

# ΤΡΙΤΗ ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ

## ΟΜΑΔΙΚΗ

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2025-2026

### Εφαρμογή Προβολής Πληροφοριών WikiViewer

#### Εκφώνηση της εργασίας

##### *Ομαδική Εργασία – Εφαρμογή Προβολής Πληροφοριών Wikipedia (WikiViewer)*

Η 3η εργασία είναι ομαδική εργασία και έχει ως στόχο την υλοποίηση της εφαρμογής «WikiViewer», η οποία επεξεργάζεται δεδομένα που αφορούν σε άρθρα της ελληνικής εγκυκλοπαίδειας Wikipedia. Τα δεδομένα λαμβάνονται από το επίσημο API της MediaWiki που περιγράφεται στην ιστοσελίδα: [https://www.mediawiki.org/wiki/API:Main\\_page](https://www.mediawiki.org/wiki/API:Main_page).

Η συγκεκριμένη υπηρεσία παρέχεται και μέσω υπηρεσιών ιστού (web services) τύπου REST (Representational State Transfer), προκειμένου να χρησιμοποιηθούν από οποιοδήποτε ενδιαφερόμενο μέρος. Τα σχετικά δεδομένα επιστρέφονται σε διάφορες μορφές, ωστόσο εμάς μας ενδιαφέρει η μορφή JSON (JavaScript Object Notation), η οποία και υποστηρίζεται.

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 **WikiViewer** που θα πρέπει να κατασκευαστεί, θα είναι μια εφαρμογή πελάτη, η οποία θα καταναλώνει τα Web Services της υπηρεσίας που παρέχεται από το προαναφερόμενο web service.

Περισσότερες πληροφορίες για τη χρήση του API (Sandbox) μπορείτε να βρείτε στο URL: <https://el.wikipedia.org/wiki/Special:ApiSandbox>.

Η συγκεκριμένη υπηρεσία παρέχεται δωρεάν και δεν απαιτείται η δημιουργία λογαριασμού ή/και API KEY. Τα δεδομένα επιστρέφονται σε μορφή JSON. Η εφαρμογή WikiViewer που θα κατασκευαστεί, θα είναι μια εφαρμογή πελάτη (client-app), η οποία θα ανακτά τίτλους και περιλήψεις άρθρων.

Κατά τη φάση του σχεδιασμού, στην παρούσα εργασία μας ενδιαφέρει η υλοποίηση ενός γραφικού συστήματος διεπαφής (GUI), το οποίο θα εξυπηρετεί την παρουσίαση πληροφοριών άρθρων της Wikipedia για τις επιλογές:

1. Αναζήτηση άρθρου (Live από το API)
2. Ενημέρωση πληροφοριών άρθρου (στην τοπική ΒΔ)
3. Προσθήκη πληροφοριών για άρθρο (π.χ. σχόλια, βαθμολογία)
4. Προβολή αποθηκευμένων άρθρων ανά κατηγορία
5. Αναζήτηση άρθρου βάσει keyword (σε τίτλο ή κείμενο)

6. Προβολή στατιστικών
7. Προσθήκη νέου άρθρου σε κατηγορία (Χειροκίνητη εισαγωγή)
8. Τροποποίηση άρθρων σε κατηγορία
9. Προσθήκη κατηγορίας

Ωστόσο, κατά τη φάση της υλοποίησης, στην παρούσα εργασία **δεν μας ενδιαφέρει να υλοποιηθούν όλες οι επιλογές, αλλά οι πρώτες 6**. Με απλά λόγια ο **σχεδιασμός** θα περιλαμβάνει περισσότερες λειτουργίες, ενώ η **υλοποίηση** θα εστιάσει σε ένα υποσύνολο της λειτουργικότητας.

#### Παράδειγμα κλήσης του web service:

Σύμφωνα με τα παραπάνω, για να αναζητήσουμε άρθρα που σχετίζονται με τη λέξη "Ελλάδα", θα καλέσουμε το API ως εξής:

<https://el.wikipedia.org/w/api.php?action=query&list=search&srsearch=Ελλάδα&format=json>

Το οποίο επιστρέφει τα αποτελέσματα στην παρακάτω μορφή (JSON format) (ενδεικτική εκτύπωση από μέρος των αποτελεσμάτων):

```
{
  "batchcomplete": "",
  "query": {
    "searchinfo": {
      "totalhits": 28450
    },
    "search": [
      {
        "ns": 0,
        "title": "Ελλάδα",
        "pageid": 412,
        "size": 185432,
        "wordcount": 18210,
        "snippet": "Η <span class=\"searchmatch\">Ελλάδα</span> (επίσημα: Ελληνική Δημοκρατία), είναι χώρα της Νοτιοανατολικής Ευρώπης...",
        "timestamp": "2024-03-15T10:00:00Z"
      },
      {
        "ns": 0,
        "title": "Ιστορία της Ελλάδας",
        "pageid": 12345,
        "size": 54321,
        "wordcount": 5000,
        "snippet": "Η ιστορία της <span class=\"searchmatch\">Ελλάδας</span> αφορά την ιστορία του ελληνικού έθνους...",
        "timestamp": "2024-02-20T14:30:00Z"
      }
    ]
  }
}
```

#### Κλήση HTTP

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

```
String urlToCall =
"your url goes 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<sup>1</sup>. 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 ] <sup>2</sup>.

#### Επιπλέον Προδιαγραφές Λειτουργικότητας του Συστήματος

No	Περιγραφή
R1	<p>Το σύστημα περιλαμβάνει τις παρακάτω επιλογές:</p> <ol style="list-style-type: none"> <li>1. Αναζήτηση άρθρου (Live από API)</li> <li>2. Προβολή αποθηκευμένων άρθρων / Προβολή αποθηκευμένων άρθρων ανά κατηγορία</li> <li>3. Ενημέρωση/Τροποποίηση στοιχείων άρθρου</li> <li>4. Αναζήτηση άρθρων βάσει λέξης-κλειδιού (Keyword Search)</li> <li>5. Προβολή στατιστικών</li> <li>6. Έξοδος</li> </ol>
R2	<p>Κατά την αναζήτηση άρθρων βάσει ενός keyword, δύναται να επιστραφούν πολλά αποτελέσματα (λίστα τίτλων/αποσπασμάτων) ή κανένα. Η προβολή των πληροφοριών (Τίτλος, Snippet, Timestamp) θα πρέπει να γίνεται σε επεξεργασμένη μορφή στο GUI της εφαρμογής (π.χ. αφαίρεση των HTML tags όπως &lt;span class="searchmatch"&gt;) και όχι στο αρχικό JSON format.</p>

<sup>1</sup> Javascript Object Notation

<sup>2</sup> 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>

	<p><u>Σημείωση:</u> Η αφαίρεση των HTML tags μπορεί να γίνει</p> <ul style="list-style-type: none"> <li>• με χρήση Regular Expressions (Regex) και τη χρήση της σχετικής βιβλιοθήκης (<a href="https://www.w3schools.com/java/java_regex.asp">https://www.w3schools.com/java/java_regex.asp</a>), ή</li> <li>• με πιο εξειδικευμένες βιβλιοθήκες όπως η Jsoup (<a href="https://jsoup.org/">https://jsoup.org/</a>). Αυτό είναι ένα σημείο όπου η κάθε ομάδα θα πρέπει να κάνει τις επιλογές της.</li> </ul>
<b>R3</b>	<p>Στην εφαρμογή σας θα διατηρείται μια ΒΔ στην οποία θα αποθηκεύονται τα άρθρα που επιλέγει ο χρήστης από τα αποτελέσματα του API. Η εφαρμογή θα έχει τη δυνατότητα να «επεκτείνει» την πληροφορία.</p> <p>Συγκεκριμένα, για κάθε αποθηκευμένο άρθρο, ο χρήστης θα μπορεί να προσθέσει πεδία όπως:</p> <ul style="list-style-type: none"> <li>- Κατηγορία (π.χ. Ιστορία, Επιστήμη)</li> <li>- Προσωπικά Σχόλια/Σημειώσεις</li> <li>- Βαθμολογία ενδιαφέροντος (1-5)</li> </ul>
<b>R4</b>	<p>Κατά την αναζήτηση (διαδικτυακά μέσω API), ο χρήστης θα πληκτρολογεί έναν όρο αναζήτησης. Τα αποτελέσματα θα εμφανίζονται σε λίστα. Ο χρήστης θα μπορεί να επιλέξει ένα άρθρο από τη λίστα για να το αποθηκεύσει στη ΒΔ ή να δει τις λεπτομέρειές του.</p> <p><u>Σημείωση:</u> Σε περίπτωση που ένα άρθρο έχει ήδη αποθηκευτεί και τροποποιηθεί (π.χ. έχουν προστεθεί σχόλια), κατά την προβολή του θα πρέπει να εμφανίζονται τα εμπλουτισμένα δεδομένα από τη ΒΔ σας και όχι μόνο τα αρχικά από το web service.</p>
<b>R5</b>	<p>Με την επιλογή προβολής στατιστικών και εκτύπωση σε PDF, αναζητάμε μέσα στην ΒΔ:</p> <ol style="list-style-type: none"> <li>1. Ποιες είναι οι πιο συχνές λέξεις-κλειδιά που έχει αναζητήσει ο χρήστης.</li> <li>2. Πόσα άρθρα έχουν αποθηκευτεί ανά Κατηγορία (που όρισε ο χρήστης στο R3).</li> </ol> <p>Τα στατιστικά αυτά θα προβάλλονται σε πίνακα και θα δίνεται δυνατότητα εξαγωγής σε PDF.</p> <p><u>Σημείωση:</u> Η ομάδα θα πρέπει να επιλέξει τη βιβλιοθήκη που θα χρησιμοποιηθεί. Δημοφιλείς βιβλιοθήκες είναι:</p> <ul style="list-style-type: none"> <li>• IText <a href="https://kb.itextpdf.com/itext/examples">https://kb.itextpdf.com/itext/examples</a></li> <li>• PDFBox <a href="https://pdfbox.apache.org/index.html">https://pdfbox.apache.org/index.html</a></li> </ul>

### Σημειώσεις

- 1) Οι απαιτήσεις εμπεριέχουν ένα βαθμό “δημιουργικής ασάφειας”. Η ασάφεια δίνει τη δυνατότητα στην κάθε ομάδα να αντιμετωπίσει την εργασία κάνοντας τις αναγκαίες παραδοχές, τεκμηριώνοντας αυτές κατάλληλα. Οι παραδοχές θα αναφέρονται στην αρχή της εργασίας στο σχετικό σημείο.
- 2) Θα πρέπει να υπάρχει πλήρης διαχείριση σφαλμάτων με χρήση exceptions που προέρχονται είτε από τη χρήση APIs, βιβλιοθηκών ή χρήση ΒΔ.
- 3) Προσθέστε αναλυτικό σχολιασμό σε κάθε κλάση και μέθοδο, ώστε να δημιουργήσετε αυτόματα τεκμηρίωση. Παράδειγμα:

```
/**
 * This class handles user authentication.
 */
public class AuthService {
    /**
     * An example of documentation
     * Authenticates a user based on username and password.
     * @param username the user's username
     * @param password the user's password
     * @return true if authentication is successful, false otherwise
     */
    public boolean authenticate(String username, String password) {
        // implementation
    }
}
```

- 4) Χρησιμοποιήστε εργαλεία όπως Javadoc Generator για να παραχθεί HTML-based τεκμηρίωση.

### ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1 – Διαχείριση Έργου

Η διαχείριση έργου θα πρέπει να γίνει με βάση τη μέθοδο Scrum. Το Scrum είναι μια σύγχρονη ευέλικτη μέθοδος, μια σύντομη εισαγωγή της οποίας δίνεται στο υποστηρικτικό υλικό 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 τελευταία θα είναι διάρκειας δέκα ημερών. Σε κάθε επανάληψη η ομάδα θα πρέπει να αποφασίζει για το ποιες απαιτήσεις του product backlog θα συμπεριληφθούν στο sprint. Για το λόγο αυτό θα πρέπει να οργανωθεί ένα sprint planning meeting στο οποίο θα γίνει η ιεράρχηση των ιστοριών χρηστών. Στην απάντησή σας θα πρέπει να συμπεριλάβετε στιγμιότυπα οθόνης (screenshots) όπου θα φαίνεται η συμμετοχή του κάθε μέλους της ομάδας καθώς και **η ημερομηνία/ώρα της συνάντησης εντός της εικόνας**.

Για την ευκολότερη διαχείριση του έργου προτείνεται η 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=3ISp-T4Fk5U>

Καλό θα ήταν να γίνει χρήση ενός template που να υποβοηθά την όλη διαδικασία (π.χ. scrum template). 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 ιεράρχηση, ώστε να αποφασιστεί ποιες απαιτήσεις θα συμπεριληφθούν σε κάθε sprint. Για την ιεράρχηση 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. 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.

4. Για καθένα από τα user stories του product backlog να περιγράψετε τα **κριτήρια αποδοχής** (acceptance criteria).
5. **Οι αναθέσεις των αρμοδιοτήτων** στα μέλη της ομάδας σε κάθε επανάληψη (sprint). Δηλαδή ποιο user story υλοποίησε ο κάθε φοιτητής.
6. Ο **χρόνος υλοποίησης** ανά απαίτηση/ανά επανάληψη (sprint) σε σύγκριση με αυτά που είχαν προϋπολογισθεί, καθώς και το διάγραμμα κατανάλωσης προσπάθειας (burndown 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, καθώς και,
- Της εκτίμησης του κόστους.

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2 – Υλοποίηση Συστήματος

### Ερώτημα Α – Διάγραμμα E-R, Διάγραμμα Κλάσεων και Υλοποίηση Κλάσεων σε Java.

Στο ερώτημα αυτό θα πρέπει:

- Να σχεδιάσετε το διάγραμμα Entity-Relationship (ER) της ΒΔ 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.
- Να σχεδιάσετε το **συγκεντρωτικό διάγραμμα κλάσεων του συστήματος WikiViewer** που θα ικανοποιεί όλες τις απαιτήσεις που περιγράφονται και όχι μόνο ζητούνται να υλοποιηθούν. Στο διάγραμμα θα πρέπει εκτός από τις κλάσεις του εννοιολογικού μοντέλου να εμφανίζονται οι κλάσεις του GUI, οι POJO κλάσεις<sup>3</sup> και όποιες άλλες κλάσεις κρίνετε απαραίτητες. Επίσης το διάγραμμα θα πρέπει να είναι σε πλήρη αντιστοιχία με τον κώδικα του συστήματος στο τμήμα που έχει υλοποιηθεί
- Να εισάγετε τον κώδικα για τις κλάσεις του συστήματος.
- 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) (προαιρετικό).

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.

<sup>3</sup> Υποχρεωτική χρήση Java Persistence API (JPA) για τη διασύνδεση της εφαρμογής με τη ΒΔ



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 IDE τη γραφική διεπαφή χρήστη,
- να υλοποιήσετε τη διασύνδεση της εφαρμογής με τη ΒΔ.

#### **Ερώτημα Γ – Υλοποίηση συστήματος**

Να υλοποιήσετε την παρουσίαση αποτελεσμάτων / στατιστικών στοιχείων που περιγράφεται στις απαιτήσεις R1 έως και R5.

#### **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 να εισάγετε 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.Δ θα σας δοθεί η δυνατότητα:

- να κατανοήσετε την έννοια του ελέγχου ποιότητας λογισμικού

### ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3 – Κριτικός Απολογισμός του Έργου

Στο τέλος του έργου να κάνετε έναν αναλυτικό και κριτικό απολογισμό του έργου αναφέροντας:

- 1) τα προβλήματα που αντιμετωπίσατε, τι θα αλλάζατε στο έργο αν είχατε τη δυνατότητα,
- 2) τις αποκλίσεις που είχατε στις επαναλήψεις σε σχέση με αυτά που ορίζονται στην εργασία,
- 3) το χρόνο που απαιτήθηκε για την κάθε δραστηριότητα με βάση τα στοιχεία που κρατήσατε και σε σχέση με το χρόνο που προϋπολογίσατε,
- 4) πως αξιολογείτε το εργαλείο συνεργασίας Trello και με ποιο τρόπο σας βοήθησε στην καθημερινή σας εργασία
- 5) τους κινδύνους που πραγματώθηκαν, καθώς και πώς αποκριθήκατε σε αυτούς,
- 6) τι θα αλλάζατε σε ένα επόμενο αντίστοιχο έργο, καθώς και
- 7) ποια ήταν τα θετικά σημεία που αποκομίσατε από αυτή την εργασία.

### ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4 – Παρουσίαση του έργου με μορφή video

Η ολοκλήρωση του έργου απαιτεί τη δημιουργία video διάρκειας έως 15 λεπτά το οποίο θα παρουσιάζει με σύντομο τρόπο

- 1) Το σχεδιασμό του συστήματος καθώς και τις παραδοχές που έγιναν,
- 2) τη λειτουργία του συστήματος σε μορφή (user guide), καθώς και
- 3) τον κριτικό απολογισμό.

**Αναλυτική Αξιολόγηση**

Άσκηση	Περιγραφή	Ποσοστό
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1	Διαχείριση του Έργου με έμφαση στη χρήση του εργαλείου συνεργασίας	15
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Α	Διάγραμμα ER, Διάγραμμα Κλάσεων και Υλοποίηση Κλάσεων σε Java (περιλαμβάνει χρήση API και επεξεργασία JSON)	15
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Β	Δημιουργία GUI Εφαρμογής	20
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Γ	Παρουσίαση Στατιστικών Στοιχείων	20
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2.Δ	Συνολικός Έλεγχος και Εκτέλεση της Εφαρμογής	15
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3	Κριτικός Απολογισμός του Έργου	5
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4	Δημιουργία video	10
Σύνολο		100

Ημερομηνία ανακοίνωσης εργασίας: 21/01/2026

Ημερομηνία παράδοσης εργασίας: 04/03/2026

**ΔΕΝ ΘΑ ΔΟΘΟΥΝ ΠΑΡΑΤΑΣΕΙΣ**

### Υποδείξεις για τη συγγραφή της εργασίας

- 1) Στο study.eap.gr θα βρείτε τα αρχεία:
  - PLH24\_3ERG\_PROTYPO\_SYGGRAPHHS.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. Τη μέθοδο Scrum από το βιβλίο του Πάνου Φιτσιλή «Ευέλικτες Μέθοδοι Διοίκησης και Διαχείρισης Έργων» (2023) (<https://repository.kallipos.gr/handle/11419/9306>).
- 2) 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://Trello.com>
- 3) 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.
- 4) 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 ομάδας.
  - Η εργασία θα πρέπει να υποβληθεί με τη μορφή συμπιεσμένου αρχείου zip ή rar. Το όνομα του αρχείου θα είναι: PLH24\_3ERG\_ΕΡΙΤΗΤΟ1.<rar|zip>. Το ΕΡΙΤΗΤΟ1 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 10MB**, 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.
  - Το 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 html κατάλογο/σελίδα με Javadoc documentation
  - Τον κατάλογο που θα περιέχει τη ΒΔ Derby καθώς και το script δημιουργίας της ΒΔ.
- Το video θα μεταφορτωθεί είτε στο YouTube, είτε στο teams, είτε σε άλλη δικτυακή πλατφόρμα και θα δοθεί το Link.
- Η 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
  - Πάνο Φιτσιλή
  - Χρήστο Γκουμόπουλο

**ΚΑΛΗ ΕΠΙΤΥΧΙΑ!**