**3η ΕΡΓΑΣΙΑ – ΟΜΑΔΙΚΗ**

**Αποτελέσματα του παιχνιδιού Joker της ΟΠΑΠ Α.Ε.**

**Εκφώνηση της εργασίας**

|  |
| --- |
| ***Ομαδική Εργασία – Ανάλυση αποτελεσμάτων παιχνιδιών ΤΖΟΚΕΡ*** |

Η 3η εργασία είναι ομαδική εργασία και έχει ως στόχο την υλοποίηση της εφαρμογής JokerGame-Stats, η οποία επεξεργάζεται δεδομένα που λαμβάνει από την υπηρεσία ιστού (web service) που περιγράφονται στην ιστοσελίδα του ΟΠΑΠ **https://www.opap.gr/web-services**.

Η ΟΠΑΠ ΑΕ παρέχει τις εν λόγω διαδικτυακές υπηρεσίες μέσω web services τύπου REST (Representational State Transfer), προκειμένου να χρησιμοποιηθούν από οποιοδήποτε ενδιαφερόμενο μέρος. Τα σχετικά δεδομένα επιστρέφονται σε μορφή JSON (JavaScript Object Notation).

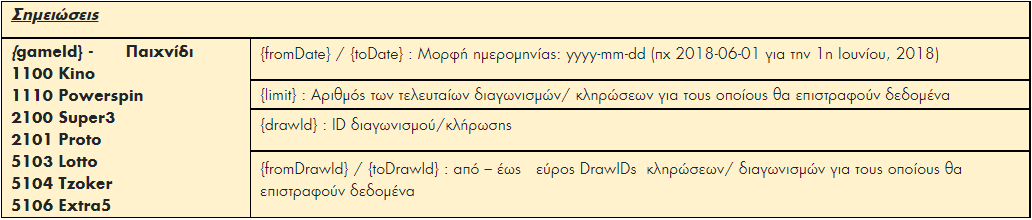
Ένα API είναι ένας κατάλογος λειτουργιών (**operations**) με τη μορφή web διευθύνσεων (Uniform Resource Identifiers - URIs) μαζί με τις παραμέτρους κλήσης αυτών (**request parameters**) των λειτουργιών. Επίσης, ένα API προσδιορίζει τις αναμενόμενες απαντήσεις (**responses**), ακολουθώντας την αρχιτεκτονική REST. Συνήθως, η κλήση του API γίνεται από ένα άλλο σύστημα-πελάτη (client) που το χρησιμοποιεί. Συνεπώς, η ζητούμενη εφαρμογή **JokerGame-Stats** που θα πρέπει να κατασκευαστεί, θαείναι μια εφαρμογή πελάτης, η οποία θα καταναλώνει τα Web Services της ΟΠΑΠ Α.Ε. σχετικά με τα αποτελέσματα του παιχνιδιού Joker.

Για τη χρήση των web services **https://www.opap.gr/web-services** δεν απαιτείται η δημιουργία λογαριασμού ή/και API KEY. Το Swagger αρχείο (σύστημα τεκμηρίωσης) για τα Web Services βρίσκεται στο: https://api.opap.gr/numerics/v1.0/api-swagger

Για τις διαθέσιμες ενέργειες χρήσης του web service δίνεται ο παρακάτω πίνακας:



Και οι συνοδευτικές σημειώσεις για τον πίνακα:



Κατά τη φάση του σχεδιασμού, στην παρούσα εργασία μας ενδιαφέρει η υλοποίηση ενός γραφικού συστήματος διεπαφής (GUI), το οποίο θα εξυπηρετεί όλες τις παραπάνω υπηρεσίες.

Ωστόσο, κατά τη φάση της υλοποίησης, στην παρούσα εργασία μας ενδιαφέρει μόνο το παιχνίδι Joker (Tzoker), οπότε και σύμφωνα με τον παραπάνω πίνακα σημειώσεων πρόκειται για το gameId: 5104

**Παράδειγμα 1:** Σύμφωνα με τα παραπάνω, για να λάβουμε τα αποτελέσματα της πιο πρόσφατης κλήρωσης/ διαγωνισμού του παιχνιδιού Joker θα χρησιμοποιήσουμε την παρακάτω κλήση:

<https://api.opap.gr/draws/v3.0/5104/last-result-and-active>

η οποία επιστρέφει τα αποτελέσματα στην παρακάτω μορφή (JSON format) (τα αποτελέσματα ενδέχεται να διαφέρουν ανάλογα με το πότε πραγματοποιήθηκε η κλήση, αφού αφορούν στο πιο πρόσφατο παιχνίδι):

|  |
| --- |
| {  "last": {  "gameId": 5104,  "drawId": 2390,  "drawTime": 1642363200000,  "status": "results",  "drawBreak": 1800000,  "visualDraw": 2390,  "pricePoints": {  "amount": 0.5  },  "winningNumbers": {  "list": [  1,  29,  26,  24,  17  ],  "bonus": [  6  ]  },  "prizeCategories": [  {  "id": 1,  "divident": 0.0,  "winners": 0,  "distributed": 356871.53,  "jackpot": 748954.15,  "fixed": 0.0,  "categoryType": 0,  "gameType": "Normal",  "minimumDistributed": 0.0  },  {  "id": 2,  "divident": 22575.97,  "winners": 4,  "distributed": 55178.93,  "jackpot": 35124.97,  "fixed": 0.0,  "categoryType": 0,  "gameType": "Normal"  },  {  "id": 3,  "divident": 2500.0,  "winners": 20,  "distributed": 50000.0,  "jackpot": 0.0,  "fixed": 2500.0,  "categoryType": 1,  "gameType": "Normal"  },  {  "id": 4,  "divident": 50.0,  "winners": 326,  "distributed": 16300.0,  "jackpot": 0.0,  "fixed": 50.0,  "categoryType": 1,  "gameType": "Normal"  },  {  "id": 5,  "divident": 50.0,  "winners": 816,  "distributed": 40800.0,  "jackpot": 0.0,  "fixed": 50.0,  "categoryType": 1,  "gameType": "Normal"  },  {  "id": 6,  "divident": 2.0,  "winners": 16634,  "distributed": 33268.0,  "jackpot": 0.0,  "fixed": 2.0,  "categoryType": 1,  "gameType": "Normal"  },  {  "id": 7,  "divident": 2.0,  "winners": 10341,  "distributed": 20682.0,  "jackpot": 0.0,  "fixed": 2.0,  "categoryType": 1,  "gameType": "Normal"  },  {  "id": 8,  "divident": 1.5,  "winners": 49233,  "distributed": 73849.5,  "jackpot": 0.0,  "fixed": 1.5,  "categoryType": 1,  "gameType": "Normal"  }  ],  "wagerStatistics": {  "columns": 2866438,  "wagers": 503579,  "addOn": []  }  },  "active": {  "gameId": 5104,  "drawId": 2391,  "drawTime": 1642536000000,  "status": "active",  "drawBreak": 1800000,  "visualDraw": 2391,  "pricePoints": {  "amount": 0.5  },  "prizeCategories": [  {  "id": 1,  "divident": 0.0,  "winners": 0,  "distributed": 0.0,  "jackpot": 1105825.68,  "fixed": 0.0,  "categoryType": 0,  "gameType": "Normal",  "minimumDistributed": 1300000.0  },  {  "id": 2,  "divident": 0.0,  "winners": 0,  "distributed": 0.0,  "jackpot": 0.0,  "fixed": 0.0,  "categoryType": 0,  "gameType": "Normal"  },  {  "id": 3,  "divident": 0.0,  "winners": 0,  "distributed": 0.0,  "jackpot": 0.0,  "fixed": 2500.0,  "categoryType": 1,  "gameType": "Normal"  },  {  "id": 4,  "divident": 0.0,  "winners": 0,  "distributed": 0.0,  "jackpot": 0.0,  "fixed": 50.0,  "categoryType": 1,  "gameType": "Normal"  },  {  "id": 5,  "divident": 0.0,  "winners": 0,  "distributed": 0.0,  "jackpot": 0.0,  "fixed": 50.0,  "categoryType": 1,  "gameType": "Normal"  },  {  "id": 6,  "divident": 0.0,  "winners": 0,  "distributed": 0.0,  "jackpot": 0.0,  "fixed": 2.0,  "categoryType": 1,  "gameType": "Normal"  },  {  "id": 7,  "divident": 0.0,  "winners": 0,  "distributed": 0.0,  "jackpot": 0.0,  "fixed": 2.0,  "categoryType": 1,  "gameType": "Normal"  },  {  "id": 8,  "divident": 0.0,  "winners": 0,  "distributed": 0.0,  "jackpot": 0.0,  "fixed": 1.5,  "categoryType": 1,  "gameType": "Normal"  }  ],  "wagerStatistics": {  "columns": 0,  "wagers": 0,  "addOn": []  }  }  } |

Σημείωση: Το παραπάνω αρχείο JSON μπορεί να μελετηθεί και σε συνδυασμό με την ιστοσελίδα (<https://www.opap.gr/apotelesmata-klhrwsewn-joker>) ως προς την πιο εύκολη κατανόηση της απεικονιζόμενης πληροφορίας.

**Παράδειγμα 2:** Αντίστοιχα, για να λάβουμε τους κωδικούς αποτελεσμάτων, DrawIds, του παιχνιδιού ΤΖΟΚΕΡ για συγκεκριμένο εύρος ημερομηνιών (συγκεκριμένα για 01-12-2021 έως και 31-12-2021) θα χρησιμοποιήσουμε την παρακάτω κλήση:

<https://api.opap.gr/draws/v3.0/5104/draw-date/2021-12-01/2021-12-31/draw-id>

**Κλήση HTTP**

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

String **urlToCall** =

" https://api.opap.gr/draws/v3.0/5104/last-result-and-active";

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. 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).

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

Επιπλέον, δίδονται οι παρακάτω προδιαγραφές που αφορούν τη λειτουργικότητα τους συστήματος.

|  |  |
| --- | --- |
| Νο | Περιγραφή |
| R1 | Το σύστημα περιλαμβάνει τις παρακάτω επιλογές:   1. Διαχείριση δεδομένων ΤΖΟΚΕΡ 2. Προβολή δεδομένων ΤΖΟΚΕΡ ανά μήνα για συγκεκριμένο έτος 3. Προβολή στατιστικών δεδομένων ΤΖΟΚΕΡ και εκτύπωση σε αρχείο pdf 4. Έξοδος |
| R2 | Με τη διαχείριση των δεδομένων ΤΖΟΚΕΡ εμφανίζεται οθόνη στην οποία ο χρήστης επιλέγει τα δεδομένα που επιθυμεί να αντληθούν από το api. Συγκεκριμένα, επιλέγει είτε το id ενός παιχνιδιού, ώστε να προβληθούν οι λεπτομέρειες για το εν λόγω παιχνίδι, είτε εύρος ημερομηνιών, οπότε προβάλλονται οι λεπτομέρειες για όσα παιχνίδια έχουν γίνει εντός τους εύρους ημερομηνιών.  Οι λεπτομέρειες που μας ενδιαφέρουν να αντληθούν για κάθε παιχνίδι είναι οι ακόλουθες:   * Τυχεροί αριθμοί (νικητήρια στήλη) * Βασικές κατηγορίες επιτυχιών, όπου για κάθε κατηγορία θέλουμε να γνωρίζουμε τον αριθμό επιτυχιών και τα κέρδη ανά επιτυχία   Εν συνεχεία, ο χρήστης έχει τη δυνατότητα να αποθηκεύσει τα δεδομένα που ανακτήθηκαν από το api σε ΒΔ. Σε αυτή την περίπτωση προηγείται έλεγχος ύπαρξης στη βάση δεδομένων των δεδομένων αυτών (από προηγούμενη αναζήτηση) και αν δεν υπάρχουν, αποθηκεύονται στη βάση δεδομένων.  Στην οθόνη διαχείρισης των δεδομένων εμφανίζονται επιπλέον τα πλήκτρα «Διαγραφή δεδομένων παιχνιδιού» και «Διαγραφή δεδομένων παιχνιδιών εντός εύρους ημερομηνιών»  Με την επιλογή της διαγραφής δεδομένων παιχνιδιού διαγράφονται από τη βάση δεδομένων τα σχετικά δεδομένα βάζει του id του εν λόγω παιχνιδιού και αντίστοιχα υλοποιείται και η διαγραφή παιχνιδιών εντός ορισμένου εύρους ημερομηνιών.  Η ολοκλήρωση οποιασδήποτε διαγραφής δεδομένων γίνεται αφού προηγηθεί ερώτημα επαλήθευσης με επιβεβαίωση από το χρήστη. |
| R3 | Στην οθόνη προβολής δεδομένων ΤΖΟΚΕΡ ανά μήνα για συγκεκριμένο έτος ο χρήστης έχει τη δυνατότητα να επιλέξει από μια λίστα τα διαθέσιμα έτη για τα οποία μπορούν να προβληθούν τα δεδομένα του κάθε μήνα. Με την επιλογή ενός έτους, εμφανίζονται οι 12 μήνες και έπειτα σε νέα σελίδα μπορεί ο χρήστης να δει τα συγκεντρωτικά δεδομένα ανά μήνα.  Για κάθε μήνα μας ενδιαφέρουν συγκεντρωτικά δεδομένα, συγκεκριμένα:   * Πόσα παιχνίδια έγιναν * Πόσα χρήματα διανεμήθηκαν * Πόσα ΤΖΑΚ-ΠΟΤ έγιναν |
| R4 | Με την επιλογή προβολής στατιστικών δεδομένων ΤΖΟΚΕΡ και εκτύπωση σε αρχείο pdf, ουσιαστικά χρησιμοποιούμε το api call [https://api.opap.gr/games/v1.0/{gameId}/statistics](https://api.opap.gr/games/v1.0/%7bgameId%7d/statistics) με το gameid του ΤΖΟΚΕΡ, το οποίο επιστρέφει τα στατιστικά των αριθμών που έχουν κληρωθεί σε ένα δοθέν εύρος ημερομηνιών. Τα στατιστικά των αριθμών πρέπει να μπορούν να προβληθούν υπό τη μορφή πίνακα, ενώ στη συνέχεια με την ύπαρξη αντίστοιχης επιλογής, δίνεται η δυνατότητα εκτύπωσης του παραγόμενου πίνακα σε αρχείο PDF. |
| R5 | Αντίστοιχα με την απαίτηση R4, όταν ο χρήστης πατήσει το πλήκτρο «Προβολή στατιστικών στοιχείων κληρώσεων σε γραφική μορφή» εμφανίζεται οθόνη, η οποία παρουσιάζει τα στατιστικά στοιχεία για τα δεδομένα που είναι αποθηκευμένα στη βάση δεδομένων:   * Συχνότητας εμφάνισης αριθμών (πέντε πιο συχνά εμφανιζόμενοι αριθμοί) * Συχνότητα εμφάνισης αριθμών joker (πέντε πιο συχνά εμφανιζόμενοι αριθμοί) * Μέσος όρος κερδών ανά κατηγορία   Επιπλέον,   * Ο χρήστης επιλέγει κάθε φορά ποια στατιστικά στοιχεία θα παραχθούν. * Υπάρχει δυνατότητα ορισμού του χρονικού διαστήματος για το οποίο θα εμφανιστούν τα στατιστικά στοιχεία. Και εδώ ισχύουν οι κανόνες διαμόρφωσης ημερομηνιών που αναφέρθηκαν στην απαίτηση R2. |

|  |
| --- |
| **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 φοιτητές (κατ’ εξαίρεση 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, οι δύο πρώτες επαναλήψεις θα είναι διάρκειας δύο εβδομάδων, ενώ η άλλη διάρκειας δέκα ημερών.

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=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 Entity-Relationship (ER) του συστήματος και να κατασκευάσετε το script δημιουργίας των πινάκων της βάσης. Επεξηγήστε σύντομα το διάγραμμα καθώς και το περιεχόμενο των πεδίων του κάθε πίνακα.
2. Να σχεδιάσετε το τελικό διάγραμμα κλάσεων του συστήματος JokerGame-Stats. 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.
3. Εισάγετε τον κώδικα για τις κλάσεις του συστήματος.
4. 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 – Υλοποίηση συστήματος*** |

Να υλοποιήσετε την παρουσίαση αποτελεσμάτων / στατιστικών στοιχείων που περιγράφεται στις απαιτήσεις R1 έως και R4.

|  |
| --- |
| ***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 ER, Διάγραμμα Κλάσεων και Υλοποίηση Κλάσεων σε Java (περιλαμβάνει χρήση API και επεξεργασία json) | 20 |
| ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Β | Δημιουργία GUI Εφαρμογής | 15 |
| ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Γ | Παρουσίαση Στατιστικών Στοιχείωνκαι Γραφημάτων | 20 |
| ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Δ | Συνολικός Έλεγχος και Εκτέλεση της Εφαρμογής | 15 |
| ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3 | Κριτικός Απολογισμός του Έργου | 5 |
|  | **Σύνολο** | 100 |

|  |
| --- |
| Ημερομηνία ανακοίνωσης εργασίας: 24/01/2022  Ημερομηνία παράδοσης εργασίας: 09/03/2022  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 το συνεργατικό εργαλείο <http://Trellο.com>

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,** αλλά μπορούν να ανεβούν πολλά αρχεία.
* 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 καθώς και το script δημιουργίας της ΒΔ.
* H εφαρμογή των παραπάνω κανόνων είναι **ΥΠΟΧΡΕΩΤΙΚΗ**
* 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, ενώ θα δοθεί σχετικό πιστοποιητικό. Η τριμελής κριτική επιτροπή αποτελείται από τους:
  + Ευθύμιο Αλέπη
  + Συρμακέση Σπύρο
  + Γκουμόπουλο Χρήστο

**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)