**3η ΕΡΓΑΣΙΑ – ΟΜΑΔΙΚΗ**

**Σύστημα Econometrica**

**Εκφώνηση της εργασίας**

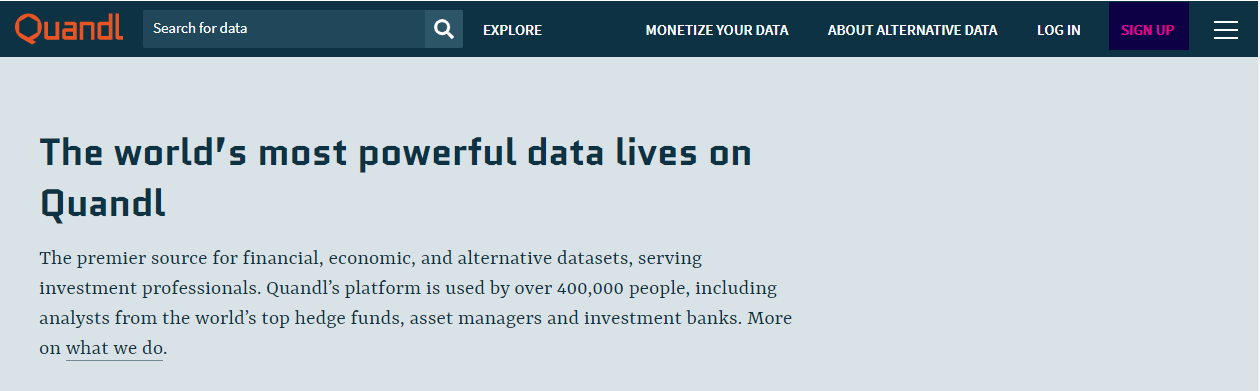
|  |
| --- |
| ***Ομαδική Εργασία – Ανάλυση Οικονομικών στοιχείων*** |

Η 3η εργασία είναι ομαδική εργασία και έχει ως στόχο την υλοποίηση της εφαρμογής Econometrica η οποία επεξεργάζεται δεδομένα που λαμβάνει από την υπηρεσία ιστού (web service) **quandl.com**

Η υπηρεσία **ndl.com** είναι μια βάση δεδομένων, η οποία περιέχει οικονομικά στοιχεία (π.χ. τιμές μετοχών) για όλες τις χώρες του κόσμου. Η πρόσβαση στην υπηρεσία μπορεί να γίνει, είτε μέσω της ιστοσελίδας [https://www.quandl.com](https://www.quandl.com/), είτε μέσω ενός ειδικού API (Application Programming Interface).

Ένα API είναι ένας κατάλογος λειτουργιών (**operations**) με τη μορφή web διευθύνσεων (Uniform Resource Identifiers - URIs) μαζί με τις παραμέτρους κλήσης αυτών (**request parameters**) των λειτουργιών. Επίσης, ένα API προσδιορίζει τις αναμενόμενες απαντήσεις (**responses**), ακολουθώντας την αρχιτεκτονική REST. Συνήθως, η κλήση του API γίνεται από ένα άλλο σύστημα-πελάτη (client) που το χρησιμοποιεί. Συνεπώς, η ζητούμενη εφαρμογή **Econometrica** που θα πρέπει να κατασκευαστεί, θαείναι μια εφαρμογή πελάτης του **quandl.com** API**.**

Για τη χρήση του **quandl.com** θα πρέπει να δημιουργήσετε έναν λογαριασμό. (αρκεί ένας λογαριασμός για κάθε ομάδα φοιτητών) ώστε να πάρετε ένα API KEY. Για τη δημιουργία του λογαριασμού αρχικά επιλέγετε το SIGN UP επάνω δεξιά.

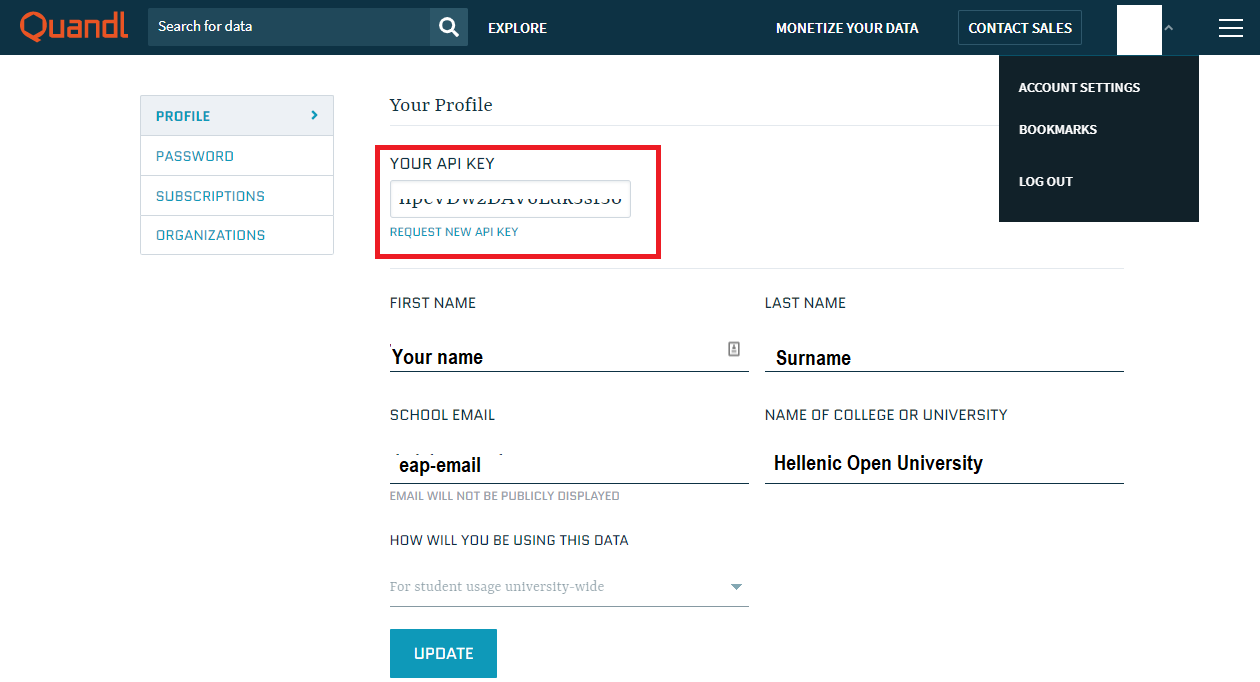


Εικόνα 1 – Κουμπί εγγραφής

Αφού δώσετε τα στοιχεία σας, θα αποσταλεί ένα e-mail επιβεβαίωσης που θα περιέχει σύνδεσμο τον οποίο πρέπει να χρησιμοποιήσετε ώστε να ενεργοποιήσετε τη σύνδεσή σας. Αφού συνδεθείτε, θα εμφανιστεί η εικόνα προφίλ επάνω δεξιά (βλέπε Εικόνα 2).

Αυτό που χρειάζεται για να γίνουν κλήσεις προς το API είναι ένα API KEY. To API KEY είναι απαραίτητο σε κάθε κλήση του API, ώστε να διασφαλιστεί ότι η σύνδεση γίνεται μόνο από τους εγγεγραμμένους χρήστες του συστήματος. Μετά τη λήψη του API KEY θα είστε σε θέση να καλείτε το API και να λαμβάνετε δεδομένα.

Για να δούμε το API KEY, επιλέγουμε την εικόνα του λογαριασμού και από το πτυσσόμενο μενού επιλέγουμε **ACCOUNT SETTINGS**.



Εικόνα 2 – Μενού επιλογών προφίλ στο Quandl.com

Η κλήση μιας λειτουργίας του API γίνεται με τη δημιουργία μιας web διεύθυνσης της παρακάτω μορφής:

operation\_URI?**parameter1**=value1&**parameter2**=value2& … &**parameterN**=valueN&**api\_key**=xxx

όπου parameterX είναι μια παράμετρος της λειτουργίας και valueX η τιμή της παραμέτρου. Για τη χρήση του API, όπως αναφέραμε, απαιτείται το API KEY που δημιουργήσατε, το οποίο δίνεται σε κάθε κλήση και αποτελεί την τιμή της παραμέτρου **api\_key**.

Για παράδειγμα για να πάρουμε μια λίστα με όλες τις καταγραμμένες τιμές ΑΕΠ (Ακαθάριστο Εθνικό Προϊόν – Gross Domestic Product (GDP)) ανά έτος για την Ελλάδα θα χρησιμοποιήσουμε την παρακάτω κλήση:

https://www.quandl.com/api/v3/datasets/WWDI/**GRC**\_NY\_GDP\_MKTP\_CN.json?api\_key=**ΤΟ\_ΚΛΕΙΔΙ\_ΕΔΩ**,

η οποία επιστρέφει τις τιμές του ΑΕΠ της Ελλάδας από το 1960 έως το 2017 στην παρακάτω μορφή (JSON format):

|  |
| --- |
| {  "dataset": {  "id": 22163003,  "dataset\_code": "GRC\_NY\_GDP\_MKTP\_CN",  "database\_code": "WWDI",  "name": "GDP (current LCU) - Greece",  "description": "GDP (current LCU) - Greece. ",  "refreshed\_at": "2019-06-25T21:19:10.949Z",  "newest\_available\_date": "2017-12-31",  "oldest\_available\_date": "1960-12-31",  "column\_names": [  "Date",  "Value"  ],  "frequency": "annual",  "type": "Time Series",  "premium": false,  "limit": null,  "transform": null,  "column\_index": null,  "start\_date": "1960-12-31",  "end\_date": "2017-12-31",  "data": [  [  "2017-12-31",  180217594600.0  ],  …  [  "1961-12-31",  441618400.0  ],  [  "1960-12-31",  391477184.0  ]  ],  "collapse": null,  "order": null,  "database\_id": 13565  }  } |

Για την εργασία θα χρειαστείτε 2 REST Endpoints (τα API operations τα λέμε και endpoints, γιατί αποτελούν την μία «άκρη» της επικοινωνίας του client (στην περίπτωσή μας η εφαρμογή econometrica) με το web service), ένα για το ΑΕΠ (Gross Domestic Product - GDP) και ένα για την κατανάλωση καυσίμων (Oil Consumption).

**Για το GDP:**

https://www.quandl.com/api/v3/datasets/WWDI/**GRC**\_NY\_GDP\_MKTP\_CN.json?api\_key=*ΚΛΕΙΔΙ\_ΕΔΩ*

**Για Oil Consumption:**

https://www.quandl.com/api/v3/datasets/BP/OIL\_CONSUM\_**GRC**.json?api\_key=*ΚΛΕΙΔΙ\_ΕΔΩ*

Παρατηρώντας προσεκτικά, παρατηρούμε πως στο σημείο **GRC**\_NY\_GDP\_MKTP\_CN περνάμε σαν παράμετρο τον κωδικό χώρας της οποίας τα στοιχεία θέλουμε να αντλήσουμε. Ο κωδικός χώρας είναι alpha-3 ISO Country Code, όπως περιγράφεται [εδώ](https://en.wikipedia.org/wiki/ISO_3166-1_alpha-3). Συνημμένο μαζί με την εκφώνηση της εργασίας θα βρείτε ένα αρχείο CSV στο οποίο περιέχονται όλες οι χώρες μαζί με τους κωδικούς ISO alpha-3.

**Κλήση HTTP**

Για την κλήση HTTP προτείνουμε τη χρήση της βιβλιοθήκης OkHttp. Ενδεικτικό παράδειγμα:

String **urlToCall** =

"https://www.quandl.com/api/v3/datasets/WWDI/**GRC**\_NY\_GDP\_MKTP\_CN.json ?api\_key=THE\_KEY\_HERE";

OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder().url(**urlToCall**).build();

try (Response response = client.newCall(request).execute()) {

if(response.isSuccessful() && response.body() != null) {

String responseString = response.body().string();

System.out.println(responseString);

return responseString;

}

} catch (IOException e) {

e.printStackTrace();

return null;

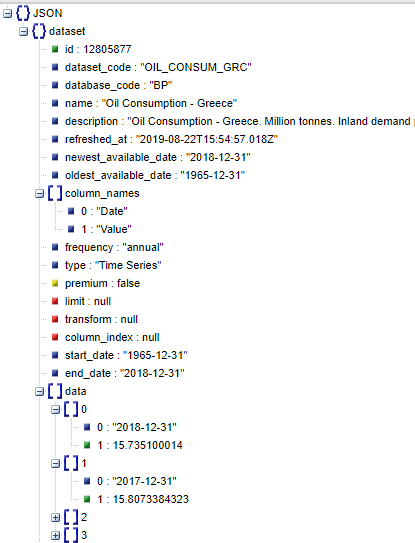
}

Όλες οι απαντήσεις έρχονται σε μορφή JSON[[1]](#footnote-2). Το JSON είναι ένας δομημένος τρόπος περιγραφής πληροφοριών που βασίζεται στη δομή του Map (βλέπε Εικόνα 3). Κάθε JSON Object αποτελείται από σύνολα Key-Value με δυνατότητα φωλιάσματος (nesting), όπου κάθε Value είναι είτε ένα άλλο JSON Object ή μια λίστα από JSON Objects κ.ο.κ. Για καλύτερη κατανόηση της δομής των αρχείων JSON μπορούν να χρησιμοποιηθούν και εργαλεία διαθέσιμα στο διαδίκτυο όπως το <http://jsonviewer.stack.hu>, τα οποία οπτικοποιούν τη δομή του αρχείου σε δεντρική μορφή (tree view).

Στην περίπτωση χρήσης του web service Quandl, βλέπουμε ότι σε πρώτο επίπεδο υπάρχει η ιδιότητα **dataset**. Στη συνέχεια ακολουθούν πολλές ιδιότητες περιγραφής του dataset, όπως οι ιδιότητες **start\_date** και **end\_date** που δηλώνουν το αρχικό και τελικό έτος καταγραφής δεδομένων. Να σημειώσουμε πως μας ενδιαφέρει μόνο το έτος, καθώς η ημερομηνία είναι πάντα η 31η Δεκεμβρίου. Τέλος, υπάρχει η λίστα δεδομένων **data** με τις τιμές που μας ενδιαφέρουν.

Παρατηρούμε πως τα JSON Objects περιέχονται μέσα στα σύμβολα { και }, ενώ οι λίστες περιέχονται μέσα στα σύμβολα [ και ] [[2]](#footnote-3).

Η γενική δομή του παραπάνω JSON αντικειμένου είναι η εξής:



Εικόνα 3 - Γενική δομή JSON Response για τα οικονομικά στοιχεία

Αναπτύσσοντας το πρώτο στοιχείο της λίστας ([0]) βλέπουμε είναι μια άλλη λίστα (φωλιασμένη) με 2 πάντα τιμές. Η 1η τιμή είναι η ημερομηνία και η δεύτερη είναι μέτρηση. Στο πεδίο description αναφέρεται η μονάδα μέτρησης, στο παράδειγμα εκατομμύρια τόνοι.

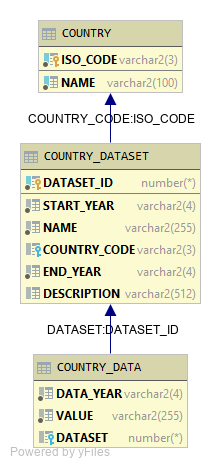
Επιπλέον, δίδονται οι παρακάτω προδιαγραφές που αφορούν τη λειτουργικότητα τους συστήματος καθώς και μια ενδεικτική διεπαφή χρήστη (Βλέπε Εικόνα 4).

|  |  |
| --- | --- |
| Νο | Περιγραφή |
| R1 | Το σύστημα περιλαμβάνει τη βασική οθόνη όπως φαίνεται στην Εικόνα 4. Κατά τη φόρτωση της εφαρμογής, το σύστημα διαβάζει από το αρχείο iso-countries.csv τις διαθέσιμές χώρες (μαζί με το alpha3 iso code) και φορτώνει μόνο τα ονόματα των χωρών στο σχετικό Drop Down μενού. |
| R2 | Με την επιλογή του πλήκτρου Fetch Data το σύστημα ελέγχει αν για την επιλεγμένη χώρα υπάρχουν διαθέσιμα τα δεδομένα στη ΒΔ.   1. Αν δεν υπάρχουν, το σύστημα χρησιμοποιεί τα 2 Endpoints για να κάνει download τα δεδομένα κατανάλωσης καυσίμων και ΑΕΠ. Στη συνέχεια τα εμφανίζει στον αριστερό και δεξιό πίνακα αντίστοιχα (Βλέπε Εικόνα 4). 2. Αν τα δεδομένα ήταν ήδη αποθηκευμένα, το σύστημα τα διαβάζει από τη ΒΔ, τα εμφανίζει στον αριστερό και δεξιό πίνακα αντίστοιχα και σημειώνει τη σχετική ένδειξη:  **Already Saved to Database.** |
| R3 | Με την επιλογή του πλήκτρου SAVE το σύστημα αποθηκεύει τα δεδομένα στη ΒΔ (αν έχουν μεταφορτωθεί κάποια από το δίκτυο) στη ΒΔ, και σημειώνει τη σχετική ένδειξη:  **Already Saved to Database**  Σημείωση 1: Όταν η ένδειξη Already Saved to Database είναι ενεργοποιημένη τότε το κουμπί SAVE είναι απενεργοποιημένο.  . |
| R4 | Με την επιλογή του πλήκτρου DELETE ALL, το σύστημα διαγράφει όλα τα δεδομένα από όλους τους πίνακες της βάσης, αφού προηγηθεί ερώτημα επαλήθευσης με επιβεβαίωση από την χρήστη. |
| R5 | Με την επιλογή του πλήκτρου Plot το σύστημα εμφανίζει σε αντιπαραβολή (στο ίδιο διάγραμμα δηλαδή για να είναι δυνατή η σύγκριση) σε ξεχωριστό παράθυρο τις 2 χρονοσειρές όπως στο παρακάτω παράδειγμα: |



Εικόνα 4 - Ενδεικτική διεπαφή χρήστη

Το Entity-Relationship (ER) διάγραμμα δίνεται στην Εικόνα 5. Το αρχείο δημιουργίας των πινάκων της βάσης (init-table-script.sql) δίδεται μαζί με την εκφώνηση της εργασίας.



Εικόνα 5 - Entity - Relationship Diagram

**Επεξήγηση Σχήματος ΒΔ**

* Πίνακας **COUNTRY**
  + **ISO\_CODE**: Ο κωδικός ISO Alpha3 της χώρας (PK)
  + **ΝΑΜΕ**: Το όνομα της χώρας
* Πίνακας **COUNTRY\_DATASET**
  + **DATASET\_ID**: Το πρωτεύων κλειδί – AUTO Incremented
  + **START\_YEAR**: Το πρώτο έτος για το οποίο υπάρχουν διαθέσιμα δεδομένα
  + **START\_YEAR**: Το τελευταίο έτος για το οποίο υπάρχουν διαθέσιμα δεδομένα
  + **NAME:** Το όνομα του Dataset πχ. Oil Consumption for Greece
  + **DESCRIPTION:** Περιγραφή του Dataset , συνήθως πιο λεπτομερής από το Name.
  + **COUNTRY\_CODE**: Ο κωδικός ISO Alpha3 της χώρας (Foreign Key)
* Πίνακας **COUNTRY\_DATA**
  + **DATASET**: Το Dataset στο οποίο αναφέρεται αυτή η μέτρηση. (Foreign Key)
  + **DATA\_YEAR**: Έτος καταγραφής
  + **VALUE**: Τιμή καταγραφής.

|  |
| --- |
| **ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1 – Διαχείριση Έργου** |

H διαχείριση έργου θα πρέπει να γίνει με βάση τη μέθοδο Scrum. H Scrum είναι μια σύγχρονη ευέλικτη μέθοδος, μια σύντομη εισαγωγή της οποίας δίνεται στο παράρτημα της παρούσας εργασίας.

Για τη διαχείριση έργου οι βασικοί ρόλοι είναι οι ακόλουθοι:

* Ιδιοκτήτης προϊόντος (Product owner): Το ρόλο αυτό παίζει ο οικείος ΣΕΠ
* Ομάδα ανάπτυξης (Scrum Team): Η κάθε ομάδα φοιτητών αποτελούμενη από 3 έως 4 φοιτητές
* Ο διαχειριστής του έργου (Scrum Master): Ένας φοιτητής από την κάθε ομάδα θα αναλάβει αυτό το ρόλο

Το έργο θα εκτελεστεί σε τρεις επαναλήψεις (sprints). Πιο συγκεκριμένα, η πρώτη επανάληψη θα είναι διάρκειας δύο εβδομάδων, ενώ οι άλλες δύο μιας εβδομάδας εκάστη.

Σε κάθε επανάληψη η ομάδα θα πρέπει να αποφασίζει σε συνεργασία με τον product owner για το ποιες απαιτήσεις του product backlog θα συμπεριληφθούν στο sprint.

Για την ευκολότερη διαχείριση του έργου προτείνεται τη χρήση του [Trello](https://trello.com/). Το Trello είναι ένα Agile-style online εργαλείο διαχείρισης και προγραμματισμού εργασιών. Μπορείτε να βρείτε περισσότερες πληροφορίες και να κάνετε δωρεάν εγγραφή στην ιστοσελίδα του https://trello.com, καθώς και να παρακολουθήσετε διαλέξεις που εξηγούν πως μπορεί να γίνει χρήση του Trello για έργα Scrum:

<https://www.youtube.com/watch?v=pFbVxXOvQ8I>

<https://www.youtube.com/watch?v=_iodOh-QTww>

<https://www.youtube.com/watch?v=3lSp-T4Fk5U>

Κατά την έναρξη του έργου θα πρέπει να γίνει αξιολόγηση των απαιτήσεων ώστε να υπολογιστεί η απαιτούμενη προσπάθεια (effort) για κάθε απαίτηση με χρήση της μεθόδου planning poker - <http://www.planningpoker.com/>.

Στη συνέχεια θα πρέπει να γίνει προτεραιοποίηση, ώστε να αποφασιστεί ποιες απαιτήσεις θα συμπεριληφθούν σε κάθε sprint. Για την προτεραιοποίηση θα πρέπει να χρησιμοποιηθεί η μέθοδος priority poker (<http://www.uxforthemasses.com/priority-poker/>) που αποτελεί μια παραλλαγή της planning poker, όπου οι συμμετέχοντες αντί να επιλέγουν εκτιμήσεις προσπάθειας για κάθε απαίτηση, επιλέγουν προτεραιότητες. Η μέθοδος είναι δυνατό να εκτελεστεί και μέσω του <http://www.planningpoker.com/> χρησιμοποιώντας τις κάρτες «T-Shirt Sizes» με την παραδοχή ότι το XXS αντιστοιχεί στην μικρότερη προτεραιότητα και το XXL στη μεγαλύτερη. Στα πλαίσια αυτής της δραστηριότητας θα πρέπει να παραδοθούν:

1. **Υπολογισμός της απαιτούμενης προσπάθειας** ανά απαίτηση τεκμηριώνοντας αναλυτικά τον τρόπο υπολογισμού της προσπάθειας.
2. **Υπολογισμός των προτεραιοτήτων των απαιτήσεων** τεκμηριώνοντας αναλυτικά τον τρόπο υπολογισμού της προτεραιότητας.
3. **To διάγραμμα GANNT**, όπου θα παρουσιάζονται, τόσο οι επαναλήψεις (sprints), όσο και οι βασικές δραστηριότητες (tasks) της κάθε επανάληψης.
4. Το **product backlog** όπως αυτό διαμορφώθηκε σε κάθε επανάληψη σε μορφή φύλλου εργασίας (θα δοθεί σε μορφή xls και θα περιέχει τρία φύλλα εργασίας, ένα για κάθε επανάληψη) μαζί με τις συμπληρωματικές/βοηθητικές πληροφορίες που αποφασίσατε να συλλέξετε.
5. Για καθένα από τα user stories του product backlog να περιγράψετε τα **κριτήρια αποδοχής** (acceptance criteria).
6. **Οι αναθέσεις των αρμοδιοτήτων** στα μέλη της ομάδας σε κάθε επανάληψη (sprint).
7. O **χρόνος υλοποίησης** ανά απαίτηση/ανά επανάληψη (sprint) σε σύγκριση με αυτά που είχαν προϋπολογισθεί, καθώς και το διάγραμμα κατανάλωσης προσπάθειας (burnt down chart). Να υπολογιστεί η ταχύτητα παράδοσης (velocity) για κάθε επανάληψη.

Η εργασία θα πρέπει να κατανεμηθεί με τέτοιο τρόπο ώστε **ΚΑΘΕ ΜΕΛΟΣ** της ομάδας να αναλάβει την υλοποίηση (ανάπτυξη κώδικα) που αφορά κάποιες από τις απαιτήσεις.

|  |
| --- |
| ***Μαθησιακά Αποτελέσματα***  Στην δραστηριότητα 1 θα σας δοθεί η δυνατότητα να κατανοήσετε και να μελετήσετε τις έννοιες:   * Του κύκλου ζωής έργου λογισμικού, * Της οργανωτικής δομής ενός έργου, * Των σύγχρονων εργαλείων συνεργασίας σε ένα έργο λογισμικού, * Του ρόλου σε ένα έργο λογισμικού, * Της διαχείριση του αντικειμένου εργασιών του έργου, * Της διαχείρισης απαιτήσεων χρήστη, * Του χρονοδιαγράμματος του έργου, * Του διαγράμματος Gantt, καθώς και, * Της εκτίμησης του κόστους. |

|  |
| --- |
| **ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2 – Υλοποίηση Συστήματος** |

|  |
| --- |
| ***Ερώτημα Α – Διάγραμμα Κλάσεων και Υλοποίηση Κλάσεων σε Java*** |

Στο ερώτημα αυτό θα πρέπει:

1. Να σχεδιάσετε το διάγραμμα κλάσεων του συστήματος Econometrica. Στο διάγραμμα θα πρέπει εκτός από τις κλάσεις του εννοιολογικού μοντέλου να εμφανίζονται οι κλάσεις του GUI, οι POJO κλάσεις και όποιες άλλες κλάσεις κρίνετε απαραίτητες. Επίσης το διάγραμμα θα πρέπει να είναι σε πλήρη αντιστοιχία με τον κώδικα του συστήματος.
2. Εισάγεται των κώδικα για τις κλάσεις του συστήματος.
3. Να δώσετε τις αντίστοιχες κλάσεις οντοτήτων (entity beans - POJOs) σε κώδικα java. Θα πρέπει να εισάγετε ΜΟΝΟ τον κώδικα που έχετε γράψει εσείς και όχι αυτόν που παράγεται αυτόματα προσδιορίζοντας ευκρινώς το όνομα της κλάσης και της μεθόδου.

|  |
| --- |
| ***Μαθησιακά Αποτελέσματα***  Στη δραστηριότητα 2.Α θα σας δοθεί η δυνατότητα να κατανοήσετε:   * τη δημιουργία διαγράμματος κλάσεων, * την έννοια της Βάσης Δεδομένων (ΒΔ), * την έννοια της διατήρησης δεδομένων, * το Java Persistence API (JPA).   Σχετικά με τις «Βάσεις Δεδομένων» θα μπορέσετε:   * να ορίσετε τους πίνακες μιας ΒΔ, * να αναγνωρίσετε τις σχέσεις μεταξύ πινάκων σε ένα σχήμα ΒΔ.   Σχετικά με τη «διατήρηση δεδομένων» θα μπορέσετε:   * να δώσετε 2 παραδείγματα οντοτήτων που διατηρούνται σε μια εφαρμογή.   Σχετικά με το «Java Persistence API» θα μπορέσετε:   * να ορίσετε κλάσεις οντότητας στο JPA, * να κατανοήσετε όλες τις συσχετίσεις που υποστηρίζει το JPA, * να αναγνωρίσετε τις φάσεις του κύκλου ζωής μιας οντότητας στο JPA, * να κατανοήσετε την αρχιτεκτονική του JPA και, * να κατασκευάσετε μια οντότητα JPA ορίζοντας τα κατάλληλα annotations. |

|  |
| --- |
| ***Ερώτημα Β – Δημιουργία GUI Εφαρμογής*** |

Να υλοποιήσετε τις συνοριακές κλάσεις, καθώς και το GUI της εφαρμογής.

|  |
| --- |
| ***Μαθησιακά Αποτελέσματα***  Στη δραστηριότητα 2.Β θα σας δοθεί η δυνατότητα:   * να αναπτύξετε μια σύνθετη εφαρμογή σε Java με γραφικό περιβάλλον που να αλληλοεπιδρά με βάση δεδομένων με βάση ένα σύνολο δεδομένων απαιτήσεων.   Πιο συγκεκριμένα θα σας δοθεί η δυνατότητα:   * να κατανοήσετε την έννοια της Γραφικής Διεπαφής Χρήστη, * να κατανοήσετε τη βιβλιοθήκη Swing, * να κατανοήσετε τα βασικά γραφικά στοιχεία του Swing, * να κατασκευάσετε με χρήση του περιβάλλοντος σχεδίασης του Netbeans γραφικά στοιχεία του Swing, * να κατασκευάσετε με χρήση του περιβάλλοντος σχεδίασης του Netbeans και του Netbeans binding ένα πίνακα του Swing που να αντλεί δεδομένα από μια οντότητα JPA * να κατασκευάσετε με χρήση του περιβάλλοντος σχεδίασης του Netbeans και του netbeans binding μια γραφική διεπαφή χρήστη με στοιχεία του Swing που να αντλούν δεδομένα από οντότητες του JPA. |

|  |
| --- |
| ***Ερώτημα Γ – Παρουσίαση Γραφήματος*** |

Να υλοποιήσετε την παρουσίαση αποτελεσμάτων / στατιστικών στοιχείων που περιγράφεται στην απαίτηση R4.

Ένα πλήρες παράδειγμα χρήσης του JfreeChart μπορείτε να βρείτε εδώ:

<http://www.java2s.com/Code/Java/Chart/JFreeChartDualAxisDemo2.htm>

|  |
| --- |
| ***Μαθησιακά Αποτελέσματα***  Στη δραστηριότητα 2.Γ θα σας δοθεί η δυνατότητα:   * να κατανοήσετε τη χρήση βιβλιοθηκών της Java. * να κατανοήσετε την επαναχρησιμοποίηση κώδικα. |

|  |
| --- |
| ***Ερώτημα Δ – Συνολικός Έλεγχος και Εκτέλεση της Εφαρμογής*** |

Για κάθε απαίτηση του πίνακα απαιτήσεων εισάγεται σχετική εικόνα που να δείχνει τη λειτουργία της εφαρμογής. Η κάθε εικόνα θα συνοδεύεται από σύντομο σχολιασμό, ώστε να αποτελεί μια πολύ σύντομη περιγραφή έκτασης μιας παραγράφου, του τρόπου λειτουργίας του συστήματος (σύντομη έκδοση εγχειριδίου χρήσης – user guide).

|  |
| --- |
| ***Μαθησιακά Αποτελέσματα***  Στη δραστηριότητα 2.Ε θα σας δοθεί η δυνατότητα:   * να κατανοήσετε την έννοια του ελέγχου ποιότητας λογισμικού |

|  |
| --- |
| **ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3 – Κριτικός Απολογισμός του Έργου** |

#### Στο τέλος του έργου κάντε έναν αναλυτικό και κριτικό απολογισμό του έργου αναφέροντας:

#### τα προβλήματα που αντιμετωπίσατε, τι θα αλλάζατε στο έργο αν είχατε τη δυνατότητα,

#### τις αποκλίσεις που είχατε στις επαναλήψεις σε σχέση με αυτά που ορίζονται στην εργασία,

#### το χρόνο που απαιτήθηκε για την κάθε δραστηριότητα με βάση τα στοιχεία που κρατήσατε και σε σχέση με το χρόνο που προϋπολογίσατε,

#### πως αξιολογείτε το εργαλείο συνεργασίας trello και με ποιο τρόπο σας βοήθησε στην καθημερινή σας εργασία

#### το πιο πολύτιμο μέλος της ομάδας σας και τον τρόπο που το επιλέξατε,

#### τους κινδύνους που πραγματώθηκαν, καθώς και πώς αποκριθήκατε σε αυτούς,

#### τι θα αλλάζατε σε ένα επόμενο αντίστοιχο έργο, καθώς και

#### ποια ήταν τα θετικά σημεία που αποκομίσατε από αυτή την εργασία.

##### Αναλυτική Αξιολόγηση

|  |  |  |
| --- | --- | --- |
| **Άσκηση** | **Περιγραφή** | **Ποσοστό** |
| ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1 | Διαχείριση του Έργου με έμφαση στη χρήση του εργαλείου συνεργασίας | 25 |
| ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Α | Διάγραμμα Κλάσεων, Υλοποίηση Κλάσεων σε Java | 20 |
| ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Β | Δημιουργία GUI Εφαρμογής | 30 |
| ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Γ | Παρουσίαση Γραφημάτων | 10 |
| ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Δ | Συνολικός Έλεγχος και Εκτέλεση της Εφαρμογής | 10 |
| ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3 | Κριτικός Απολογισμός του Έργου | 5 |
|  | **Σύνολο** | 100 |

|  |
| --- |
| Ημερομηνία ανακοίνωσης εργασίας:, 26/01/2020  Ημερομηνία παράδοσης εργασίας: 11/03/2020  Καταληκτική ημερομηνία παραλαβής σε περίπτωση ατομικής παράδοσης:  ΔΕΝ ΘΑ ΔOΘΟΥΝ ΠΑΡΑΤΑΣΕΙΣ |

**Υποδείξεις για τη συγγραφή της εργασίας**

1. Στο study.eap.gr θα βρείτε τα αρχεία:

* PLH24\_3ERG\_PROTYPO\_SYGGRAFHS.doc που περιέχει το υπόδειγμα συγγραφής της εργασίας. Στο υπόδειγμα θα πρέπει:
  + Να συμπληρώσετε όλα τα στοιχεία με κίτρινο.
  + Μην ξεχάσετε να δηλώσετε εάν η εργασία αποτελεί προϊόν αποκλειστικά δικής σας εργασίας.
  + Αν δεν έχετε απαντήσει σε ένα ερώτημα γράψτε «**ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ**».
  + Αν απαντήσατε με ελλείψεις σε ένα ερώτημα γράψτε «**ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ**».
* Στο υποστηρικτικό υλικό θα βρείτε:
  + 1. Το έγγραφο Netbeans\_Binding που περιέχει παρουσίαση για το Netbeans Binding που είναι απαραίτητο για την εύκολη διασύνδεση των GUI συστατικών με το μοντέλο.
    2. Ένα σύντομο παράδειγμα δημιουργίας βάσης δεδομένων στο Netbeans, δημιουργία entity class από αυτή και αποθήκευσης αντικειμένων στη βάση.
    3. Μια συνοπτική περιγραφή της μεθόδου Scrum.

1. Η εκπόνηση της εργασίας θα πρέπει να γίνει αποκλειστικά με τα εργαλεία:

* Visual Paradigm για τη UML,
* Netbeans για τη Java,
* Derby για ΒΔ.
* Ως εργαλείο για τη διαχείριση έργων θα πρέπει να χρησιμοποιήσετε ένα από τα ακόλουθα εργαλεία.
  + - το ProjectLibre (<https://www.projectlibre.com>). Μπορείτε να κατεβάσετε δωρεάν και να εγκαταστήσετε το λογισμικό στον υπολογιστή σας.
    - το EasyProject (<https://www.easyproject.com>). Μπορείτε να χρησιμοποιήσετε το εργαλείο αφού πρώτα εγγραφείτε στον πιο πάνω δικτυακό τόπο.
    - Ή άλλο αντίστοιχο

1. Βιβλιοθήκες που θα χρειαστείτε για την υλοποίηση:

|  |
| --- |
| <dependencies>  <!—Βιβλιοθήκη για γραφήματα -- >  <dependency>  <groupId>org.jfree</groupId>  <artifactId>jfreechart</artifactId>  <version>1.0.19</version>  </dependency>  <!—Βιβλιοθήκη για εκτέλεση μεθόδων HTTP -- >  <dependency>  <groupId>com.squareup.okhttp3</groupId>  <artifactId>okhttp</artifactId>  <version>3.12.1</version>  </dependency>  <!—Βιβλιοθήκη για JSON Parsing -- >  <dependency>  <groupId>com.google.code.gson</groupId>  <artifactId>gson</artifactId>  <version>2.8.5</version>  </dependency>  </dependencies> |

1. Η εργασία θα εκπονηθεί σε ομάδες:

* Οι ομάδες **θα αποτελούνται από 3 άτομα**. Οι φοιτητές θα αυτό-οργανωθούν και θα επιλέξουν τα μέλη της ομάδας μόνοι τους, σύμφωνα με τις προσωπικές τους προτιμήσεις. Οι ομάδες θα δηλωθούν στον καθηγητή σύμβουλο του κάθε τμήματος μέχρι και την 3η ΟΣΣ του κάθε τμήματος, με την αποστολή σχετικού e-mail (ένα ανά ομάδα). Αν δεν έχετε/μπορείτε να βρείτε ομάδα συνιστούμε να αποστείλετε e-mail στον οικείο ΣΕΠ.
* Η βαθμολογία θα είναι κοινή για όλα τα μέλη της ομάδας.

1. Υποβολή εργασίας:

* Η υποβολή κοινών απαντήσεων μεταξύ ομάδων από διαφορετικές ομάδες φοιτητών που συνεργάστηκαν δεν επιτρέπεται και θεωρείται ως **ΑΝΤΙΓΡΑΦΗ**. Η αντιγραφή έχει ως αποτέλεσμα το **ΜΗΔΕΝΙΣΜΟ ΤΗΣ ΕΡΓΑΣΙΑΣ ΣΥΝΟΛΙΚΑ** και την παραπομπή των παραβατών στην Κοσμητεία της Σχολής Θετικών Επιστημών & Τεχνολογίας, σύμφωνα με τον εσωτερικό κανονισμό του ΕΑΠ. Η μη εφαρμογή αυτού του κανόνα μπορεί να οδηγήσει σε συνολική απόρριψη της εργασίας.
* Η **ίδια** εργασία θα υποβληθεί από όλους τους φοιτητές της κάθε ομάδος. Αυτό χρειάζεται διότι στο Moodle στην παρούσα φάση δεν υποστηρίζει ομαδικές εργασίες.
* Η εργασία θα πρέπει να υποβληθεί με τη μορφή συμπιεσμένου αρχείου zip ή rar. To όνομα του αρχείου θα είναι: PLH24\_3ERG\_EPITHETO1.<rar|zip>. Το EPITHETO1 είναι αυτό του scrum master. Να γίνει χρήση λατινικών χαρακτήρων **ΑΠΟΚΛΕΙΣΤΙΚΑ** για την αποφυγή προβλημάτων με το moodle (study.eap.gr).
* Το μέγεθος του αρχείου **δεν μπορεί να υπερβαίνει τα 10ΜΒ.**
* Το συμπιεσμένο αρχείο θα πρέπει να αποτελείται από:
  + Ένα αρχείο κειμένου με όνομα PLH24\_3ERG\_EPITHETO.doc που θα περιέχει την απάντηση της εργασίας. Όπου επίθετο είναι το επίθετο του SCRUM master.
  + Όλα τα μέλη της ομάδας θα υποβάλουν την ίδια εργασία.
  + To product backlog σε μορφή φύλλου εργασίας xls.
  + Το χρονοδιάγραμμα του έργου σε αρχείο pdf.
  + Τον κατάλογο με τον κώδικα Java όπου θα περιλαμβάνεται το project όπως αυτό δημιουργείται από το εργαλείο Netbeans, το οποίο θα πρέπει να μπορεί να εκτελείται χωρίς αλλαγές από τον καθηγητή. **ΠΡΟΣΟΧΗ.** Για κάθε εργασία να υποβάλλεται **ΜΟΝΟ ΕΝΑ** project java που να περιλαμβάνει την απάντηση σε όλα τα ερωτήματα ΚΑΙ ΟΧΙ ένα project ανά ερώτημα.
  + Τον κατάλογο που θα περιέχει τη ΒΔ Derby.
* H εφαρμογή των παραπάνω κανόνων είναι **ΥΠΟΧΡΕΩΤΙΚΗ**

1. Δεν θα ανακοινωθεί ενδεικτική απάντηση. Αντί για αυτό, οι τρεις καλύτερες εργασίες όλων των τμημάτων, όπως θα αποφασιστεί από την ΟΔΠ, θα βραβευτούν ως οι καλύτερες εργασίες της ΘΕ ΠΛΗ24 και θα αναρτηθούν στο site. Η τριμελής κριτική επιτροπή αποτελείται από τους
   * Ιγνάτιος Δεληγιάννης
   * Τάκης Αλεφραγκής
   * Πάνος Φιτσιλής

**ΚΑΛΗ ΕΠΙΤΥΧΙΑ!**

1. Javascript Object Notation [↑](#footnote-ref-2)
2. Για τη διαχείριση αρχείων τύπου JSON, προτείνεται η χρήση της βιβλιοθήκης GSON της Google.   
   Use case example: <http://tutorials.jenkov.com/java-json/gson-jsonparser.html> [↑](#footnote-ref-3)