1.4.1 Διαφορές μεταξύ Visual Prolog και παραδοσιακής Prolog

Οι διαφορές μεταξύ της παραδοσιακής Prolog και της Visual Prolog μπορούν να χωριστούν ευρέως στις παρακάτω κατηγορίες:

- **Διαφορές στη Δομή του Προγράμματος:** Υπάρχουν ευδιάκριτες, αλλά εύκολα κατανοητές διαφορές μεταξύ της δομής που χρησιμοποιείται στην παραδοσιακή Prolog και αυτής που χρησιμοποιείται στο Visual Prolog. Ουσιαστικά περιλαμβάνει την κατανόηση του τρόπου επισήμανσης των δηλώσεων από τους ορισμούς και την ένδειξη του κύριου στόχου που πρέπει να επιδιώξει το πρόγραμμα χρησιμοποιώντας συγκεκριμένες λέξεις-κλειδιά.
- **Θέματα αρχείου:** Η Visual Prolog παρέχει διάφορες ευκολίες για την οργάνωση της δομής του προγράμματος σε διαφορετικά είδη αρχείων.
- **Ζητήματα Πρόσβασης Πεδίου:** Ένα πρόγραμμα Visual Prolog μπορεί να επιλέξει τη λειτουργικότητα που έχει αναπτυχθεί σε άλλες λειτουργικές μονάδες χρησιμοποιώντας μια έννοια που ονομάζεται αναγνώριση εύρους.
- **Αντικειμενοστραφής:** Ένα πρόγραμμα Visual Prolog μπορεί να γραφτεί ως αντικειμενοστραφή πρόγραμμα, χρησιμοποιώντας κλασικά αντικειμενοστραφή χαρακτηριστικά. [6]

1.4.2 **Δηλώσεις και ορισμοί**

Στην Prolog, όταν χρειάζεται να χρησιμοποιήσουμε ένα κατηγόρημα, το κάνουμε απλώς χωρίς καμία προηγούμενη ένδειξη στη μηχανή Prolog σχετικά με την πρόθεσή μας.

Ομοίως, όταν χρησιμοποιείται ένας σύνθετος τομέας στην παραδοσιακή Prolog, μπορούμε να τον χρησιμοποιήσουμε χωρίς να προειδοποιήσουμε πρώτα τη μηχανή Prolog σχετικά με την πρόθεσή μας να χρησιμοποιήσουμε τον τομέα. Απλώς χρησιμοποιούμε έναν τομέα τη στιγμή που νιώθουμε την ανάγκη του.

Ωστόσο, στο Visual Prolog, πριν γράψουμε τον κώδικα για το σώμα της πρότασης ενός κατηγορήματος, πρέπει πρώτα να δηλώσουμε την ύπαρξη ενός τέτοιου κατηγορήματος. Ομοίως, πριν χρησιμοποιήσετε οποιονδήποτε τομέα, πρέπει να δηλωθούν. Ο λόγος που απαιτούνται τέτοιες προειδοποιήσεις στο Visual Prolog είναι να διασφαλιστεί ότι οι εξαιρέσεις χρόνου εκτέλεσης μετατρέπονται σε σφάλματα χρόνου μεταγλώττισης όσο το δυνατόν περισσότερο.

1.4.3 **Λέξεις Κλειδιά**

Ένα πρόγραμμα Visual Prolog αποτελείται από κώδικα Prolog ο οποίος σημειώνεται σε διαφορετικές ενότητες με κατάλληλες λέξεις-κλειδιά που ενημερώνουν τον μεταγλωττιστή τον κώδικα που πρέπει να δημιουργήσει.

Για παράδειγμα, υπάρχουν λέξεις-κλειδιά που διαφοροποιούν τις δηλώσεις από τους ορισμούς κατηγορημάτων και τομέων. Συνήθως, κάθε ενότητα προηγείται από μια λέξη-κλειδί. Συνήθως δεν υπάρχει λέξη-κλειδί που να δηλώνει το τέλος μιας συγκεκριμένης ενότητας. Η παρουσία άλλης λέξης-κλειδιού υποδηλώνει το τέλος της προηγούμενης ενότητας και την αρχή της επόμενης.

Εξαίρεση σε αυτόν τον κανόνα, αποτελούν οι λέξεις-κλειδιά "implement" και "end implement". Ο κώδικας που περιέχεται μεταξύ αυτών των δύο λέξεων-κλειδιών υποδεικνύει τον κώδικα που θα χρησιμοποιηθεί για μια συγκεκριμένη κλάση.

Η λίστα με τις λέξεις-κλειδιά που πρέπει να γνωρίζετε είναι η εξής:

- **implement and end implement**

Μεταξύ όλων των λέξεων-κλειδιών που συζητούνται εδώ, αυτή είναι η μόνη που υπάρχει ως ζευγάρι. Η Visual Prolog αντιμετωπίζει τον κώδικα που γράφεται μεταξύ αυτών των δύο λέξεων-κλειδιών ως τον κώδικα που ανήκει σε μία κλάση. Το όνομα της κλάσης πρέπει να δίνεται μετά τη λέξη-κλειδί implement.

➢ **open**

Αυτή η λέξη-κλειδί χρησιμοποιείται για την επέκταση της ορατότητας του εύρους της τάξης. Πρέπει να χρησιμοποιείται αμέσως μετά τη λέξη-κλειδί implement.

➢ **domains**

Αυτή η λέξη-κλειδί χρησιμοποιείται για την επισήμανση μιας ενότητας που δηλώνει τους τομείς που θα χρησιμοποιηθούν στον κώδικα. Υπάρχουν πολλές παραλλαγές για τη σύνταξη τέτοιων δηλώσεων τομέα και καλύπτουν όλα τα πιθανά είδη τομέων που θα χρησιμοποιηθούν αργότερα στον κώδικα.

➢ **class facts**

Αυτή η λέξη-κλειδί προσδιορίζει μια ενότητα, η οποία δηλώνει τα γεγονότα που θα χρησιμοποιηθούν αργότερα στον κώδικα του προγράμματος. Κάθε γεγονός δηλώνεται με το όνομα που χρησιμοποιείται για να υποδηλώσει το γεγονός και τα ορίσματα που χρησιμοποιούνται για τα αντίστοιχα γεγονότα μαζί με τους τομείς στους οποίους ανήκουν αυτά τα ορίσματα.

➢ **class predicates**

Αυτή η ενότητα περιέχει τις δηλώσεις των κατηγορημάτων που θα οριστούν αργότερα στην ενότητα clauses του κώδικα.

➢ **clauses**

Από όλες τις ενότητες που υπάρχουν σε έναν κώδικα Visual Prolog, αυτή η ενότητα είναι αυτή που μιμείται πολύ ένα παραδοσιακό πρόγραμμα Prolog. Περιέχει τους πραγματικούς ορισμούς των προηγουμένως δηλωμένων κατηγορημάτων. Τα κατηγορήματα που χρησιμοποιούνται εδώ ακολουθούν τη σύνταξη τους όπως δηλώνεται στην ενότητα class predicates. [6]

1.4.4 Αρχειοθέτηση

Η Visual Prolog έχει τη δυνατότητα να διαιρεί τον κώδικα του προγράμματος σε ξεχωριστά αρχεία χρησιμοποιώντας το IDE (Integrated Development Environment) και χρησιμοποιώντας αυτό το IDE είναι δυνατή η εγγραφή καθαρών κομματιών κώδικα σε ξεχωριστά αρχεία. Όταν γίνεται με αυτόν τον τρόπο, τα πράγματα που πρόκειται να χρησιμοποιηθούν συνήθως είναι προσβάσιμα σε όλα τα αρχεία.

Κεφάλαιο 2. Ανάλυση Δομής και Κώδικα Εφαρμογής

Η υλοποίηση της διαγνωστικής εφαρμογής έγινε στη γλώσσα προγραμματισμού Visual Prolog σε παραθυρικό περιβάλλον.

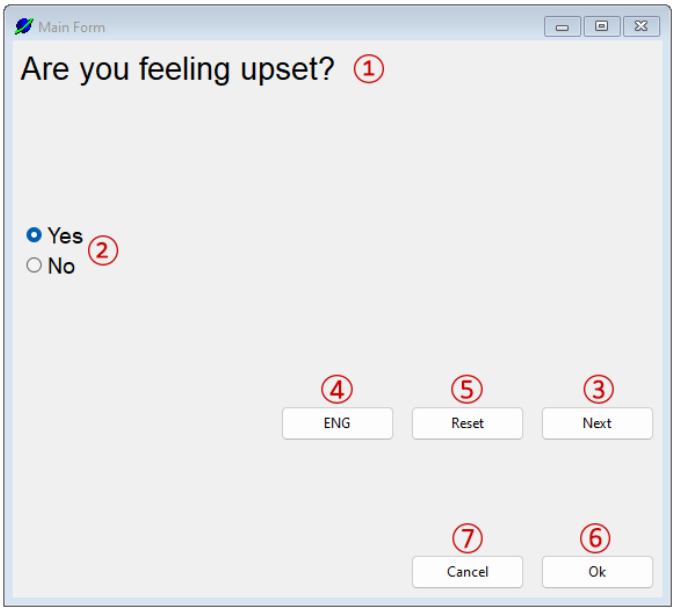
Η εφαρμογή λειτουργεί κάνοντας ερωτήσεις οι οποίες ανακτώνται από βάση δεδομένων στην οποία ορίζονται τα κείμενα, οι ερωτήσεις, απαντήσεις ακόμη και ποια είναι η επόμενη ερώτηση για κάθε πιθανή απάντηση.

2.1 Δομή Εφαρμογής

Η εφαρμογή χωρίζεται σε 3 διακριτά μέρη, τα οποία είναι:

2.1.1 Κεντρική Φόρμα

Στη Κεντρική Φόρμα πραγματοποιούνται όλες οι ερωτήσεις και απαντήσεις του διαγνωστικού προγράμματος.

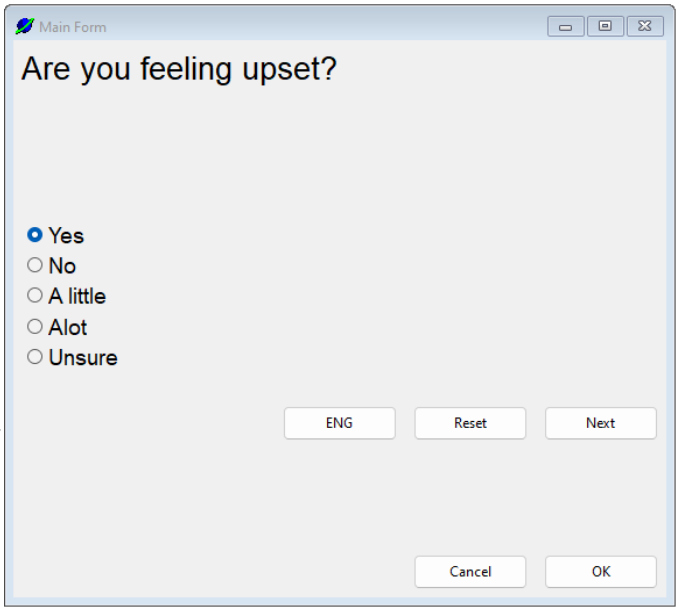


Τα Στοιχεία της Κεντρικής Φόρμας είναι:

1) Το κείμενο της τρέχουσας ερώτησης.
2) Τα κουμπιά επιλογών.
3) Το κουμπί επόμενο.
4) Το κουμπί αλλαγής γλώσσας.
5) Το κουμπί Επαναφοράς.
6) Το κουμπί Κλείσιμο με εμφάνιση αποτελεσμάτων.
7) Το κουμπί Κλείσιμο χωρίς εμφάνιση αποτελεσμάτων.

➢ **Κείμενο Τρέχουσας Ερώτησης και Κουμπιά επιλογών**

Η τρέχον ερώτηση εμφανίζεται πάντα στο πεδίο κειμένου τρέχουσας ερώτησης το οποίο ενημερώνεται κάθε φορά που αλλάζει η ερώτηση



Τα κουμπιά επιλογών χρησιμοποιούνται για την επιλογή της απάντησης που ταιριάζει καλύτερα στην ερώτηση που γίνεται. Για το παράδειγμα που θα ακολουθήσει θα χρησιμοποιούνται μόνο 2 απαντήσεις ανά ερώτηση αλλά η εφαρμογή υποστηρίζει έως και 5 πιθανές απαντήσεις σε κάθε ερώτηση τα κουμπιά των οποίων εμφανίζονται μόνο όταν χρειάζεται.

➢ **Κουμπί Επόμενο.**

Το κουμπί επόμενο βρίσκει πρώτα ποια απάντηση έχει επιλεχθεί, μετά καταχωρεί το αντίστοιχο σύμπτωμα τέλος υπολογίζει ποια ερώτηση ακολουθεί και θέτει νέα κείμενα και απαντήσεις.

Σε περίπτωση που δεν υπάρχει άλλη ερώτηση η λειτουργία του κουμπιού αλλάζει ώστε κατά το πάτημα του να τυπώνει όσα αποτελέσματα έχουν βρεθεί.

➢ **Κουμπί Αλλαγής Γλώσσας**

Αυτό το κουμπί αλλάζει τη γλώσσα και όλα τα κείμενα από Αγγλικά σε Ελληνικά και αντίστροφα το οποίο είναι εφικτό σε οποιαδήποτε στιγμή κατά τη διάρκεια της διάγνωσης.

➢ **Κουμπί Επαναφοράς**

Το κουμπί επαναφοράς γυρίζει τη φόρμα πίσω στην αρχική της κατάσταση και καταργεί όσα συμπτώματα έχουν καταχωρηθεί ως τώρα.

➢ **Κουμπιά Κλεισίματος**

Το κουμπί Οκ θα τυπώσει ότι αποτελέσματα μπορεί να αντιστοιχίσει με βάσει τα συμπτώματα που έχουν καταχωρηθεί μέχρι στιγμής και θα κλείσει τη φόρμα.

Το κουμπί Άκυρο θα κλείσει τη φόρμα και δεν θα τυπώσει τίποτα.

2.1.2 Βάση Δεδομένων

Τη Βάση Δεδομένων όπου βρίσκονται όλες οι πληροφορίες σχετικά με τις ερωτήσεις και απαντήσεις που θα κάνει η κεντρική φόρμα. Η βάση δεδομένων περιέχει σύνολο 6 πίνακες οι οποίοι είναι οι εξής:

- **Πίνακας Περιγραφών**

  Ο πίνακας Περιγραφών περιέχει το κείμενο μιας ερώτησης τα κείμενα περιγραφών κάθε απάντησης τα οποία εμφανίζονται δεξιά των επιλογών απαντήσεων κατά την επιλογή τους στη κεντρική φόρμα.

- **Πίνακας Τίτλων Κουμπιών Επιλογής**

  Ο πίνακας τίτλων κουμπιών ορίζει τι όνομα θα έχει το κάθε κουμπί επιλογής σε μια ερώτηση.

- **Πίνακας Κατάστασης Κουμπιών Επιλογής**

  Έπειτα ο πίνακας κατάστασης κουμπιών επιλογής ορίζει ποια κουμπιά επιλογής θα είναι ενεργά και ποια όχι.

- **Πίνακας Επόμενης Κατάστασης**

  Ο πίνακας επόμενης κατάστασης περιέχει ένα αναγνωριστικό απάντησης για κάθε απάντηση σε μια ερώτηση με σκοπό να χρησιμοποιηθούν στον πίνακα αντιστοίχισης επόμενης κατάστασης.

- **Πίνακας Αντιστοίχισης Απαντήσης**

  Ο πίνακας αντιστοίχισης απάντησης αντιστοιχεί το αναγνωριστικό απάντησης από τον πίνακα επόμενης κατάστασης με σύμπτωμα και την ερώτηση που ακολουθεί.

  Αυτό δίνει την δυνατότητα πολλές απαντήσεις να δίνουν το ίδιο σύμπτωμα αλλά να ακολουθεί διαφορετική ερώτηση όπως και το αντίστροφο, δηλαδή πολλές απαντήσεις να δίνουν διαφορετικό σύμπτωμα αλλά να ακολουθεί η ίδια ερώτηση με τις άλλες.

- **Πίνακας Αντιστοίχισης Αποτελεσμάτων**

  Τέλος ο πίνακας αντιστοίχισης αποτελεσμάτων κάνει αντιστοίχιση τιμή από τον Υπολογισμό Αποτελεσμάτων μαζί με την επιλεγμένη Γλώσσα με το κείμενο αποτελέσματος προς τύπωση δίνοντας δυνατότητα τύπωσης των αποτελεσμάτων σε διαφορετικές Γλώσσες.

2.1.3 Υπολογισμός Αποτελεσμάτων

Τέλος ο Υπολογισμός Αποτελεσμάτων όπου η εφαρμογή αποθηκεύει τα συμπεράσματα και αργότερα υπολογίζει τα αποτελέσματα.
Ο υπολογισμός αποτελεσμάτων περιέχει τα παρακάτω:

- Αποτελέσματα τα Συμπτώματα τους
- Καταχώρηση Συμπτωμάτων
- Υπολογισμός και Τύπωση Αποτελεσμάτων

2.2 Ανάλυση Κώδικα

Παρακάτω γίνεται ανάλυση του κώδικα των κατηγορημάτων της εφαρμογής.

2.2.1 Κατηγορήματα Βάσης Δεδομένων

➢ Κατηγορήματα Ανάκτησης Δεδομένων από τους Πίνακες

Τα κατηγορήματα Ανάκτησης Περιγραφών και Ονομάτων Κουμπιών λαμβάνουν ως είσοδο αναγνωριστικό ερώτησης και γλώσσα και βγάζουν έξοδο το κείμενο της ερώτησης συν τις περιγραφές απαντήσεων και τους τίτλους κουμπιών αντίστοιχα. Σε περίπτωση που δεν βρεθεί αντιστοίχιση επιστρέφουν κενό ως τιμές.

Πίνακας 1. Κώδικας Κατηγορημάτων Κειμένων Κεντρικής Φόρμας

```
getDesc(ID, Lang, Description, Ds1, Ds2, Ds3, Ds4, Ds5) :-
    description(ID, Lang, Description, Ds1, Ds2, Ds3, Ds4, Ds5),
    !.
  getDesc(_ID, _Lang, " ", " ", " ", " ", " ", " ").

      getRname(ID, Lang, Rd1, Rd2, Rd3, Rd4, Rd5) :-
        radioname(ID, Lang, Rd1, Rd2, Rd3, Rd4, Rd5),
        !.
  getRname(_ID, _Lang, " ", " ", " ", " ", " ").
```

Τα κατηγορήματα Κατάστασης Κουμπιών Επιλογών λαμβάνουν ως είσοδο μόνο το αναγνωριστικό ερώτησης βγάζουν έξοδο τις καταστάσεις κουμπιών επιλογής και τα αναγνωριστικά απαντήσεων. Αν δεν βρεθεί αντιστοίχιση τότε επιστρέφει ως απάντηση τέλος διάγνωσης.

Πίνακας 2. Κώδικας Κατηγορημάτων Κατάστασης και Αναγνωριστικών Κουμπιών Επιλογής

```
      getRstate(ID, SRd1, SRd2, SRd3, SRd4, SRd5) :-
        radiostate(ID, SRd1, SRd2, SRd3, SRd4, SRd5),
        !.
      getRstate(_ID, false, false, false, false, false).

      getNstate(ID, Nx1, Nx2, Nx3, Nx4, Nx5) :-
        nextstate(ID, Nx1, Nx2, Nx3, Nx4, Nx5),
        !.
      getNstate(_ID, "END", "END", "END", "END", "END").
```

Το Κατηγόρημα Αντιστοίχισης Απαντήσεων λαμβάνει ως είσοδο Αναγνωριστικό Απάντησης και βγάζει έξοδο Σύμπτωμα και Αναγνωριστικό Ερώτησης.

Πίνακας 3. Κώδικας Κατηγορήματος Αντιστοίχισης Απαντήσεων

```
getID(ID, SYMPTOM, NEXTID) :-
   symptomID(ID, SYMPTOM, NEXTID),
   !.
getID(_ID, "END", "END").
```

Το κατηγόρημα Αντιστοίχισης Αποτελεσμάτων λαμβάνει ως είσοδο αναγνωριστικό αποτελέσματος με γλώσσα και το αντιστοιχεί με αποτέλεσμα προς τύπωση.

Πίνακας 4. Κώδικας Κατηγορήματος Αντιστοίχισης Αποτελεσμάτων

```
getResult(ID, LANG, RESULT) :-
   endresults(ID, LANG, RESULT),
   !.
getResult(_ID, _LANG, "").
```

➢ Κατηγόρημα Ενημέρωσης Πινάκων από Αρχείο

Το κατηγόρημα ενημέρωσης πινάκων ξεκινάει ελέγχοντας αν το αρχείο idDatabase.txt υπάρχει στον ίδιο φάκελο με την εφαρμογή, έπειτα καθαρίζει όλα τα υπάρχον δεδομένα από όλους τους πίνακες στη βάση δεδομένων και τέλος εισάγει νέα δεδομένα από το αρχείο. Αν το αρχείο idDatabase.txt δεν βρεθεί τότε βγάζει προειδοποιητικό μήνυμα ότι το αρχείο δεν βρέθηκε.

Πίνακας 5. Κώδικας Κατηγορήματος Βάσης Δεδομένων 5

```
consultDatabase() :-
    file::existExactFile("idDatabase.txt"),
    !,
    getData().
  consultDatabase() :-
    vpiCommonDialogs::error("WARNING", "File idDatabase.txt not
found.\nDatabase is Empty"). %Ensure the file idDatabase.txt exists in the executable
directory
  /* Delete all existing entries from database then Fill with new ones from File */
  getData() :-
    retractFactDB(descriptionDB),
    file::consult("idDatabase.txt", descriptionDB).
```

2.2.2 **Κατηγορήματα Αποτελεσμάτων**

➢ Κατηγόρημα go

Το Κατηγόρημα go δέχεται είσοδο επιλογή γλώσσας, ελέγχει ποια γλώσσα είναι επιλεγμένη έπειτα τυπώνει κείμενο ποια είναι τα αποτελέσματα στην αντίστοιχη γλώσσα, τέλος καλεί το κατηγόρημα gradeUser με είσοδο την επιλογή γλώσσας.

Πίνακας 6. Κώδικας Κατηγορήματος go Αποτελεσμάτων

```
go(LANG) :-
    if LANG = "ENG" then
        stdio::write("The Results Are: \n"),
        gradeUser(LANG)
    elseif LANG = "GRE" then
        stdio::write("Τα Αποτελέσματα είναι: \n"),
        gradeUser(LANG)
    end if,
    fail.
go(_LANG).
```

➢ Κατηγόρημα assertsymptom

Το Κατηγόρημα assertsymptom κάνει έλεγχο πρώτα ότι το σύμπτωμα προς καταχώρηση δεν είναι κενό και ότι δεν έχει καταχωρηθεί ήδη, και έπειτα το καταχωρεί.

Πίνακας 7. Κώδικας Κατηγορήματος Καταχώρησης Συμπτωμάτος

```
assertsymptom(Symptom) :-
    if not(Symptom = "") and not(symptom(Symptom)) then
        assert(symptom(Symptom))
    end if.
```

➢ Κατηγόρημα gradeUser

Το Κατηγόρημα gradeUser είναι υπεύθυνο για την εύρεση και τύπωση των αποτελεσμάτων και έχει είσοδο επιλογή γλώσσας, πρώτα ψάχνει για έγκυρο αποτέλεσμα βάσει τον συμπτωμάτων που καταχωρήθηκαν αυτό επιστρέφει αναγνωριστικό αποτελέσματος, έπειτα με κάνει κλήση το κατηγόρημα getResult από τη βάση δεδομένων δίνοντας ως είσοδο το αναγνωριστικό αποτελέσματος και την γλώσσα, τυπώνει το αποτέλεσμα και επαναλαμβάνει για κάθε έγκυρο αποτέλεσμα.

Πίνακας 8. Κώδικας Κατηγορήματος Τύπωσης Αποτελεσμάτων

```
gradeUser(LANG) :-
    hypothesis(ResultID),
    database::getResult(ResultID, LANG, Review),
    stdIo::write(Review, "\n").
```

➢ Κατηγόρημα estimateUser

Το Κατηγόρημα estimateUser έχει ως σκοπό τον έλεγχο αν υπάρχει τουλάχιστον ένα έγκυρο αποτέλεσμα, αν ναι τερματίζει και επιστρέφει τον αναγνωριστικό του, αν δεν βρεθεί κανένα αποτέλεσμα επιστρέφει κενό.

Πίνακας 9. Κώδικας Κατηγορήματος Ελέγχου Αποτελεσμάτος

```
estimateUser(ResultID) :-
    hypothesis(ResultID),
    !.
estimateUser("").
```

2.2.3 **Κατηγορήματα Κεντρικής Φόρμας**

Η κεντρική φόρμα αποτελείται από τα παρακάτω κατηγορήματα:

➢ Κώδικας Αρχικοποίησης

Πίνακας 10. Κώδικας Αρχικοποίησης Φόρμας

```
new(Parent) :-
    resultsDB::initial(),
    description::consultDatabase(),
    [...]
    initForm("BEGIN", lang:getText()),
    rbuttons:selectAt(1, true),
    rbuttons:addList(["NoSel", "c1", "c2", "c3", "c4", "c5"]),
    ch1_ctl:setRadioState(radioButton::checked).
```

Οι παραπάνω εντολές εκτελούνται κατά το άνοιγμα νέας φόρμας με την επιλογή του κουμπιού New Form οι οποίες ετοιμάζουν τη φόρμα σε μια αρχική κατάσταση ώστε να ξεκινήσει μια νέα διάγνωση.α

Αρχικά γίνεται εκκαθάριση των καταχωρημένων συμπτωμάτων και πραγματοποιείται ενημέρωση της βάσης δεδομένων από το αρχείο.

Έπειτα γίνεται αρχικοποίηση των κουμπιών και κειμένων της φόρμας. Δημιουργείται λίστα με τη χρήση της οποίας γίνεται ανάκτηση της επιλεγμένης απάντησης σε μορφή αριθμού με βάση τη σειρά εμφάνισης π.χ. 1η απάντηση επιστρέφει 1, 2η 2 κτλ. και επιλέγεται η επιλογή 1 στη λίστα.

Τέλος γίνεται προεπιλογή της απάντησης Ναι στη Φόρμα.

➢ Κατηγόρημα Go

Το κατηγόρημα Go πραγματοποιείται σε 3 μέρη, έχει ως είσοδο την μεταβλητή Calc που είναι αριθμός ο οποίος αντιστοιχεί στην επιλεγμένη απάντηση στη φόρμα.

Στο πρώτο μέρος γίνεται η καταχώρηση του συμπτώματος, πρώτα απενεργοποιείται το κουμπί επόμενο για αποφυγή διπλού πατήματος του πριν ολοκληρωθεί η εκτέλεση του κατηγορήματος, έπειτα διαβάζει από τη φόρμα σε ποια ερώτηση βρίσκεται τη στιγμή της κλήσης του Go και το καταχωρεί στην μεταβλητή ID, ελέγχει ότι δεν βρίσκεται στο τέλος της διάγνωσης, αν όχι τότε διαβάζει από τη βάση δεδομένων τις απαντήσεις που αντιστοιχούν σε κάθε επιλογή απάντησης, μετά χρησιμοποιώντας τον αριθμό που εισάγεται στην κλήση του κατηγορήματος βρίσκει ποιο σύμπτωμα αντιστοιχεί στην απάντηση που δόθηκε και το καταχωρεί στο αρχείο αποτελεσμάτων.

Πίνακας 11. Μέρος 1ο Κώδικα Κατηγορήματος Go της Φόρμας

```
go(Calc) :-
    %Assert Symptom
    next_ctl:setEnabled(false), %DISABLE NEXT BUTTON
    ID = currentstate:getText(),
    if not(ID = "END") then
        database::getNstate(ID, Nx1, Nx2, Nx3, Nx4, Nx5),
        if Calc = 1 then
            database::getID(NX1, Symptom, _),
            resultsDB::assertsymptom(Symptom)
        elseif Calc = 2 then
            database::getID(NX2, Symptom, _),
            resultsDB::assertsymptom(Symptom)
        elseif Calc = 3 then
            database::getID(NX3, Symptom, _),
            resultsDB::assertsymptom(Symptom)
        elseif Calc = 4 then
            database::getID(NX4, Symptom, _),
            resultsDB::assertsymptom(Symptom)
        elseif Calc = 5 then
            database::getID(NX5, Symptom, _),
            resultsDB::assertsymptom(Symptom)
        end if
    end if,
    fail.
```

Έπειτα στο δεύτερο μέρος αρχικά όπως και στο πρώτο μέρος διαβάζει σε ποια ερώτηση βρίσκεται, διαβάζει ποια γλώσσα είναι επιλεγμένη, ελέγχει αν βρίσκεται στο τέλος της διάγνωσης ή όχι, αν ναι τότε ελέγχει ότι υπάρχει τουλάχιστον 1 αποτέλεσμα που αντιστοιχεί στα συμπτώματα, αν ναι τυπώνει όσα αποτελέσματα αντιστοιχούν, αν όχι τυπώνει ότι δεν βρέθηκαν αποτελέσματα.

Αν δεν βρίσκεται στο τέλος της διάγνωσης διαβάζει τις απαντήσεις που αντιστοιχούν στις επιλογές απάντησης και αναλόγως με ποια επιλογή είναι αντιστοιχεί από τη βάση ποια είναι η επόμενη ερώτηση και θέτει τη νέα ερώτηση και κείμενα.

Πίνακας 12. Μέρος 2ο Κώδικα Κατηγορήματος Go της Φόρμας

```
go(Calc) :-
    %Read data needed
    ID = currentstate:getText(),
    LANG = lang:getText(),
    %If at end of diagnosis check for and print results
    if ID = "END" then
        %Check if at least one result exists
        resultsDB::estimateUser(Result),
        if not(Result = "") then
            resultsDB::go(LANG),
            stdio::write("\n")
        elseif LANG = "ENG" then
            stdio::write("No Results Were Found \n")
        elseif LANG = "GRE" then
            stdio::write("Δεν Βρέθηκαν Αποτελέσματα \n")
        end if
    end if,

    if not(ID = "END") then
        database::getNstate(ID, Nx1, Nx2, Nx3, Nx4, Nx5),
        % Sets New Texts Based on Selected Button
        if Calc = 1 then
            database::getID(NX1, _, ST),
            currentstate:setText(ST),
            initForm(ST, LANG)
        […]
        elseif Calc = 5 then
            database::getID(NX5, _, ST),
            currentstate:setText(ST),
            initForm(ST, LANG)
        end if
    end if,
    fail.
```

Τέλος στο τρίτο μέρος ελέγχει αν βρίσκεται στο τέλος της διάγνωσης, αν ναι θέτει τα κείμενα των βασικών κουμπιών Επόμενο, OK και Άκυρο σε Αποτελέσματα, Τέλος ή στα Αγγλικά αν η επιλεγμένη γλώσσα είναι η Αγγλική.

Πίνακας 13. Μέρος 3ο Κώδικα Κατηγορήματος Go της Φόρμας

```
go(_Calc) :-
    %Change main buttons text if at end of diagnosis
    if currentstate:getText() = "END" then
      next_ctl:setEnabled(true),
      LANG = lang:getText(),
      if LANG = "ENG" then
        cancel_ctl:setText("Cancel"),
        ok_ctl:setText("END"),
        next_ctl:setText("Results")
      else
        ok_ctl:setText("Τέλος"),
        cancel_ctl:setText("Άκυρο"),
        next_ctl:setText("Αποτελέσματα")
      end if
    end if,
    next_ctl:setEnabled(true).
```

➢ Κατηγόρημα Reset

Το κατηγόρημα Reset προστέθηκε ως ένας τρόπος επαναφοράς της φόρμας στην αρχική κατάσταση χωρίς να χρειάζεται ο χρήστης να κλείσει τη φόρμα και να ανοίξει καινούργια.

Αυτό που κάνει είναι το ίδιο με τον κώδικα της αρχικοποίησης με το επιπλέον βήμα ότι ελέγχει ποια γλώσσα είναι επιλεγμένη και διατηρεί την ίδια θέτοντας τα κείμενα βάση αυτής, Επίσης επαναφέρει τα κείμενα των βασικών κουμπιών πίσω σε Επόμενο, Οκ και Άκυρο σε περίπτωση που είχε φτάσει στο τέλος της διάγνωσης πριν την επαναφορά.

Πίνακας 14. Κώδικας Κατηγορήματος Reset

```
reset() :-
    resultsDB::initial(),
    rbuttons:selectAt(1, true),
    initForm("BEGIN", lang:getText()),
    LANG = lang:getText(),
    if LANG = "ENG" then
       cancel_ctl:setText("Cancel"),
       ok_ctl:setText("OK"),
       next_ctl:setText("Next")
    else
       ok_ctl:setText("OK"),
       cancel_ctl:setText("Άκυρο"),
       next_ctl:setText("Επόμενο")
    end if,
    ch1_ctl:setRadioState(radioButton::checked).
```

➢ Κατηγόρημα initForm

Το κατηγόρημα initForm έχει ως είσοδο το αναγνωριστικό της επόμενης ερώτησης και την επιλεγμένη γλώσσα. Ξεκινάει θέτοντας σε κρυφό πεδίο στη φόρμα το αναγνωριστικό της νέας ερώτησης ως την τρέχον ερώτηση, έπειτα κάνει κλήση του κατηγορήματος αλλαγής γλώσσας για αλλαγή κειμένων φόρμας, τέλος διαβάζει από τη βάση ποια κουμπιά επιλογών είναι ενεργά και κάνει τις απαραίτητες αλλαγές απενεργοποιώντας και κρύβοντας τα ανενεργά κουμπιά επιλογών.

Πίνακας 15. Κώδικας Κατηγορήματος Αλλαγής Κειμένων Πεδίων

```
  initForm(ID, LANG) :-
    currentstate:setText(ID),
    langchange(ID, LANG),
    database::getRstate(ID, RS1, RS2, RS3, RS4, RS5),
    ch1_ctl:setEnabled(RS1), %Button State
    ch2_ctl:setEnabled(RS2),
    ch3_ctl:setEnabled(RS3),
    ch4_ctl:setEnabled(RS4),
    ch5_ctl:setEnabled(RS5),
    ch1_ctl:setVisible(RS1), %Button Visibility
    ch2_ctl:setVisible(RS2),
    ch3_ctl:setVisible(RS3),
    ch4_ctl:setVisible(RS4),
    ch5_ctl:setVisible(RS5).
```

➢ Κώδικας Κουμπιού Αλλαγής Γλώσσας

Στο κουμπί αλλαγής διαβάζει πρώτα ποια είναι η τρέχον ερώτηση από τη φόρμα, έπειτα γίνεται έλεγχος αν η επιλεγμένη γλώσσα είναι Αγγλικά ή Ελληνικά.

Αν είναι Αγγλικά κάνει κλήση του κατηγορήματος langchange και ως είσοδο του δίνει την τρέχον ερώτηση και την επιλογή Ελληνικά, μετά αλλάζει το κείμενο του κουμπιού αλλαγής γλώσσας σε GRE, και τέλος αλλάζει το κείμενο στα κουμπιά Επόμενο, Επαναφορά, Τέλος ώστε να είναι στα Ελληνικά, με επιπλέον έλεγχο αν βρίσκεται ήδη στο τέλος της διάγνωσης ή όχι ώστε να δώσει τα αντίστοιχα ονόματα.

Ομοίως αν η γλώσσα είναι Ελληνικά θα ακολουθήσει τα ίδια βήματα αλλά θα αλλάξει τα κείμενα στα Αγγλικά.

Πίνακας 16. Κώδικας Κουμπιού Αλλαγής Γλώσσας

```
onLangswitchClick(_Source) = button::defaultAction :-
    ID = currentstate:getText(),
    if lang:getText() = "ENG" then
      langchange(ID, "GRE"),
      langswitch_ctl:setText("GRE"),
      lang:setText("GRE"),
      reset_ctl:setText("Επαναφορά"),
      cancel_ctl:setText("Άκυρο"),
      if ID = "END" then
        next_ctl:setText("Αποτελέσματα"),
        ok_ctl:setText("Τέλος")
      else
        next_ctl:setText("Επόμενο"),
        ok_ctl:setText("OK")
      end if
    elseif lang:getText() = "GRE" then
      langchange(ID, "ENG"),
      langswitch_ctl:setText("ENG"),
      lang:setText("ENG"),
      reset_ctl:setText("Reset"),
      cancel_ctl:setText("Cancel"),
      if ID = "END" then
        next_ctl:setText("Results"),
        ok_ctl:setText("END")
      else
        next_ctl:setText("Next"),
        ok_ctl:setText("OK")
      end if
    end if.
```

➢ Κατηγόρημα langchange

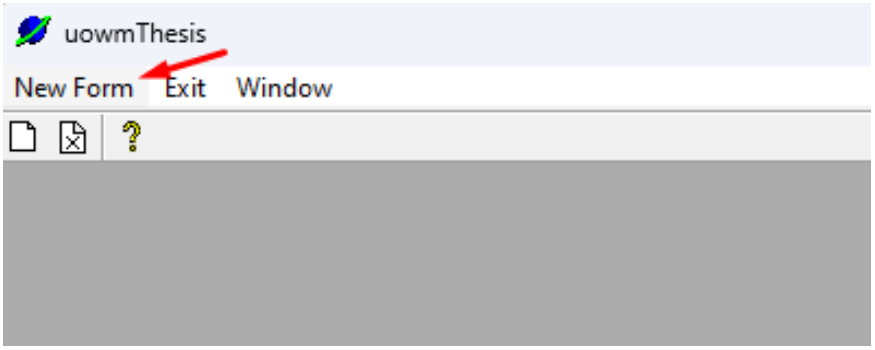
Τέλος το κατηγόρημα αλλαγής γλώσσας, ως είσοδο παίρνει αναγνωριστικό ερώτησης και επιλογή γλώσσας, ξεκινάει διαβάζοντας από τη βάση δεδομένων τα κείμενα χρησιμοποιώντας σαν κριτήριο αναγνωριστικό και γλώσσα έπειτα ενημερώνει το κείμενο της ερώτησης, τα κουμπιά επιλογών

Πίνακας 17. Κώδικας Κατηγορήματος Αλλαγής Γλώσσας

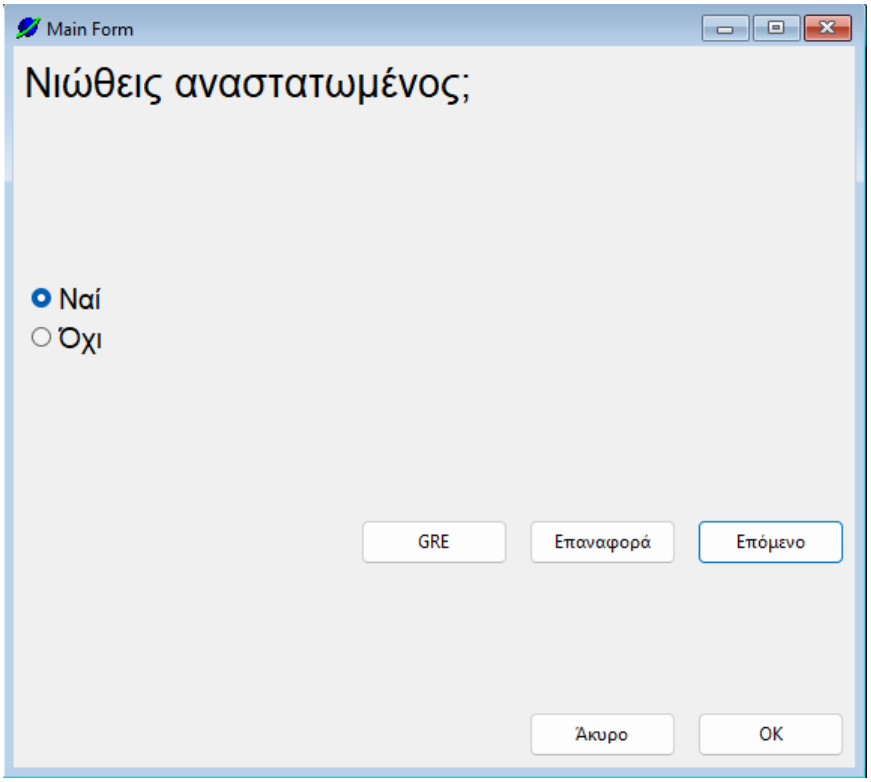
```
langchange(ID, LANG) :-
  %Set Texts
  database::getDesc(ID, LANG, DESC, RD1, RD2, RD3, RD4, RD5),
  database::getRname(ID, LANG, RN1, RN2, RN3, RN4, RN5),
  description_ctl:setText(DESC),
  %Context Texts
  con1:setText(RD1),
  con2:setText(RD2),
  con3:setText(RD3),
  con4:setText(RD4),
  con5:setText(RD5),
  %Button Texts
  ch1_ctl:setText(RN1),
  ch2_ctl:setText(RN2),
  ch3_ctl:setText(RN3),
  ch4_ctl:setText(RN4),
  ch5_ctl:setText(RN5),
  Selected = ictl_ctl:getInteger(),
  if Selected = 1 then
    context_ctl:setText(RD1)
  elseif Selected = 2 then
    context_ctl:setText(RD2)
  elseif Selected = 3 then
    context_ctl:setText(RD3)
  elseif Selected = 4 then
    context_ctl:setText(RD4)
  elseif Selected = 5 then
    context_ctl:setText(RD5)
  end if.
```

Κεφάλαιο 3. Εκτέλεση Εφαρμογής με Παραδείγματα

Για την εκτέλεση νέας διάγνωσης ο Χρήστης πρέπει αρχικά να πατήσει το κουμπί New Form αυτό θα ανοίξει τη φόρμα διάγνωσης σε νέο παράθυρο και έπειτα μπορεί να ξεκινήσει να απαντάει σε ερωτήματα.



Εικόνα 7. Κεντρικό Παράθυρο της Εφαρμογής



Εικόνα 8. Κεντρική Φόρμα Στη Πρώτη Ερώτηση

3.1 Αποτελέσματα και τα Συμπτώματα τους

Για τα παρακάτω παραδείγματα ως παράδειγμα διαγνωστικού θα χρησιμοποιηθούν ερωτήσεις που έχουν να κάνουν με ψυχικές ασθένειες.

Παρακάτω στον Πίνακα 18 Τα πιθανά αποτελέσματα και τα που αντιστοιχούν σε αυτά:

Πίνακας 18. Λίστα Πιθανών Αποτελεσμάτων με Συμπώματα

hypothesis('Psychological-Anxiety') :- symptom("upset"), symptom("persistentfear"), symptom("expecttohappen"), symptom("musclespasms"), symptom("increasedandrenaline"), symptom("fearofunknown"), symptom("hypertension").
hypothesis('Obsessive-Compulsive') :- symptom("dirt"), symptom("shutdoors"), symptom("decisions"), symptom("ideasquestions"), symptom("bathe"), symptom("washhands").
hypothesis('Hysteria') :- symptom("hatred"), symptom("absence"), symptom("memoryloss"), symptom("misalignmentoflimbs").
hypothesis('Depression') :- symptom("hatred"), symptom("foodimbalance"), symptom("losehope"), symptom("inactivityofbody"), symptom("lifepressure").

3.2 Παράδειγμα 1ο

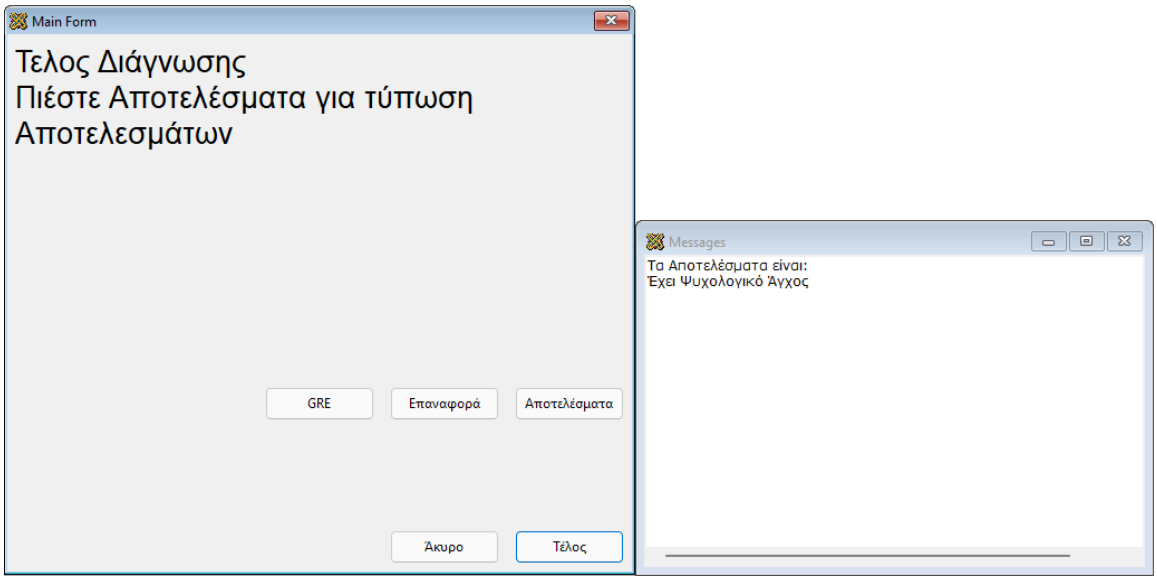
Στο Πρώτο παράδειγμα θα δούμε την περίπτωση όπου ο Χρήστης απαντάει στα ερωτήματα όπως φαίνεται η σειρά στον Πίνακα 19 και η τελική διάγνωση να βγάλει το αποτέλεσμα «Έχει Ψυχολογικό Άγχος».

Πίνακας 19. Συμπώματα για Αποτέλεσμα Phychological-Anxiety(Παράδειγμα 1ο)

Ερώτηση	Απάντηση
Νιώθεις αναστατωμένος;	Ναι
Έχεις επίμονο φόβο;	Ναι
Ανησυχείς ότι θα συμβεί κάτι;	Ναι
Έχεις μυϊκούς σπασμούς;	Ναι
Έχεις αυξημένη έκκριση αδρεναλίνης;	Ναι
Φοβάσαι για το άγνωστο;	Ναι
Έχεις υπέρταση;	Ναι
Φοβάσαι τη συσσώρευση βρωμιάς;	Όχι
Έχεις το αίσθημα του μίσους;	Όχι

Εφόσον οι απαντήσεις στα ερωτήματα «Φοβάσαι τη συσσώρευση βρωμιάς;» και «Έχεις το αίσθημα του μίσους;» ήτανε όχι παρατηρούμε ότι έγινε αποκοπή των υπόλοιπων ερωτημάτων, επιπλέον επειδή το αποτέλεσμα “Depression” έχει κοινό σύμπτωμα το «Αίσθημα του Μίσους;» αποκόβονται και οι ερωτήσεις αυτού.

Στην παρακάτω εικόνα βλέπουμε το αποτέλεσμα από την παραπάνω σειρά απαντήσεων με αποτέλεσμα «Έχει Ψυχολογικό Άγχος».



Εικόνα 9. Στιγμιότυπο Αποτέλεσματος 1ου Παραδείγματος

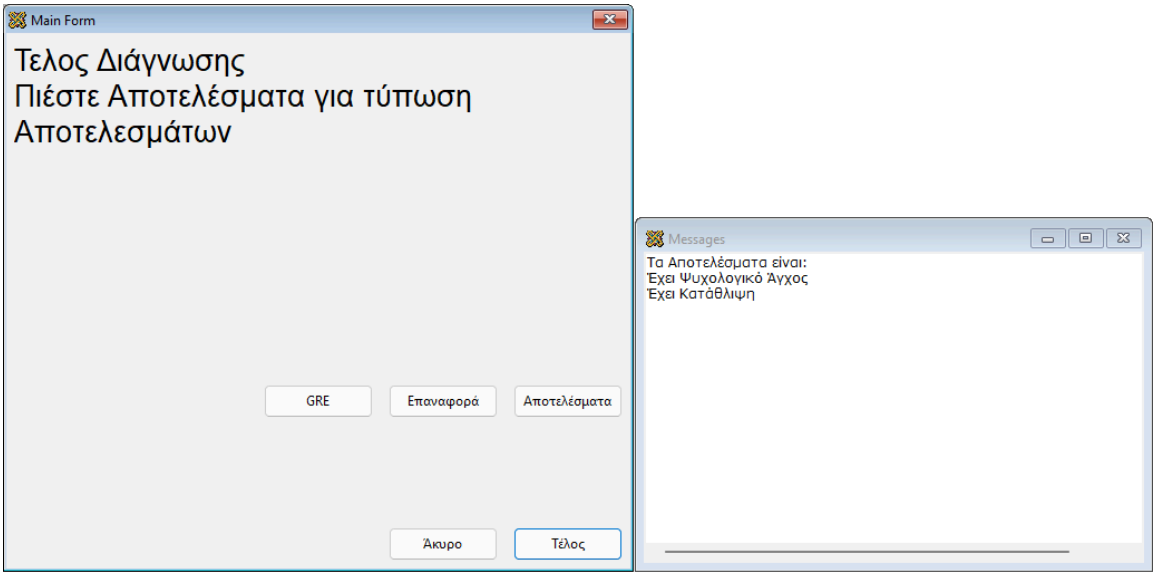
3.3 Παράδειγμα 2ο

Στο Δεύτερο παράδειγμα ο Χρήστης απαντάει της ερωτήσεις με βάσει τη σειρά στον Πίνακα 20 έτσι θα ισχύουν 2 αποτελέσματα, το «Έχει Ψυχολογικό Άγχος» και το «Έχει Κατάθλιψη».

Πίνακας 20. Συμπώματα για Αποτέλεσματα Phychological-Anxiety και Depression(Παράδειγμα 2^{ο})

Ερώτηση	Απάντηση
Νιώθεις αναστατωμένος;	Ναι
Έχεις επίμονο φόβο;	Ναι
Ανησυχείς ότι θα συμβεί κάτι;	Ναι
Έχεις μυϊκούς σπασμούς;	Ναι
Έχεις αυξημένη έκκριση αδρεναλίνης;	Ναι
Φοβάσαι για το άγνωστο;	Ναι
Έχεις υπέρταση;	Ναι
Φοβάσαι τη συσσώρευση βρωμιάς;	Όχι
Έχεις το αίσθημα του μίσους;	Ναι
Έχεις απουσία συνείδησης;	Όχι
Έχεις διατροφική ανισορροπία;	Ναι
Χάνεις την ελπίδα σου;	Ναι
Νιώθεις αδράνεια του σώματος;	Ναι
Έχεις πιέσεις στη ζωή;	Ναι

Ακόμη και όταν η εφαρμογή έχει βρει περισσότερα από 1 αποτελέσματα τυπώνει όλα τα αποτελέσματα που έχουν βρεθεί.



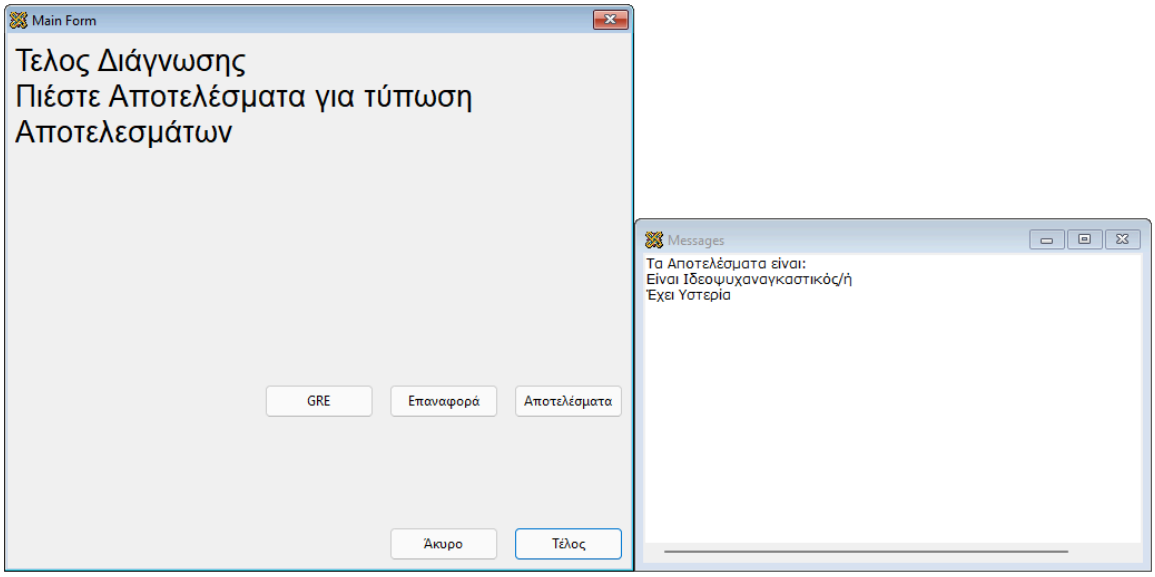
Εικόνα 10. Στιγμιότυπο Αποτελέσματος 2^{ου} Παραδείγματος

3.4 Παράδειγμα 3ο

Στο Τρίτο παράδειγμα ο Χρήστης απαντάει στις ερωτήσεις με βάση τον πίνακα 21 ώστε τα αποτελέσματα να είναι τα «Είναι Ιδεοψυχαναγκαστικός/ή» και «Έχει Υστερία» σε αντίθεση με το 2^{ο} παράδειγμα θα απαντήσει με Ναι σε μερικές ερωτήσεις και από τα άλλα αποτελέσματα προτού απαντήσει Όχι και γίνει αποκοπή ερωτημάτων.

Πίνακας 21. Συμπώματα για Αποτέλεσματα Obsessive-Compulsive και Hysteria (Παράδειγμα 3^{ο})

Ερώτηση	Απάντηση
Νιώθεις αναστατωμένος;	Ναι
Έχεις επίμονο φόβο;	Ναι
Ανησυχείς ότι θα συμβεί κάτι;	Ναι
Έχεις μυϊκούς σπασμούς;	Όχι
Φοβάσαι τη συσσώρευση βρωμιάς;	Ναι
Έχεις την ανάγκη να κλείνεις τις πόρτες συνεχώς;	Ναι
Έχεις την ανάγκη να παίρνεις συχνά αποφάσεις;	Ναι
Έχεις ιδέες και ερωτήσεις;	Ναι
Έχεις την ανάγκη να κάνεις μπάνιο δέκα φορές;	Ναι
Έχεις την ανάγκη να πλένεις επανειλημμένα τα χέρια σου;	Ναι
Έχεις το αίσθημα του μίσους;	Ναι
Έχεις απουσία συνείδησης;	Ναι
Έχεις προσωρινή απώλεια μνήμης;	Ναι
Έχεις κακή ευθυγράμμιση των άκρων;	Ναι
Έχεις διατροφική ανισορροπία;	Ναι
Χάνεις την ελπίδα σου;	Ναι
Νιώθεις αδράνεια του σώματος;	Όχι



Εικόνα 11. Στιγμιότυπο Αποτελέσματος 3ου Παραδείγματος

Συμπεράσματα

Στις μέρες μας οι διαγνωστικές εφαρμογές είναι πολύ χρήσιμες καθώς βοηθάνε στη προσέγγιση ή εύρεση προβλημάτων. Ο Κύριος σκοπός της εργασίας ήταν η δημιουργία μιας διαγνωστικής εφαρμογής υλοποιημένη σε περιβάλλον Visual Prolog.

Η εφαρμογή λειτουργεί κάνοντας χρήση παραθυρικού περιβάλλοντος καθιστώντας τη χρήση της εύκολη και χωρίς επιπλέον βήματα για τον χρήστη, επίσης η δυνατότητα παροχής πάνω από δυο απαντήσεις ανά ερώτηση θα μπορούσε να καλύψει ερωτήσεις με ποιο ευρύτερο τύπου απαντήσεις, το αρνητικό με το εξής είναι ότι απαιτεί περισσότερη μελέτη από τον προγραμματιστή κατά τη προετοιμασία των ερωτήσεων.

Μερικές από τις δυσκολίες που συναντήθηκαν κατά την εκπόνηση της εργασίας ήταν η έλλειψη διαθεσιμότητας υλικού πάνω στη γλώσσα Visual Prolog με μερικές από τις ποιο έμπιστες πηγές να είναι τα ηλεκτρονικά βιβλία που μπορεί να βρει κάποιος μέσω της ιστοσελίδας της Visual Prolog.

Τέλος ως μελλοντική επέκταση της εφαρμογής προτείνεται η προσθήκη δυνατότητας φόρτωσης αποτελεσμάτων και συμπτωμάτων από αρχείο ώστε ο χρήστης να έχει δυνατότητα επιλογής τι διάγνωση θέλει να κάνει.

Βιβλιογραφία

1] Wikipedia, «Diagnostic program,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Diagnostic_program.

2] SoftwareGeekBytes Team, «Object Oriented Programming,» 5 1 2023. [Ηλεκτρονικό]. Available: https://softwaregeekbytes.com/object-oriented-programming-simple-words.

3] A. Kay, «Meaning of Object-Oriented Programming,» 2003. [Ηλεκτρονικό]. Available: https://www.purl.org/stefan_ram/pub/doc_kay_oop_en.

4] Oracle, «What is a Database,» [Ηλεκτρονικό]. Available: https://www.oracle.com/database/what-is-database/.

5] B. Lutkevich και A. Hughes, «techtarget,» [Ηλεκτρονικό]. Available: https://www.techtarget.com/searchdatamanagement/definition/database.

6] Visual Prolog, «Fundamental Visual Prolog,» [Ηλεκτρονικό]. Available: https://wiki.visual-prolog.com/index.php?title=Fundamental_Visual_Prolog.

Παράρτημα Κώδικα

➢ Αρχείο mainForm.pro

```
predicates
   onDescriptionValidate : control::validateResponder.
   onResetClick : button::clickResponder.
   onOkClick : button::clickResponder.
   onNextClick : button::clickResponder.
   onCh5StateChanged : radioButton::stateChangedListener.
   onCh4StateChanged : radioButton::stateChangedListener.
   onCh3StateChanged : radioButton::stateChangedListener.
   onCh2StateChanged : radioButton::stateChangedListener.
   onCh1StateChanged : radioButton::stateChangedListener.
   go : (integer Calc).
   initForm : (string ID, string LANG).
   langchange : (string ID, string LANG).
   handleListBoxIndex : (listBox Changed, positive Index [out]).
   getListBoxIndex : (listBox Changed, integercontrol).
   onLangswitchClick : button::clickResponder.
   reset : ().

clauses
   display(Parent) = Form :-
      Form = new(Parent),
      Form:show().

/* On Window Creation Do the Following */
   new(Parent) :-
      resultsDB::initial(),
      database::consultDatabase(),
      formWindow::new(Parent),
      generatedInitialize(),
      initForm("BEGIN", lang:getText()),
      rbuttons:addList(["NoSel", "c1", "c2", "c3", "c4", "c5"]),
      rbuttons:selectAt(1, true),
      ch1_ctl:setRadioState(radioButton::checked).

/* Reset Button */
   reset() :-
      resultsDB::initial(),
      rbuttons:selectAt(1, true),
      initForm("BEGIN", lang:getText()),
      LANG = lang:getText(),
      if LANG = "ENG" then
         cancel_ctl:setText("Cancel"),
```

```
        ok_ctl:setText("OK"),
        next_ctl:setText("Next")
    else
        ok_ctl:setText("OK"),
        cancel_ctl:setText("Άκυρο"),
        next_ctl:setText("Επόμενο")
    end if,
    ch1_ctl:setRadioState(radioButton::checked).

/* Initialize Form */
  initForm(ID, LANG) :-
    currentstate:setText(ID),
    langchange(ID, LANG),
    database::getRstate(ID, RS1, RS2, RS3, RS4, RS5),
    ch1_ctl:setEnabled(RS1), %Button State
    ch2_ctl:setEnabled(RS2),
    ch3_ctl:setEnabled(RS3),
    ch4_ctl:setEnabled(RS4),
    ch5_ctl:setEnabled(RS5),
    ch1_ctl:setVisible(RS1), %Button Visibility
    ch2_ctl:setVisible(RS2),
    ch3_ctl:setVisible(RS3),
    ch4_ctl:setVisible(RS4),
    ch5_ctl:setVisible(RS5).

/* Language Change */
  langchange(ID, LANG) :-
    %Set Texts
    database::getDesc(ID, LANG, DESC, RD1, RD2, RD3, RD4, RD5),
    database::getRname(ID, LANG, RN1, RN2, RN3, RN4, RN5),
    description_ctl:setText(DESC),
    %Context Texts
    con1:setText(RD1),
    con2:setText(RD2),
    con3:setText(RD3),
    con4:setText(RD4),
    con5:setText(RD5),
    %Button Texts
    ch1_ctl:setText(RN1),
    ch2_ctl:setText(RN2),
    ch3_ctl:setText(RN3),
    ch4_ctl:setText(RN4),
    ch5_ctl:setText(RN5),
    Selected = ictl_ctl:getInteger(),
    if Selected = 1 then
        context_ctl:setText(RD1)
    elseif Selected = 2 then
        context_ctl:setText(RD2)
    elseif Selected = 3 then
```

```
        context_ctl:setText(RD3)
    elseif Selected = 4 then
        context_ctl:setText(RD4)
    elseif Selected = 5 then
        context_ctl:setText(RD5)
    end if.

/* Next Button */
  onNextClick(_Source) = button::defaultAction :-
    getListBoxIndex(rbuttons, ictl_ctl),
    Next = ictl_ctl:getInteger(),
    go(Next).

/* Go */
  go(Calc) :-
    %Assert Symptom
    next_ctl:setEnabled(false), %DISABLE NEXT BUTTON
    ID = currentstate:getText(),
    if not(ID = "END") then
      database::getNstate(ID, Nx1, Nx2, Nx3, Nx4, Nx5),
      if Calc = 1 then
        database::getID(NX1, Symptom, _),
        resultsDB::assertsymptom(Symptom)
      elseif Calc = 2 then
        database::getID(NX2, Symptom, _),
        resultsDB::assertsymptom(Symptom)
      elseif Calc = 3 then
        database::getID(NX3, Symptom, _),
        resultsDB::assertsymptom(Symptom)
      elseif Calc = 4 then
        database::getID(NX4, Symptom, _),
        resultsDB::assertsymptom(Symptom)
      elseif Calc = 5 then
        database::getID(NX5, Symptom, _),
        resultsDB::assertsymptom(Symptom)
      end if
    end if,
    fail.
  go(Calc) :-
    %Read data needed
    ID = currentstate:getText(),
    LANG = lang:getText(),
    %If at end of diagnosis check for and print results
    if ID = "END" then
      %Check if at least one result exists
      resultsDB::estimateUser(Result),
      if not(Result = "") then
        resultsDB::go(LANG),
        stdio::write("\n")
```

```
        elseif LANG = "ENG" then
            stdio::write("No Results Were Found \n")
        elseif LANG = "GRE" then
            stdio::write("Δεν Βρέθηκαν Αποτελέσματα \n")
        end if
    end if,
    %if not at end of diagnosis set new texts for next question
    if not(ID = "END") then
        database::getNstate(ID, Nx1, Nx2, Nx3, Nx4, Nx5),
        % Sets New Texts Based on Selected Button
        if Calc = 1 then
            database::getID(NX1, _, ST),
            currentstate:setText(ST),
            initForm(ST, LANG)
        elseif Calc = 2 then
            database::getID(NX2, _, ST),
            currentstate:setText(ST),
            initForm(ST, LANG)
        elseif Calc = 3 then
            database::getID(NX3, _, ST),
            currentstate:setText(ST),
            initForm(ST, LANG)
        elseif Calc = 4 then
            database::getID(NX4, _, ST),
            currentstate:setText(ST),
            initForm(ST, LANG)
        elseif Calc = 5 then
            database::getID(NX5, _, ST),
            currentstate:setText(ST),
            initForm(ST, LANG)
        end if
    end if,
    fail.
go(_Calc) :-
    %Change main buttons text if at end of diagnosis
    if currentstate:getText() = "END" then
        next_ctl:setEnabled(true),
        LANG = lang:getText(),
        if LANG = "ENG" then
            cancel_ctl:setText("Cancel"),
            ok_ctl:setText("END"),
            next_ctl:setText("Results")
        else
            ok_ctl:setText("Τέλος"),
            cancel_ctl:setText("Άκυρο"),
            next_ctl:setText("Αποτελέσματα")
        end if
    end if,
    next_ctl:setEnabled(true).
```

```
/* Get ListBox Values */
   %Returns Index Value without getting determ error, no idea why.
   handleListBoxIndex(Changed, Index) :-
      Index = Changed:tryGetSelectedIndex(),
      !.
   handleListBoxIndex(_, 0).

/* Returns Integer Value to Integer Control */
   getListBoxIndex(Changed, Ictl) :-
      handleListBoxIndex(Changed, Index),
      Index1 = Index,
      Ictl:setInteger(Index1).

/* Radio Button on Select Actions */
   onCh5StateChanged(_Source, _OldState, _NewState) :-
      rbuttons:selectAt(5, true),
      context_ctl:setText(con5:getText()).
   onCh4StateChanged(_Source, _OldState, _NewState) :-
      rbuttons:selectAt(4, true),
      context_ctl:setText(con4:getText()).
   onCh3StateChanged(_Source, _OldState, _NewState) :-
      rbuttons:selectAt(3, true),
      context_ctl:setText(con3:getText()).
   onCh2StateChanged(_Source, _OldState, _NewState) :-
      rbuttons:selectAt(2, true),
      context_ctl:setText(con2:getText()).
   onCh1StateChanged(_Source, _OldState, _NewState) :-
      rbuttons:selectAt(1, true),
      context_ctl:setText(con1:getText()).

/* Ok Button */
   onOkClick(_Source) = button::defaultAction :-
      go(1).
      %description::listDatabase().

/* Language Switch Button */
   onLangswitchClick(_Source) = button::defaultAction :-
      ID = currentstate:getText(),
      if lang:getText() = "ENG" then
         langchange(ID, "GRE"),
         langswitch_ctl:setText("GRE"),
         lang:setText("GRE"),
         reset_ctl:setText("Επαναφορά"),
         cancel_ctl:setText("Άκυρο"),
         if ID = "END" then
            next_ctl:setText("Αποτελέσματα"),
            ok_ctl:setText("Τέλος")
         else
```

```
            next_ctl:setText("Επόμενο"),
            ok_ctl:setText("OK")
        end if
    elseif lang:getText() = "GRE" then
        langchange(ID, "ENG"),
        langswitch_ctl:setText("ENG"),
        lang:setText("ENG"),
        reset_ctl:setText("Reset"),
        cancel_ctl:setText("Cancel"),
        if ID = "END" then
            next_ctl:setText("Results"),
            ok_ctl:setText("END")
        else
            next_ctl:setText("Next"),
            ok_ctl:setText("OK")
        end if
    end if.

  onDescriptionValidate(_Source) = control::contentsOk.

/* Reset Button */
  onResetClick(_Source) = button::defaultAction :-
    reset().
```

➢ Αρχείο idDatabase.cl

```
class database

predicates
  getDesc : (string ID, string Lang, string Description [out], string Ds1 [out], string Ds2 [out], string Ds3
[out], string Ds4 [out],
    string Ds5 [out]).
  getRname : (string ID, string Lang, string Rd1 [out], string Rd2 [out], string Rd3 [out], string Rd4 [out],
string Rd5 [out]).
  getRstate : (string ID, boolean SRd1 [out], boolean SRd2 [out], boolean SRd3 [out], boolean SRd4 [out],
boolean SRd5 [out]).
  getNstate : (string ID, string Nx1 [out], string Nx2 [out], string Nx3 [out], string Nx4 [out], string Nx5
[out]).
  getResult : (string ID, string LANG, string RESULT [out]).
  consultDatabase : ().
  getID : (string ID, string SYMPTOM [out], string NEXTID [out]).
  getData : ().

end class database
```

➢ Αρχείο idDatabase.pro

```
implement database
   /* Used to read data from file */
   open file

class facts - descriptionDB
   /* Create Tables */
   description : (string ID, string Lang, string Description, string Drd1, string Drd2, string Drd3, string Drd4,
string Drd5).
   radioname : (string ID, string Lang, string Rd1, string Rd2, string Rd3, string Rd4, string Rd5).
   nextstate : (string ID, string Nx1, string Nx2, string Nx3, string Nx4, string Nx5).
   radiostate : (string ID, boolean Rd1, boolean Rd2, boolean Rd3, boolean Rd4, boolean Rd5).
   symptomID : (string ID, string SYMPTOM, string NEXTID).
   endresults : (string ID, string LANG, string RESULT).

clauses
   /* Reads and Returns Data From Tables */
   getDesc(ID, Lang, Description, Ds1, Ds2, Ds3, Ds4, Ds5) :-
      description(ID, Lang, Description, Ds1, Ds2, Ds3, Ds4, Ds5),
      !.
   getDesc(_ID, _Lang, " ", " ", " ", " ", " ", " ", " ").

   getRname(ID, Lang, Rd1, Rd2, Rd3, Rd4, Rd5) :-
      radioname(ID, Lang, Rd1, Rd2, Rd3, Rd4, Rd5),
      !.
   getRname(_ID, _Lang, " ", " ", " ", " ", " ", " ").

   getRstate(ID, SRd1, SRd2, SRd3, SRd4, SRd5) :-
      radiostate(ID, SRd1, SRd2, SRd3, SRd4, SRd5),
      !.
   getRstate(_ID, false, false, false, false, false).

   getNstate(ID, Nx1, Nx2, Nx3, Nx4, Nx5) :-
      nextstate(ID, Nx1, Nx2, Nx3, Nx4, Nx5),
      !.
   getNstate(_ID, "END", "END", "END", "END", "END").

   getID(ID, SYMPTOM, NEXTID) :-
      symptomID(ID, SYMPTOM, NEXTID),
      !.
   getID(_ID, "END", "END").

   getResult(ID, LANG, RESULT) :-
      endresults(ID, LANG, RESULT),
      !.
   getResult(_ID, _LANG, "").

   /* Load data from File */
```

```
   consultDatabase() :-
      file::existExactFile("idDatabase.txt"),
      !,
      getData().
   consultDatabase() :-
      vpiCommonDialogs::error("WARNING", "File idDatabase.txt not found.\nDatabase is Empty").
%Ensure file idDatabase.txt exists in the executable directory

   /* Delete all existing entries from database then Fill with new ones from File */
   getData() :-
      retractFactDB(descriptionDB),
      file::consult("idDatabase.txt", descriptionDB).

end implement database
```

➢ Αρχείο results.cl

```
class resultsDB

predicates
   initial : ().
   go : (string Lang).
   assertsymptom : (string Symptom).
   estimateUser : (string ResultID [out]).

end class resultsDB
```

➢ Αρχείο results.pro

```
implement resultsDB

domains
   review = symbol.

class predicates
   hypothesis : (review Performance [out]) nondeterm.
   gradeUser : (string LANG) nondeterm.

class facts
   symptom : (string) nondeterm.

clauses
   initial() :-
      retractAll(symptom(_)).
/* Go */
   go(LANG) :-
      if LANG = "ENG" then
```

```
        stdio::write("The Results Are: \n"),
        gradeUser(LANG)
    elseif LANG = "GRE" then
        stdio::write("Τα Αποτελέσματα είναι: \n"),
        gradeUser(LANG)
    end if,
    fail.
go(_LANG).

/* Asserts given Symptom */
  assertsymptom(Symptom) :-
    if not(Symptom = "") and not(symptom(Symptom)) then
      assert(symptom(Symptom))
    end if.

/* Returns any Hypothesis that match a combination of asserted Symptoms */
  hypothesis('Psychological-Anxiety') :-
    symptom("upset"),
    symptom("persistentfear"),
    symptom("expecttohappen"),
    symptom("musclespasms"),
    symptom("increasedandrenaline"),
    symptom("fearofunknown"),
    symptom("hypertension").

  hypothesis('Obsessive-Compulsive') :-
    symptom("dirt"),
    symptom("shutdoors"),
    symptom("decisions"),
    symptom("ideasquestions"),
    symptom("bathe"),
    symptom("washhands").

  hypothesis('Hysteria') :-
    symptom("hatred"),
    symptom("absence"),
    symptom("memoryloss"),
    symptom("misalignmentoflimbs").

  hypothesis('Depression') :-
    symptom("hatred"),
    symptom("foodimbalance"),
    symptom("losehope"),
    symptom("inactivityofbody"),
    symptom("lifepressure").

/* For Every Hypothesis  */
  gradeUser(LANG) :-
    hypothesis(ResultID),
```

```
    database::getResult(ResultID, LANG, Review),
    stdIo::write(Review, "\n").

/* Find First Valid Hypothesis */
  estimateUser(ResultID) :-
    hypothesis(ResultID),
    !.
  estimateUser("").

end implement resultsDB
```

➢ Αρχείο idDatabase.txt

```
clauses
description("BEGIN","ENG","Are you feeling upset?","","","","","")
description("BEGIN","GRE","Νιώθεις αναστατωμένος;","","","","","")
description("END","ENG","End of Diagnosis\nPress Results to get your Results.","","","","","")
description("END","GRE","Τελος Διάγνωσης\nΠιέστε Αποτελέσματα για τύπωση
Αποτελεσμάτων","","","","","")
description("DIRT","ENG","Do you have fear of accumulation of dirt?","","","","","")
description("DIRT","GRE","Φοβάσαι τη συσσώρευση βρωμιάς;","","","","","")
description("HATRED","ENG","Do you have the feeling of hatred?","","","","","")
description("HATRED","GRE","Έχεις το αίσθημα του μίσους;","","","","","")
description("PERSISTENTFEAR","ENG","Do you have persistent fear?","","","","","")
description("PERSISTENTFEAR","GRE","Έχεις επίμονο φόβο;","","","","","")
description("EXPECTTOHAPPEN","ENG","Do you expect something to happen?","","","","","")
description("EXPECTTOHAPPEN","GRE","Ανησυχείς ότι θα συμβεί κάτι;","","","","","")
description("MUSCLESPASMS","ENG","Do you have muscle spasms?","","","","","")
description("MUSCLESPASMS","GRE","Έχεις μυϊκούς σπασμούς;","","","","","")
description("ANDRENALINE","ENG","Do you have increased adrenaline secretion?","","","","","")
description("ANDRENALINE","GRE","Έχεις αυξημένη έκκριση αδρεναλίνης;","","","","","")
description("FEAROFUNKNOWN","ENG","Do you have fear of the unknown?","","","","","")
description("FEAROFUNKNOWN","GRE","Φοβάσαι για το άγνωστο;","","","","","")
description("HYPERTENSION","ENG","Do you have hypertension?","","","","","")
description("HYPERTENSION","GRE","Έχεις υπέρταση;","","","","","")
description("SHUTDOORS","ENG","Do you have the need to shut the doors continually?","","","","","")
description("SHUTDOORS","GRE","Έχεις την ανάγκη να κλείνεις τις πόρτες συνεχώς;","","","","","")
description("DECISIONS","ENG","Do you have the need to frequently make decisions?","","","","","")
description("DECISIONS","GRE","Έχεις την ανάγκη να παίρνεις συχνά αποφάσεις;","","","","","")
description("IDEASQUESTIONS","ENG","Do you have ideas and questions?","","","","","")
description("IDEASQUESTIONS","GRE","Έχεις ιδέες και ερωτήσεις;","","","","","")
description("BATHE","ENG","Do you have the need of bathing ten times?","","","","","")
description("BATHE","GRE","Έχεις την ανάγκη να κάνεις μπάνιο δέκα φορές;","","","","","")
description("WASHHANDS","ENG","Do you have the need of repeatedly your washing
hands?","","","","","")
description("WASHHANDS","GRE","Έχεις την ανάγκη να πλένεις επανειλημμένα τα χέρια
σου;","","","","","")
description("CONSCABSENCE","ENG","Do you have absence of consciousness?","","","","","")
```

```
description("CONSCABSENCE","GRE","Έχεις απουσία συνείδησης;","","","","","")
description("MEMORYLOSS","ENG","Do you have temporary loss of memory?","","","","","","")
description("MEMORYLOSS","GRE","Έχεις προσωρινή απώλεια μνήμης;","","","","","","")
description("LIMBS","ENG","Do you have misalignment of limbs?","","","","","","")
description("LIMBS","GRE","Έχεις κακή ευθυγράμμιση των άκρων;","","","","","","")
description("FOODIMBALANCE","ENG","Do you have food imbalance?","","","","","","")
description("FOODIMBALANCE","GRE","Έχεις διατροφική ανισορροπία;","","","","","","")
description("LOSEHOPE","ENG","Do you lose hope?","","","","","","")
description("LOSEHOPE","GRE","Χάνεις την ελπίδα σου;","","","","","","")
description("INACTIVITY","ENG","Do you feel inactivity of body?","","","","","","")
description("INACTIVITY","GRE","Νιώθεις αδράνεια του σώματος;","","","","","","")
description("LIFEPRESSURE","ENG","Do you have life pressures?","","","","","","")
description("LIFEPRESSURE","GRE","Έχεις πιέσεις στη ζωή;","","","","","","")


%IDLANGUAGEBUTTON1BUTTON2BUTTON3BUTTON4BUTTON5
radioname("BEGIN","ENG","Yes","No","","","","")
radioname("BEGIN","GRE","Ναί","Όχι","","","","")
radioname("PERSISTENTFEAR","ENG","Yes","No","","","","")
radioname("PERSISTENTFEAR","GRE","Ναί","Όχι","","","","")
radioname("EXPECTTOHAPPEN","ENG","Yes","No","","","","")
radioname("EXPECTTOHAPPEN","GRE","Ναί","Όχι","","","","")
radioname("MUSCLESPASMS","ENG","Yes","No","","","","")
radioname("MUSCLESPASMS","GRE","Ναί","Όχι","","","","")
radioname("ANDRENALINE","ENG","Yes","No","","","","")
radioname("ANDRENALINE","GRE","Ναί","Όχι","","","","")
radioname("FEAROFUNKNOWN","ENG","Yes","No","","","","")
radioname("FEAROFUNKNOWN","GRE","Ναί","Όχι","","","","")
radioname("HYPERTENSION","ENG","Yes","No","","","","")
radioname("HYPERTENSION","GRE","Ναί","Όχι","","","","")
radioname("SHUTDOORS","ENG","Yes","No","","","","")
radioname("SHUTDOORS","GRE","Ναί","Όχι","","","","")
radioname("DECISIONS","ENG","Yes","No","","","","")
radioname("DECISIONS","GRE","Ναί","Όχι","","","","")
radioname("IDEASQUESTIONS","ENG","Yes","No","","","","")
radioname("IDEASQUESTIONS","GRE","Ναί","Όχι","","","","")
radioname("BATHE","ENG","Yes","No","","","","")
radioname("BATHE","GRE","Ναί","Όχι","","","","")
radioname("WASHHANDS","ENG","Yes","No","","","","")
radioname("WASHHANDS","GRE","Ναί","Όχι","","","","")
radioname("HATRED","ENG","Yes","No","","","","")
radioname("HATRED","GRE","Ναί","Όχι","","","","")
radioname("CONSCABSENCE","ENG","Yes","No","","","","")
radioname("CONSCABSENCE","GRE","Ναί","Όχι","","","","")
radioname("FOODIMBALANCE","ENG","Yes","No","","","","")
radioname("FOODIMBALANCE","GRE","Ναί","Όχι","","","","")
radioname("DIRT","ENG","Yes","No","","","","")
radioname("DIRT","GRE","Ναί","Όχι","","","","")
radioname("MEMORYLOSS","ENG","Yes","No","","","","")
```

```
radioname("MEMORYLOSS","GRE","Ναί","Όχι","","","")
radioname("LIMBS","ENG","Yes","No","","","")
radioname("LIMBS","GRE","Ναί","Όχι","","","")
radioname("LOSEHOPE","ENG","Yes","No","","","")
radioname("LOSEHOPE","GRE","Ναί","Όχι","","","")
radioname("INACTIVITY","ENG","Yes","No","","","")
radioname("INACTIVITY","GRE","Ναί","Όχι","","","")
radioname("LIFEPRESSURE","ENG","Yes","No","","","")
radioname("LIFEPRESSURE","GRE","Ναί","Όχι","","","")

radiostate("BEGIN",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("END",FALSE,FALSE,FALSE,FALSE,FALSE)
radiostate("PERSISTENTFEAR",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("EXPECTTOHAPPEN",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("MUSCLESPASMS",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("ANDRENALINE",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("FEAROFUNKNOWN",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("HYPERTENSION",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("HATRED",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("SHUTDOORS",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("DECISIONS",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("IDEASQUESTIONS",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("BATHE",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("WASHHANDS",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("DIRT",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("CONSCABSENCE",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("FOODIMBALANCE",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("MEMORYLOSS",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("LIMBS",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("LOSEHOPE",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("INACTIVITY",TRUE,TRUE,FALSE,FALSE,FALSE)
radiostate("LIFEPRESSURE",TRUE,TRUE,FALSE,FALSE,FALSE)

nextstate("BEGIN","Upset","SKIPTODIRT","END","END","END")
nextstate("PERSISTENTFEAR","Persistent Fear","SKIPTODIRT","END","END","END")
nextstate("DIRT","Dirt","SKIPTOHATRED","END","END","END")
nextstate("HATRED","Hatred","END","END","END","END")
nextstate("EXPECTTOHAPPEN","Expect Something To Happen","SKIPTODIRT","END","END","END")
nextstate("MUSCLESPASMS","Muscle Spasms","SKIPTODIRT","END","END","END")
nextstate("ANDRENALINE","Increased Andrenaline","SKIPTODIRT","END","END","END")
nextstate("FEAROFUNKNOWN","Fear Of The Unknown","SKIPTODIRT","END","END","END")
nextstate("HYPERTENSION","Hypertension","SKIPTODIRT","END","END","END")
nextstate("SHUTDOORS","Shutdoors","SKIPTOHATRED","END","END","END")
nextstate("DECISIONS","Decisions","SKIPTOHATRED","END","END","END")
nextstate("IDEASQUESTIONS","IdeasQuestions","SKIPTOHATRED","END","END","END")
nextstate("BATHE","Bathe","SKIPTOHATRED","END","END","END")
nextstate("WASHHANDS","Wash Hands","SKIPTOHATRED","END","END","END")
nextstate("CONSCABSENCE","Abscence Of
Consciousness","SKIPTOFOODIMBALANCE","END","END","END")
```

```
nextstate("FOODIMBALANCE","Food Imbalance","END","END","END","END")
nextstate("MEMORYLOSS","Memory Loss","SKIPTOFOODIMBALANCE","END","END","END")
nextstate("LIMBS","Misalignment of Limbs","SKIPTOFOODIMBALANCE","END","END","END")
nextstate("LOSEHOPE","Lose Hope","END","END","END","END")
nextstate("INACTIVITY","Inactivity of Body","END","END","END","END")
nextstate("LIFEPRESSURE","Life Pressure","END","END","END","END")

symptomID("NULL","","END")
symptomID("","","END")
symptomID("END","","END")
symptomID("SKIPTODIRT","","DIRT")
symptomID("SKIPTOHATRED","","HATRED")
symptomID("SKIPTOFOODIMBALANCE","","FOODIMBALANCE")
symptomID("Upset","upset","PERSISTENTFEAR")
symptomID("Dirt","dirt","SHUTDOORS")
symptomID("Hatred","hatred","CONSCABSENCE")
symptomID("Persistent Fear","persistentfear","EXPECTTOHAPPEN")
symptomID("Expect Something To Happen","expecttohappen","MUSCLESPASMS")
symptomID("Muscle Spasms","musclespasms","ANDRENALINE")
symptomID("Increased Andrenaline","increasedandrenaline","FEAROFUNKNOWN")
symptomID("Fear Of The Unknown","fearofunknown","HYPERTENSION")
symptomID("Hypertension","hypertension","DIRT")
symptomID("Shutdoors","shutdoors","DECISIONS")
symptomID("Decisions","decisions","IDEASQUESTIONS")
symptomID("IdeasQuestions","ideasquestions","BATHE")
symptomID("Bathe","bathe","WASHHANDS")
symptomID("Wash Hands","washhands","HATRED")
symptomID("Abscence Of Consciousness","absence","MEMORYLOSS")
symptomID("Food Imbalance","foodimbalance","LOSEHOPE")
symptomID("Memory Loss","memoryloss","LIMBS")
symptomID("Misalignment of Limbs","misalignmentoflimbs","FOODIMBALANCE")
symptomID("Lose Hope","losehope","INACTIVITY")
symptomID("Inactivity of Body","inactivityofbody","LIFEPRESSURE")
symptomID("Life Pressure","lifepressure","END")

endresults("Psychological-Anxiety","ENG","Has Phychological Anxiety")
endresults("Psychological-Anxiety","GRE","Έχει Ψυχολογικό Άγχος")
endresults("Obsessive-Compulsive","ENG","Is Obsessive Compulsive")
endresults("Obsessive-Compulsive","GRE","Είναι Ιδεοψυχαναγκαστικός/ή")
endresults("Hysteria","ENG","Has Hysteria")
endresults("Hysteria","GRE","Έχει Υστερία")
endresults("Depression","ENG","Has Depression")
endresults("Depression","GRE","Έχει Κατάθλιψη")
```