

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Πτυχιακή Εργασία

Νούτση Ουρανία

**Web-based εφαρμογή εξαγωγής στατιστικών δεδομένων από την Eurostat,
στατιστικής ανάλυσής τους και εξαγωγής συμπερασμάτων**

Επιβλέπων Καθηγητής: Γ. Δημητρακόπουλος

Αθήνα, 2016

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα επίκουρο καθηγητή κ. Γεώργιο Δημητρακόπουλο για την δυνατότητα που μου έδωσε να πραγματοποιήσω την πτυχιακή μου εργασία και για την πολύ σημαντική του καθοδήγηση και υποστήριξη. Επίσης θα ήθελα να ευχαριστήσω την Καθηγήτρια κα. Μ. Νικολαΐδου και τον Λέκτορα κ. Χ. Μιχαλακέλη για την συνολική τους βοήθεια στην υλοποίηση της παρούσας εργασίας. Θα ήθελα να εκφράσω τις ευχαριστίες μου σε όλους τους καθηγητές του τμήματος Πληροφορικής και Τηλεματικής για τις σημαντικές υποδείξεις και συμβουλές που μου προσέφεραν όλα τα χρόνια των σπουδών μου.

Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου και την αδερφή μου για την αμέριστη στήριξη, συμπαράσταση και κατανόηση σε όλες τις δύσκολες στιγμές. Επίσης θα ήταν παράλειψη μου να μην ευχαριστήσω όλους τους φίλους μου, που με βοήθησαν και με στήριζαν όλα αυτά τα χρόνια.

Περίληψη

Σήμερα είναι σημαντική η ανάπτυξη εφαρμογών μέσω των οποίων αυτοματοποιούνται διάφορες λειτουργίες που στο παρελθόν γίνονταν χειροκίνητα από τον χρήστη όπως για παράδειγμα η εισαγωγή δεδομένων σε κάποια βάση δεδομένων.

Αντικείμενο της παρούσας πτυχιακή εργασίας είναι η μελέτη, σχεδίαση και υλοποίηση ενός Web Service Client μέσω του οποίου θα γίνεται η αποθήκευση των τιμών του ακαθάριστου εγχώριου προϊόντος (ή ΑΕΠ) (αγγλ. Gross Domestic Product – GDP) όλων των περιφερειών των χωρών της Ευρωπαϊκής Ένωσης στη βάση δεδομένων. Εκτός από τη τιμή του ΑΕΠ θα αποθηκεύονται και άλλες χρήσιμες πληροφορίες που παρέχονται από τους πίνακες της Eurostat. Επίσης τα δεδομένα θα μελετηθούν με τη χρήση του στατιστικού προγράμματος, SPSS με σκοπό την εξαγωγή συμπερασμάτων για την οικονομία των χωρών της Ευρωπαϊκής Ένωσης. Με αυτό τον τρόπο θα δημιουργηθεί μια γραφική αναπαράσταση των πληροφοριών που υπάρχουν στη βάση, ώστε να είναι πιο κατανοητά και διαχειρίσιμα προς τον χρήστη.

Κατά την υλοποίηση της εργασίας δόθηκε ιδιαίτερη μέριμνα στην προσέγγιση της έρευνας και της επιλογής των πρωτοκόλλων επικοινωνίας που επιτρέπουν την ταχύτερη και ασφαλέστερη ανταλλαγή πληροφοριών.

Επιπλέον, δόθηκε μεγάλη βαρύτητα στην μέθοδο εξόρυξης των δεδομένων καθώς και στην επιλογή των εργαλείων που χρησιμοποιήθηκαν για την ανάλυση των δεδομένων εξαιτίας την ανάγκης για οπτικοποίηση των αποτελεσμάτων ώστε να δοθεί στον χρήστη άμεση δυνατότητα κατανόηση της μοντελοποίησης των οικονομικών δεικτών.

Abstract

Nowadays is important to develop applications through which we can automate various functions which over the past were applied manually such as the process of entering data into a computerized database.

Purpose of this thesis is the research, the design and implementation of a Web Service Client through which prices of gross domestic product (or GDP) (in English. Gross Domestic Product - GDP) of all European Union countries will be stored in a new database. Apart from the value of GDP also will be stored other useful information provided by the Eurostat tables. Also the data will be modelled using statistical program, SPSS to draw conclusions for the economy of the European Union. This will create a graphical representation of the information available in the database, to make it comprehensible and manageable for the user.

During the implementation of the work was given special attention in the research approach and choice of communication protocols that allow fast and secure information exchange.

Moreover, it was given great emphasis on the data mining method and the choice of tools used to analyse the data because of the need for visualization of the results to give the user immediate modelling representation of the economic indicators.

Περιεχόμενα

| | |
|--|-----------|
| 1 Εισαγωγή..... | 7 |
| 1.1 Προσδιορισμός προβλήματος..... | 7 |
| 1.2 Κίνητρο..... | 7 |
| 1.3 Σκοπός..... | 7 |
| 1.4 Δομή..... | 8 |
| 2 Βιβλιογραφική Ανασκόπηση..... | 9 |
| 2.1 Εισαγωγή στα Web Services..... | 9 |
| 2.1.1 Χαρακτηριστικά των WS..... | 10 |
| 2.1.2 Εφαρμογές των WS..... | 11 |
| 2.1.3 Τεχνολογίες των WS..... | 11 |
| 2.2 WSDL..... | 12 |
| 2.2.1 WSDL Versions..... | 12 |
| 2.2.2 Δομή WSDL..... | 12 |
| 2.3 SOAP..... | 13 |
| 2.3.1 SOAP Versions..... | 14 |
| 2.3.2 Δομή SOAP..... | 14 |
| 2.4 UDDI..... | 15 |
| 2.4.1 Δομή UDDI..... | 16 |
| 2.5 XML..... | 17 |
| 2.5.1 Δομή XML..... | 17 |
| 2.6 MYSQL..... | 18 |
| 2.7 SDMX..... | 20 |
| 2.7.1 Ορισμοί SDMX..... | 20 |
| 3 Βάση Δεδομένων..... | 22 |
| 3.1 Σκοπός Eurostat..... | 22 |
| 3.2 Κατηγορίες Δεδομένων..... | 22 |
| 3.2.1 Web Services Eurostat..... | 23 |
| 3.3 Ανάλυση και Υλοποίηση Βάσης..... | 24 |
| 3.3.1 Εύρεση Dataset..... | 25 |
| 3.3.2 Ανάλυση δεδομένων..... | 26 |
| 3.3.3 Σχεδίαση βάσης δεδομένων..... | 26 |
| A) GDP_Basic_Table..... | 26 |
| B) Geo_Nuts..... | 27 |

| | |
|--|-----------|
| 3.4 Σύνδεση Βάσης..... | 28 |
| 4 Αντικειμενοστρεφής Προγραμματισμός..... | 30 |
| 4.1 Ιστορία του Α.Π..... | 30 |
| 4.2 Βασικές έννοιες..... | 30 |
| 4.3 Πλεονεκτήματα..... | 31 |
| 4.4 Java..... | 31 |
| 4.4.1 Εκδόσεις της Java..... | 31 |
| 4.4.2 Java Platforms..... | 32 |
| 4.5 Δομή της Java..... | 33 |
| 4.5.1 Κλάση (Class)..... | 33 |
| 4.5.2 Αντικείμενο (Object)..... | 33 |
| 4.5.3 Μεταβλητές (Variables)..... | 33 |
| 4.5.4 Κατασκευαστές (Constructors)..... | 33 |
| 4.6 JDBC API..... | 34 |
| 4.6.1 Αρχιτεκτονική του JDBC..... | 34 |
| 4.7 JAXP API..... | 36 |
| 4.7.1 DOM API..... | 36 |
| 4.7.2 SAX API..... | 37 |
| 5 Στατιστική Ανάλυση Δεδομένων..... | 39 |
| 5.1 Εισαγωγή στο SPSS..... | 39 |
| 5.2 Εγκατάσταση λογισμικού..... | 39 |
| 5.2.1 Προδιαγραφές..... | 39 |
| 5.2.2 Βήματα εγκατάστασης..... | 40 |
| 5.3 Περιβάλλον SPSS..... | 45 |
| 5.3.1 Καταχώρηση δεδομένων..... | 45 |
| 5.3.2 Μεταβλητές..... | 46 |
| 5.3.3 Παρουσίαση δεδομένων..... | 47 |
| 5.3.4 Υπολογισμός στατιστικών μέτρων..... | 49 |
| 5.4 Ανάλυση δεδομένων..... | 50 |
| 6 Συμπεράσματα..... | 53 |
| 6.1 Συμπεράσματα..... | 53 |
| 6.2 Μελλοντική εργασία..... | 53 |
| 7 Βιβλιογραφία..... | 54 |
| Παράρτημα Κώδικα | |

Κεφάλαιο 1: Εισαγωγή

1.1 Προσδιορισμός προβλήματος

Η εύρεση και η ανάλυση των δεδομένων σήμερα είναι ένα πολύ σημαντικό θέμα. Ιδιαίτερα όμως περίπλοκη είναι η ανάλυση δεδομένων τα οποία είτε μεταβάλλονται είτε επαυξάνονται με την πάροδο του χρόνου. Για παράδειγμα τα οικονομικά δεδομένα με την πάροδο των χρόνων επαυξάνονται σε μια βάση δεδομένων έτσι ώστε ο χρήστης να μπορεί να έχει μια συνολική εικόνα της οικονομικής ανάπτυξης μιας χώρας συγκρίνοντας για παράδειγμα την τιμή του ακαθάριστου εγχώριου προϊόντος (ή ΑΕΠ) (αγγλ. *Gross Domestic Product – GDP*). Συνεπώς είναι χρήσιμη η αναπαράσταση του συνόλου των δεδομένων με τρόπο απλό και κατανοητό με τη χρήση στατιστικών εργαλείων ώστε να γίνεται εύκολα η σύγκριση.

Για το λόγο αυτό πλέον πολλές πηγές δεδομένων όπως η Eurostat διαθέτουν πλήθος ανοιχτών δεδομένων τα οποία είναι προσβάσιμα μέσω των Web Services. Η Eurostat προσθέτει καθημερινά δεδομένα στην βάση της, που σημαίνει ότι πρέπει να υπάρχει συγχρονισμός μεταξύ της βάσης της Eurostat και της βάσης που αποθηκεύει ο χρήστης αυτά τα δεδομένα. Έτσι δημιουργείται η απαίτηση εύρεσης ενός εύκολου τρόπου μέσω του οποίου θα γίνεται η ενημέρωση της βάσης του χρήστη κάθε φορά που αυτό απαιτείται.

1.2 Κίνητρο

Η ενημέρωση των εγγραφών των πινάκων σε μια βάση δεδομένων είναι μια χρονοβόρα διαδικασία, ειδικά όταν τα δεδομένα πρέπει να ενημερώνονται καθημερινά. Η παρούσα πτυχιακή στοχεύει στην επίλυση του παραπάνω προβλήματος με την ανάπτυξη ενός προγράμματος για την αυτοματοποιημένη ενημέρωση της βάσης δεδομένων.

Πρόβλημα όμως αποτελεί και η εξαγωγή γνώσης από ένα τεράστιο πλήθος δεδομένων. Για το λόγο αυτό πραγματοποιείται στατιστική ανάλυση των οικονομικών δεδομένων που υπάρχουν στη βάση και οπτικοποίηση των αποτελεσμάτων με τη βοήθεια στατιστικών οργάνων όπως πίνακες, καμπύλες. Τα συμπεράσματα αυτής της ανάλυσης θα βοηθήσουν στην καλύτερη κατανόηση της οικονομικής πορείας των χωρών με βάση το ΑΕΠ.

1.3 Σκοπός

Ο σκοπός της παρούσας εργασίας είναι η μελέτη και ανάπτυξη ενός Web Service Client μέσω του οποίου θα γίνεται η αποθήκευση των τιμών του ακαθάριστου εγχώριου προϊόντος (ή ΑΕΠ) (αγγλ. *Gross Domestic Product – GDP*) όλων των περιφερειών των χωρών της Ευρωπαϊκής Ένωσης στη βάση δεδομένων. Εκτός από την τιμή του ΑΕΠ θα αποθηκεύονται και άλλες χρήσιμες πληροφορίες που παρέχονται από τους πίνακες της Eurostat.

Επίσης τα δεδομένα θα μελετηθούν με τη χρήση κατάλληλου στατιστικού προγράμματος με

σκοπό την εξαγωγή συμπερασμάτων για την οικονομία των χωρών της ΕΕ. Με αυτό τον τρόπο θα δημιουργηθεί μια γραφική αναπαράσταση των πληροφοριών που υπάρχουν στη βάση, ώστε να είναι πιο κατανοητά και διαχειρίσιμα προς τον χρήστη.

1.4 Δομή

- Κεφάλαιο 2: Στο δεύτερο κεφάλαιο αναλύονται κάποιες βασικές έννοιες οι οποίες είναι απαραίτητες για τον αναγνώστη ώστε να μπορεί να κατανοήσει διάφορες ορολογίες που χρησιμοποιούνται στα υπόλοιπα κεφάλαια της πτυχιακής εργασίας.
- Κεφάλαιο 3: Στο τρίτο κεφάλαιο παρουσιάζεται ο τρόπος σχεδίασης της βάσης δεδομένων όπου αποθηκεύονται τα δεδομένα. Επίσης γίνεται ανάλυση του είδους των δεδομένων.
- Κεφάλαιο 4: Στο τέταρτο κεφάλαιο γίνεται αναφορά στον αντικειμενοστρεφή προγραμματισμό και δίνεται ιδιαίτερη έμφαση στην αντικειμενοστρεφή γλώσσα προγραμματισμού, Java.
- Κεφάλαιο 5: Στο πέμπτο κεφάλαιο πραγματοποιείται η στατιστική ανάλυση των δεδομένων και παρουσιάζονται τα αποτελέσματα με τη χρήση γραφικών παραστάσεων.
- Κεφάλαιο 6: Στο τελευταίο κεφάλαιο παρουσιάζονται τα συμπεράσματα της πτυχιακής εργασίας και μελλοντικές επεκτάσεις.

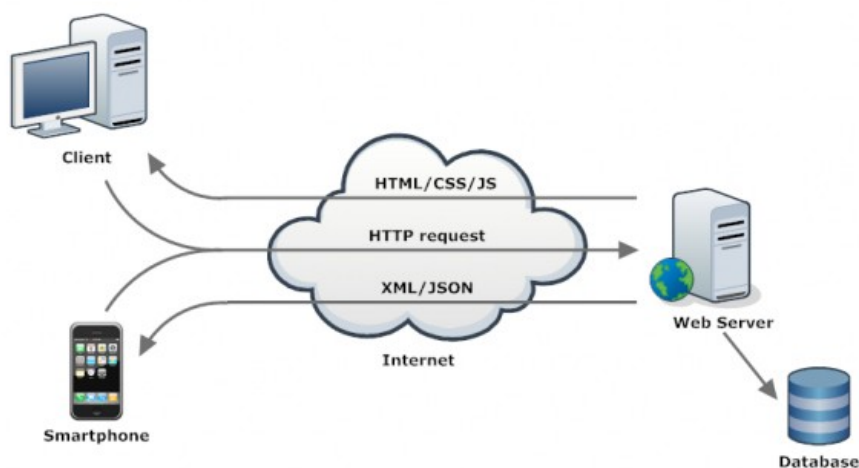
Κεφάλαιο 2: Βιβλιογραφική Ανασκόπηση

Στο παρόν κεφάλαιο παρουσιάζεται η ανασκόπηση της βιβλιογραφίας που χρησιμοποιήθηκε κατά την συγγραφή της συγκεκριμένης πτυχιακής εργασίας. Παρουσιάζονται τα χαρακτηριστικά των Web Services καθώς επίσης και οι τεχνολογίες που χρησιμοποιούν. Γίνεται ανάλυση των βασικών λειτουργιών των WS, όπως SOAP, WSDL, UDDI και XML. Επίσης αναλύονται διάφορες εφαρμογές των WS. Τέλος γίνεται αναφορά στην βάση που χρησιμοποιήθηκε για την αποθήκευση των δεδομένων.

2.1 Εισαγωγή στα Web Services

Η IBM ορίζει τα Web Services ως μία τεχνολογία που επιτρέπει στις εφαρμογές να επικοινωνούν μεταξύ τους ανεξαρτήτως πλατφόρμας και γλώσσας προγραμματισμού. Ένα Web Service είναι μία διεπαφή λογισμικού (software interface) που περιγράφει μια συλλογή από λειτουργίες οι οποίες μπορούν να προσεγγιστούν από το δίκτυο μέσω πρότυπων μηνυμάτων XML. Χρησιμοποιεί πρότυπα βασισμένα στη γλώσσα XML για να περιγράψει μία λειτουργία (operation) προς εκτέλεση και τα δεδομένα προς ανταλλαγή με κάποια άλλη εφαρμογή. Μία ομάδα από WS οι οποίες αλληλεπιδρούν μεταξύ τους καθορίζει μία εφαρμογή WS.

Το Web Service πρέπει να περιλαμβάνει ένα Web-based πρωτόκολλο, όπως το HTTP ή το SMTP – Simple Transfer Protocol. Μπορούν να χρησιμοποιηθούν και άλλα πρωτόκολλα όμως το πιο γνωστό είναι το HTTP. Υπάρχει και το HTTPS που διατηρεί ασφαλή την μεταφορά δεδομένων χρησιμοποιώντας SSL – Secure Socket Layer ή TLS – Transport Secure Layer. [2]



Σχήμα 2.1

Τα τελευταία δυο χρόνια έχουν οριστεί τρεις βασικές τεχνολογίες οι οποίες έχουν αναδειχθεί ως παγκόσμια πρότυπα και αποτελούν τον πυρήνα των σημερινών Web Services.

- SOAP - Simple Object Access Protocol
- WSDL - Web Service Description Language
- UDDI – Universal Description, Discovery and Intergation

Η σχέση μεταξύ αυτών των τριών τεχνολογιών μπορεί να περιγραφεί ως εξής: μια εφαρμογή η οποία έχει το ρόλο του Web Service Client πρέπει να συνδέεται με μια άλλη εφαρμογή (Web Service) που βρίσκεται κάπου στο δίκτυο. Ο Client κάνει ερωτήματα (queries) στο UDDI ψάχνοντας την υπηρεσία με βάση το όνομα, την κατηγορία ή το αναγνωριστικό. Μόλις εντοπίσει την υπηρεσία που ψάχνει, ο Client λαμβάνει πληροφορίες σχετικά με τη θέση του αρχείου WSDL από το μητρώο του UDDI. Το αρχείο WSDL περιέχει πληροφορίες σχετικά με τον τρόπο επικοινωνίας με το WS και τη μορφή των μηνυμάτων σε μορφή XML. Ο Client συντάσσει το μήνυμα σύμφωνα με το XML σχήμα που υπάρχει στο WSDL και τέλος στέλνει το αίτημα του στον κεντρικό υπολογιστή όπου βρίσκεται το WS.[1]

2.1.1 Χαρακτηριστικά των Web Services

- **XML Based:** Η χρήση της XML – eXtensible Markup Language είναι ένα από τα βασικά χαρακτηριστικά των WS. Με την XML πραγματοποιείται η μεταφορά δεδομένων αλλά και η αναπαράστασή τους. Στο επίπεδο της μεταφοράς των δεδομένων, η XML απαλείφει κάθε δικτύωση, λειτουργικό σύστημα ή δέσμευση πλατφόρμας που έχει ένα πρωτόκολλο.
- **Loosely Coupled:** Ένας χρήστης ενός Web Service δεν είναι άμεσα δεσμευμένος με το συγκεκριμένο WS. Το interface του Service μπορεί να αλλάξει με την πάροδο του χρόνου, χωρίς όμως να διακυβεύεται η δυνατότητα που έχει ο client να αλληλεπιδρά με την υπηρεσία. Ένα στενά συνδεδεμένο (tightly coupled) σύστημα δεσμεύει τον client με τον server και σε περίπτωση που αλλάξει το interface του ενός πρέπει να ενημερωθεί άμεσα ο άλλος. Υιοθετώντας όμως μια loosely coupled αρχιτεκτονική, τα συστήματα γίνονται πιο εύχρηστα και επιτρέπεται η απλούστερη ενσωμάτωση μεταξύ διαφορετικών συστημάτων.
- **Coarse Grained:** Οι αντικειμενοστρεφείς τεχνολογίες όπως η JAVA, υλοποιούν τις υπηρεσίες τους μέσω μεμονωμένων μεθόδων. Μια μέθοδος είναι μια πολύ λεπτή λειτουργία που παρέχει κάθε χρήσιμη ικανότητα σε ένα συνεταιρικό επίπεδο. Η ανάπτυξη ενός προγράμματος σε Java από το μηδέν απαιτεί τη δημιουργία πολλών “fine-grained” μεθόδων οι οποίες στη συνέχεια ενσωματώνονται σε μια “coarse-grained” υπηρεσία που εκτελείται είτε από έναν πελάτη είτε από μια άλλη υπηρεσία. Τα web services παρέχουν έναν φυσικό τρόπο ορισμού υπηρεσιών “coarse-grained”, που έχουν πρόσβαση στην επιχειρησιακή λογική.
- **Ability to be synchronous or asynchronous:** Ο συγχρονισμός αναφέρεται στη σύνδεση του πελάτη με την εκτέλεση ενός service. Στα σύγχρονα WS, ο client στέλνει ένα αίτημα σε μορφή μηνύματος και περιμένει να λάβει από το service ένα μήνυμα σχετικά με την ολοκλήρωση της λειτουργίας, όταν το λάβει συνεχίζει. Στα ασύγχρονα WS, ο client μπορεί να καλέσει μια υπηρεσία και στη συνέχεια να εκτελέσει άλλες λειτουργίες. Τα ασύγχρονα web services αποτελούν βασικό παράγοντα που επιτρέπει την λειτουργία “Loosely Coupled” συστημάτων.
- **RPCs – Supports Remote Procedure Calls:** Τα WS επιτρέπουν στους clients να καλούν

διαδικασίες, λειτουργίες και μεθόδους σε απομακρυσμένα αντικείμενα χρησιμοποιώντας ένα “XML-based” πρωτόκολλο. Οι απομακρυσμένες διαδικασίες εκθέτουν τις παραμέτρους εισαγωγής και εξαγωγής που ένα WS πρέπει να υποστηρίξει. Ένα web service υποστηρίζει RPC με την παροχή υπηρεσιών του, ισοδύναμων με εκείνες ενός παραδοσιακού συστατικού, ή με την μετάφραση των εισερχομένων κλήσεων σε μια κλήση ενός EJB – Enterprise JavaBeans ή ενός .NET συστατικού.

- **Supports document exchange:** Ένα από τα βασικότερα χαρακτηριστικά της XML είναι ο γενικός τρόπος αναπαράστασης όχι μόνο των δεδομένων αλλά και σύνθετων εγγράφων. Αυτά τα έγγραφα μπορούν να είναι απλά, όπως όταν δείχνουν την τρέχουσα διεύθυνση ή μπορούν να είναι και σύνθετα, αντιπροσωπεύοντας ένα ολόκληρο βιβλίο ή ένα RFQ. Τα WS υποστηρίζουν τη διαφανή ανταλλαγή των εγγράφων για να διευκολύνουν την επιχειρησιακή ενοποίηση.[1]

2.1.2 Εφαρμογές των Web Services

Τα πρώτα Web Services είχαν σκοπό να παρέχουν πληροφορίες που θα μπορούσαν να ενσωματωθούν εύκολα σε εφαρμογές μετοχών, δελτίων καιρού, αποτελεσμάτων αγώνων κλπ. Είναι εύκολο να φανταστεί κανείς μια ολόκληρη κατηγορία εφαρμογών που θα μπορούσε να κατασκευάσει ώστε να αναλύσει και να συγκεντρώσει τις πληροφορίες που τον ενδιαφέρουν και να τις παρουσιάσει με διαφορετικούς τρόπους. Για παράδειγμα, θα μπορούσε να έχει ένα λογιστικό φύλλο το οποίο συνοψίζει όλη την οικονομική εικόνα μιας εταιρείας: μετοχές, τραπεζικούς λογαριασμούς, δάνεια κλπ. Αν αυτή η πληροφορία ήταν διαθέσιμη μέσω Web Services το λογιστικό φύλλο θα μπορούσε να ενημερώνεται αυτοματοποιημένα συνεχώς. Οι περισσότερες από αυτές τις πληροφορίες είναι ήδη διαθέσιμες στον παγκόσμιο ιστό αλλά τα WS κάνουν την προγραμματιστική πρόσβαση σε αυτές πιο εύκολη και πιο αξιόπιστη.

Εκθέτοντας ήδη υπάρχουσες εφαρμογές σαν WS θα επιτρέψει στους χρήστες να κατασκευάσουν νέες πιο ισχυρές εφαρμογές οι οποίες χρησιμοποιούν τα WS σαν δομικά στοιχεία. Για παράδειγμα, ένας χρήστης θα μπορούσε να αναπτύξει μια εφαρμογή προμηθειών η οποία να παίρνει αυτόματα τιμές από προμηθευτές, να επιτρέπει στο χρήστη να υποβάλει την παραγγελία του και να παρακολουθεί την αποστολή έως ότου γίνει η παραλαβή της. Η εφαρμογή του προμηθευτή θα μπορούσε να χρησιμοποιήσει άλλα Web Services για να ελέγξει την πιστοληπτική ικανότητα του πελάτη, να χρεώσει τον τραπεζικό λογαριασμό του πελάτη και να καθορίσει την αποστολή σε μια εταιρεία μεταφορών.[9]

Τα WS κοινώς ομολογούμενα είναι μία τεχνολογία που θα χρησιμοποιηθεί αναγκαστικά, στην ανάπτυξη λογισμικού. Πρόσφατα άρχισαν παρόλα αυτά να κεντρίζουν την προσοχή όλων.

Ένα από τα σπουδαιότερα θέματα τα οποία αναδεικνύονται από την χρήση WS είναι η χρησιμοποίηση του Διαδικτύου ως ο παράγων που επιτρέπει την επαναχρησιμοποίηση και σύνθεση των υπηρεσιών. Υπάρχοντες πάροχοι υπηρεσιών θα συνδυάζουν τις βασικές τους δυνατότητες και πόρους, για να δημιουργήσουν σύνθετες υπηρεσίες.

Με άλλα λόγια, μία σύνθετη υπηρεσία είναι μία συσσωμάτωση διαφορετικών υπηρεσιών που εργάζονται συνδυαστικά για να παρέχουν μία υπηρεσία προστιθέμενης αξίας. Οι προσπάθειες προκειμένου να ορισθούν πρότυπα για την σύνθεση των WS βρίσκονται εν εξελίξει. Αυτές

περιλαμβάνουν την WSFL (Web Service Flow Language), XLANG και BPEL4WS (Business Process Execution Language for Web Services) . Αν και τυγχάνουν καινούργια τεχνολογία, τα WS χρησιμοποιούνται για να καλύψουν επιχειρησιακές ανάγκες, συμπεριλαμβανομένων των ακόλουθων:

- Ολοκλήρωση επιχειρηματικής εφαρμογής (Enterprise Application Integration)
- Ολοκλήρωση επιχειρησιακών συνεργατών (BusinessPartnerIntegration)
- Portal, ταμπλώ, έλεγχος επιχειρησιακής δραστηριότητας (Portal Integration, Dashboards, Business Activity Monitoring)
- Εκτεταμένη λειτουργία για τις εφαρμογές Ιστού (Extended Functionality for Web Applications)
- Βελτιωμένη αποδοτικότητα ανάπτυξης εφαρμογών (Improved Application Development Efficiency) [8]

2.1.3 Τεχνολογίες Web Services

To framework του W3C για τα web services, αποτελεί μια συνένωση τριών βασικών XML προδιαγραφών:

- WSDL - Web Service Description Language
- SOAP - Simple Object Access Protocol
- UDDI – Universal Description, Discovery and Intergation

2.2 WSDL

Η WSDL είναι μια γλώσσα η οποία ακολουθεί τη δομή της XML. Περιγράφει τέσσερα σημαντικά κομμάτια δεδομένων:

1. Πληροφορία για όλες τις διαθέσιμες λειτουργίες (operations).
2. Πληροφορίες τύπων δεδομένων για όλα τα μηνύματα αίτησης και απάντησης(types).
3. Πληροφορίες τύπων σύνδεσης σχετικά με τα πρωτόκολλα μεταφοράς(binding).
4. Πληροφορίες διευθύνσεων URL για τον εντοπισμό της συγκεκριμένης υπηρεσίας.[3]

Η WSDL είναι ένας τρόπος με τον οποίο περιγράφετε η διεπαφή (interface) ενός Web Service. Είναι ένας τρόπος περιγραφής των υπηρεσιών του WS και πως θα πρέπει να συνδέεται με συγκεκριμένες διευθύνσεις δικτύου. [4]

2.2.1 WSDL Versions

Το WSDL έχει οριστεί ως πρότυπο από το W3C. Το έργο αυτό έχει ξεκινήσει από την άνοιξη του 2001, και η ανάπτυξη του συνεχίζεται ακόμη. Η πρώτη έκδοση, το WSDL 1.0 αναπτύχθηκε από την IBM, Microsoft, και Ariba για να περιγράψουν τα Web Services για το SOAP toolkit τους. Χτίστηκε συνδυάζοντας δυο περιγραφικές γλώσσες: την NASSL (Network Application

Service Specification Language) της IBM και την SDL (Service Description Language) της Microsoft.

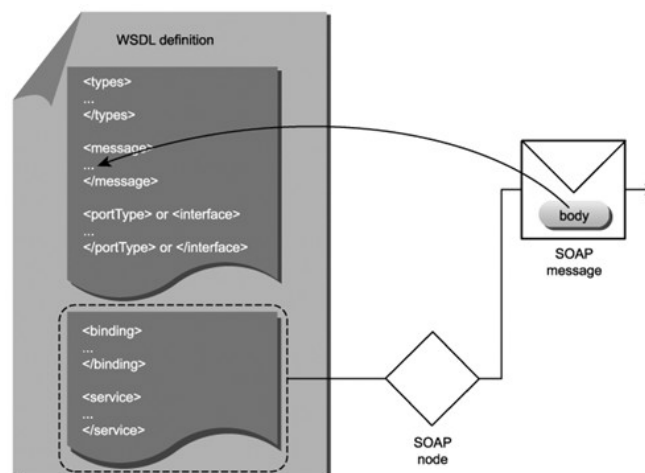
Το WSDL 1.1, που δημοσιεύθηκε τον Μάρτιο του 2001, είναι η επισημοποίηση της WSDL 1.0. Δεν εισήχθησαν σημαντικές αλλαγές μεταξύ 1.0 και 1.1.

Η τελευταία έκδοση είναι η WSDL 2.0, επισημοποιήθηκε από το W3C τον Ιουνίου του 2007. Οι αλλαγές μεταξύ της προηγούμενης έκδοσης και της νεότερης ήταν πολύ μεγάλες για αυτό το λόγο ονομάστηκε σε WSDL 2.0 αντί για WSDL 1.2.

Οι αλλαγές είναι επιγραμματικά οι εξής: προστέθηκε επιπλέον σημασιολογία στην περιγραφή της γλώσσας, αφαιρέθηκαν κατασκευαστικά μηνύματα, δεν υποστηρίζεται η υπερφόρτωση λειτουργιών, τα PortTypes μετονομάστηκαν σε Interfaces, και τέλος τα Ports μετονομάστηκαν σε EndPoints.[7]

2.2.2 Δομή WSDL

Τα έγγραφα WSDL περιέχουν συνήθως δύο ομάδες ορισμών: ένα αφηρημένο μέρος (abstract) και ένα συγκεκριμένο μέρος (concrete). Το αφηρημένο μέρος περιγράφει τι κάνει το Web Service από άποψη των μηνυμάτων που καταναλώνει και παράγει, χωρίς να εξετάζει το πώς και το που παρέχεται. Το άλλο τμήμα καλύπτεται από τα στοιχεία <types>, <messages> και <portType>. Τα τρία αυτά αντικείμενα μας δίνουν πληροφορίες σχετικά με “τι” κάνει το WS. Το “πώς” και το “που” καλύπτονται από τα στοιχεία <binding> και <service>.



Σχήμα 2.2

Πιο αναλυτικά παρουσιάζονται τα στοιχεία (elements) του WSDL.

1. **<types>**: Το στοιχείο αυτό χρησιμοποιείται για τη φιλοξενία δηλώσεων δομών δεδομένων που αναφέρονται στη συνέχεια του WSDL αρχείου, με σκοπό τον ορισμό των μηνυμάτων που ανταλλάσσει μια υπηρεσία.
2. **<message>**: Το στοιχείο αυτό επιτρέπει την περιγραφή του μηνύματος που ανταλλάσσει το WS. Στη WSDL, τα μηνύματα μπορούν να αποτελούνται από ένα ή περισσότερα μέρη, που κάθε μέρος αντιπροσωπεύει ένα μοναδικό στοιχείο το οποίο είτε πρόκειται να σταλεί είτε να ληφθεί.

3. **<portType>**: Το portType ορίζει ένα σύνολο σχετικών <operations> (λειτουργίες), τις οποίες υποστηρίζει το Web Service. Μια λειτουργία είναι μια ομαδοποίηση από σχετικά σύνολα μηνυμάτων που ανταλλάσσονται. Κάθε λειτουργία μπορεί να στείλει και να λάβει το πολύ ένα μήνυμα προς κάθε κατεύθυνση.
4. **<binding>**: Ο σκοπός του στοιχείου <binding> είναι να περιγράψει τον τρόπο μορφοποίησης όλων των μηνυμάτων ώστε να μπορούν να αλληλεπιδρούν με μια συγκεκριμένη υπηρεσία. Δεν υπάρχει σταθερός τρόπος μορφοποίησης των μηνυμάτων αντίθετα τα μηνύματα μπορούν να είναι σε μορφή SOAP, HTTP ή MIME.
5. **<service>**: Είναι το τελευταίο μέρος της περιγραφής του WS. Δείχνει που βρίσκεται το service που θέλουμε να βρούμε.[6]

2.3 SOAP

Το SOAP είναι το βασικό framework ανταλλαγής μηνυμάτων μεταξύ των Web Services. Με την χρήση του SOAP, μπορεί ο χρήστης να έχει πρόσβαση στα WS μέσω “loosely coupled” υποδομές που παρέχουν ανθεκτικότητα, επεκτασιμότητα και ευελιξία στην εγκατάσταση της σύνδεσης χρησιμοποιώντας διαφορετικές τεχνολογίες υλοποίησης και μεταφορές μηνυμάτων εντός ενός δικτύου.

Το SOAP παρέχει τέσσερις βασικές δυνατότητες:

1. Μια τυποποιημένη δομή μηνυμάτων βασισμένη στο XML Infoset.
2. Ένα μοντέλο επεξεργασίας, το οποίο περιγράφει πως πρέπει να επεξεργάζεται τα μηνύματα μια υπηρεσία.
3. Έναν μηχανισμό για τη σύνδεση(bind) μηνυμάτων SOAP σε διαφορετικά δικτυακά πρωτόκολλα μεταφοράς.
4. Έναν τρόπο σύνδεσης κωδικοποιημένων πληροφοριών μη-XML σε μηνύματα SOAP.

2.3.1 SOAP Versions

Το SOAP ξεκίνησε ως το Απλό Πρωτόκολλο Προσπέλασης Αντικειμένων (Simple Object Access Protocol), που αναπτύχθηκε από τις εταιρείες Microsoft, DevelopMentor και Userland. Στη συνέχεια, η IBM και η Lotus συνέβαλλαν σε μια αναθεωρημένη προδιαγραφή, που είχε αποτέλεσμα την δημιουργία της έκδοσης SOAP 1.1 τον Απρίλιο του 2000. Αυτή η προδιαγραφή έγινε ευρέως αποδεκτή από το σύνολο της βιομηχανίας και αποτέλεσε τη βάση πολλών δια λειτουργικών υλοποιήσεων ανοιχτού κώδικα. Υιοθετήθηκε από τον Οργανισμό Διαλειτουργικότητας Υπηρεσιών Ιστού (WS-I) ως μέρος του βασικού προφίλ του.

Τον Μάιο του 2000, οι συγγραφείς υπέβαλαν την προδιαγραφή SOAP 1.1 στην Κοινοπραξία του Παγκόσμιου Ιστού (W3C), όπου αντιμετωπίστηκε ως έγγραφο εκκίνησης στο σχηματισμό της Ομάδας Εργασίας για το Πρωτόκολλο XML(αργότερα έγινε γνωστό ως SOAP) τον Σεπτέμβριο του 2000. Το καταστατικό αυτής της ομάδας εργασίας επρόκειτο να επιφέρει την

τυποποίηση του SOAP.

Υπήρχε μια σύγχυση σχετικά με το τι αντιπροσώπευε το ακρόνυμο SOAP, με αποτέλεσμα να χρησιμοποιούνται διάφορες δημοφιλείς ερμηνείες. Αρκετά νωρίς, η ομάδα εργασίας αποφάσισε ότι το SOAP δεν θα ήταν πλέον ακρόνυμο, από εκεί και πέρα το πρωτόκολλο ήταν απλώς γνωστό ως SOAP. Η ομάδα εργασίας της W3C δημιούργησε τελικά την Εισήγηση SOAP 1.2, τον Ιούνιο του 2003. Η Εισήγηση αποτελείτο από μια εισαγωγή [SOAP 1.2 Part 0], ένα πλαίσιο εργασίας ανταλλαγής μηνυμάτων [SOAP 1.2 Part 1], και παραρτήματα [SOAP 1.2 Part 2]. Η ομάδα εργασίας έχει αναπτύξει επίσης προδιαγραφές για προσαρτήματα [MTOM],[XOP].

2.3.2 Δομή SOAP

Το SOAP είναι ένα σύνολο συμβάσεων, οι οποίες καθορίζουν μια μορφή μηνύματος μαζί με ένα σύνολο κανόνων που διέπουν την επεξεργασία του μηνύματος, καθώς αυτό μεταβιβάζεται κατά μήκος μιας διαδρομής. Οι συμβάσεις περιγράφουν πως συναρμολογείται ένα μήνυμα, και τι είδους αλληλεπιδράσεις μπορούν να εκτελέσουν οι κόμβοι SOAP που επεξεργάζονται τα μηνύματα κατά μήκος της διαδρομής του.

Ένα SOAP μήνυμα αποτελείται από τρία βασικά μέρη. Το πιο ακραίο στοιχείο είναι το **env:Envelope**, το οποίο περιλαμβάνει το χώρο ονομάτων URL για το SOAP. Μέσα σε αυτό, εσωκλείονται δυο υποστοιχεία που καθορίζει το SOAP. Πρόκειται για τα στοιχεία **env:Header** και **env:Body**. Το SOAP δεν καθορίζει τα περιεχόμενα τους. Τα στοιχεία είναι συγκεκριμένα για την εφαρμογή η οποία δημιουργεί και επεξεργάζεται το μήνυμα SOAP. Ωστόσο, η προδιαγραφή SOAP καθορίζει πως γίνεται η επεξεργασία των εν λόγω στοιχείων από έναν κόμβο SOAP. Το στοιχείο env:Header (κεφαλίδα) είναι προαιρετικό στο SOAP, αλλά συμπεριλαμβάνεται στο παράδειγμα που παρουσιάζεται στην εικόνα 2.3 ώστε να εξηγηθεί η χρήση του στο μήνυμα.

Η κεφαλίδα SOAP είναι ένας μηχανισμός επέκτασης, ο οποίος παρέχει έναν τρόπο για τη μεταβίβαση πληροφοριών μέσα σε ένα μήνυμα SOAP, οι οποίες δεν αποτελούν μέρος του επιχειρηματικού ωφέλιμου φορτίου του μηνύματος. Το στοιχείο env:Header περιλαμβάνει ένα άμεσο θυγατρικό στοιχείο, το οποίο περιέχει ένα σύνολο παραμέτρων ποιότητας. Στην προδιαγραφή SOAP, αυτό το άμεσο θυγατρικό στοιχείο λέγεται τμήμα κεφαλίδας (header block).

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <pns:qualityOfService xmlns:pns="http://example.org/qos">
      <pns:priority>3</pns:priority>
      <pns:timestamp>2004-02-25T01:00:00-00:00</pns:timestamp>
      <pns:persist>true</pns:persist>
    </pns:appHeaderBlock>
  </env:Header>
  <env:Body>
    <bmns:businessPO xmlns:env="http://example.org/po">
      <bmns:description>Widgets</bmns:description>
      <bmns:quantity>100</bmns:quantity>
      <bmns:price>20.5</bmns:price>
    </bmns:businessPO>
  </env:Body>
</env:Envelope>
```

Σχήμα 2.3

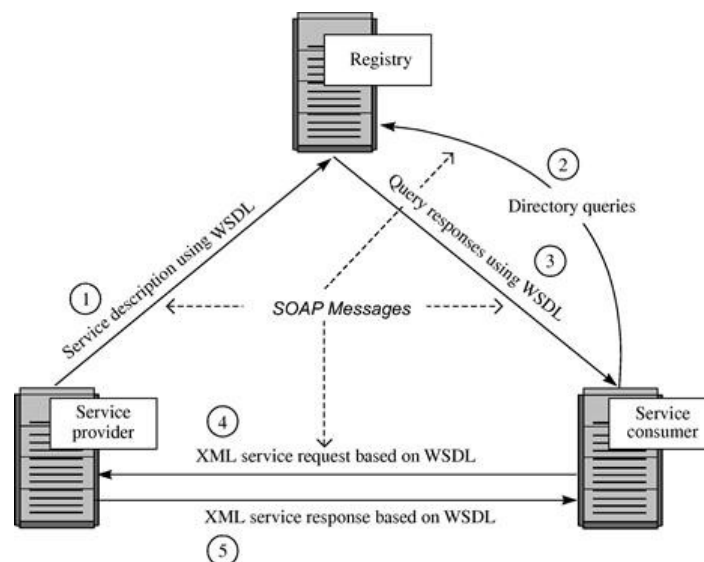
Το τμήμα της κεφαλίδας διαθέτει τον δικό του χώρο ονομάτων XML, στον οποίο εμφανίζονται τα εξής στοιχεία:

1. προτεραιότητας (priority)
2. ένδειξη ώρας (timestamp)
3. μονιμότητας (persist)

Το στοιχείο `env:Body` (σώμα) μέσα στο μήνυμα SOAP είναι υποχρεωτικό. Περιέχει το ωφέλιμο φορτίο του μηνύματος, δηλαδή τις πληροφορίες που μεταφέρονται από τον αρχικό αποστολέα στον τελικό παραλήπτη. Η επιλογή σχετικά με το ποιες πληροφορίες τοποθετούνται στο στοιχείο `env:Header` και ποιες στο `env:Body`, είναι θέμα επιχειρηματικής εφαρμογής και σχεδίασης του συστήματος. Το στοιχείο `env:Body` χρησιμοποιείται για τη φιλοξενία απαραίτητων πληροφοριών που αφορούν τη συγκεκριμένη εφαρμογή, τις οποίες επεξεργάζεται ο τελικός παραλήπτης στην υλοποίηση της απομακρυσμένης υπηρεσίας.[6]

2.4 UDDI

Το UDDI έχει ως στόχο να χρησιμοποιείται ως μέσο εύρεσης των Web Services που περιγράφονται στο αρχείο WSDL. Η κεντρική ιδέα είναι ότι το μητρώο UDDI θα μπορούσε να ψάξει με ποικίλους τρόπους προκειμένου να βρει τρόπο επικοινωνίας με την υπηρεσία και να βρει όσες υπηρεσίες είναι διαθέσιμες από διάφορους οργανισμούς. Ο όρος μητρώο αντικαθίσταται συχνά από τον όρο αποθετήριο υπηρεσίας. Σήμερα τα αποθετήρια χρησιμοποιούνται κυρίως κατά την σχεδίαση και ως βοηθητικό μέσο για τη διαχείριση του συστήματος.[4]



Σχήμα 2.2

2.4.1 Δομή UDDI

Τα κύρια στοιχεία του μοντέλου δεδομένων UDDI είναι τα εξής:

- businessEntity (επιχειρηματική οντότητα)
- businessService (επιχειρηματική υπηρεσία)
- bindingTemplate (πρότυπο σύνδεσης)
- tModel (τεχνικό μοντέλο)
- publisherAssestion (ισχυρισμός εκδότη)

Η κατηγοριοποίηση αποτελεί μία από τις σημαντικότερες πλευρές του UDDI. Στο μοντέλο δεδομένων UDDI, αναπαρίσταται από το στοιχείο categoryBag (σάκος κατηγοριών). Αυτό το στοιχείο μπορεί να περιέχει πολλά keyedReferences (ευρετηριασμένες αναφορές), με κάθε keyedReference να δείχνει μια κατηγοριοποίηση από ένα συγκεκριμένο σύστημα κατηγοριών. Κάθε σύστημα κατηγοριών αναπαρίστανται με ένα tModel(tModelKey) που αναπαριστά το σύστημα κατηγοριών. Το στοιχείο keyedReference μπορεί να περιέχει δυο επιπλέον πληροφορίες. Η πρώτη είναι ένα keyName (όνομα κλειδιού), το οποίο στα περισσότερα συστήματα κατηγοριών υπάρχει μόνο για λόγους τεκμηρίωσης, δεν χρησιμοποιείται σε ερωτήματα άλλα εκτός ενός προκαθορισμένου συστήματος κατηγοριών. Η δεύτερη είναι ένα keyValue(τιμή κλειδιού), το οποίο είναι η συγκεκριμένη τιμή που χρησιμοποιείται για το δεδομένο στιγμιότυπο του συστήματος κατηγοριών.

Για παράδειγμα, ένα από τα συστήματα κατηγοριών που περιγράφονται στην προδιαγραφή UDDI είναι το σύστημα γεωγραφικής κατηγοριοποίησης ISO 3166. Το κλειδί του tModel που αναπαριστά αυτό το σύστημα κατηγοριών είναι το uddi:uddi.org:ubr:categorization:iso3166. Αν μια επιχείρηση θέλει να δείξει ότι λειτουργεί στην πολιτεία της Καλιφόρνιας των ΗΠΑ, περιλαμβάνει ένα keyedReference με αυτο το tModelKey και ένα keyValue με τιμή “US-CA”.

Αν ένας πελάτης θέλει να ψάξει για μια επιχείρηση στην Καλιφόρνια, περιλαμβάνει στην αίτηση αναζήτησης του το ίδιο keyReference. Κατά την αναζήτηση ενός keyReference όπως αυτό, η προεπιλογή είναι μια ακριβής ταύτιση στην οποία λαμβάνεται υπόψη η διαφορά πεζών-κεφαλαίων. Ωστόσο, υπάρχει επίσης η επιλογή της εκτέλεσης μιας ταύτισης χωρίς διάκριση πεζών-κεφαλαίων (με το “us-ca”), ή μιας κατά προσέγγιση ταύτισης με τη χρήση χαρακτήρων-μπαλαντέρ, για παραδείγματα “US%”, ώστε να εντοπιστεί οποιαδήποτε πολιτεία των ΗΠΑ.[6]

2.5 XML

Ο κόσμος των Web Services βασίζεται σε εξαιρετικά μεγάλο βαθμό στο κύριο σύνολο των προδιαγραφών XML. Η XML στην πραγματικότητα δεν είναι μια προγραμματιστική γλώσσα αλλά μια μεταγλώσσα η οποία βοηθά στον καθορισμό νέων γλωσσών. Ο ορισμός της XML είναι ανεξάρτητος από πλατφόρμες και καθορίζεται με τη χρήση του προτύπου Unicode, το οποίο επιτρέπει την αναπαράσταση περιεχομένου από πολλές φυσικές γλώσσες. Λόγω αυτών των παραγόντων και της ευρείας υποστήριξης που έχει η XML από κάθε κατασκευαστή λογισμικού, η XML εξελίχθηκε γρήγορα στη de facto μορφή για την ανταλλαγή δεδομένων μεταξύ διαφορετικών οντοτήτων.

2.5.1 Δομή XML

Η XML παρέχει ένα μικρό σύνολο βασικών εννοιών για τον καθορισμό νέων γλώσσών:

- **Elements:** Ένα XML Element είναι μια επώνυμη δομή, η οποία διαθέτει ένα σύνολο χαρακτηριστικών (attributes) και ορισμένα θυγατρικά στοιχεία. Τα θυγατρικά στοιχεία ενός element μπορεί να είναι άλλα elements, κείμενο, σχόλια και ορισμένοι άλλοι τύποι. Τα elements γράφονται με την χρήση γωνιακών αγκυλών: Αρχικό element foo *<foo>* και τελικό element foo *</foo>*.
- **Attributes:** Τα attributes είναι ζεύγη ονόματος-τιμής, που σχετίζονται με ένα element. Ένα element μπορεί να έχει οποιοδήποτε πλήθος attributes και γράφεται ως εξής:
<foo name1="value1" name2="value2" ... >
- **Comments:** Τα σχόλια εσωκλείονται μέσα σε ακολουθίες χαρακτήρων, “<!-->” και “-->”, και αγνοούνται από τον επεξεργαστή.
- **Literal Text:** Ένα element μπορεί να περιέχει ακολουθίες χαρακτήρων Unicode. Ένα βασικό χαρακτηριστικό της XML είναι ότι όλοι οι χαρακτήρες που περιλαμβάνονται μέσα σε έγγραφο XML, αναπαρίστανται σε Unicode. Με τη χρήση Unicode, η XML μπορεί να αποθηκεύει χαρακτήρες από σχεδόν οποιαδήποτε γλώσσα. Επομένως, η XML είναι εξ ορισμού διεθνοποιημένη.
- **Document:** Ένα έγγραφο XML είναι μια μονάδα συσκευασίας XML, η οποία αποτελείται από ακριβώς ένα στοιχείο (λέγεται στοιχείο εγγράφου), και μπορεί να περιέχει σχόλια και ορισμένα άλλα στοιχεία.

Χρησιμοποιώντας αυτές τις έννοιες, μπορούν να οριστούν νέες γλώσσες, παρέχοντας απλώς ένα σύνολο ονομάτων στοιχείων, το έγκυρο περιεχόμενο τους και τα είδη του κυριολεκτικού κειμένου που είναι επιτρεπτά ως τιμές χαρακτηριστικών και περιεχόμενο στοιχείων. Μια βασική βελτίωση της XML σε σχέση με την SGML (το πρόδρομο της), είναι η έννοια των καλά διαμορφωμένων αλλά όχι απαραίτητως και έγκυρων εγγράφων. Ένα έγγραφο θεωρείται καλά διαμορφωμένο εφόσον τηρεί όλους τους συντακτικούς κανόνες της XML. Ένα καλά διαμορφωμένο έγγραφο είναι επίσης έγκυρο, αν ακολουθεί κάποιο DTD ή σχήμα XML, ή μια άλλη γλώσσα καθορισμού δομής εγγράφων.

Ένα παράδειγμα λεξιλογίου XML αποτελούν τα στοιχεία που καθορίζουν τη μορφή των μηνυμάτων SOAP. Η WSDL είναι ένα ακόμα παράδειγμα. Έτσι, οι περισσότερες από τις προδιαγραφές των WS καθορίζουν μία ή περισσότερες γλώσσες XML.[6]

2.6 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Η MySQL είναι δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως το YouTube, Facebook και Twitter. [5]



Τα δεδομένα σε μια MySQL βάση δεδομένων αποθηκεύονται σε πίνακες. Ένας πίνακας είναι μια συλλογή από σχετικά δεδομένα και αποτελείται από στήλες και γραμμές. Για παράδειγμα μια πολύ απλή βάση δεδομένων με πωλήσεις, μπορεί να έχει τρεις πίνακες για την αποθήκευση δεδομένων σχετικά με τα προϊόντα, τους πελάτες και τις παραγγελίες.

Η δημιουργία της βάσης γίνεται με την εντολή:

CREATE DATABASE database_name;

Η δημιουργία των πινάκων με την εντολή:

***CREATE TABLE database_name.table_name
(column_1 data_type_1,
column_2 data_type_2, ...
[PRIMARY KEY (columns)]
);***

Όταν γίνεται προσθήκη μιας εγγραφής δεδομένων σε έναν πίνακα (table), το αρχείο αποθηκεύεται ως μια γραμμή του πίνακα. Ένα προϊόν έχει πεδία, για παράδειγμα θα μπορούσε να έχει τέσσερα πεδία: product_code, name, value, launch_date. [6]

Η εισαγωγή γίνεται ως εξής:

***INSERT INTO table
(column_1,column_2,...)
VALUES (value_1, value_2, ...) ;***

Η MySQL υποστηρίζει πολλούς διαφορετικούς τύπους δεδομένων. Ο όρος τύπος δεδομένων (data type), αναφέρεται στην κατηγοριοποίηση των δεδομένων με βάση την τιμή που παίρνουν, το σύνολο των πράξεων που μπορούν να εκτελεστούν και τις απαιτήσεις αποθήκευσης. [13]

A) Αριθμητικοί Τύποι

| Data Type | Storage Used (Bytes) | Min. Signed | Max. Signed | Min. Unsigned | Max. Unsigned |
|-----------|----------------------|----------------------------|---------------------------|---------------|----------------------------|
| TINYINT | 1 | -128 127 | 127 | 0 | 255 |
| SMALLINT | 2 | -32,768 | 32,767 | 0 | 65,535 |
| MEDIUMINT | 3 | -8,388,608 | 8,388,607 | 0 | 6,777,215 |
| INTEGER | 4 | -2,147,483,648 | 2,147,483,647 | 0 | 4,294,967,295 |
| BIGINT | 8 | -9,223,372,036,854,775,808 | 9,223,372,036,854,775,807 | 0 | 18,446,744,073,709,551,615 |

B) Αλφαριθμητικοί Τύποι

| Data Type | Storage Used (Bytes) | Maximum |
|-----------------------|--|---------------------|
| CHAR(<i>m</i>) | <i>m</i> × maximum-size character in the character set | 255 chars |
| VARCHAR(<i>m</i>) | up to 2 bytes + <i>m</i> × maximum-size character in the character set | 65,535 chars |
| TEXT | size of string in bytes + 2 | 65,535 chars |
| TINYTEXT | size of string in bytes + 1 | 255 chars |
| MEDIUMTEXT | size of string in bytes + 3 | 16,777,215 chars |
| LONGTEXT | size of string in bytes + 4 | 4,294,967,295 chars |
| BINARY(<i>m</i>) | <i>m</i> | 255 bytes |
| VARBINARY(<i>m</i>) | up to 2 bytes + <i>m</i> | 65,535 bytes |
| BLOB | size of string in bytes + 2 | 65,535 bytes |
| TINYBLOB | size of string in bytes + 1 | 255 bytes |
| MEDIUMBLOB | size of string in bytes + 3 | 16,777,215 bytes |
| LOB | size of string in bytes + 4 | 4,294,967,295 bytes |
| ENUM | up to 2 bytes | 65,535 values |
| SET | up to 8 bytes | 64 members |

Επίσης η MySQL ορίζει κάποιους περιορισμούς (constraints), που έχουν σκοπό την διαφύλαξη της ακεραιότητας των δεδομένων με βάση κάποιους κανόνες. Μετά από κάθε ενημέρωση, ελέγχει εάν τα δεδομένα συμμορφώνονται στους σχετικούς περιορισμούς που έχουν οριστεί κατά την δημιουργία της βάσης.[17]

- **Πρωτεύον Κλειδί (Primary Key)**

Ως πρωτεύον κλειδί ορίζεται μια στήλη ή μια ομάδα στηλών ενός πίνακα του οποίου οι τιμές είναι πάντα μοναδικές. Κενές τιμές (null) δεν επιτρέπονται σε στήλες που αποτελούν μέρος ενός πρωτεύοντος κλειδιού.

- **Ξένο Κλειδί (Foreign Key)**

Το ξένο κλειδί σε ένα πίνακα συνδέεται με το πρωτεύον κλειδί σε έναν άλλο πίνακα. Το ξένο κλειδί χρησιμοποιείται για να αποτρέψει ενέργειες οι οποίες θα μπορούσαν να καταστρέψουν τις συνδέσεις μεταξύ πινάκων. [16]

2.7 SDMX

Η SDMX- Statistical Data and Metadata Exchange επιχορηγείται από επτά ιδρύματα(BIS, ECB, Eurostat, IMF, OECD, UN και World Bank), προκειμένου να παρέχει πρότυπα για την ανταλλαγή στατιστικών πληροφοριών. Αυτό το πρότυπο περιγράφει και οικουμενοποιεί τον τρόπο για την ανταλλαγή στατιστικών στοιχείων και προβλέπει ένα τυποποιημένο τρόπο μορφοποίησης για τα δεδομένα, τα μεταδεδομένα, τις κατευθυντήριες γραμμές όπως για την αρχιτεκτονική του συστήματος για την ανταλλαγή δεδομένων και μεταδεδομένων.

Ο τρόπος αναπαράστασης ενός SDMX μηνύματος έχει δυο βασικές εκφράσεις, SDMX-ML (χρησιμοποιώντας XML) και SDMX-EDI (χρησιμοποιώντας EDIFACT).

2.7.1 Ορισμοί SDMX

Μερικοί ορισμοί που χρησιμοποιούνται σε SDMX παρουσιάζονται παρακάτω,

- **Σύνολο Δεδομένων – Dataset:** Μια συλλογή σχετικών δεδομένων, που οργανώνονται σύμφωνα με μια προκαθορισμένη δομή.
- **Ορισμός Δομής Δεδομένων – Data Structure Definition (DSD):** Μεταδεδομένα που περιγράφουν τη δομή και την οργάνωση ενός συνόλου δεδομένων, τις στατιστικές έννοιες και την σύνδεση τους με έναν κατάλογο κωδικών που χρησιμοποιούνται μέσα στο dataset.
- **Διαστάσεις – Dimensions:** Έννοιες που καθορίζουν τη φυσική δομή του dataset.
- **Κωδικός Καταλόγου – Code lists:** Ένας κατάλογος κωδικών είναι μια προκαθορισμένη λίστα από την οποία κάποιοι στατιστικοί κωδικοί παίρνουν την τιμή τους. Κάθε κωδικός της λίστας έχει τις ακόλουθες ιδιότητες: α) αναγνωριστικό, β) όνομα, γ)περιγραφή και

δ)μήκος της τιμής του κωδικού.

- **Χαρακτηριστικά – Attributes:** Δίνουν περισσότερες πληροφορίες σχετικά με τις έννοιες που χρησιμοποιούνται και δεν επηρεάζουν την δομή του dataset.
- **Ροή Δεδομένων – Dataflow:** Μια δομή που περιγράφει, κατηγοριοποιεί και περιορίζει την επιτρεπόμενη περιεκτικότητα του συνόλου δεδομένων.
- **Σχέδια Εννοιών – Concept Schemes:** Οι περιγραφικές πληροφορίες για την σύνθεση ή την διαίρεση των εννοιών σε ομάδες με βάση τα χαρακτηριστικά, που έχουν ως κοινό τα αντικείμενα. Ένα σχέδιο εννοιών διατηρεί καταλόγους των εννοιών που χρησιμοποιούνται στην οικογένεια των κλειδιών και ορισμούς της δομής των μεταδεδομένων.[11]

Κεφάλαιο 3: Βάση Δεδομένων

Στο τρίτο κεφάλαιο γίνεται ανάλυση και σχεδίαση της βάσης δεδομένων. Παρουσιάζεται αναλυτικά η δομή και το περιεχόμενο της βάσης, καθώς επίσης και η σύνδεση της MySQL βάσης με την εφαρμογή. Τέλος επεξηγείται η αξία και το είδος των δεδομένων που χρησιμοποιήθηκαν στην παρούσα πτυχιακή από την Eurostat.

3.1 Σκοπός Eurostat

Η Ευρωπαϊκή Στατιστική Υπηρεσία (Eurostat) είναι υπηρεσία της Ευρωπαϊκής Επιτροπής και κύριος σκοπός της είναι η συλλογή και δημοσίευση στατιστικών στοιχείων που αφορούν τις χώρες της Ευρωπαϊκής Ένωσης. Η αποστολή της είναι η θεμελίωση της ως ο κορυφαίος πάροχος στατιστικών δεδομένων υψηλής ποιότητας στην Ευρώπη.

Η Eurostat ιδρύθηκε το 1953 για να καλύψει τις απαιτήσεις της Κοινότητας Άνθρακα και Χάλυβα. Με την πάροδο των χρόνων το έργο της έχει διευρυνθεί και όταν ιδρύθηκε το 1958 η Ευρωπαϊκή Κοινότητα μετατράπηκε σε Γενική Διεύθυνση της Ευρωπαϊκής Επιτροπής. Σήμερα η Eurostat αποτελεί μέρος του χαρτοφυλακίου της Marianne Thyssen, Επίτροπος για την Απασχόληση, τις Κοινωνικές Υποθέσεις, Δεξιότητες και την Κινητικότητα Εργατικού Δυναμικού. Βασικός ρόλος της ΕΕ είναι η παροχή στατιστικών στοιχείων σε άλλες Γενικές Διευθύνσεις, στην Ευρωπαϊκή Επιτροπή και σε άλλα θεσμικά όργανα ώστε με την αξιοποίηση αυτών των δεδομένων να μπορούν να καθορίζουν, να εφαρμόζουν και να αναλύουν τις κοινοτικές πολιτικές.

Οι δημοκρατικές κοινωνίες δεν μπορούν να λειτουργήσουν σωστά χωρίς μια βάση από αξιόπιστες και αντικειμενικές στατιστικές. Από τη μία πλευρά, οι υπεύθυνοι φορείς που λαμβάνουν αποφάσεις σε επίπεδο ΕΕ, τα κράτη μέλη, οι τοπικές αυτοδιοικήσεις των χωρών και οι επιχειρήσεις χρειάζονται τα στατιστικά δεδομένα για να πάρουν τις κατάλληλες αποφάσεις. Από την άλλη πλευρά, το κοινό και τα μέσα ενημέρωσης χρειάζονται τα στατιστικά στοιχεία ώστε να διαμορφώσουν εικόνα για την σύγχρονη κοινωνία και να μπορούν να αξιολογήσουν την απόδοση των πολιτικών και όχι μόνο. Φυσικά, τα εθνικά στατιστικά στοιχεία εξακολουθούν να είναι σημαντικά για εθνικούς σκοπούς στα κράτη μέλη της ΕΕ, ενώ τα ευρωπαϊκά στατιστικά δεδομένα είναι απαραίτητα για τη λήψη αποφάσεων και την αξιολόγηση σε ευρωπαϊκό επίπεδο.

Τέλος η σημασία των δεδομένων που προσφέρει η Eurostat σήμερα είναι πολύ πιο σημαντική από ότι πριν από δέκα χρόνια. Με την ανάπτυξη των κοινοτικών πολιτικών ο ρόλος της Eurostat έχει αλλάξει. Σήμερα η συλλογή δεδομένων για την ΟΝΕ και την ανάπτυξη στατιστικών συστημάτων στις υποψήφιες χώρες για την ένταξη στην ΕΕ καθιστά την Eurostat μια υψίστης σημασίας υπηρεσία.

3.2 Κατηγορίες δεδομένων

Η κεντρική βάση δεδομένων της Eurostat αποτελείται από ορισμένες βασικές κατηγορίες, οι οποίες περιέχουν πολλαπλούς υποπίνακες. Οι βασικές κατηγορίες είναι οι εξής,

- Γενικές και Περιφερειακές στατιστικές/πολιτικές της ΕΕ

- Οικονομία και Χρηματοοικονομικά
- Πληθυσμός και κοινωνικές συνθήκες
- Βιομηχανία και υπηρεσίες
- Γεωργία, δασοκομία και αλιεία
- Διεθνές εμπόριο
- Μεταφορές
- Περιβάλλον και ενέργεια
- Καινοτομία και κοινωνία της πληροφορίας

Τα δεδομένα που περιέχονται στους πίνακες της Eurostat, δεν συλλέγονται από την ίδια την υπηρεσία αλλά από τις στατιστικές αρχές των κρατών μελών. Αφού ελέγξει και αναλύσει τα δεδομένα η εκάστοτε εθνική υπηρεσία τα στέλνει στην Eurostat. Ο ρόλος της Eurostat στη συλλογή των δεδομένων είναι η συγκέντρωση των στοιχείων και η διασφάλιση ότι είναι συγκρίσιμα, χρησιμοποιώντας μια εναρμονισμένη μεθοδολογία.

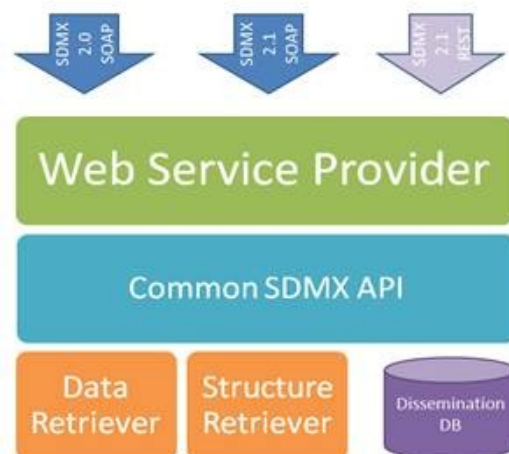
3.2.1 Web Services Eurostat

Η πρόσβαση στα δεδομένα της Eurostat, μπορεί να πραγματοποιηθεί με πολλαπλούς τρόπους. Ο πρώτος τρόπος ονομάζεται συγκεντρωτική αποθήκευση δεδομένων (bulk download). Η Eurostat δίνει την δυνατότητα αποθήκευσης μεγάλων datasets σε μορφή tsv (tab separated values), dft και sdmx. Με τον τρόπο αυτό παρέχονται επίσης οδηγίες για το πως μπορεί ο χρήστης να κάνει αυτοματοποιημένα το “κατέβασμα” των δεδομένων, ένας οδηγός που περιέχει λεπτομερείς πληροφορίες σχετικά με την υπηρεσία, τον πίνακα με όλα τα δεδομένα και τέλος ένα λεξιλόγιο όλων των κωδικών που υπάρχουν στο dataset. Ο άλλος τρόπος, ο οποίος χρησιμοποιήθηκε και στην παρούσα πτυχιακή, έχει σαν βάση τα SDMX Web Services.

Τα SDMX Web Service δίνουν προγραμματιστική πρόσβαση στα δεδομένα της Eurostat, παρέχοντας τις εξής δυνατότητες:

- να πάρει ο χρήστης μια ολοκληρωμένη λίστα με το σύνολο των διαθέσιμων σύνολων δεδομένων
- να πάρει έναν λεπτομερή ορισμό την δομής ενός συνόλου δεδομένων
- να “κατεβάσει” ένα υποσύνολο του συνόλου δεδομένων ή και ολόκληρο το σύνολο.

Τα SDMX WS υποστηρίζουν και το REST - Representation State Transfer και το SOAP - Simple Object Access Protocol. Τα δεδομένα υποστηρίζουν μόνο την αγγλική γλώσσα. Ένα από τα σημαντικά πλεονεκτήματα της υπηρεσίας είναι ότι δεν υπάρχει κόστος χρήσης.



Στην παρούσα πτυχιακή εργασία χρησιμοποιήθηκε το SOAP πρωτόκολλο και η μορφή των επιστρεφόμενων μηνυμάτων επιλέχθηκε να είναι σε SDMX 2.1.

3.3 Ανάλυση και Υλοποίηση Βάσης

Η αποθήκευση των δεδομένων στο πρόγραμμα πραγματοποιήθηκε με την χρήση της υπηρεσίας των Web Services που προσφέρει η Eurostat. Προτού όμως γίνει το “κατέβασμα” και η αποθήκευση των δεδομένων υπάρχουν κάποια βασικά βήματα, τα οποία αναλύονται στις επόμενες ενότητες.

3.3.1 Εύρεση Dataset

Όπως αναφέρθηκε και σε προηγούμενη ενότητα, η Eurostat παρέχει πολύ μεγάλο θεματικό εύρος δεδομένων. Ιδιαίτερη όμως είναι η συμβολή της σε οικονομικό επίπεδο. Τα δεδομένα που πραγματεύεται η συγκεκριμένη πτυχιακή αφορούν το ακαθάριστο εγχώριο προϊόν (ή ΑΕΠ) (αγγλ. *Gross Domestic Product – GDP*) όλων των περιφερειών των χωρών της Ευρωπαϊκής Ένωσης. Το ΑΕΠ είναι το σύνολο όλων των προϊόντων και αγαθών που παράγει μια οικονομία, εκφρασμένο σε χρηματικές μονάδες. Με άλλα λόγια είναι η συνολική αξία όλων των τελικών αγαθών (υλικών και άυλων) που παρήχθησαν εντός μιας χώρας σε διάστημα ενός έτους, ακόμα και αν μέρος αυτού παρήχθη από παραγωγικές μονάδες που ανήκουν σε κατοίκους του εξωτερικού. [14]

Η εύρεση του κατάλληλου dataset, έγινε με την βοήθεια του navigation tree, που προσφέρει η eurostat στο site της.

DATABASE

Your search has matched 1 node.

Here > **Data Navigation Tree**

Here > **Database by themes**

Here > **General and regional statistics**

Here > **European and national indicators for short-term analysis (euroind)** M

Here > **Regional statistics by NUTS classification (reg)**

Here > **Regional agriculture statistics (reg_agr)**

Here > **Regional demographic statistics (reg_dem)**

Here > **Regional economic accounts - ESA2010 (reg_eco10)**

Here > **Regional economic accounts - ESA95 (reg_eco)**

Here > **Gross domestic product indicators - ESA95 (reg_ecogdp)**

Here > **Gross domestic product (GDP) at current market prices by NUTS 2 regions (nama_r_e2gdp)** M

Here > **Gross domestic product (GDP) at current market prices by NUTS 3 regions (nama_r_e3gdp)** M

Here > **Real growth rate of regional gross value added (GVA) at basic prices by NUTS 2 regions - percentage change on previous year (nama_r_e2grgdp)** M

Here > **Dispersion of regional GDP by NUTS 3 regions (%) (nama_r_e0digdp)** M i

Here > **Average annual population to calculate regional GDP data (thousand persons), by NUTS 3 regions (nama_r_e3popgdp)** M

Here > **Household accounts - ESA95 (reg_ecohh)** M

Here > **Regional education statistics (reg_educ)** M

Όπως φαίνεται και στην εικόνα τα data που επιθυμούμε σχετικά με το ΑΕΠ (αγγλ GDP), βρίσκονται στον υποφάκελλο Gross domestic product indicators-ESA95. Το κατάλληλο dataset βρίσκεται στον εσωτερικό φάκελο, με όνομα Gross domestic product (GDP) at current market prices by NUTS 3 regions (nama_r_e3gdp).

Gross domestic product (GDP) at current market prices by NUTS 3 regions [nama_r_e3gdp]

Last update: 28-02-2014

Table Customization [show](#)

TIME: [] GEO: [] Unit of measure: Euro per inhabitant

| | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 |
|---|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|------|
| European Union (28 countries) | 19,000 | 19,700 | 20,400 | 20,600 | 21,600 | 22,400 | 23,600 | 24,900 | 25,000 | 23,400 | |
| European Union (27 countries) | 19,100 | 19,800 | 20,500 | 20,800 | 21,700 | 22,500 | 23,700 | 25,100 | 25,100 | 23,500 | |
| Belgium | 24,600 | 25,300 | 26,000 | 26,600 | 28,000 | 29,000 | 30,200 | 31,600 | 32,400 | 31,600 | |
| Bulgaria | 1,700 | 2,000 | 2,200 | 2,400 | 2,600 | 3,000 | 3,400 | 4,000 | 4,600 | 4,600 | |
| Czech Republic | 6,200 | 7,000 | 8,200 | 8,300 | 9,000 | 10,200 | 11,500 | 12,800 | 14,800 | 13,600 | |
| Denmark | 32,500 | 33,500 | 34,400 | 35,000 | 36,500 | 38,300 | 40,200 | 41,700 | 42,800 | 40,500 | |
| Germany (until 1990 former territories) | 24,900 | 25,500 | 25,900 | 26,000 | 26,600 | 27,000 | 28,100 | 29,500 | 30,100 | 29,000 | |
| Estonia | 4,500 | 5,100 | 5,700 | 6,400 | 7,200 | 8,300 | 10,000 | 12,000 | 12,100 | 10,400 | |
| Ireland | 27,800 | 30,400 | 33,200 | 35,200 | 36,900 | 39,200 | 41,600 | 43,100 | 40,100 | 35,800 | |
| Greece | 12,600 | 13,400 | 14,300 | 15,600 | 16,700 | 17,400 | 18,700 | 19,900 | 20,800 | 20,500 | |
| Spain | 15,600 | 16,700 | 17,700 | 18,600 | 19,700 | 21,000 | 22,400 | 23,500 | 23,900 | 22,800 | |
| France | 23,700 | 24,500 | 25,000 | 25,600 | 26,500 | 27,300 | 28,400 | 29,600 | 30,100 | 29,300 | |
| Croatia | 5,300 | 5,800 | 6,300 | 6,800 | 7,400 | 8,100 | 8,900 | 9,800 | 10,700 | 10,100 | |
| Italy | 21,000 | 22,000 | 22,800 | 23,300 | 24,000 | 24,900 | 25,300 | 26,200 | 26,300 | 25,200 | |
| Cyprus | 14,300 | 15,300 | 15,600 | 16,200 | 17,300 | 18,400 | 19,500 | 20,700 | 21,800 | 20,900 | |
| Latvia | 3,600 | 3,900 | 4,200 | 4,300 | 4,900 | 5,800 | 7,200 | 8,600 | 10,500 | 8,600 | |
| Lithuania | 3,600 | 3,900 | 4,400 | 4,900 | 5,400 | 6,300 | 7,400 | 8,900 | 10,100 | 8,400 | |
| Luxembourg | 50,300 | 51,100 | 53,700 | 57,100 | 59,900 | 65,000 | 71,700 | 78,000 | 76,400 | 71,400 | |
| Hungary | 4,900 | 5,800 | 6,900 | 7,300 | 8,100 | 8,800 | 9,900 | 9,900 | 10,500 | 9,100 | |
| Malta | 11,200 | 11,600 | 11,800 | 11,600 | 11,600 | 12,200 | 12,800 | 13,700 | 14,600 | 14,400 | |
| Netherlands | 26,300 | 27,900 | 28,800 | 29,400 | 30,200 | 31,500 | 33,100 | 34,900 | 36,200 | 34,700 | |
| Austria | 26,000 | 26,600 | 27,300 | 27,700 | 28,700 | 29,800 | 31,300 | 33,000 | 34,000 | 33,100 | |

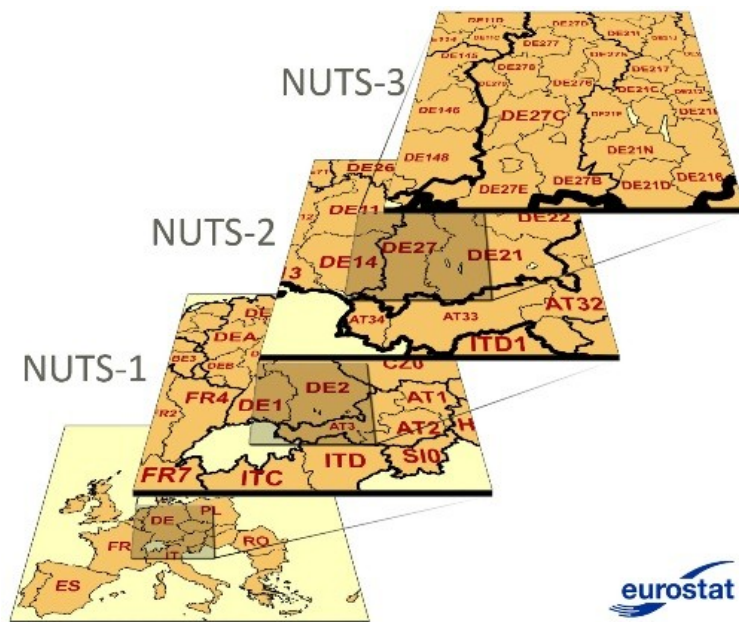
Ο πίνακας δίνει μια αναπαράσταση των δεδομένων GDP για κάθε χώρα και κάθε χρονιά.

3.3.2 Ανάλυση δεδομένων

Μετά την εύρεση του κατάλληλου dataset, σειρά έχει η ανάλυση των δεδομένων προκειμένου να σχεδιαστούν οι πίνακες της βάσης και τα αντίστοιχα πεδία τους.

Στο dataset όπως είναι λογικό ένα πεδίο περιέχει τα ονόματα των χωρών της ΕΕ. Η Eurostat χρησιμοποιεί για την κατηγοριοποίηση (classification) των χωρών της ΕΕ, το σύστημα NUTS. Το NUTS (Nomenclature of territorial units for statistics) είναι ένα ιεραρχικό σύστημα που διαιρεί ολόκληρη την επικράτεια της ΕΕ σε υποκατηγορίες. Χωρίζεται σε τρεις κατηγορίες:

1. NUTS 1: μεγάλες κοινωνικό-οικονομικές περιφέρειες
2. NUTS 2: τις βασικές περιοχές για την εφαρμογή των περιφερειακών πολιτικών
3. NUTS 3: μικρές περιοχές για συγκεκριμένες διαγνώσεις



Το σύνολο των χωρών της ΕΕ είναι 28. Το πρώτο επίπεδο (NUTS1) περιλαμβάνει 98 περιφέρειες, το δεύτερο επίπεδο περιλαμβάνει 276 περιφέρειες και το NUTS3 και περιλαμβάνει 1,342 περιφέρειες όλων των χωρών της Ε.Ε. Τα δεδομένα που θέλουμε αφορούν τις περιφέρειες σε επίπεδο NUTS3. Προφανώς οι αριθμοί αυτοί είναι μεταβαλλόμενοι λόγω της εισόδου νέων χωρών στην ΕΕ. Εκτός από τα τρία επίπεδα NUTS υπάρχουν και άλλα δυο, τα οποία αφορούν τις τοπικές διοικητικές μονάδες (Local Administrative Units-LAUs). Τα δεδομένα που χρησιμοποιήθηκαν για την εξαγωγή των τελικών συμπερασμάτων αφορούν τις περιφέρειες των χωρών της ΕΕ σε επίπεδο NUTS3.

Τα δεδομένα που μας παρέχει η Eurostat όπως προαναφέρθηκε είναι ιδιαίτερα χρήσιμα διότι μας δείχνουν τις αλλαγές των τιμών με την πάροδο των χρόνων. Στην αξιολόγηση των οικονομικών δεδομένων και ιδιαίτερα του ΑΕΠ, η παρατήρηση της μεταβολής των τιμών με την πάροδο των χρόνων βοηθά στην καλύτερη ανάλυση των δεδομένων. Τα δεδομένα που

αποθηκεύσαμε στη παρούσα εφαρμογή καλύπτουν το χρονικό εύρος από την χρονιά 2000 έως και την χρονιά 2010.

Εκτός από τον χρόνο και τον γεωγραφικό κωδικό (NUTS) που αναλύθηκε προηγουμένως, στη βάση δεδομένων αποθηκεύονται και κάποια άλλα στοιχεία. Υπάρχουν ακόμη δυο πεδία το Unit και το Freq. Το Unit είναι ένας δείκτης, που μας δείχνει με βάση ποια μονάδα μετράται η τιμή του ΑΕΠ. Έχει σταθερά την τιμή EUR_HAB (Euro per inhabitant). Ο άλλος δείκτης FREQ μας δείχνει την συχνότητα των μετρήσεων και έχει πολλαπλές επιλογές όπως είναι καθημερινά, μηνιαία, ετήσια. Εμείς θέλουμε τα δεδομένα να είναι ετήσια και για αυτό το λόγο χρησιμοποιούμε στο πεδίο FREQ σταθερά την τιμή “Α”.

3.3.3 Σχεδίαση Βάσης Δεδομένων

Η διαχείριση της βάσης τύπου MySQL η οποία είναι εγκατεστημένη στον database server της σχολής, πραγματοποιήθηκε με την βοήθεια του περιβάλλοντος του phpMyAdmin. Το phpMyAdmin είναι ένα δωρεάν εργαλείο λογισμικού γραμμένο σε PHP που προορίζεται για να χειριστεί την διαχείριση της MySQL, μέσω του World Wide Web. Το phpMyAdmin υποστηρίζει ένα ευρύ φάσμα δραστηριοτήτων με τη MySQL. Οι πιο συχνά χρησιμοποιούμενες λειτουργίες που υποστηρίζονται από το περιβάλλον εργασίας χρήστη (διαχείριση βάσεων δεδομένων, πινάκων, πεδίων, σχέσεων, ευρετηρίων κλπ), επίσης παρέχει τη δυνατότητα εκτέλεσης άμεσα οποιασδήποτε SQL δήλωσης. [12]

Για την αποθήκευση των δεδομένων δημιουργήθηκαν δυο πίνακες. Ο ένας πίνακας με όνομα GDP_Basic_Table είναι ο βασικός που έχει το μεγαλύτερο όγκο δεδομένων. Ο δεύτερος πίνακας με όνομα Geo_Nuts είναι βοηθητικός.

A) GDP_Basic_Table

Ο πίνακας αυτός περιέχει το μεγαλύτερο πλήθος των δεδομένων. Σε αυτόν βρίσκεται όλη η πληροφορία που χρειάζεται, για την εξαγωγή των συμπερασμάτων.

Τα πεδία του είναι τα εξής:

- Year: Η χρονιά στην οποία αναφέρεται η εγγραφή. Ο τύπος του πεδίου είναι καθαρός αριθμός και έχει δηλωθεί ως int (Integer).
- Value: Η τιμή του ΑΕΠ - Ακαθάριστο Εγχώριο Προϊόν. Είναι επίσης καθαρός αριθμός όμως σε κάποιες εγγραφές που δεν υπάρχει διαθέσιμη τιμή από την Eurostat, υπάρχει η λέξη “NAN”, εξαιτίας αυτής της περίπτωσης, δηλώνεται ως Varchar.
- Freq: Είναι σταθερά και έχει την τιμή “Α”. Δηλώνεται επίσης ως Varchar.
- Unit: Είναι σταθερά και έχει την τιμή “EUR_HAB”. Δηλώνεται ως Varchar.
- Nuts_Code: Περιέχει τον κωδικό σύμφωνα με τη λίστα NUTS 3, που αντιστοιχεί στην περιφέρεια μιας χώρας.

Εκτός από τα πεδία, ορίζεται πρωτεύον κλειδί στο πεδίο Nuts_Code.

B) Geo_Nuts

Ο δεύτερος πίνακας είναι ένας παραμετρικός πίνακας που λειτουργεί βοηθητικά στον πρώτο. Όπως είναι λογικό ο χρήστης δεν μπορεί να είναι σε θέση να γνωρίζει τις αντιστοιχίες των κωδικών της κατηγοριοποίησης NUTS 3 με τις περιφέρειες των χωρών. Για αυτό το λόγο δημιουργήθηκε ο πίνακας Geo_Nuts.


























Τα πεδία του είναι τα εξής:






- Nuts_Code: Περιέχει τον κωδικό σύμφωνα με τη λίστα NUTS 3, που αντιστοιχεί στην περιφέρεια μιας χώρας. Δηλώνεται ως Varchar.
- Nuts_Country: Το όνομα της χώρας στην οποία ανήκει η περιφέρεια. Δηλώνεται ως Varchar.
- Nuts_Name: Το πλήρες όνομα της εκάστοτε περιφέρειας. Δηλώνεται ως Varchar.




Εκτός από τα πεδία, ορίζεται πρωτεύον κλειδί στο πεδίο Nuts_Code.

Προκειμένου να συσχετίσουμε τους δυο πίνακες χρησιμοποιήθηκαν ξένα κλειδιά. Το ξένο κλειδί είναι το πεδίο Nuts_Code στον πίνακα GDP_Basic_Table αναφερόμενο στον πίνακα Geo_Nuts.


Πίνακας GDP_Basic_Table



| Πεδίο | Τύπος | Σύνθεση | Χαρακτηριστικά | Κενό | Προκαθορισμένο | Πρόσθετα | Ενέργεια |
|------------------------------------|-------------|-----------------|----------------|------|----------------|----------|---|
| <input type="checkbox"/> Value | int(50) | | | Όχι | Κανένα | |      |
| <input type="checkbox"/> Year | int(50) | | | Όχι | Κανένα | |      |
| <input type="checkbox"/> Unit | varchar(50) | utf8_unicode_ci | | Όχι | EUR_HAB | |      |
| <input type="checkbox"/> Freq | varchar(50) | utf8_unicode_ci | | Όχι | A | |      |
| <input type="checkbox"/> Nuts_Code | varchar(50) | utf8_unicode_ci | | Όχι | Κανένα | |      |

Επιλογή όλων / Απεπιλογή όλων Με τους επιλεγμένους:     
















Εμφάνιση για εκτύπωση  Εμφάνιση συσχετίσεων  Προτεινόμενη δομή πίνακα  Παρακολούθηση πίνακα






Προσθήκη 1 πεδίου(ων) ☒ Στο τέλος του Πίνακα ☐ Στην αρχή του Πίνακα ☐ Μετά το Value




Ευρετήρια: 

| Ενέργεια | Όνομα κλειδιού | Τύπος | Μοναδικό | Συμπιεσμένο | Πεδίο | Μοναδικότητα | Σύνθεση | Κενό | Σχόλιο |
|---|----------------|-------|----------|-------------|-----------|--------------|---------|------|--------|
|   | PRIMARY | BTREE | Ναι | Όχι | Nuts_Code | 0 | A | | |


Πίνακας Geo_Nuts



| Πεδίο | Τύπος | Σύνθεση | Χαρακτηριστικά | Κενό | Προκαθορισμένο | Πρόσθετα | Ενέργεια |
|---------------------------------------|-------------|-----------------|----------------|------|----------------|----------|---|
| <input type="checkbox"/> Nuts_Code | varchar(50) | utf8_unicode_ci | | Όχι | Κανένα | |      |
| <input type="checkbox"/> Nuts_Country | varchar(50) | utf8_unicode_ci | | Όχι | Κανένα | |      |
| <input type="checkbox"/> Nuts_Name | varchar(50) | utf8_unicode_ci | | Όχι | Κανένα | |      |

Επιλογή όλων / Απεπιλογή όλων Με τους επιλεγμένους:     

Εμφάνιση για εκτύπωση  Εμφάνιση συσχετίσεων  Προτεινόμενη δομή πίνακα  Παρακολούθηση πίνακα

Προσθήκη 1 πεδίου(ων) ☒ Στο τέλος του Πίνακα ☐ Στην αρχή του Πίνακα ☐ Μετά το Nuts_Code

Ευρετήρια: 

| Ενέργεια | Όνομα κλειδιού | Τύπος | Μοναδικό | Συμπιεσμένο | Πεδίο | Μοναδικότητα | Σύνθεση | Κενό | Σχόλιο |
|---|----------------|-------|----------|-------------|-----------|--------------|---------|------|--------|
|   | PRIMARY | BTREE | Ναι | Όχι | Nuts_Code | 0 | A | | |

Συσχέτιση Πινάκων



Στην εικόνα φαίνεται η σχέση των δυο πινάκων και η αλληλεξάρτηση με την χρήση ξένων κλειδιών.

3.4 Σύνδεση Βάσης

Στο κομμάτι αυτό αναλύεται η σύνδεση της εφαρμογής με την MySQL βάση που σχεδιάστηκε προηγουμένως. Αρχικά δημιουργήθηκε το αρχείο config.ini το οποίο περιέχει όλα τα στοιχεία που χρειάζονται για την πραγματοποίηση της σύνδεσης με την βάση. Παρακάτω φαίνεται ο κώδικας που περιέχει το αρχείο.

```
[database]
server = 83.212.105.20
port = 3306
databaseName = it21121
user = it21121
password = *****
```

Τα απαραίτητα στοιχεία όπως φαίνεται και στον κώδικα είναι η διεύθυνση του server στον οποίο είναι καταχωρημένη η βάση, ο αριθμός της θύρας (port, το όνομα της βάσης, και τέλος το όνομα του χρήστη και ο κωδικός πρόσβασης.

Αρχικά πρέπει να περαστούν τα στοιχεία που βρίσκονται στο αρχείο config.ini μέσα στον κώδικα. Τα δεδομένα του αρχείου αποθηκεύονται σε τοπικές μεταβλητές.

```
Properties pro = new Properties();
InputStream is;
pro.load(new FileInputStream("database_config.ini"));
String port = pro.getProperty("port");
String server = pro.getProperty("server");
String database = pro.getProperty("databaseName");
String user = pro.getProperty("user");
```

```
String passwd = pro.getProperty("password");
```

Στη συνέχεια γίνεται η επικοινωνία του προγράμματος με την βάση χρησιμοποιώντας τον κατάλληλο οδηγό (driver). Οι παράμετροι της σύνδεσης δίνονται με το κατάλληλο url.

```
Class.forName("com.mysql.jdbc.Driver");  
String connectionUrl = "jdbc:mysql://" + server + ":" + port + "?" + database + "user=" +  
user + ";password=" + passwd + ";";  
Connection conn = DriverManager.getConnection(connectionUrl);
```

Κεφάλαιο 4: Αντικειμενοστρεφής Προγραμματισμός

Στο τέταρτο κεφάλαιο αναλύεται η έννοια του αντικειμενοστρεφή προγραμματισμού και επεξηγούνται κάποιες βασικές έννοιες. Ειδικότερα όμως παρουσιάζεται η γλώσσα Java, μια πολύ χρήσιμη αντικειμενοστρεφής γλώσσα προγραμματισμού. Τέλος αναλύονται ορισμένα βασικά Java APIs όπως το JDBC και το JAXP.

4.1 Ιστορία του Αντικειμενοστρεφή Προγραμματισμού

Ο αντικειμενοστρεφής προγραμματισμός (Object Oriented Programming – OOP), είναι ένας τρόπος προγραμματισμού με τον οποίο οι απαιτήσεις ενός προβλήματος σπάνε σε πιο μικρά και διαχειρίσιμα τμήματα σε σχέση με το συνολικό πρόβλημα. Κάθε κομμάτι είναι αυτόνομο αλλά μπορεί παράλληλα να συνεργάζεται με τα υπόλοιπα τμήματα του προγράμματος. [17]

Η έννοια του αντικειμενοστρεφή προγραμματισμού άρχισε να αναδύεται στα μέσα της δεκαετίας του 1960 με μια γλώσσα που ονομάζονταν Simula και στη συνέχεια εξελίχθηκε το 1970 με την έλευση της γλώσσας Smalltalk. Οι προγραμματιστές λογισμικού δεν αποδέχθηκαν τις αρχικές εκδόσεις των αντικειμενοστρεφών γλωσσών, παρόλα αυτά ο αντικειμενοστρεφής προγραμματισμός εξελίχθηκε.

Το 1980 υπήρξε μια αναζωπύρωση του ενδιαφέροντος για την αντικειμενοστρεφή μεθοδολογία. Συγκεκριμένα, αντικειμενοστρεφείς γλώσσες όπως C++ και Eiffel έγιναν πολύ δημοφιλείς και χρησιμοποιήθηκαν από προγραμματιστές υπολογιστών. Η δημοτικότητα αυτών των γλωσσών άρχισε να αυξάνεται στη δεκαετία του 90, κυρίως με την έλευση της Java.

Το 2002 σε συνδυασμό με την κυκλοφορία του .NET framework, η Microsoft εισήγαγε μια νέα γλώσσα, την C# (αντικειμενοστρεφής C) και ανανέωσε την ήδη υπάρχουσα διάσημη γλώσσα, Visual Basic, με αποτέλεσμα τώρα να είναι πλήρης αντικειμενοστρεφής. Σήμερα οι αντικειμενοστρεφείς γλώσσες συνεχίζουν να ακμάζουν και αποτελούν τη βάση του μοντέρνου προγραμματισμού. [18]

4.2 Βασικές Έννοιες

- **Αντικείμενο (Object):** Ένα αντικείμενο είναι μια δομή που χρησιμοποιείται για την ενσωμάτωση των δεδομένων και των διαδικασιών που χρειάζονται για την επεξεργασία των δεδομένων. Για παράδειγμα, για τη διαχείριση των δεδομένων ενός e-shop, πρέπει αρχικά να σχεδιαστεί ένα αντικείμενο με όνομα Προϊόν, το οποίο είναι υπεύθυνο για την διατήρηση και χρήση των δεδομένων που αφορούν τα προϊόντα. Για την χρήση της δυνατότητας εκτύπωσης θα έπρεπε το αντικείμενο Προϊόν να συνεργαστεί με το αντικείμενο Εκτυπωτής, που είναι υπεύθυνο για τα δεδομένα και τις μεθόδους που χρησιμοποιούνται για την αλληλεπίδραση με τους εκτυπωτές.[18]
- **Ενθυλάκωση (Encapsulation):** Ενθυλάκωση είναι η διατήρηση της ασφάλειας της εσωτερικής λειτουργίας του κώδικα από τις παρεμβολές των προγραμμάτων που το χρησιμοποιούν. Επιτρέπει μόνο σε συγκεκριμένες μεταβλητές και μεθόδους να έχουν

πρόσβαση. Αυτό σημαίνει ότι ο κώδικας μπορεί να ενημερωθεί, επεκταθεί ή να βελτιωθεί χωρίς να επηρεάζονται τα προγράμματα που το χρησιμοποιούν, εφόσον τα εκτεθειμένα μέρη εξακολουθούν να είναι προσβάσιμα με τον ίδιο τρόπο.[17]

- **Κληρονομικότητα (Inheritance):** Κληρονομικότητα σημαίνει ότι μπορούν να αξιοποιηθούν όλες οι δυνατότητες και τα οφέλη της δουλειάς άλλων ανθρώπων, συμπεριλαμβανομένων της ενθυλάκωσης και του πολυμορφισμού από άλλα προγράμματα, ενώ προσαρμόζεται ο κώδικας στις απαιτήσεις του κυρίως προγράμματος. Στην πραγματικότητα, αυτό γίνεται κάθε φορά που χρησιμοποιείται η κωδική λέξη “extends”.[17]
- **Πολυμορφισμός (Polymorphism):** Είναι η ικανότητα δυο διαφορετικών αντικειμένων να ανταποκρίνονται στο ίδιο αιτούμενο μήνυμα με το δικό τους μοναδικό τρόπο.[18]
- **Αφαίρεση (Abstraction):** Είναι η λειτουργία όπου αντικείμενα αντιπροσωπεύουν πραγματικά αντικείμενα. Τα προγραμματιστικά αντικείμενα δεν χρειάζεται να έχουν όλα τα στοιχεία που έχουν τα πραγματικά αντικείμενα.[17]

4.3 Πλεονεκτήματα

Τα πλεονεκτήματα του αντικειμενοστρεφή προγραμματισμού σε σχέση με άλλες γλώσσες είναι η εύκολη συντήρηση του κώδικα, η επαναχρησιμοποίηση κώδικα και η επεκτασιμότητα. Τα πλεονεκτήματα αυτά παρουσιάζονται με περισσότερες λεπτομέρειες στην συνέχεια.

1. **Ευκολία Συντήρησης:** Οι σύγχρονες εφαρμογές αποτελούνται από πολύ μεγάλο πλήθος γραμμών κώδικα και η αλληλεξάρτηση των τμημάτων της εφαρμογής είναι πολύ μεγάλη. Οι αλλαγές σε ένα τμήμα της εφαρμογής επηρεάζουν και τα υπόλοιπα. Ο αντικειμενοστρεφής προγραμματισμός κάνει εύκολα τις εφαρμογές δομημένες και καθιστά την συντήρηση εύκολη.
2. **Επαναχρησιμοποίηση Κώδικα:** Επαναχρησιμοποίηση σημαίνει ότι ο κώδικας που έχει γραφτεί μπορεί να χρησιμοποιηθεί ξανά από τον ίδιο τον συντάκτη του κώδικα, αλλά και από άλλους που χρειάζονται την ίδια λειτουργικότητα που υπάρχει στον αρχικό κώδικα. Είναι λογικό μια αντικειμενοστρεφής γλώσσα συχνά να περιλαμβάνει ένα σύνολο έτοιμων προς χρήση βιβλιοθηκών. Στην περίπτωση της Java, η συγκεκριμένη γλώσσα συνοδεύεται από εκατοντάδες βιβλιοθήκες ή διασυνδέσεις προγραμματιστικών εφαρμογών (API) που έχουν σχεδιαστεί και δοκιμαστεί προσεκτικά. Είναι επίσης εύκολο να γράψει και να διανείμει ο χρήστης την δική του βιβλιοθήκη. Η υποστήριξη της επαναχρησιμοποίησης σε μια πλατφόρμα προγραμματισμού είναι πολύ ελκυστική, επειδή μειώνει το χρόνο ανάπτυξης.
3. **Επεκτασιμότητα:** Κάθε εφαρμογή είναι μοναδική. Έχει τις δικές της απαιτήσεις και προδιαγραφές. Όσον αφορά τη δυνατότητα επαναχρησιμοποίησης, μερικές φορές δεν μπορεί να βρεθεί μια κλάση που παρέχει την ακριβή λειτουργικότητα που απαιτεί η εφαρμογή. Επεκτασιμότητα σημαίνει ότι μπορούν να χρησιμοποιηθούν αυτές οι κλάσεις επεκτείνοντας τις (extend) ώστε να ταιριάζουν στις απαιτήσεις της εφαρμογής. Με αυτό τον τρόπο εξοικονομείται πολύς χρόνος διότι δεν χρειάζεται να γραφεί κώδικας από το μηδέν. Στον αντικειμενοστρεφή προγραμματισμό η επεκτασιμότητα επιτυγχάνεται μέσω της

κληρονομικότητας. [19]

4.4 Java

Η Java δεν είναι μόνο μια αντικειμενοστρεφής γλώσσα προγραμματισμού, είναι επίσης ένα σύνολο τεχνολογιών που συμβάλλουν στην ταχύτερη ανάπτυξη λογισμικού και προκύπτουν πολύ πιο ισχυρές και ασφαλείς εφαρμογές.[19]

4.4.1 Εκδόσεις της Java

Η Sun Microsystems παρουσίασε το 1995 την Java και σύντομα έγινε γνωστή ως η προγραμματιστική γλώσσα για την γραφή μικροεφαρμογών (applets), μικρών προγραμμάτων και μικρά προγράμματα που τρέχουν μέσα σε web browsers. Η ανάπτυξη του διαδικτύου συνέβαλλε πολύ στην πρόωρη επιτυχία της Java.

Το πιο ελκυστικό χαρακτηριστικό της Java ήταν η υπόσχεση της ότι θα υποστηρίζει την ανεξαρτησίας της από την πλατφόρμα (platform -independence) που τρέχει, εξ ου και το σύνθημα “Write once, Run anywhere”. Αυτό σημαίνει ότι το ίδιο πρόγραμμα θα μπορεί να τρέχει σε Windows, Linux, Mac, Unix και άλλα λειτουργικά συστήματα. Αυτό ήταν κάτι που καμία γλώσσα προγραμματισμού δεν μπορούσε να κάνει μέχρι τότε. Εκείνη την εποχή, οι C και C++ ήταν οι δυο πιο συχνά χρησιμοποιούμενες γλώσσες για την ανάπτυξη σοβαρών εφαρμογών. Η Java κατάφερε να κλέψει μέρος από την δύναμη τους.

Το 1997 κυκλοφόρησε η Java 1.1, προσθέτοντας σημαντικά χαρακτηριστικά όπως τα Java Beans. Το 1998 εκδόθηκε η Java 1.2, μετά από τρεις μέρες άλλαξε ο αριθμός έκδοσης σε 2, που σηματοδότησε την αρχή μιας τεράστιας εκστρατείας μάρκετινγκ με σκοπό να προωθήσει την Java ως μια τεχνολογία επόμενης γενιάς (next generation). Η Java 2 πουλήθηκε σε τέσσερις κατηγορίες: τη Standard Edition(J2SE), τη Enterprise Edition (J2EE) , τη Micro Edition (J2ME) και τη Java Card.

Τον Μάιο του 2007 η Sun απελευθέρωσε τον πηγαίο κώδικα της Java στην κοινότητα OpenJDK ως ελεύθερο λογισμικό. Η IBM, η Oracle και η Apple συμμετείχαν στην κοινότητα OpenJDK. Το 2010 η Oracle απέκτησε την Sun. Τον Ιούλιο του 2011 κυκλοφόρησε η Java 7 και το Μάρτη του 2014 κυκλοφόρησε η Java 8, και οι δυο εκδόσεις είναι αποτέλεσμα της συνεργασίας ανοιχτού κώδικα (open-source) μέσα από το OpenJDK.[19]

4.4.2 Java Platforms

Όλες οι Java πλατφόρμες αποτελούνται από μια Java Virtual Machine – Εικονική Μηχανή (JVM) και μια διεπαφή προγραμματισμού εφαρμογών (API). Το JVM είναι ένα πρόγραμμα, για μια συγκεκριμένη software και hardware πλατφόρμα, που τρέχει Java εφαρμογές. Ένα API είναι μια συλλογή από στοιχεία λογισμικού (software components) που μπορούν να χρησιμοποιηθούν ώστε να δημιουργηθούν άλλα στοιχεία λογισμικού ή εφαρμογές. Κάθε Java πλατφόρμα παρέχει μια εικονική μηχανή και ένα API, που επιτρέπουν στις εφαρμογές που έχουν γραφεί για αυτή τη πλατφόρμα να τρέχουν σε οποιοδήποτε συμβατό σύστημα με όλα τα πλεονεκτήματα της γλώσσας

προγραμματισμού Java.[20]

Java SE – Standard Edition: Το API της JAVA SE παρέχει τη βασική λειτουργικότητα της γλώσσας προγραμματισμού, Java. Ορίζει τα πάντα, από τα βασικά είδη και αντικείμενα της Java μέχρι υψηλού επιπέδου κλάσεις που χρησιμοποιούνται για τη δικτύωση, την ασφάλεια, τη πρόσβαση σε βάσεις δεδομένων, την ανάπτυξη γραφικών διεπαφών (GUI) και την ανάλυση της XML. Εκτός από το API, η πλατφόρμα Java SE αποτελείται από μια εικονική μηχανή, εργαλεία ανάπτυξης, βιβλιοθήκες και εργαλεία κλάσεων που χρησιμοποιούνται συνήθως σε εφαρμογές Java.

Java EE – Enterprise Edition: Η πλατφόρμα Java EE είναι κατασκευασμένη με βάση την πλατφόρμα Java SE. Η Java EE παρέχει ένα API και ένα runtime περιβάλλον για την ανάπτυξη και τη λειτουργία μεγάλων, πολυεπίπεδων, επεκτάσιμων, αξιόπιστων και ασφαλών δικτυακών εφαρμογών.

Java ME – Micro Edition: Η πλατφόρμα Java ME παρέχει ένα API και ένα μικρό μέρος μιας εικονικής μηχανής για το τρέξιμο εφαρμογών Java σε μικρές συσκευές όπως κινητά τηλέφωνα. Το API είναι ένα υποσύνολο του Java SE API, μαζί με ειδικές βιβλιοθήκες κλάσεων που είναι χρήσιμες για την ανάπτυξη εφαρμογών για μικρές συσκευές. [20]

4.5 Δομή της Java

4.5.1 Κλάση (Class)

Η κλάση είναι ένα θεμελιώδες δομικό στοιχείο των προγραμμάτων αντικειμενοστρεφούς προγραμματισμού. Κάθε κλάση αντιπροσωπεύει ένα πραγματικό αντικείμενο. Είναι ένα πρότυπο για την δήλωση και τη δημιουργία μιας κατηγορίας αντικειμένων. Μια ιδιότητα ενός αντικειμένου ορίζεται από ένα πεδίο, το οποίο είναι μια μεταβλητή που μπορεί να αποθηκεύσει την τιμή που αντιπροσωπεύει το συγκεκριμένο πεδίο. Η συμπεριφορά ενός αντικειμένου ορίζεται από μια μέθοδο. Τα πεδία και οι μέθοδοι της κλάσης είναι γνωστά ως τα μέλη της. Ο ορισμός μιας κλάσης στην Java αποτελείται από την δήλωση των μελών της (μεταβλητές και μέθοδοι) και αρχίζει με τη λέξη κλειδί “class”. [21]

```
class ClassA {  
    // members declarations  
}
```

4.5.2 Αντικείμενο (Object)

Ένα αντικείμενο είναι το στιγμιοτύπο μιας κλάσης. Ένα αντικείμενο πρέπει να δημιουργηθεί πριν χρησιμοποιηθεί μέσα σε ένα πρόγραμμα. Η διαδικασία δημιουργίας αντικειμένων από μια κλάση ονομάζεται στιγμιοτύπο. Όταν μια κλάση αρχικοποιείται, επιστρέφει μια τιμή αναφοράς που υποδηλώνει το αντικείμενο που δημιουργείται. Μια τιμή αναφοράς δηλώνει ένα συγκεκριμένο αντικείμενο. Ένα αντικείμενο αναφοράς είναι μια μεταβλητή που μπορεί να

αποθηκεύσει μια τιμή αναφοράς και ένα δείκτη σε ένα αντικείμενο.

```
ClassA var1 ;  
var1 = new ClassA();
```

Η διαδικασία δημιουργίας αντικειμένων περιλαμβάνει την δημιουργία μιας μεταβλητής αναφοράς στην οποία αποθηκεύεται η τιμή αναφοράς του αντικειμένου και στη συνέχεια η δημιουργία του αντικείμενου χρησιμοποιώντας την λέξη κλειδί “new”. [21]

4.5.3 Μεταβλητές (Variables)

Στην Java οι μεταβλητές αποθηκεύουν τιμές των βασικών (primitive) τύπων δεδομένων και των αναφοράς (reference) τύπων δεδομένων. [21]

```
int a, b, c; // a, b και c είναι ακέραιες μεταβλητές  
boolean flag; // η flag είναι αληθείας  
int i = 10; // η int είναι ακέραια με αρχική τιμή 10
```

```
ClassA var1 ; // Η μεταβλητή var1 αναφέρεται σε αντικείμενα της κλάσης ClassA
```

4.5.4 Κατασκευαστές (Constructors)

Όταν ένα αντικείμενο δημιουργείται χρησιμοποιώντας τη λέξη κλειδί “new”, καλούνται οι κατασκευαστές ώστε να ορίσουν την αρχική κατάσταση ενός αντικειμένου. Εκτός από τις μεθόδους και τις μεταβλητές που ορίζουμε εμείς, κάθε κλάση περιλαμβάνει και μία ειδική μέθοδο που αφορά ένα σύνολο εντολών το οποίο εκτελείται κάθε φορά που ένα νέο αντικείμενο δημιουργείται. Η ειδική αυτή μέθοδος καλείται **constructor** και ξεχωρίζει από τις υπόλοιπες μεθόδους της Κλάσης, αφού η ονομασία που της δίνουμε είναι αυτή της κλάσης. [27]

Όταν δεν υπάρχουν κατασκευαστές που ορίζονται σε μια κλάση, τότε χρησιμοποιείται ένας προεπιλεγμένος (default) κατασκευαστής, δηλαδή ένας κατασκευαστής χωρίς παραμέτρους και δημιουργείται για την κλάση από τον compiler. [21]

```
1. class ClassA {  
2. int i;  
3.  
4. ClassA(int i) {  
5.     this.i = i;  
6. }  
7.}  
8. class ClassB {  
9.  
10. ClassA var1 = new ClassA(2);  
11.}
```

4.6 JDBC API

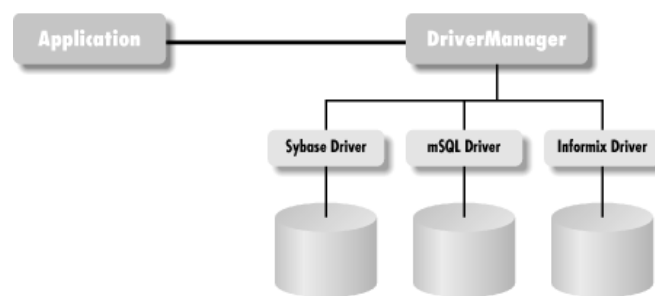
Μια εφαρμογή δημιουργεί ή χρησιμοποιεί συχνά μεγάλο όγκο δεδομένων, που είναι αποθηκευμένα σε μια βάση δεδομένων. Αν και έχει παρατηρηθεί τα τελευταία χρόνια μεγάλη αύξηση της χρήσης NoSQL βάσεων δεδομένων, παρόλα αυτά οι πιο ευρέως χρησιμοποιούμενες βάσεις δεδομένων παραμένουν οι σχεσιακές βάσεις, και μερικά παραδείγματα σχεσιακών βάσεων δεδομένων είναι της Oracle, DB2, Sybase, MySQL και Microsoft SQL Server.

Η Java υποστηρίζει τις σχεσιακές βάσεις δεδομένων μέσω του JDBC (Java Database Connectivity) API, που υπάρχει στο πακέτο `java.sql` και αποτελείται από ορισμένες διεπαφές και κάποιες απλές κλάσεις. Ακριβώς όπως και τα προγράμματα σε Java προορίζονται ώστε να λειτουργούν σε πολλές διαφορετικές πλατφόρμες, έτσι και το JDBC έχει σχεδιαστεί για να επιτρέπει σε μια εφαρμογή να επικοινωνεί με πολλά διαφορετικά συστήματα βάσεων δεδομένων. [23]

4.6.1 Αρχιτεκτονική του JDBC

Το JDBC επιτυγχάνει το στόχο του μέσα από ένα σύνολο διασυνδέσεων (Interfaces) σε Java, όπου καθεμία εφαρμόζεται διαφορετικά από μεμονωμένους προμηθευτές. Το σύνολο των κλάσεων που υλοποιούν τις JDBC διασυνδέσεις για ένα συγκεκριμένο είδος βάσης δεδομένων, ονομάζεται JDBC driver.

Το σχήμα που ακολουθεί δείχνει πως μια εφαρμογή χρησιμοποιεί το JDBC για να μιλήσει με μία ή περισσότερες βάσεις δεδομένων χωρίς να γνωρίζει λεπτομέρειες σχετικά με τον JDBC Driver για αυτές τις βάσεις. Μια εφαρμογή χρησιμοποιεί το JDBC ως διεπαφή μέσω της οποίας περνά όλα τα αιτήματα προς την βάση δεδομένων. [24]



Όταν σχεδιάζεται μια Java εφαρμογή βάσεων δεδομένων, οι μόνες πληροφορίες σχετικά με το driver που απαιτεί το JDBC υπάρχουν στο URL της βάσης. Χρησιμοποιώντας το URL της βάσης και ότι επιπλέον πληροφορίες απαιτεί το JDBC driver (όνομα χρήστη και κωδικός πρόσβασης), η εφαρμογή θα ζητήσει πρώτα την εφαρμογή `java.sql.Connection` από τον `DriverManager`. Ο `DriverManager` με τη σειρά του θα ψάξει μέσω όλων των γνωστών εφαρμογών του `java.sql.Driver` για εκείνη που συνδέεται με το URL που παρέχεται. Σε περίπτωση που δεν βρει το κατάλληλο, θα

επιστρέψει ένα μήνυμα λάθους στην εφαρμογή.

Μόλις ο Driver αναγνωρίσει την URL διεύθυνση, δημιουργεί μια σύνδεση με την βάση δεδομένων χρησιμοποιώντας τις ιδιότητες που έχουν οριστεί στο URL.[24]

```
Connection con = DriverManager.getConnection(url, uid, password);
```

Αρα το JDBC για την σύνδεση με τη βάση δεδομένων χρησιμοποιεί μια κλάση (java.sql.DriverManager) και δυο διεπαφές (java.sql.Driver και java.sql.Connection).

Java.sql.Driver: Δίνει απλώς ένα σημείο εκκίνησης στο JDBC για τη σύνδεση με τη βάση δεδομένων, ανταποκρινόμενη στα αιτήματα σύνδεσης της DriverManager και παρέχει πληροφορίες σχετικά με την εν λόγω υλοποίηση.

Java.sql.DriverManager: Σε αντίθεση με τα άλλα μέρη του JDBC, ο DriverManager είναι μια κλάση και όχι διεπαφή. Κύρια αρμοδιότητα του είναι να διατηρεί μια λίστα από Drivers και να παρουσιάζει μια εφαρμογή που ταιριάζει με την διεύθυνση URL.

Java.sql.Connection: Η κλάση Connection αντιπροσωπεύει μια απλή σύνδεση με τη βάση. Με άλλα λόγια, μπορεί να χρησιμοποιηθεί η κλάση Connection για την αποστολή μιας σειράς SQL queries στη βάση δεδομένων και την διαχείριση αυτών των SQL queries.

Μετά τη σύνδεση με τη βάση σειρά έχει η επικοινωνία μέσω ενημερώσεων και ερωτημάτων από την εφαρμογή προς τη βάση δεδομένων. Το πιο βασικό είδος πρόσβασης στη βάση περιλαμβάνει τη σύνταξη JDBC κώδικα όταν οι δηλώσεις είναι ενημερώσεις (INSERT, UPDATE, DELETE) ή ερωτήματα (SELECT). Στη συνέχεια παρουσιάζονται οι πιο βασικές JDBC κλάσεις που χρησιμοποιούνται κάθε φορά που γράφεται JDBC κώδικας.[24]

Java.sql.Statement: Η κλάση Statement είναι η πιο βασική από τις τρεις JDBC κλάσεις που αντιπροσωπεύουν SQL δηλώσεις. Εκτελεί όλες τις βασικές SQL δηλώσεις. Γενικά, μια απλή συναλλαγή με τη βάση χρησιμοποιεί μόνο μια από τις τρεις μεθόδους εκτέλεσης ενός ερωτήματος, που υπάρχουν στην κλάση Statement. Η πρώτη μέθοδος είναι η executeQuery(), η οποία παίρνει μια συμβολοσειρά SQL ως όρισμα και επιστρέφει ένα αντικείμενο ResultSet. Αυτή η μέθοδος χρησιμοποιείται για τις SQL κλήσεις που περιμένουν την επιστροφή δεδομένων από τη βάση. Από την άλλη πλευρά, δηλώσεις όπως Update εκτελούνται με τη χρήση της μεθόδου executeUpdate(). Αυτή η μέθοδος επιστρέφει τον αριθμό των γραμμών που επηρεάζονται. Τέλος υπάρχει η μέθοδος execute(), για τις περιπτώσεις που δεν είναι γνωστό αν το SQL ερώτημα που εκτελείται είναι ερώτημα ή ενημέρωση.[24]

Java.sql.ResultSet: Ένα ResultSet είναι μια ή περισσότερες σειρές δεδομένων που επιστρέφονται από ένα ερώτημα προς την βάση. Η κλάση παρέχει απλώς μια σειρά μεθόδων για την ανάκτηση στηλών από τα αποτελέσματα ενός ερωτήματος προς την βάση. Οι μέθοδοι για την ανάκτηση μιας στήλης δεδομένων από την βάση έχουν την εξής μορφή:

```
type get type(int | String)
```

Επίσης επιτρέπει την αποθήκευση τιμών στη βάση με ένα συγκεκριμένο τύπο π.χ. χαρακτήρας και η ανάκτηση τους να γίνει σε άλλη μορφή π.χ. Αριθμοί. Για παράδειγμα, γίνεται ανάκτηση μιας ημερομηνίας από τη βάση δεδομένων ως μια συμβολοσειρά (String) με την κλήση `result_set.getString(1)` αντί της `result_set.getDate(1)`. [24]

4.7 JAXP API

Το JAXP (Java API for XML Processing) χρησιμοποιείται για την επεξεργασία των δεδομένων που είναι σε μορφή XML χρησιμοποιώντας εφαρμογές που έχουν γραφτεί σε Java. Το JAXP αξιοποιεί τις δυνατότητες του Simple API for XML Parsing(SAX) και του Document Object Model (DOM), έτσι ώστε να μπορεί να επιλέξει αν τα δεδομένα θα αναλυθούν ως μια ροή γεγονότων ή θα χτιστεί ένα αντικείμενο που τα αναπαριστά. Το JAXP υποστηρίζει επίσης το πρότυπο Extensible Stylesheet Language Transformations (XSLT), δίνοντας στον χρήστη τον έλεγχο της παρουσίασης των δεδομένων και τη δυνατότητα της μετατροπής των δεδομένων σε άλλα έγγραφα όπως XML ή σε άλλες μορφές, όπως HTML.

Σχεδιασμένο για να είναι ευέλικτο, το JAXP επιτρέπει την χρήση οποιουδήποτε XML parser. Αυτό επιτυγχάνεται με το ονομαζόμενο στρώμα “pluggability”, που επιτρέπει τη σύνδεση μιας εφαρμογής με το SAX ή DOM API. Το στρώμα “pluggability” επιτρέπει επίσης τη σύνδεση με έναν επεξεργαστή XSL, επιτρέποντας έτσι τον έλεγχο του τρόπου εμφάνισης των XML δεδομένων.

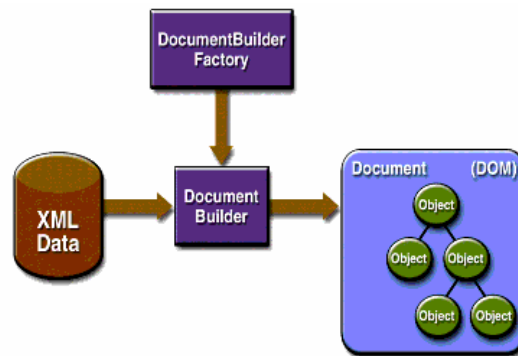
Τα κύρια JAXP APIs ορίζονται στο πακέτο `javax.xml.parsers`. Αυτό το πακέτο περιέχει ανεξάρτητες κλάσεις, `SAXParserFactory`, `DocumentBuilderFactory` και `TransformerFactory`, που δίνουν αντίστοιχα έναν `SAXParser`, `DocumentBuilder` και ένα μετασχηματιστή XSLT. Ο `DocumentBuilder` με τη σειρά του δημιουργεί ένα DOM (συμβατό αντικείμενο Document). Τα APIs επιτρέπουν τη σύνδεση τους με μια XML εφαρμογή που προσφέρεται από άλλο κατασκευαστή χωρίς να χρειαστεί αλλαγή ο πηγαίος κώδικας. [26]

4.7.1 Document Object Model (DOM) API

Το DOM παρέχει μια δομημένη παρουσίαση ενός εγγράφου (δέντρο) και ορίζει τον τρόπο που μπορεί να προσπελαστεί από προγράμματα, έτσι ώστε να μπορεί να αλλάξει το έγγραφο δομή, ύφος και περιεχόμενο. Το DOM παρέχει μια αναπαράσταση του εγγράφου ως μια δομημένη ομάδα κόμβων και αντικειμένων που έχουν ιδιότητες και μεθόδους. [25]

Στην εικόνα φαίνεται η λειτουργία του DOM API. Με την κλάση `javax.xml.parsers.DocumentBuilderFactory` δημιουργείται ένα instance ενός `DocumentBuilder`, και με το instance αυτό

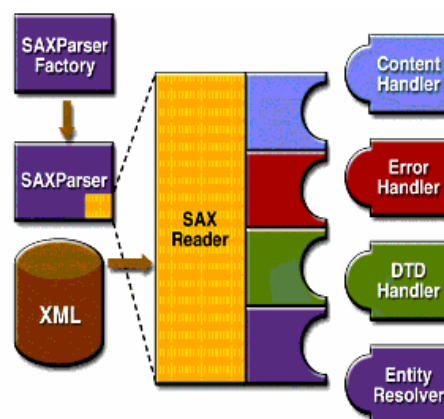
δημιουργείται ένα αντικείμενο εγγράφου (Document) που ακολουθεί τις προδιαγραφές του DOM.



Ο Builder που δημιουργείται στην πραγματικότητα, καθορίζεται από το σύστημα μέσω του `javax.xml.parsers.DocumentBuilderFactory`, το οποίο επιλέγει ποια εργοστασιακή εφαρμογή θα χρησιμοποιηθεί για την παραγωγή του κατασκευαστή (builder). Υπάρχει επίσης και η μέθοδος `DocumentBuilder newDocument()`, που δημιουργεί ένα κενό Document που υλοποιεί την διασύνδεση `org.w3c.dom.Document`. Εναλλακτικά, μπορεί να χρησιμοποιηθεί μία από τις μεθόδους του builder, ώστε να κατασκευαστεί το έγγραφο από τα υπάρχοντα XML δεδομένα. Το αποτέλεσμα είναι ένα DOM δέντρο όπως φαίνεται και στην εικόνα.[25]

4.7.2 Simple API for XML Parsing(SAX) API

Το SAX είναι ένας μηχανισμός σειριακής επεξεργασίας που επεξεργάζεται στοιχείο προς στοιχείο ένα XML. Το API αυτό διαβάζει και γράφει XML είτε στο αποθετήριο μιας βάσης είτε στο διαδίκτυο. Για server-side και εφαρμογές υψηλής απόδοσης είναι σημαντική η κατανόηση του συγκεκριμένου API. [26]



Για να ξεκινήσει η διαδικασία της επεξεργασίας, το στιγμιότυπο της κλάσης `SAXParserFactory` χρησιμοποιείται για την δημιουργία ενός στιγμιότυπου του parser. Ο parser τυλίγει ένα αντικείμενο `SAXReader`. Όταν καλείται η μέθοδος `parse()` του parser, ο reader

Web based Εφαρμογή εξαγωγής στατιστικών δεδομένων από Eurostat, στατιστικής ανάλυσής τους και εξαγωγής συμπερασμάτων.

ενεργοποιεί μια από τις μεθόδους επανάκλησης που εφαρμόζονται στην εφαρμογή. Αυτές οι μέθοδοι ορίζονται στις διεπαφές ContentHandler, ErrorHandler, DTDHandler και EntityResolver.
[26]

Κεφάλαιο 5: Στατιστική Ανάλυση Δεδομένων

5.1 Εισαγωγή στο SPSS

Στο Πανεπιστήμιο του Στάνφορντ στα τέλη της δεκαετίας του 1960, οι Norman H. Nie, C. Hadlai (Tex) Hull, και Dale H. Bent ανέπτυξαν το πρωτότυπο σύστημα λογισμικού που ονομάζεται στατιστικό πακέτο για τις Κοινωνικές Επιστήμες (SPSS). Έπρεπε να αναλύσουν ένα μεγάλο όγκο δεδομένων κοινωνικών επιστημών, έτσι έγραψαν το λογισμικό ώστε να μπορέσουν να αναλύσουν τα δεδομένα.

Οι τρεις άνδρες δημιούργησαν ένα εγχειρίδιο χρήσης στη δεκαετία του 1970, και η δημοτικότητα του λογισμικού απογειώθηκε. Η δημοτικότητα του λογισμικού εκτός από τα πανεπιστήμια επεκτάθηκε και σε τομείς της κυβέρνησης αλλά και σε ιδιωτικές επιχειρήσεις. Το 2009 η εταιρεία SPSS.Inc εξαγοράστηκε από την IBM και το όνομα του προϊόντος μετονομάστηκε σε IBM SPSS Statistics.

Η τόσο μεγάλη επιτυχία του λογισμικού οφείλετε στην πολύ καλή δουλειά που κάνει όσον αφορά τις προβλέψεις και η βοήθεια που προσφέρει ώστε να ξέρουν οι χρήστες τι πρέπει να κάνουν μετά με βάση τις ακριβείς προβλέψεις. Η πρακτική εφαρμογή του λογισμικού είχε πάντα στόχο την πρόβλεψη του μέλλοντος. Τα προβλεπτικά μοντέλα χρησιμοποιούνται σε επιχειρησιακά δεδομένα ώστε να προβλέψουν τους κινδύνους και τις ευκαιρίες. [28]

















5.2 Εγκατάσταση λογισμικού

Η έκδοση που χρησιμοποιήθηκε για την υλοποίηση της παρούσας πτυχιακής εργασίας είναι η τελευταία έκδοση, IBM SPSS 23. Το συγκεκριμένο προϊόν, SPSS Statistics Desktop 23.0.0.0 διανέμεται δωρεάν στους φοιτητές του Χαροκοπείου Πανεπιστημίου.













5.2.1 Προδιαγραφές

Στους πίνακες στην συνέχεια παρουσιάζονται αναλυτικά οι προδιαγραφές που απαιτούνται για την εγκατάσταση του προγράμματος σε έναν υπολογιστή αναλόγως το λειτουργικό σύστημα.

































Linux

| Operating System | Operating System Minimum  | Hardware | Bitness  | Product Minimum  | Components  | Notes | Details |
|--|--|----------|---|---|--|---------|--|
| Debian 6.0 | Base | x86-64 | 64-Exploit | 23.0.0.0 |  | (1) (2) |  View |
| Debian 7.0 | Base | x86-64 | 64-Exploit | 23.0.0.0 |  | (1) |  View |
| Red Hat Enterprise Linux (RHEL) 5 Desktop editions | Base | x86-64 | 64-Exploit | 23.0.0.0 |  | (1) |  View |
| Red Hat Enterprise Linux (RHEL) Client 6 | Base | x86-64 | 64-Exploit | 23.0.0.0 |  | (1) |  View |
| Red Hat Enterprise Linux (RHEL) Client 7 | Base | x86-64 | 64-Exploit | 23.0.0.0 |  | (1) |  View |
| Ubuntu 14.04 LTS | Base | x86-64 | 64-Exploit | 23.0.0.0 |  | (1) |  View |

Mac OS

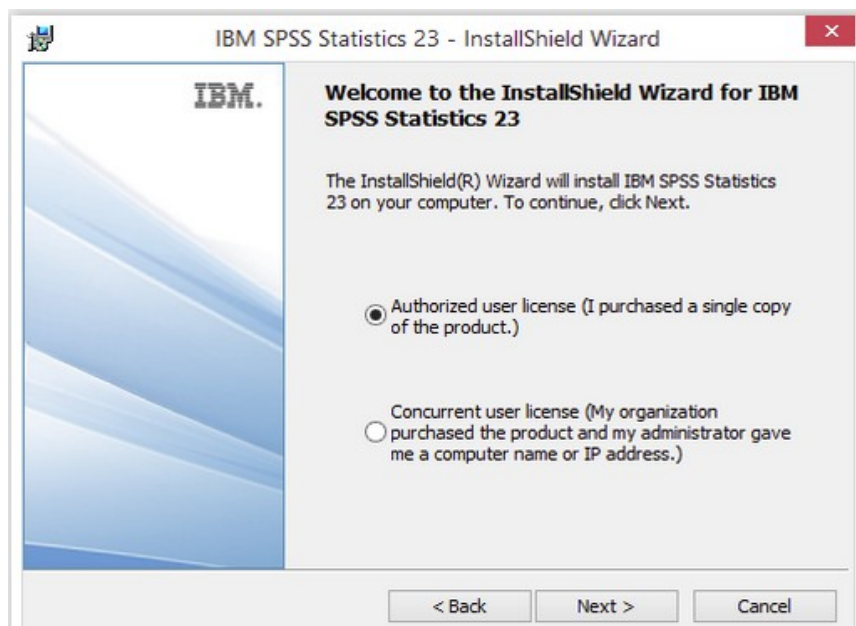
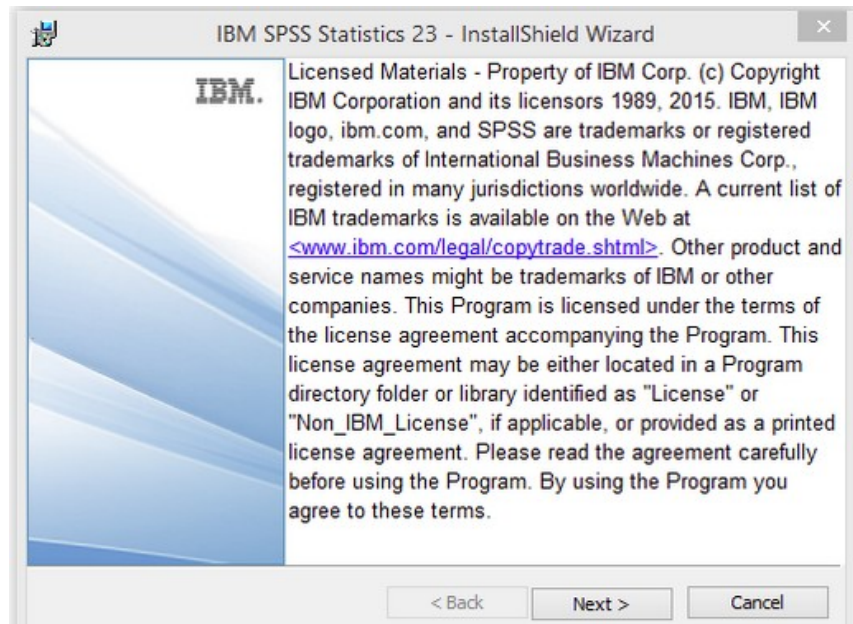
| Φίλτρο | | | | | | | |
|-------------------------|--|----------|---|---|--|-------|--|
| Operating System | Operating System Minimum  | Hardware | Bitness  | Product Minimum  | Components  | Notes | Details |
| OS X Mountain Lion 10.8 | Base | x86-64 | 64-Exploit | 23.0.0.0 |  | No |  View |
| OS X Mavericks 10.9 | Base | x86-64 | 64-Exploit | 23.0.0.0 |  | (1) |  View |
| OS X Yosemite 10.10 | Base | x86-64 | 64-Exploit | 23.0.0.0 |  | (3) |  View |
| OS X El Capitan 10.11 | Base | x86-64 | 64-Exploit | 23.0.0.0 |  | (2) |  View |

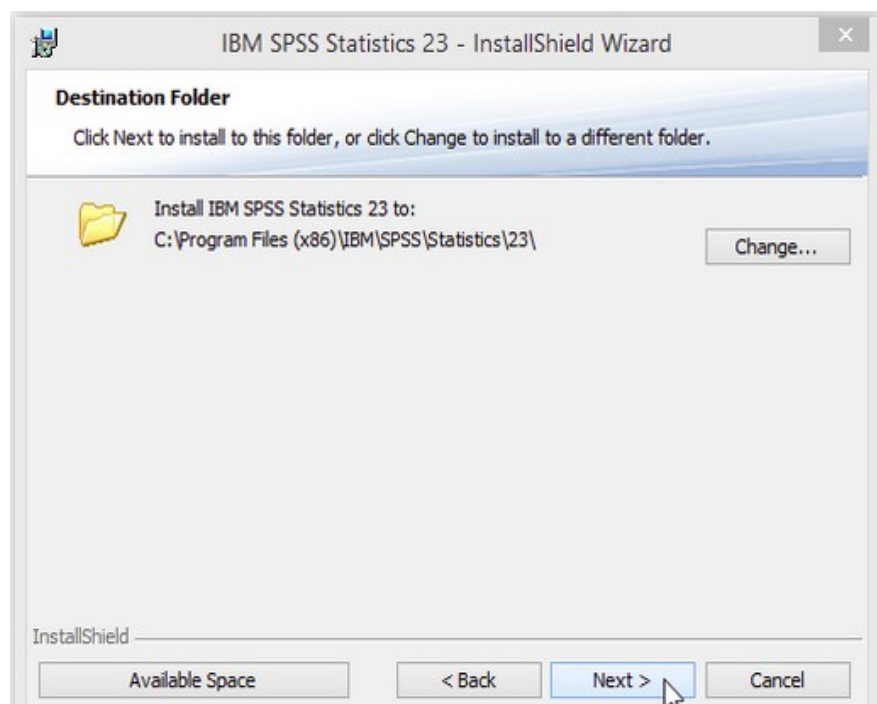
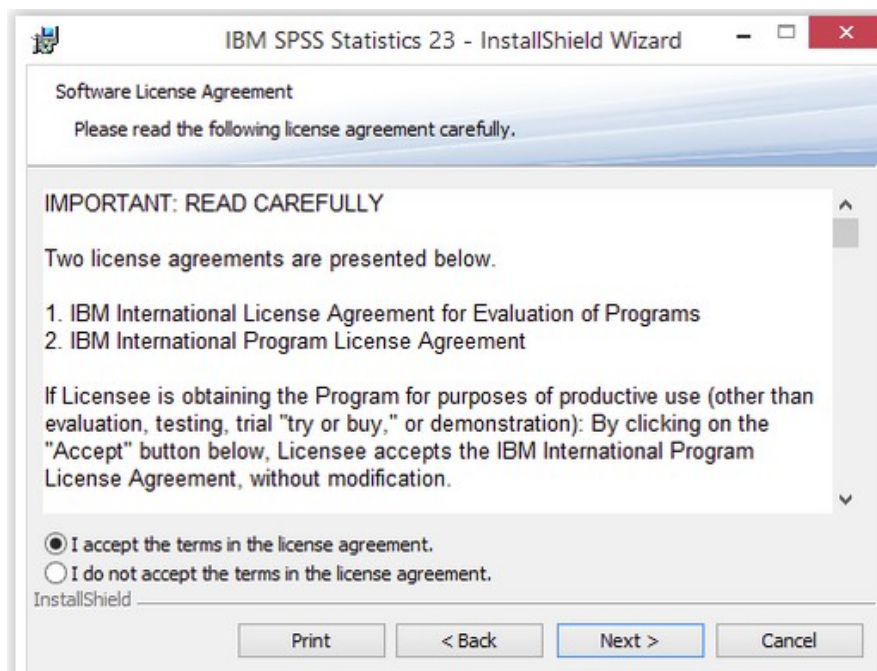
Windows

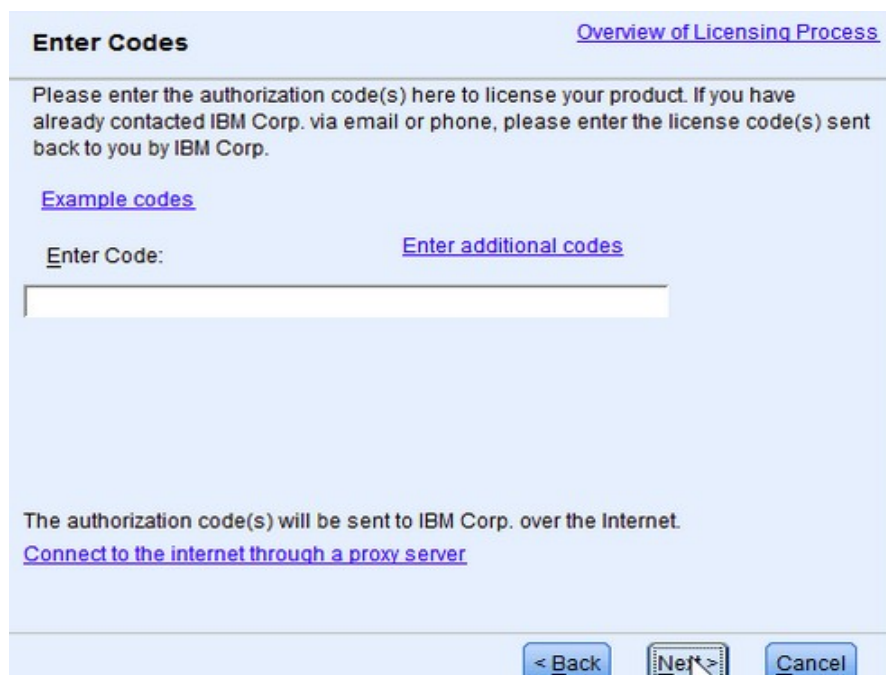
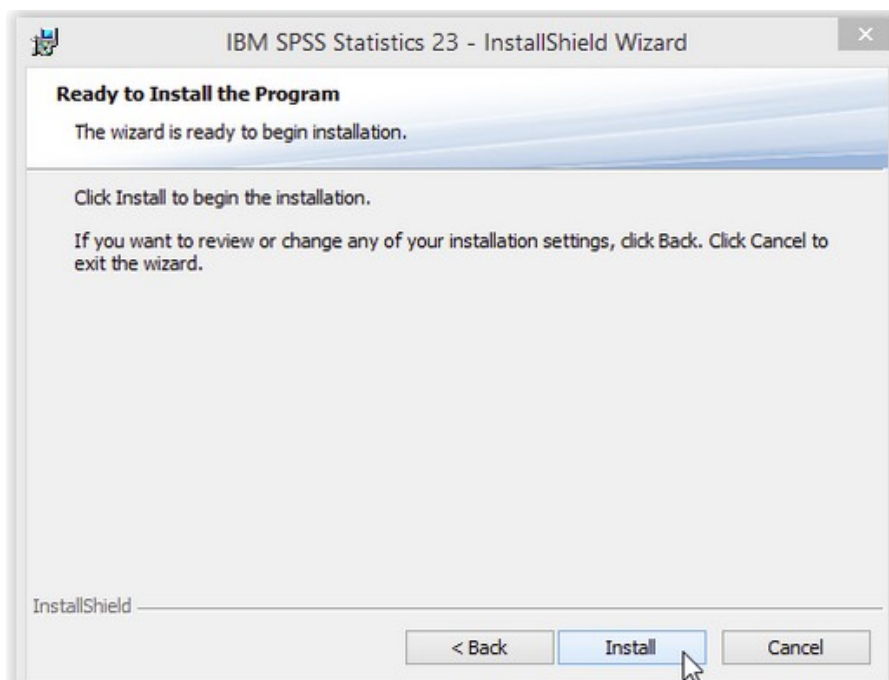
| Φίλτρο | | | | | | | |
|--------------------------|--|----------|---|---|--|-------|--|
| Operating System | Operating System Minimum  | Hardware | Bitness  | Product Minimum  | Components  | Notes | Details |
| Windows 10 Education | Base | x86-64 | 64-Exploit, 64-Tolerate | 23.0.0.0 |  | No |  View |
| Windows 8.1 Enterprise | Base | x86-32 | 32 | 23.0.0.0 |  | No |  View |
| Windows 8.1 Enterprise | Base | x86-64 | 64-Exploit, 64-Tolerate | 23.0.0.0 |  | No |  View |
| Windows 10 Enterprise | Base | x86-64 | 64-Exploit, 64-Tolerate | 23.0.0.0 |  | No |  View |
| Windows 10 Home | Base | x86-64 | 64-Exploit, 64-Tolerate | 23.0.0.0 |  | No |  View |
| Windows 10 Pro | Base | x86-64 | 64-Exploit, 64-Tolerate | 23.0.0.0 |  | No |  View |
| Windows 8.1 Professional | Base | x86-32 | 32 | 23.0.0.0 |  | No |  View |
| Windows 8.1 Professional | Base | x86-64 | 64-Exploit, 64-Tolerate | 23.0.0.0 |  | No |  View |
| Windows 8.1 Standard | Base | x86-32 | 32 | 23.0.0.0 |  | No |  View |
| Windows 8.1 Standard | Base | x86-64 | 64-Exploit, 64-Tolerate | 23.0.0.0 |  | No |  View |
| Windows 7 Enterprise | Service Pack 1 | x86-32 | 32 | 23.0.0.0 |  | No |  View |
| Windows 7 Enterprise | Service Pack 1 | x86-64 | 64-Exploit, 64-Tolerate | 23.0.0.0 |  | No |  View |
| Windows 7 Home Premium | Service Pack 1 | x86-32 | 32 | 23.0.0.0 |  | No |  View |
| Windows 7 Home Premium | Service Pack 1 | x86-64 | 64-Exploit, | 23.0.0.0 |  | No |  View |

5.2.2 Βήματα εγκατάστασης

Το πρόγραμμα, SPSS 23 εγκαταστάθηκε σε προσωπικό υπολογιστή με Windows 7 με σκοπό στη συνέχεια την ανάλυση των οικονομικών δεδομένων που έχουν αποθηκευτεί στη βάση δεδομένων. Παρουσιάζονται τα βήματα που ακολουθήθηκαν στις εξής εικόνες.







Product Authorization

Select one of the following:

☒ **License my product now**
Select this option to enter code(s) you received from IBM Corp. You should have received an authorization code and/or a license code.
[Example codes](#)

☐ **Enable for temporary usage**
Please license your product as soon as possible. When the temporary usage period is over, the software will not run.

Lock Code for this machine is: 4-2CB8B

< Back

Next >




Cancel

Internet Authorization Status
Displaying progress below.

Copy Status

Authorization in progress ... (401d817068b17168c8c3)
Authorization succeeded:
DH7Z2WU8OD5UARBZEMK5YX9YD8R5DH3UHRZOM78JQDK6LANOVBN96H
PSKKMD3DCC64NA8YUEVTFK92VH2YD3RDSTDET8ODGFPG

*** End Of Transaction. ***
Successfully processed all codes



< Back

Next >

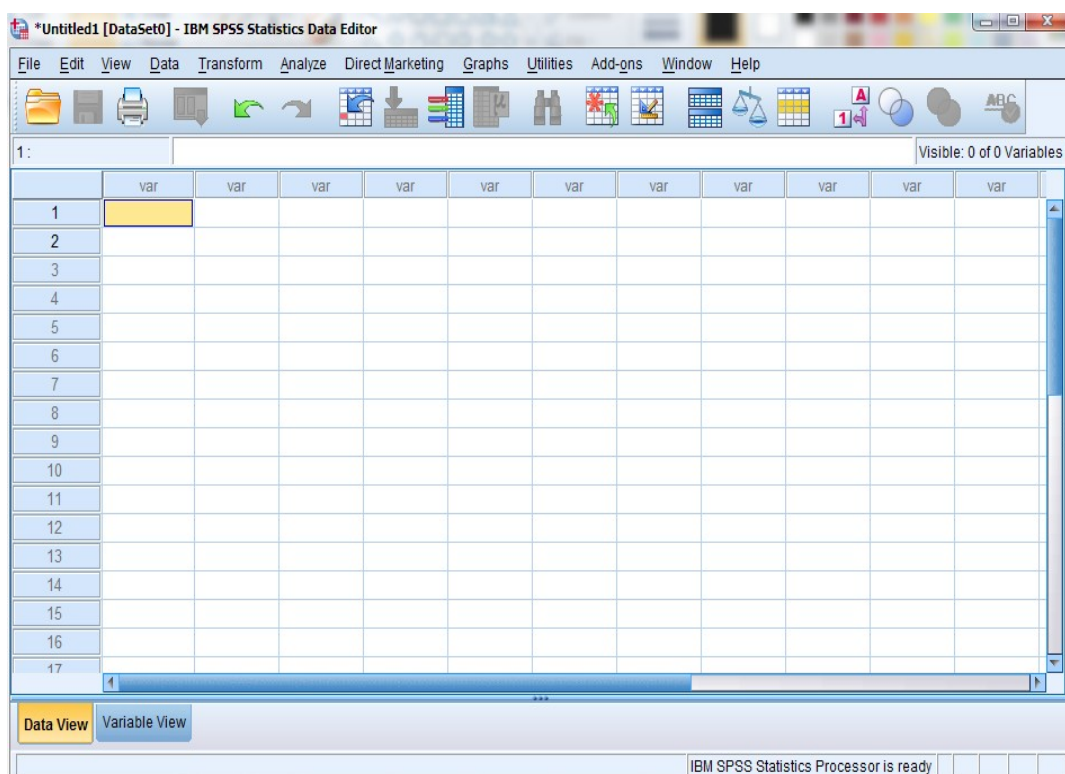
Cancel

5.3 Περιβάλλον SPSS

5.3.1 Καταχώρηση Δεδομένων

Η αρχική οθόνη που παρουσιάζεται μετά την εγκατάσταση του SPSS, αποτελείται από έναν πίνακα, που θυμίζει εμφανισιακά ένα λογιστικό φύλλο (π.χ. MS Excel). Η αρχική οθόνη αποτελείται από γραμμές και στήλες. Το ενεργό κελί (cell) έχει πιο έντονο περίγραμμα και γαλανό χρώμα. Είναι το κελί όπου θα προστεθεί ότι πληκτρολογήσει ο χρήστης.

Όπως φαίνεται και στην εικόνα που ακολουθεί, στη γραμμή μενού υπάρχουν μια σειρά από διάφορες λειτουργίες του λογισμικού. Υπάρχουν επίσης στη γραμμή εργαλείων ορισμένα εικονίδια συντόμευσης για άμεση πρόσβαση σε λειτουργίες που χρησιμοποιούνται συχνά. Στην εικόνα παρουσιάζεται ένα κενό φύλλο δεδομένων (data sheet). Υπάρχουν δυο τρόποι να εισαχθούν δεδομένα στο φύλλο, είτε πληκτρολογώντας είτε εισάγοντας απευθείας από κάποιο αρχείο. [29]



Οι επιλογές εισαγωγής δεδομένων μέσα στο πρόγραμμα εκτός από τον τρόπο της πληκτρολόγησης από τον χρήστη, είναι η εισαγωγή υπαρχόντων αρχείων απευθείας μέσα στο υπολογιστικό φύλλο του SPSS.

Στο μενού [File] υπάρχουν τρεις ξεχωριστές διέξοδοι για την ανάγνωση δεδομένων από υπάρχοντα αρχεία. Η πρώτη είναι η επιλογή [Open]. Όπως και σε άλλες εφαρμογές (π.χ. MS Word, Excel, κ.λ.π.) το SPSS έχει τη δική του μορφοποίηση για τα δεδομένα που αποθηκεύονται μέσα σε αυτό. Στην περίπτωση αυτή, η αποδεκτή προέκταση ονόματος αρχείου είναι το “sav”. Ωστόσο, αυτή η μορφοποίηση δεν είναι αναγνώσιμη από άλλα προγράμματα, γιατί δεν είναι σε δυαδική

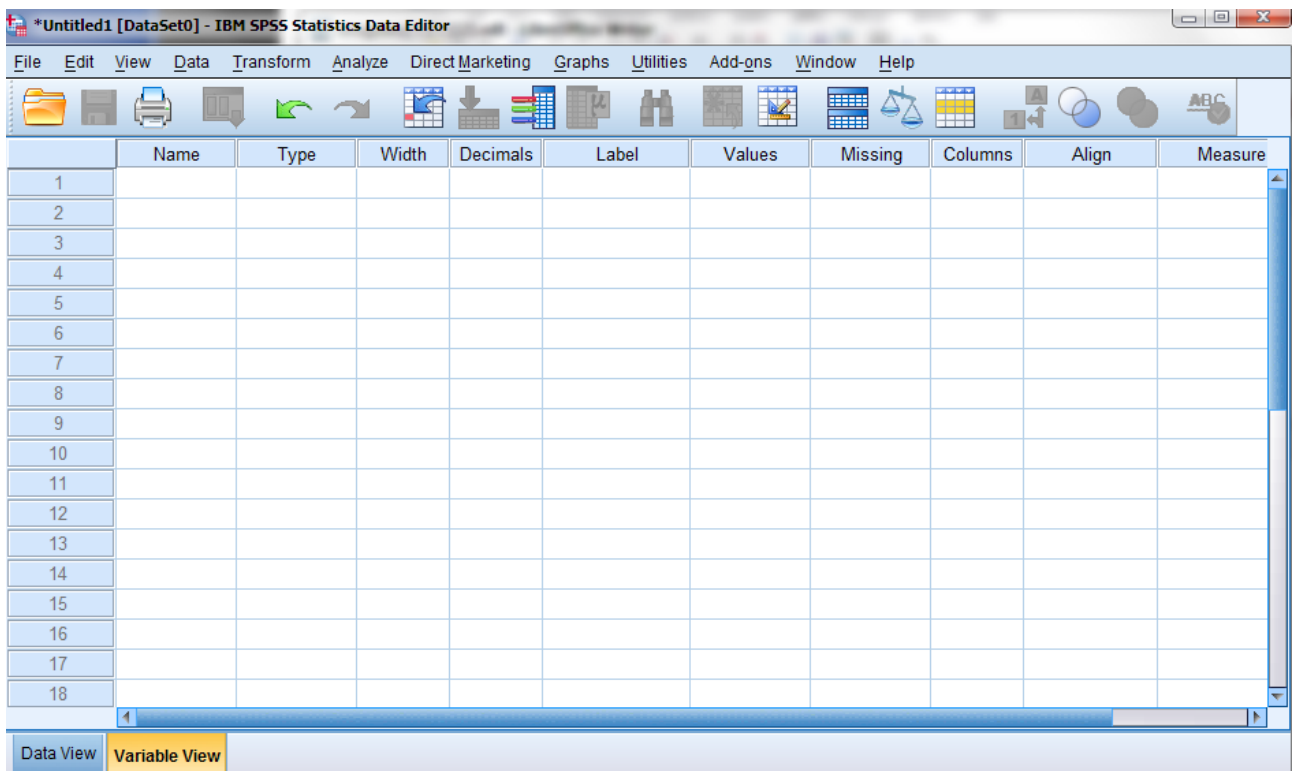
μορφή (binary format). Δεύτερη είναι η επιλογή [Open Database], που επιτρέπει την εισαγωγή δεδομένων που είναι αποθηκευμένα σε αρχεία του MS Excel ή της Access. Τέλος, η τρίτη επιλογή είναι η [Read Text Data], όπως υποδηλώνει και το όνομα της, προορίζεται για την ανάγνωση αρχείων που έχουν αποθηκευτεί σε μορφοποίηση ASCII. [29]

Στην ανάλυση των δεδομένων της παρούσα πτυχιακής θα χρησιμοποιηθεί η απευθείας εισαγωγή των δεδομένων σε μορφή .excel τα οποία έχουν εξαχθεί από την MySQL βάση δεδομένων όπου είναι αποθηκευμένα.

Επίσης αξίζει να σημειωθεί ότι στο κάτω αριστερά τμήμα της οθόνης υπάρχουν δυο καρτέλες (Data View και Variable View). Στην πρώτη από αυτές [Data View] τοποθετούνται τα αρχικά δεδομένα, ενώ στη δεύτερη [Variable View] καθορίζονται τα χαρακτηριστικά των μεταβλητών της έρευνας του χρήστη. Η μετάβαση από τη μια στην άλλη προβολή γίνεται πατώντας στο όνομα της καρτέλας που επιθυμεί ο χρήστης να μεταβεί.[29]

5.3.2 Μεταβλητές

Όπως αναφέρθηκε και προηγουμένως οι μεταβλητές βρίσκονται στην καρτέλα [Variable View]. Οι μεταβλητές εμφανίζονται στις σειρές του πίνακα, ενώ στις στήλες καθορίζονται τα διάφορα χαρακτηριστικά των μεταβλητών.



Αναλυτικά παρουσιάζονται στη συνέχεια τα διάφορα χαρακτηριστικά που μπορούν να καθοριστούν για κάθε μεταβλητή.

- **Name:** Εδώ καθορίζεται το όνομα της μεταβλητής. Οι νεώτερες εκδόσεις SPSS δέχονται ελληνικούς χαρακτήρες, αλλά προτείνεται η χρήση ονομάτων μεγέθους έως 8 λατινικούς χαρακτήρες χωρίς κενά, προκειμένου να διατηρηθεί η συμβατότητα του αρχείου με παλαιότερες εκδόσεις του SPSS. Στην περίπτωση που δοθεί το ίδιο όνομα σε δυο μεταβλητές, το πρόγραμμα θα δώσει ένα προειδοποιητικό μήνυμα ότι κάτι τέτοιο δεν επιτρέπεται. Αν ο χρήστης θέλει να δώσει ένα αναλυτικό όνομα με ελληνικούς χαρακτήρες μπορείτε να χρησιμοποιήσετε την επιλογή [Label]. Το όνομα μιας μεταβλητής δεν επιτρέπεται να ξεκινάει από αριθμό και δεν μπορεί να περιέχει κενά.
- **Type:** Είναι το είδος των δεδομένων για κάθε μεταβλητή. Εξορισμού οι νέες μεταβλητές έχουν ως type το [numeric], δηλαδή θεωρείται ότι το είδος των δεδομένων είναι αριθμοί. Ενδιαφέρον παρουσιάζει και η επιλογή [string], που σημαίνει ότι η μεταβλητή θα δεχθεί αλφαριθμητικά δεδομένα (δηλαδή οποιονδήποτε συνδυασμό γραμμάτων και αριθμών), τα οποία δεν μπορούν να χρησιμοποιηθούν σε κάποια είδη αναλύσεων. Οι [dot], [comma] και [scientific notation] αφορούν τον τρόπο παρουσίασης των αριθμών στα κελιά, ενώ οι υπόλοιπες επιλογές δεν απαιτούν κάποια ιδιαίτερη εξήγηση.
- **Width:** Εδώ επιλέγεται ο μέγιστος αριθμός χαρακτήρων που μπορούν να καταχωρηθούν για κάθε περίπτωση στη μεταβλητή του χρήστη.
- **Decimals:** Εδώ επιλέγεται το μέγιστο αριθμό δεκαδικών που μπορούν να καταχωρηθούν για κάθε περίπτωση στη μεταβλητή του χρήστη.
- **Label:** Εδώ δίνετε ένα “δεύτερο όνομα” στη μεταβλητή με ελληνικούς χαρακτήρες, κενά ή ότι άλλο θέλετε με μήκος έως 256 χαρακτήρες. Είναι σημαντικό να προσεχθεί η ορθογραφία των περιγραφών, πεζά – κεφαλαία γράμματα, ώστε να βελτιωθεί η αναγνωσιμότητα των αποτελεσμάτων, καθώς το Label χρησιμοποιείται στους πίνακες που θα κατασκευάσει το SPSS.
- **Values:** Συχνά, απαιτείται για λόγους ευκολίας να κωδικοποιηθούν οι κατηγορίες/συνθήκες των μεταβλητών σε αριθμητική μορφή. Για παράδειγμα, το “άνδρας” και το “γυναίκα” μπορεί να κωδικοποιηθεί ως “1” και “2” αντίστοιχα.
- **Missing:** Εδώ μπορούν να οριστούν συγκεκριμένες τιμές που το SPSS θα εκλαμβάνει ως ελλειπείς κατά την ανάλυση των δεδομένων. Αν για μια περίπτωση αφήσετε κενό το κελί που αντιστοιχεί στη στήλη μιας μεταβλητής το SPSS αυτόματα θα θεωρήσει ότι η συγκεκριμένη περίπτωση είναι ελλιπής ως προς μια συγκεκριμένη μεταβλητή. Οι περιπτώσεις αυτές εμφανίζονται στο παράθυρο Viewer ως system missing.
- **Columns:** Με αυτή τη ρύθμιση μπορεί αν διαφοροποιηθεί το μέγεθος της κάθε στήλης όπως θα εμφανίζεται στην κατάσταση προβολής [Data View]. Η αλλαγή του πλάτους της στήλης από αυτή την επιλογή δεν πρόκειται να επηρεάσει το μέγεθος της μεταβλητής, όπως αυτό ρυθμίστηκε προηγουμένως μέσω της ρύθμισης [Width].
- **Align:** Πρόκειται για τη ρύθμιση του τρόπου με τον οποίο θα στοιχιστούν τα δεδομένα στη στήλη. Υπάρχουν τρεις επιλογές: Left, Right και Center.
- **Measure:** Αυτή η ρύθμιση είναι από τις πιο σημαντικές καθώς δηλώνετε στο SPSS με ποιο τύπο κλίμακας έχει γίνει η μέτρηση της μεταβλητής. Υπάρχουν τρεις επιλογές: Scale, Ordinal και Nominal. Αυτές αντιστοιχούν σε Κλίμακα Ίσων Διαστημάτων (ή αναλογική), Ιεραρχική (ή τακτική) και Ονομαστική (ή κατηγορική) αντίστοιχα.[29]

5.3.3 Παρουσίαση Δεδομένων

Η παρουσίαση των δεδομένων που είναι αποθηκευμένα στο Data View μπορούν να αναπαρασταθούν είτε με την χρήση πινάκων κατανομής συχνοτήτων είτε με τη χρήση γραφικών παραστάσεων.

Οι πίνακες συχνοτήτων που εξάγονται περιέχουν την στήλη Frequency (συχνότητα), Percent (ποσοστό), Valid Percent (ποσοστό επί του συνόλου έγκυρων απαντήσεων) και Cumulative Percent (αθροιστικό ποσοστό).[30] Στο παράδειγμα που ακολουθεί βλέπουμε πως ο πίνακας συχνοτήτων μας διευκολύνει στην κατανόηση των δεδομένων.

| ΦΥΛΟ | | | | | |
|-------|---------|-----------|---------|---------------|--------------------|
| | | Frequency | Percent | Valid Percent | Cumulative Percent |
| Valid | ΑΓΟΡΙ | 6 | 42,9 | 42,9 | 42,9 |
| | ΚΟΡΙΤΣΙ | 8 | 57,1 | 57,1 | 100,0 |
| | Total | 14 | 100,0 | 100,0 | |

Ο δεύτερος τρόπος παρουσίασης δεδομένων είναι οι γραφικές παραστάσεις. Τα διαγράμματα είναι πολύ σημαντικά αλλά συχνά παραμελημένα. Αν και είναι πολύ δύσκολο να αναπαραστήσει κάποιος γραφικά όλες μαζί τις μεταβλητές του, μπορεί εναλλακτικά να κάνει διαγράμματα για την κάθε μεταβλητή ξεχωριστά και για ζευγάρια μεταβλητών. Είναι σημαντική η πληροφορία που περικλείουν καθώς μπορούν να εντοπιστούν τις σχέσεις μεταβλητών και να ερμηνεύσουν τη συμπεριφορά των υπό μελέτη αντικειμένων.[33]

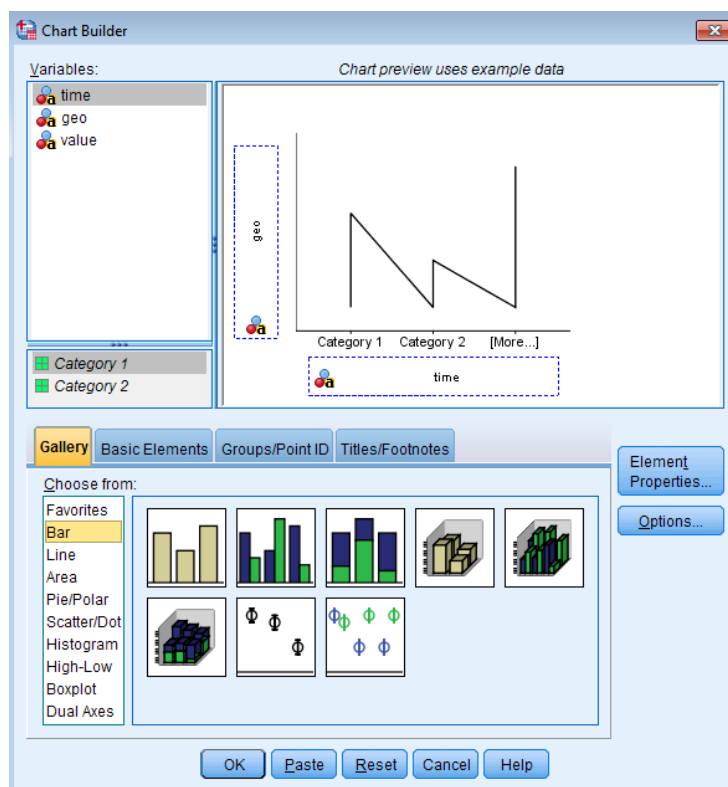
Επιπλέον, με τη χρήση των σύγχρονων H/Y μπορεί κανείς να δημιουργήσει πολύπλοκα τρισδιάστατα διαγράμματα, ενώ υπάρχουν και μερικά γραφήματα για την ταυτόχρονη απεικόνιση πολλών μεταβλητών (scatter plot matrix, star plot, chernoff faces). Ένας οδηγός διαγραμμάτων δίνεται από τον Πίνακα που ακολουθεί. [33]

| Τύπος γραφήματος | Τύπος μεταβλητών | Διάγραμμα |
|-----------------------------------|-------------------------------|--|
| Μιας διάστασης (μεταβλητής) | Κατηγορική (Ποιοτική) | Ραβδόγραμμα (Bar chart) Κυκλικό ή Πίτας (Pie chart) |
| | Αριθμητική (Ποσοτική) | Ιστόγραμμα (Histogram) Διάγραμμα μίσχου-φύλλου (Steam-and-Leaf plot) Διάγραμμα πλαισίου και απολήξεων (Box plot) |
| Δύο διαστάσεων (μεταβλητών) | Δύο ποσοτικές | Διάγραμμα σημείων (Scatter plot) |
| | Δύο ποιοτικές | Ραβδόγραμμα (Bar plot) |
| | Μια ποσοτική και μια ποιοτική | Διάγραμμα πλαισίου και απολήξεων (Box plot) Διάγραμμα σφαλμάτων (Error bars) |
| Πολλών διαστάσεων ή Πολυμεταβλητά | Ποσοτικές ή Διατάξιμες | Πίνακας διαγραμμάτων σημείων (scatter plot matrix) Αστεροειδή γραφήματα (Star plot) Τα πρόσωπα του Chernoff (Chernoff faces) |

Για την καλύτερη κατανόηση του προηγούμενου πίνακα αξίζει να δοθούν οι εξής ορισμοί

1. Ποσοτική μεταβλητή: Οι ποσοτικές μεταβλητές είναι μία από τις δυο βασικές κατηγορίες των μεταβλητών στο επιστημονικό πεδίο της έρευνας, της παρατήρησης και γενικότερα του πειράματος. Οι ποσοτικές μεταβλητές είναι άμεσα μετρήσιμες καθώς είναι αυτές που δείχνουν ότι οι διάφοροι παράγοντες μεταβάλλονται κατά αριθμητική ποσότητα, αφού πρόκειται πάντα για ποσοτικά μεγέθη, π.χ. μάζα, χρόνος, ταχύτητα κλπ. [32]
2. Ποιοτική μεταβλητή: Οι ποιοτικές μεταβλητές, καλούμενες και κατηγορικές μεταβλητές. Οι κατηγορικές μεταβλητές είναι αυτές που δείχνουν ότι οι διάφοροι παράγοντες μεταβάλλονται κατά είδος, ειδικότερα σε παρατηρήσεις ενδιαφέροντος των Συμπεριφορικών επιστημών. Στις κατηγορικές μεταβλητές περιλαμβάνονται για παράδειγμα: το “φύλο” και η “επαγγελματική κατάσταση”. [31]

Στο SPSS το σύνολο των διαθέσιμων γραφημάτων είναι προσβάσιμο διαλέγοντας από το μενού, την επιλογή Graphs και στη συνέχεια την επιλογή Chart Builder.



5.3.4 Υπολογισμός Στατιστικών Μέτρων

Τα στατιστικά μέτρα είναι αριθμοί που υπολογίζονται από τα δεδομένα και η τιμή τους αντιπροσωπεύει κάποια τάση ή συμπεριφορά του δείγματος. Τα στατιστικά μέτρα διακρίνονται στα μέτρα κεντρικής τάσης (measures of central tendency), στα μέτρα μεταβλητότητας (measures of

dispersion) και στα μέτρα σχήματος (measures of shape). Οι τρεις αυτές κατηγορίες περιγράφονται αναλυτικά στη συνέχεια. [34]

Μέτρα κεντρικής τάσης

Χρησιμοποιούνται για να περιγράψουν τη θέση του συνόλου των δεδομένων. Δηλαδή, η τιμή τους είναι η πιο αντιπροσωπευτική για να σχηματίσει κάποιος μια πρώτη εικόνα για το ύψος των τιμών των παρατηρήσεων μιας μεταβλητής. Στην κατηγορία αυτή ανήκουν η μέση τιμή (Mean), διάμεσος (Median) και επικρατούσα τιμή (Mode).

Μέτρα μεταβλητότητας / διασποράς

Χρησιμοποιούνται για να περιγράψουν τις διαφορές που υπάρχουν ανάμεσα στις τιμές μιας μεταβλητής. Δηλαδή, προσδιορίζουν αν οι παρατηρήσεις είναι συγκεντρωμένες γύρω από μια αντιπροσωπευτική τιμή ή παρουσιάζουν μεγάλη διασπορά. Στην κατηγορία αυτή ανήκουν το εύρος (range), η διασπορά / διακύμανση (variance), τυπική απόκλιση (st. deviation), τεταρτημόρια (quartiles) και ενδοτεταρτημοριακό εύρος (interquartile range).

Μέτρα σχήματος

Είναι μέτρα που περιγράφουν το σχήμα της μεταβλητής όταν αυτή παρασταθεί με ιστόγραμμα ή ραβδόγραμμα. Στην κατηγορία αυτή ανήκουν η λοξότητα (skewness) και η κυρτότητα (kurtosis),. [34]

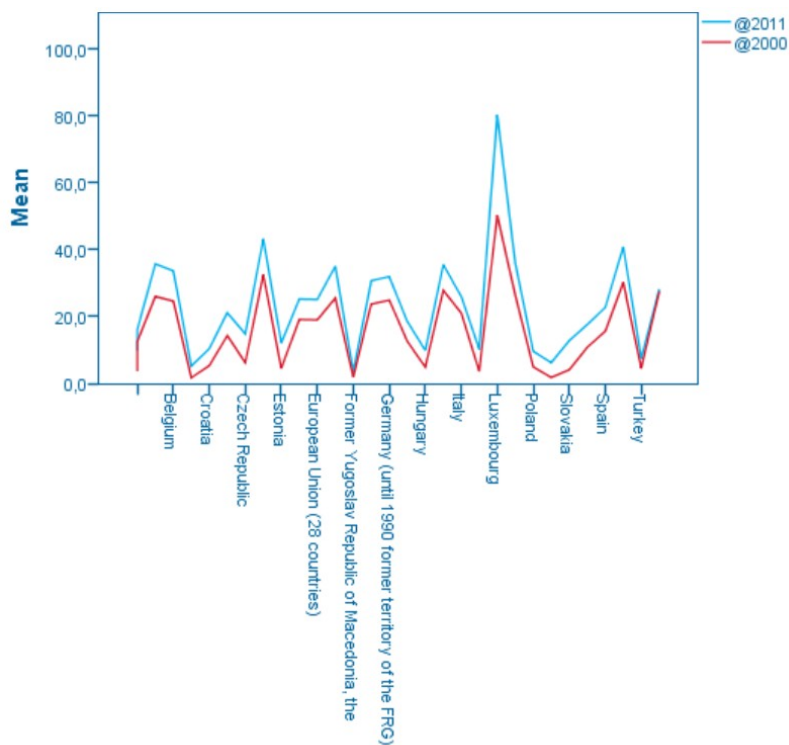
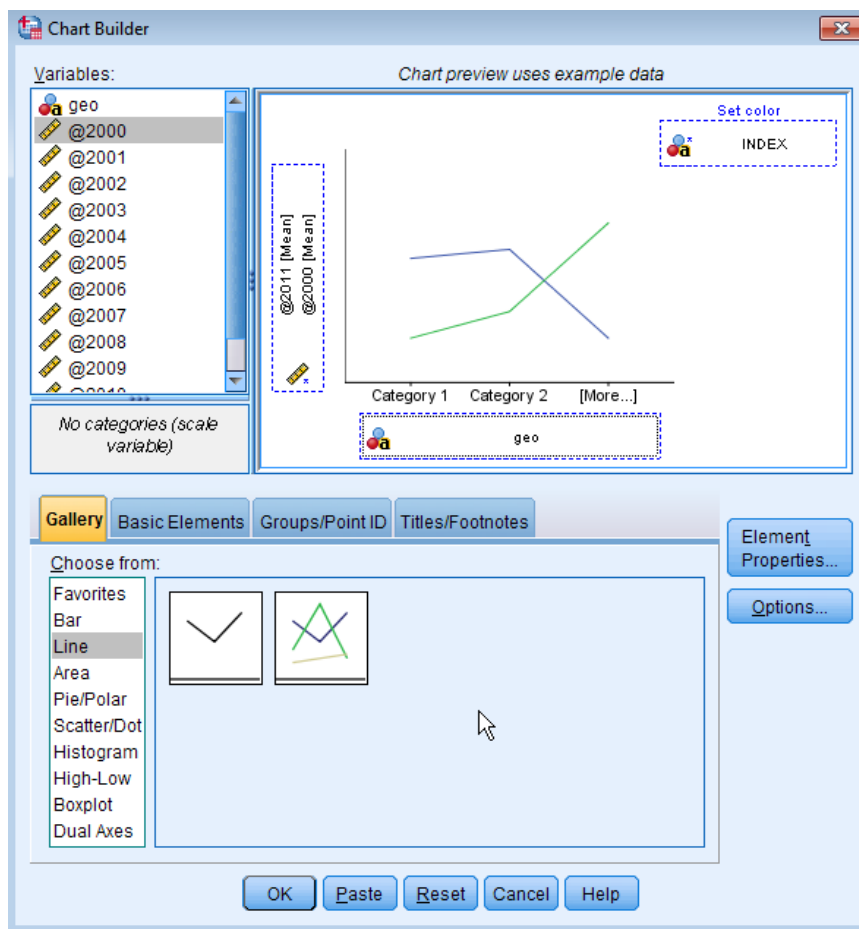
5.4 Ανάλυση Δεδομένων

Σε προηγούμενο κεφάλαιο έγινε ανάλυση του είδους των δεδομένων που αποθηκεύτηκαν στη βάση δεδομένων. Οι μεταβλητές που θα χρησιμοποιηθούν για την στατιστική ανάλυση των συμπερασμάτων είναι ο χρόνος, η χώρα και φυσικά η αντίστοιχη τιμή του ΑΕΠ.

Με τη χρήση του στατιστικού εργαλείου SPSS έγινε η αναπαράσταση των αποτελεσμάτων σε γραφικές παραστάσεις. Όπως αναφέρθηκε και προηγουμένως, με την επιλογή Chart Builder βλέπουμε όλα τα διαθέσιμα διαγράμματα που μπορούν να χρησιμοποιηθούν.

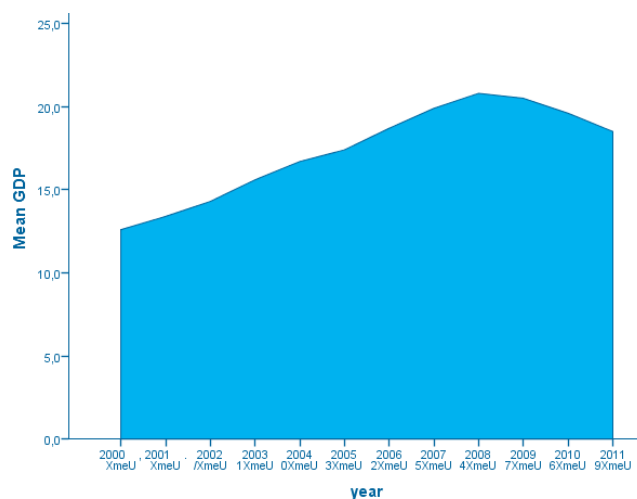
1. Αρχικά χρησιμοποιήθηκε η γραφική παράσταση Line και ειδικότερα Multiple Line. Όπως φαίνεται και στην εικόνα που ακολουθεί η χρονιές που τοποθετήθηκαν στον άξονα Y είναι η χρονιά 2000 και η χρονιά 2011. Στον άξονα X τοποθετήθηκαν όλες οι διαθέσιμες χώρες. Στόχος αυτής της ανάλυσης είναι η παρουσίαση της διαφοράς των τιμών ανάμεσα στις δυο χρονιές.

Web based Εφαρμογή εξαγωγής στατιστικών δεδομένων από Eurostat, στατιστικής ανάλυσής τους και εξαγωγής συμπερασμάτων.

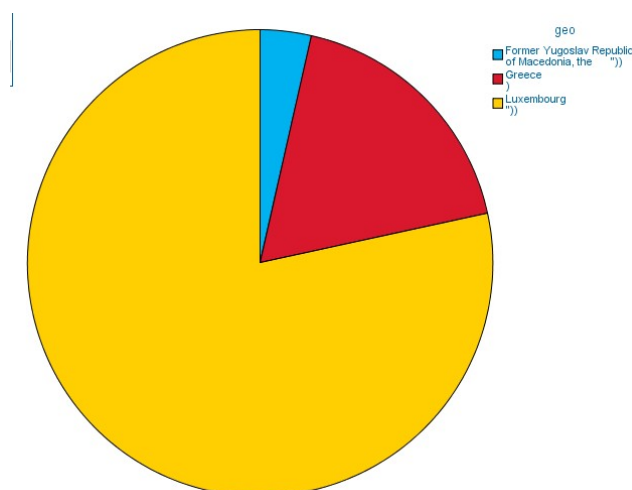


Φαίνεται αισθητά η αύξηση της τιμής του ΑΕΠ ανάμεσα στις δυο χρονιές. Η μεγαλύτερη αύξηση παρατηρείται στην χώρα του Λουξεμβούργου. Το Λουξεμβούργο έχει με διαφορά το υψηλότερο κατά κεφαλήν ΑΕΠ. Όπως φαίνεται και στο διάγραμμα οι βαλκανικές χώρες παρουσιάζουν το χαμηλότερο κατά κεφαλήν ΑΕΠ σε αντίθεση με τις χώρες της κεντρικής Ευρώπης.

2. Στη συνέχεια παρουσιάζεται η οικονομική πορεία της Ελλάδας μέσα στο χρονικό εύρος 2000 – 2011. Καταλαβαίνουμε ότι από την χρονιά 2008 όπου ξεκίνησε και η οικονομική κρίση στην Ελλάδα αλλά και σε όλη την Ευρώπη είναι καθοδική η πορεία της τιμής του ΑΕΠ.



3. Η τελευταία γραφική παράσταση δείχνει το ποσοστό του ΑΕΠ με τη βοήθεια του διαγράμματος πίτας την χρονιά 2011 για την χώρα με την μεγαλύτερη τιμή του ΑΕΠ που είναι το Λουξεμβούργο, την χώρα με τη μικρότερη τιμή του ΑΕΠ που είναι η FYROM και την Ελλάδα.



Κεφάλαιο 6: Συμπεράσματα

Στο κεφάλαιο αυτό παρουσιάζονται τα βασικότερα συμπεράσματα της παρούσας πτυχιακής εργασίας καθώς επίσης προτείνονται και ορισμένες επιπρόσθετες μελλοντικές επεκτάσεις του έργου που πραγματοποιήθηκε στα πλαίσια της εργασίας

6.1 Συμπεράσματα

Σκοπός της εργασίας αυτής είναι η ανάπτυξη μιας web-based εφαρμογής ώστε να ενημερώνονται αυτόματα τα δεδομένα στη βάση δεδομένων μας και στη συνέχεια να πραγματοποιείται στατιστική ανάλυση των δεδομένων.

Τα Web Services κάνουν ευκολότερο τον χειρισμό μεγάλου πλήθους δεδομένων. Επιτρέπουν την διαλειτουργικότητα και την ευκολία ανάπτυξης νέων εφαρμογών. Οι διάφορες πηγές δεδομένων όπως η Eurostat κάνουν προσβάσιμα τα δεδομένα τους με την βοήθεια των web services. Έτσι συμπεραίνουμε την σημασία και την αξία των WS στην εποχή μας, όπου η ανάγκη αυτοματοποίησης διαφόρων εφαρμογών είναι τεράστια ειδικά στο κομμάτι της αποθήκευσης δεδομένων.

Ο τελικός στόχος της πτυχιακής είναι η ανάλυση των οικονομικών δεδομένων ώστε να εξαχθούν ορισμένα συμπεράσματα σχετικά με το ΑΕΠ των χωρών της Ευρωπαϊκής Ένωσης. Από την ανάλυση των δεδομένων με τη βοήθεια του στατιστικού προγράμματος SPSS προκύπτουν τα εξής συμπεράσματα:

- Η διαφορά της τιμής του ΑΕΠ μεταξύ των Βαλκανικών χωρών και των χωρών της κεντρικής Ευρώπης
- Η οικονομική κάθοδος της Ελλάδας από την έναρξη της κρίσης την χρονιά 2008 έως και το 2011 (έως τότε είναι διαθέσιμα τα δεδομένα)

6.2 Μελλοντική εργασία

Μια πρόταση για επέκταση της παρούσας εργασίας είναι να μπορεί μελλοντικά ο χρήστης εκτός από τον συγκεκριμένο πίνακα από τον οποίο κατεβάζουμε τα οικονομικά δεδομένα, να μπορεί να παίρνει πληροφορίες και από άλλους πίνακες αναλόγως τις ανάγκες του. Επίσης θα μπορούσαν να χρησιμοποιηθούν και άλλες πηγές ανοιχτών δεδομένων εκτός από την Eurostat όπως για παράδειγμα ο κεντρικός κατάλογος δημοσίων δεδομένων της ελληνικής κυβέρνησης (data.gov.gr)

Επίσης σημαντική επέκταση θα ήταν η πλήρης αυτοματοποίηση της εφαρμογής με την χρήση scripts. Θα μπορούσε το .jar αρχείο του project να τοποθετηθεί σε ένα web server και με την χρήση script να τρέχει αυτόματα για παράδειγμα κάθε μέρα και να ενημερώνει την βάση. Η Eurostat ορισμένους πίνακες τους ενημερώνει δυο φορές καθημερινά κάτι που κάνει απαραίτητη την αυτόματη ενημέρωση.

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

- [1] CHAPPELL, D. A. & JEWELL, T. 2002. Java Web Services, O'Reilly & Associates
- [2] Kiet T. Tran, 2013, Introduction to Web Services with Java
- [3] <http://www.w3.org/TR/wsdl>
- [4] Douglas K. Barry, 2012, Web Services, Service-Oriented Architectures, and Cloud Computing
- [5] www.w3schools.com/php/php_mysql_intro.asp
- [6] Frank Leymann; Donald F. Ferguson; Francisco Curbera; Tony Storey; Sanjiva Weerawarana, 2005, Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More
- [7] https://en.wikipedia.org/wiki/Web_Services_Description_Language
- [8] http://aetos.it.teithe.gr/~iliou/cs4804/dialexeis/tmp/Technology_environment.pdf
- [9] <https://msdn.microsoft.com/el-gr/library/ms996507.aspx>
- [10] <http://www.altova.com/whitepapers/webservices.pdf>
- [11] <http://ec.europa.eu/eurostat/web/sdmx-web-services/sdmx>
- [12] <http://www.hostnous.com/clients/knowledgebase.php?action=displayarticle&id=48>
- [13] Timothy Boronczyk, 2015 , Jump Start MySQL
- [14] https://el.wikipedia.org/wiki/Ακαθάριστο_Εγχώριο_Προϊόν
- [15] http://www.w3schools.com/sql/sql_foreignkey.asp
- [16] SQL for MySQL Developers: A Comprehensive Tutorial and Reference ,2007, Rick F. van der Lans, Diane Cools
- [17] Learning Java by Building Android Games , 2015 , John Horton
- [18] Beginning C# Object-Oriented Programming, Second Edition , 2013 , Dan Clark
- [19] Java: A Beginner's Tutorial, 4th Edition (Updated for Java SE 8) , 2015 ,Budi Kurniawan
- [20] Your First Cup: An Introduction to the Java EE Platform, 2007, Sun Microsystems
- [21] Learn Java for Web Development: Modern Java Web Development , 2014, Vishal Layka
- [22] Java Programming with Oracle JDBC, 2001, Donald Bales
- [23] Pro Java 8 Programming, Third Edition , 2015, Brett Spell

- [24] Database Programming with JDBC & Java, Second Edition ,2000, George Reese
- [25] https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model
- [26] <http://www.oracle.com/technetwork/java/intro-140052.html>
- [27] <http://users.teicrete.gr/taxd/04/jnotes/j02.htm>
- [28] SPSS For Dummies®, 2nd Edition, 2010, Arthur Griffith
- [29] http://old.psych.uoa.gr/~roussosp//stats/Manual_SPSS16.pdf
- [30] <http://www.rhodes.aegean.gr/ptde/statistiki/FYLLA%20ERGASIAS/FE28-spss-pinakes-syxnotiton.pdf>
- [31] https://el.wikipedia.org/wiki/Ποιοτική_μεταβλητή
- [32] https://el.wikipedia.org/wiki/Ποσοτική_μεταβλητή
- [33] <http://www.math.upatras.gr/~adk/lectures/ida/lab1/tutor6%28gr%29.pdf>
- [34] <http://users.auth.gr/agpapana/SPSS>

Παράρτημα: Κώδικας

Στο συγκεκριμένο παράρτημα παρουσιάζεται ο κώδικας που υλοποιήθηκε για την ανάπτυξη της Web-based εφαρμογής. Στον πίνακα που ακολουθεί αναλύονται επιγραμματικά οι κλάσεις που υλοποιήθηκαν.

| Κλάσεις | Περιγραφή |
|---------------------|---|
| Main_Soap.java | Είναι η αρχική κλάση όπου γίνεται η εκκίνηση του προγράμματος και υλοποιούνται όλα τα στιγμιότυπα των κλάσεων που θα καλέσει η main. |
| Request.java | Στην κλάση Request, υλοποιείται το SOAP request με τις κατάλληλες παραμέτρους στο αντίστοιχο SDMX Web Service της Eurostat. |
| Retrieve_Data.java | Στην κλάση Retrieve_Data, γίνεται η επεξεργασία της συμβολοσειράς που επιστρέφεται ως response στο request που κάνουμε στο Web Service της βάσης δεδομένων της Eurostat. |
| Retrieve_Data2.java | Στην κλάση Retrieve_Data, γίνεται η επεξεργασία της συμβολοσειράς που επιστρέφεται ως response στο request που κάνουμε στο Web Service της βάσης δεδομένων της Eurostat. Τα δεδομένα των χρονιών 2002 και 2007 θέλουν διαφορετική επεξεργασία σε σχέση με τις υπόλοιπες χρονιές γι'αυτό το λόγο υπάρχει αυτή η κλάση. |
| Data_UPD.java | Στη συγκεκριμένη κλάση πραγματοποιείται η σύνδεση με την βάση και υλοποιούνται όλα τα queries όπως η εισαγωγή και η ανανέωση των δεδομένων στον πίνακα. |

Main_Soap

```
package Main_Soap;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.sql.SQLException;
import java.util.Properties;
import javax.xml.parsers.ParserConfigurationException;
import org.xml.sax.SAXException;
/**
```

```
* @author rania
*/
public class Main_Soap {
    public static void main(String[] args) throws IOException, SAXException,
    ParserConfigurationException, ClassNotFoundException, InstantiationException,
    IllegalAccessException, SQLException {
        // TODO code application logic here

        BufferedReader reader = null;
        BufferedReader reader02 = null;
        BufferedReader reader0304 = null;
        BufferedReader reader0506 = null;
        BufferedReader reader07 = null;
        BufferedReader reader0809 = null;
        BufferedReader reader1011 = null;
        //δημιουργία αντικειμένων
        Request new_soap_request = new Request();
        Retrieve_Data new_data = new Retrieve_Data();
        retrieve_data2 new_data2 = new Retrieve_Data2();
        //κάλεσμα των μεθόδων που βρίσκονται μέσα στις κλάσεις με τα κατάλληλα ορίσματα
        reader = new_soap_request.create_request(2000,2001);
        new_data.retrieve(reader);

        reader02 = new_soap_request.create_request(2001,2002);
        new_data2.retrieve(reader02);

        reader0304 = new_soap_request.create_request(2002,2004);
        new_data.retrieve(reader0304);

        reader0506 = new_soap_request.create_request(2004,2006);
        new_data.retrieve(reader0506);

        reader07 = new_soap_request.create_request(2006,2007);
        new_data2.retrieve(reader07);
```

```
        reader0809 = new_soap_request.create_request(2007,2009);
        new_data.retrieve(reader0809);

        reader1011 = new_soap_request.create_request(2009,2011);
        new_data.retrieve(reader1011);
    }
}
```

Request

```
package Main_Soap;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URL;

/*
 * @author rania
 */
//δημιουργείται το SOAP request
public class Request {

    public BufferedReader create_request(int start_year, int end_year) throws IOException {
        // ορίζεται αρχικά το URL με το οποίο συνδέεται το πρόγραμμα μας με το SDMX WS.
        URL oURL = new URL("http://ec.europa.eu/eurostat/SDMX/diss-ws/NSISStdV20Service?wsdl");
        HttpURLConnection con = (HttpURLConnection) oURL.openConnection();
        // TODO code application logic here
        // ορίζουμε τις παραμέτρους της κεφαλίδας του request όπως για παράδειγμα σε τι μορφή θέλουμε να
        // μας επιστραφεί το response, XML ή JSON.
    }
}
```

```
con.setRequestMethod("POST");
con.setRequestProperty("Content-type", "text/xml;UTF-8");
con.setRequestProperty("SOAPAction", "");
con.setRequestProperty("Accept-Encoding", "application/vnd.sdmx.structurespecificdata+xml ");
con.setRequestProperty("User-Agent", "Apache-HttpClient/4.1.1 (java 1.5)");
//σειρά έχει η δημιουργία του SOAP request
String xmldata = "<soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/' "
    + "xmlns:ns='http://ec.europa.eu/eurostat/sri/service/2.0/' "
    + "xmlns:mes='http://www.SDMX.org/resources/SDMXML/schemas/v2_0/message' "
    + "xmlns:quer='http://www.SDMX.org/resources/SDMXML/schemas/v2_0/query'> "
    + "<soapenv:Header/> "
    + "<soapenv:Body> "
    + "<ns:GetGenericData> "
    + "<message:QueryMessage "
    + "xsi:schemaLocation='http://ec.europa.eu/eurostat/sri/service/2.0/SDMXMessage.xsd "
http://ec.europa.eu/eurostat/sri/service/2.0/%20SDMXMessage.xsd' "
    + "xmlns:generic='http://www.SDMX.org/resources/SDMXML/schemas/v2_0/generic' "
    + "xmlns:message='http://www.SDMX.org/resources/SDMXML/schemas/v2_0/message' "
    + "xmlns:query='http://www.SDMX.org/resources/SDMXML/schemas/v2_0/query' "
    + "xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'> "
    + "<message:Header> "
    + "<message:ID>onh14381631849097042</message:ID> "
    + "<message:Test>false</message:Test> "
    + "<message:Prepared>2015-07-03</message:Prepared> "
    + "<message:Sender id='online-help'/> "
    + "</message:Header> "
    + "<message:Query> "
    + "<query:DataWhere> "
    + "<query:And> "
    + "<query:Dimension id='FREQ'>A</query:Dimension> "
    + "<query:Dimension id='UNIT'>EUR_HAB</query:Dimension> "
+ "<query:Time> "
    + "<query:StartTime>"+start_year+"-07-10</query:StartTime> "
    + "<query:EndTime>"+end_year+"-07-17</query:EndTime> "
+ "</query:Time> "
```

```
//ορίζεται το dataset, στο οποίο υπάρχουν οι πληροφορίες που θέλουμε.
+ "<query:Dataflow>nama_r_e3gdp</query:Dataflow> "
+ "</query:And> "
+ "</query:DataWhere> "
+ "</message:Query> "
+ "</message:QueryMessage> "
+ "</ns:GetGenericData> "
+ "</soapenv:Body> "
+ "</soapenv:Envelope> ";

con.setDoOutput(true);

//αφού τελειώσει το request, αποθηκεύουμε το response εφόσον ήταν επιτυχής η σύνδεση, στην
μεταβλητή reader, που είναι τύπου BufferedReader και επιστρέφεται στην κλάση Main_Soap.
OutputStream reqStream = con.getOutputStream();
reqStream.write(xmldata.getBytes());
BufferedReader reader = null;
//έλεγχος εάν πήγαν όλα σωστά.
if (con.getResponseCode() == 200)
    reader = new BufferedReader(new InputStreamReader(con.getInputStream()));
else
    System.out.println("Loading Error");
return reader;
}
}
```

Retrieve_Data

```
package Main_Soap;

import java.io.BufferedReader;
```

```
import java.io.IOException;
import java.io.StringReader;
import java.sql.SQLException;
import java.util.ArrayList;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;

/**
 * @author rania
 */
public class Retrieve_Data {

    public void retrieve(BufferedReader reader) throws IOException, ParserConfigurationException,
    SAXException, ClassNotFoundException, SQLException {
        String line, GDP = null, Year = null;
        int temp = 0, count = 0;

        ArrayList<String> ar = new ArrayList<>();
        //διαβάζουμε κάθε γραμμή του BufferedReader
        while ((line = reader.readLine()) != null) {
            //δημιουργία νέου DocumentBuilder
            DocumentBuilder builder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
            InputSource src = new InputSource();
            src.setCharacterStream(new StringReader(line));
            //δημιουργία νέου document από την είσοδο src
            org.w3c.dom.Document doc = builder.parse(src);
            //αποθήκευση όλων των Elements σε λίστες
            NodeList nList = doc.getElementsByTagName("generic:Obs");
            NodeList extras = doc.getElementsByTagName("generic:Value");
            NodeList nodeList = doc.getElementsByTagName("generic:ObsValue");
```



```
for (int i = 0; i < nList.getLength(); i++) {  
    // παίρνει όλα τα παιδιά nodes  
    Node nNode = nList.item(i);  
    Element eElement = (Element) nNode;  
  
    //στη συνέχεια κάνει parsing με σκοπό να κρατήσει μόνο τις πληροφορίες που θέλει από τα Nodes.  
    int y = i % 2;  
    if (y == 0) {  
        if (nNode.getNodeType() == Node.ELEMENT_NODE) {  
            if (null != extras.item(temp).getAttributes().getNamedItem("value").getNodeValue()) {  
                switch (extras.item(temp).getAttributes().getNamedItem("value").getNodeValue()) {  
                    //αναλόγως την τιμή εδώ, θα καθοριστεί ο αριθμός των προσπελάσεων στο loop στη  
                    //συνέχεια  
                    case "na":  
                        count = 4;  
                        break;  
                    case "s":  
                        count = 4;  
                        break;  
                    default:  
                        count = 2;  
                        break;  
                }  
            }  
            for (int j = 0; j <= count; j++) {  
                if (j != 0) {  
                    temp = temp + 1;  
                }  
                //ελέγχουμε εάν υπάρχουν τιμές όπως b,c,p ώστε να τις απορρίψουμε  
                if ("c".equals(extras.item(temp).getAttributes().getNamedItem("value").getNodeValue()) ||  
                    "b".equals(extras.item(temp).getAttributes().getNamedItem("value").getNodeValue()) ||  
                    "p".equals(extras.item(temp).getAttributes().getNamedItem("value").getNodeValue())) {  
                    count = count + 1;  
                }  
            }  
            if(!"OBS_STATUS".equals(extras.item(temp).getAttributes().getNamedItem("concept").getNodeValue()))
```

```
&& !"OBS_FLAG".equals(extras.item(temp).getAttributes().getNamedItem("concept").getNodeValue())) {  
    String Freq = extras.item(temp).getAttributes().getNamedItem("value").getNodeValue();  
    ar.add(Freq);  
}  
}  
temp = temp + 1;  
}  
}  
GDP = nodeList.item(i).getAttributes().getNamedItem("value").getNodeValue();  
  
Year = eElement.getElementsByTagName("generic:Time").item(0).getTextContent();  
Data_UPD DI = new Data_UPD();  
  
// αφού έχουμε όλα τα δεδομένα που χρειαζόμαστε, καλούμε την συνάρτηση όπου θα γίνει η  
σύνδεση με την βάση.  
DI.insertquery(ar, GDP, Year);  
if (y != 0) {  
    //άδειασμα λίστας  
    ar.clear();  
}  
}  
}  
}  
}
```

Retrieve_Data2

```
package Main_Soap;  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.StringReader;  
import java.sql.SQLException;  
import java.util.ArrayList;  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;
```

```
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;

/**
 * @author rania
 */
public class Retrieve_Data2 {

    public void retrieve(BufferedReader reader) throws IOException, ParserConfigurationException,
SAXException, ClassNotFoundException, SQLException {
        String line, GDP = null, Year = null;
        int temp = 0, count = 0;

        ArrayList<String> ar = new ArrayList<>();
        //διαβάζουμε κάθε γραμμή του BufferedReader
        while ((line = reader.readLine()) != null) {
            //δημιουργία νέου DocumentBuilder
            DocumentBuilder builder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
            InputSource src = new InputSource();
            src.setCharacterStream(new StringReader(line));
            //δημιουργία νέου document από την είσοδο src
            org.w3c.dom.Document doc = builder.parse(src);
            //αποθήκευση όλων των Elements σε λίστες
            NodeList nList = doc.getElementsByTagName("generic:Obs");
            NodeList extras = doc.getElementsByTagName("generic:Value");
            NodeList nodeList = doc.getElementsByTagName("generic:ObsValue");

            for (int i = 0; i < nList.getLength(); i++) {
                // παίρνει όλα τα παιδιά nodes
                Node nNode = nList.item(i);
                Element eElement = (Element) nNode;
                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
```

```
// αλγόριθμος ώστε να τυπωθούν οι 3 τιμές FREQ,UNIT,GEO
if (null != extras.item(temp).getAttributes().getNamedItem("value").getNodeValue()) {
    switch (extras.item(temp).getAttributes().getNamedItem("value").getNodeValue()) {
        //αναλόγως την τιμή εδώ, θα καθοριστεί ο αριθμός των προσπελάσεων στο loop στη
        //συνέχεια
        case "na":
            count = 3;
            break;
        case "s":
            count = 3;
            break;
        default:
            count = 2;
            break;
    }
}
for (int j = 0; j <= count; j++) {
    if (j != 0) {
        temp = temp + 1;
    }
    if ("c".equals(extras.item(temp).getAttributes().getNamedItem("value").getNodeValue()) ||
        "b".equals(extras.item(temp).getAttributes().getNamedItem("value").getNodeValue()) ||
        "p".equals(extras.item(temp).getAttributes().getNamedItem("value").getNodeValue())) {
        count = count + 1;
    }
    if(!"OBS_STATUS".equals(extras.item(temp).getAttributes().getNamedItem("concept").getNodeValue())
        && !"OBS_FLAG".equals(extras.item(temp).getAttributes().getNamedItem("concept").getNodeValue())) {
        String Freq = extras.item(temp).getAttributes().getNamedItem("value").getNodeValue();
        ar.add(Freq);
    }
}
temp = temp + 1;
}
}
GDP = nodeList.item(i).getAttributes().getNamedItem("value").getNodeValue();
```

```
Year = eElement.getElementsByTagName("generic:Time").item(0).getTextContent();
Data_UPD DI = new Data_UPD();
// αφού έχουμε όλα τα δεδομένα που χρειαζόμαστε, καλούμε την συνάρτηση όπου θα γίνει η
σύνδεση με την βάση.
DI.insertquery(ar, GDP, Year);
if (y != 0) {
//άδειασμα λίστας
ar.clear();
}
}
}
}
```

Data_UPD

```
package Main_Soap;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Properties;

/**
 * @author rania
```

```
*/  
public class Data_UPD {  
    void insertquery(ArrayList<String> ar, String GDP, String Year) throws ClassNotFoundException,  
        SQLException, FileNotFoundException, IOException {  
        int year = 0;  
        Properties pro = new Properties();  
        InputStream is;  
        //δημιουργία σύνδεσης με την βάση, διαβάζει τα απαραίτητα στοιχεία από το αρχείο database.config  
        pro.load(new FileInputStream("database_config.ini"));  
        String port = pro.getProperty("port");  
        String server = pro.getProperty("server");  
        String database = pro.getProperty("databaseName");  
        String user = pro.getProperty("user");  
        String passwd = pro.getProperty("password");  
        //σύνδεση με την mysql βάση, χρήση του jdbc  
        Class.forName("com.mysql.jdbc.Driver");  
        String connectionUrl = "jdbc:mysql://" + server + ":" + port + "?" + database + "user=" +  
            user + ";password=" + passwd + ";";  
        Connection conn = DriverManager.getConnection(connectionUrl);  
        if (conn != null) {  
            //Update δεδομένων σε περίπτωση που υπάρχει αλλαγή  
            year = Integer.parseInt(Year);  
            Statement s2 = conn.createStatement();  
            int rs2 = s2.executeUpdate("UPDATE GDP_Basic_Table SET Value= '" + GDP + "'  
                WHERE Year='" + year + "' AND Nuts_Code='" + ar.get(1) + "'");  
        }  
  
        int count = 1;  
        conn.close();  
        Connection conn2 = DriverManager.getConnection(connectionUrl);  
        if (conn2 != null) {  
            Statement s3 = conn2.createStatement();  
            ResultSet rs1 = s3.executeQuery("SELECT * FROM GDP_Basic_Table");  
            //έλεγχος εάν πρέπει έχει προστεθεί κάτι νέο στη βάση της Eurostat  
            while (rs1.next()) {  
                if (year == rs1.getInt("Year")) {
```

```
        if (ar.get(1).equals(rs1.getString("SubGeo"))) {
            count = 0;
        }
    }
}

//Αν έχει προστεθεί, το εισάγουμε και στην δική μας βάση με το ερώτημα που ακολουθεί
if (count != 0) {
    Statement s4 = conn2.createStatement();
    int rs = s4.executeUpdate("INSERT INTO GDP_Basic_Table"
        + "(Unit,Nuts_Code,Year,Value) VALUES"
        + "(" + ar.get(0) + " , " + ar.get(1) + " , " + year + " , " + GDP + ")");
} else {
    System.out.println("Nothing changed");
}
}
conn2.close();
}
```