|  |
| --- |
| **ΠΡΩΤΗ ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ**  **ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2021-2022** |

ΣΤΟΙΧΕΙΑ ΠΟΥ ΣΥΜΠΛΗΡΩΝΕΙ Ο ΦΟΙΤΗΤΗΣ / Η ΦΟΙΤΗΤΡΙΑ

|  |  |
| --- | --- |
| Ονοματεπώνυμο Φοιτητή/Φοιτήτριας: | ΕΥΑΓΓΕΛΟΣ ΜΠΑΤΣΑΛΗΣ |
| Αριθμός Μητρώου: | ΑΜ: 119181 |
| Κωδικός Θ.Ε.: | **ΠΛΗ24** |
| Κωδικός Τμήματος: | ΗΛΕ41 |
| Α/Α Γραπτής Εργασίας: | **1η ΕΡΓΑΣΙΑ** |
| Ονοματεπώνυμο Καθηγητή: | ΦΙΤΣΙΛΗΣ ΠΑΝΑΓΙΩΤΗΣ |
| Σχόλια προς Καθηγητή: |  |

***Υπεύθυνη Δήλωση Φοιτητή/Φοιτήτριας****: Βεβαιώνω ότι είμαι συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία αυτής της εργασίας, είναι πλήρως αναγνωρισμένη και αναφέρεται, είτε στο σημείο «Σχόλια προς καθηγητή», είτε μέσα στην εργασία. Επίσης, έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς, είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τη συγκεκριμένη Θεματική Ενότητα.*

|  |  |
| --- | --- |
| NAI | Συμφωνώ και αποδέχομαι την ανωτέρω δήλωση |
|  | Δε συμφωνώ και δεν αποδέχομαι την ανωτέρω δήλωση (στην περίπτωση αυτή, ο Κ-Σ έχει δικαίωμα να μην αξιολογήσει την εργασία του φοιτητή/της φοιτήτριας) |

**ΣΤΟΙΧΕΙΑ ΠΟΥ ΣΥΜΠΛΗΡΩΝΕΙ Ο ΚΑΘΗΓΗΤΗΣ**

|  |  |
| --- | --- |
| **Ημερομηνία Αξιολόγησης:** |  |
| **Τελικός Βαθμός:** |  |

**Σχόλια προς το Φοιτητή/Φοιτήτρια:**

|  |
| --- |
| **Ημερομηνία ανακοίνωσης εργασίας:** Πέμπτη, 21/10/2021  **Ημερομηνία υποβολής εργασίας:** Τετάρτη, 17/11/2021, 11:59 μ.μ.  **Ημερομηνία υποβολής εργασίας με ατομική παράταση:** Τετάρτη 24/11/2021, 11:59 μ.μ.  *Ο κάθε φοιτητής/φοιτήτρια δικαιούται* ***μια*** *ατομική παράταση σε όλη τη διάρκεια του ακαδημαϊκού έτους* |

**Αναλυτική Αξιολόγηση:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Άσκηση** | **Περιγραφή** | **Ποσοστό %** | **Βαθμός** |
| **1.** | ***Σύστημα Πλατφόρμας Ανάπτυξης Χαμηλού Κώδικα*** | **25** |  |
| 1.Α | Δημιουργία Διαγράμματος Περιπτώσεων Χρήσης (Π.Χ.) | 15 |  |
| 1.Β | Ορισμός Ιστοριών Χρηστών | 10 |  |
| **2.** | ***Σύστημα Ηλεκτρονικού Παιχνιδιού για την Ελληνική Επανάσταση του 1821*** | **15** |  |
| 2.Α | Δημιουργία Εννοιολογικού Μοντέλου | 15 |  |
| **3.** | ***Σύστημα Προσομοίωσης Αγώνων Μπάσκετ*** | **50** |  |
| 3.A | Υλοποίηση Μεθόδου showStats | 10 |  |
| 3.B | Υλοποίηση Μεθόδου showPlayersStats | 15 |  |
| 3.Γ | Επέκταση Κώδικα Εφαρμογής | 20 |  |
| 3.Δ | Δημιουργία της main και Εκτέλεσή της | 5 |  |
| **4.** | ***Συμμόρφωση με τους Κανόνες Συγγραφής*** | **10** |  |
|  | **Σύνολο:** | | **100** |
| **Εκφώνηση 1ης Άσκησης**  **Σύστημα Πλατφόρμας Ανάπτυξης Χαμηλού Κώδικα** | | | | | |

Μια πλατφόρμα ανάπτυξης χαμηλού κώδικα (Low-Code Development Platform, LCDP) παρέχει ένα διαδικτυακό (web-based) περιβάλλον ταχείας ανάπτυξης εφαρμογών λογισμικού με ελάχιστο κώδικα ή ακόμη και χωρίς καθόλου κώδικα, χρησιμοποιώντας γραφικές διεπαφές χρήστη (Graphical User Interfaces, GUIs) για το γρήγορο ορισμό της επιχειρησιακής λογικής και την κατασκευή φορμών. Για το σύστημα αυτής της διαδικτυακής πλατφόρμας ισχύουν τα ακόλουθα:

Για να είναι σε θέση ένας χρήστης να χρησιμοποιήσει το σύστημα θα πρέπει να ***εγγραφεί*** σε αυτό καταχωρώντας ένα όνομα χρήστη και έναν κωδικό πρόσβασης και στη συνέχεια να χρησιμοποιήσει αυτά τα στοιχεία για να πραγματοποιήσει την ***είσοδό*** του ***στο σύστημα***.

Χρήστες του συστήματος μπορεί να είναι προγραμματιστές (ακόμη και με ελάχιστες γνώσεις / εμπειρία), μηχανικοί λογισμικού (οι οποίοι μπορεί να είναι και προγραμματιστές) και τελικοί χρήστες.

Ένας προγραμματιστής ***καθορίζει*** τη ***λειτουργικότητα*** μιας εφαρμογής βασιζόμενος στις ***απαιτήσεις*** που έχουν ***καταγράψει*** οι τελικοί χρήστες. Πιο συγκεκριμένα, πραγματοποιεί κάθε φορά τις ακόλουθες δύο ενέργειες:

* ***Σχεδιάζει*** το ***μοντέλο του κώδικα*** με οπτικό τρόπο και ανάλογα με τις ιδιαιτερότητες της εφαρμογής είναι πιθανό να ***αναζητήσει έτοιμες στοιχειώδεις υπηρεσίες λογισμικού*** (components) σε εξωτερική του συστήματος, κατάλληλη βιβλιοθήκη επαναχρησιμοποιήσιμου λογισμικού ή να προχωρήσει στην (περιορισμένη και στοχευμένη) ***συγγραφή κώδικα***. Με την ολοκλήρωση της σχεδίασης του μοντέλου του κώδικα ***παράγεται*** με αυτόματο τρόπο ο ***κώδικας*** που υλοποιεί την εφαρμογή.
* ***Προσδιορίζει*** μια σειρά από ***ειδικές απαιτήσει***ς που είναι πιθανό να χαρακτηρίζουν τη συγκεκριμένη εφαρμογή, όπως είναι (ανάμεσα σε άλλες απαιτήσεις) η ***υποστήριξη κινητού υπολογισμού*** (mobility), η ***υποστήριξη υπολογιστικού νέφους*** (cloud) και η ***υποστήριξη Διαδικτύου των Πραγμάτων*** (Internet of Things, IoT).

Τέλος, ο μηχανικός λογισμικού, σε συνεργασία με τον προγραμματιστή, προχωρούν στην ***ολοκλήρωση*** (integration) της ***εφαρμογής*** με άλλες πλατφόρμες λογισμικού. Για το σκοπό αυτό είναι αναγκαία, τόσο η ***δημιουργία κατάλληλων προγραμματιστικών διεπαφών εφαρμογής*** (Application Programming Interfaces, APIs), όσο και η ***υποστήριξη ανοικτών δεδομένων*** (open data). Τελικός στόχος αυτής της ολοκλήρωσης, ο οποίος όμως δεν είναι πάντα εφικτός, αλλά εξαρτάται από το είδος της εφαρμογής, είναι η ***δημιουργία*** μιας ***εφαρμογής SaaS*** (Software as a Service, SaaS).

**Ερώτημα 1.Α – Δημιουργία Διαγράμματος Περιπτώσεων Χρήσης (Π.Χ.)**

Να σχεδιάσετε το διάγραμμα περιπτώσεων χρήσης του συστήματος που αντιστοιχεί στην παραπάνω περιγραφή, στην οποία σημειώνονται με ***έντονα και πλάγια γράμματα*** οι λειτουργίες που θα πρέπει αυτό να υποστηρίζει, αφού πρώτα εντοπίσετε τους χειριστές του συστήματος και ορίσετε τα όριά του. Στη συνέχεια, να τεκμηριώσετε σύντομα τις επιλογές σας αιτιολογώντας τις σχέσεις που έχετε χρησιμοποιήσει ανάμεσα στους χειριστές και ανάμεσα στις περιπτώσεις χρήσης του συστήματος.

***Προαιρετική Δραστηριότητα***

Όσοι/όσες ενδιαφέρονται να γνωρίσουν καλύτερα τις ιδιαιτερότητες της ανάπτυξης εφαρμογών χρησιμοποιώντας πλατφόρμες ανάπτυξης χαμηλού κώδικα, όπως το σύστημα που περιγράφεται στην εκφώνηση της άσκησης, μπορούν να διαβάσουν την πιο κάτω δημοσίευση:

Talesra, Κ., Nagaraja, G.S., “*Low-Code Platform for Application Development*”, International Journal of Applied Engineering Research, Vol. 16, No. 5, 2021. pp. 346-351.

(<https://www.ripublication.com/ijaer21/ijaerv16n5_02.pdf>)

***(Σύνολο: 15 μονάδες)***

***(Σχεδιασμός Διαγράμματος Περιπτώσεων Χρήσης: 13 μονάδες)***

***(Αιτιολόγηση Σχέσεων: 2 μονάδες)***

|  |
| --- |
| ***Μαθησιακά Αποτελέσματα***  Στην Άσκηση 1.Α θα σας δοθεί η δυνατότητα να κατανοήσετε:   * τη διεργασία του προσδιορισμού των απαιτήσεων (Requirement Specification) και * τον τρόπο κατασκευής του «Διαγράμματος Περιπτώσεων Χρήσης» (Use Case Diagram).   Πιο συγκεκριμένα, σε σχέση με τη διεργασία του «Προσδιορισμού Απαιτήσεων» θα κατανοήσετε:   * τον ορισμό των απαιτήσεων και * τους δύο κύριους τύπους των απαιτήσεων.   Σε σχέση με την κατασκευή του «Διαγράμματος Περιπτώσεων Χρήσης» θα μπορέσετε:   * να παραθέσετε μία περιγραφή του διαγράμματος Π.Χ., * να παραθέσετε τα 4 μέλη που ένα διάγραμμα Π.Χ. περιλαμβάνει, * να περιγράψετε τη λειτουργικότητα του κάθε μέλους, * να παραθέσετε τα 3 είδη σχέσεων των Π.Χ., * να περιγράψετε τη λειτουργικότητα του κάθε είδους σχέσης, * να αντιστοιχίσετε το κάθε μέλος του διαγράμματος Π.Χ. με το σύμβολό του και * να αντιστοιχίσετε την κάθε σχέση του διαγράμματος Π.Χ. με το σύμβολό της. |

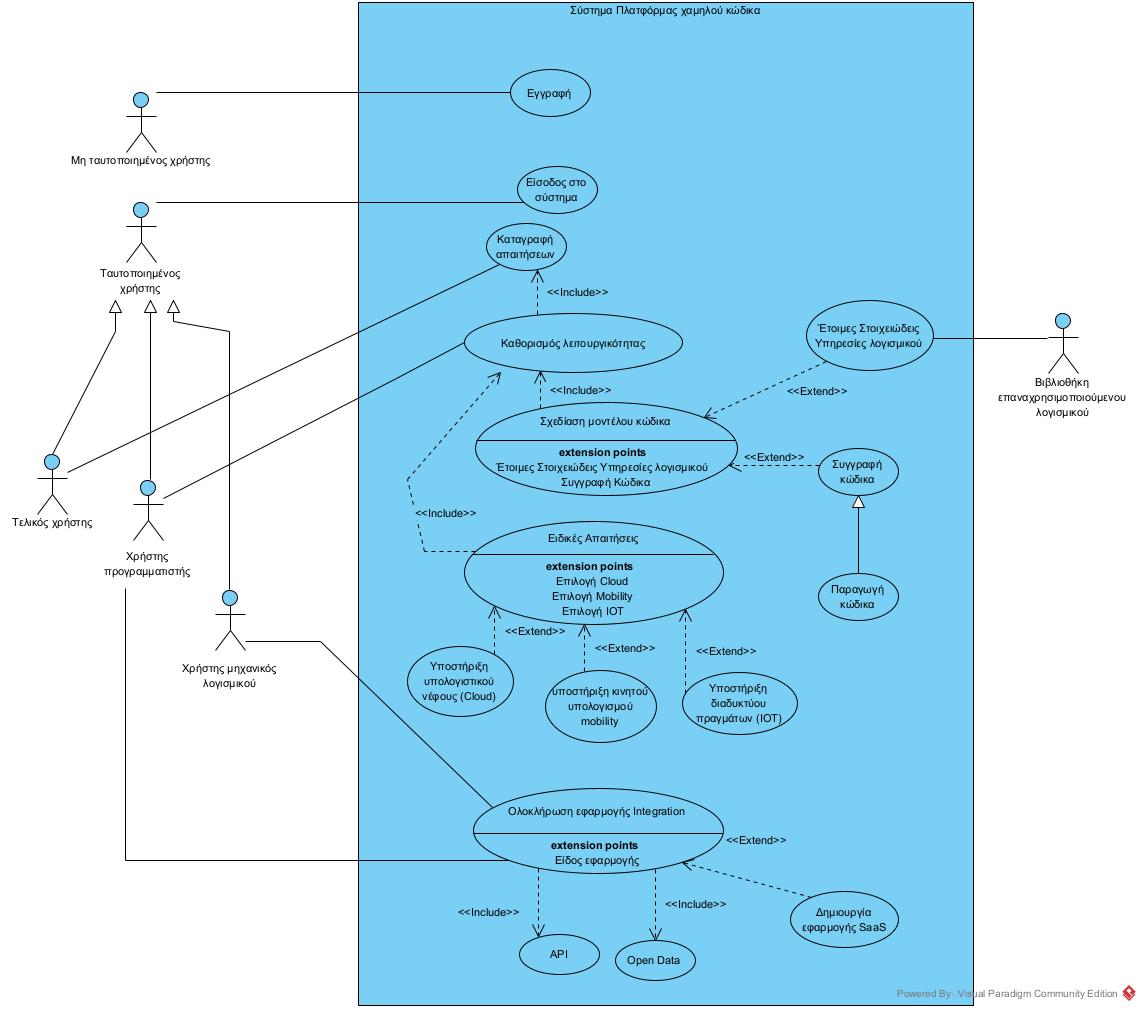
**Απάντηση 1ης Άσκησης / Ερώτημα Α**

*Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα,* ***ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ****.*

*Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα,* ***ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ****.*

*Εξηγήστε σε ποιο σημείο θεωρείτε την απάντηση ελλιπή και γιατί.*

**Διάγραμμα Περιπτώσεων Χρήσης:**



**Αιτιολόγηση Σχέσεων:**

**Σχέση μεταξύ χρηστών:**

Χρησιμοποίησα έναν χρήστη ως επιλογή use case για τους μη ταυτοποιημένους σε ένα σύστημα, όπως και για τους ταυτοποιημένους ως διαφορετική επιλογή του διαγράμματος user Case με τη λογική ότι για να μπεις σε ένα σύστημα όπως και αναγράφεται στην εκφώνηση πρέπει να είσαι ταυτοποιημένος. Αν δεν είσαι ταυτοποιημένος τότε δεν εισέρχεσαι στο σύστημα.

Οι σχέσεις μεταξύ χρηστών (Χρήστης Προγραμματιστής, Χρήστης μηχανικός και τελικός χρήστης). Χρησιμοποίησα τη σχέση κληρονομικότητας (σχέση γενίκευσης) του ταυτοποιημένου χρήστη διότι όλοι αυτοί οι χρήστες είναι σίγουρα ταυτοποιημένοι και οι οποίοι αρχικά κληρονομούν τις ιδιότητες ενός ταυτοποιημένου χρήστη επεκτείνοντας τες ως προς τις συγκεκριμένες λειτουργικές απαιτήσεις του κάθε χρήστη (Προγραμματιστή, μηχανικός ή τελικός).

**Σχέση μεταξύ περιπτώσεων χρήστης:**

* Για τη σχέση **Καθορισμός λειτουργικότητας:**

όπου συνδέεται από τον χρήστη προγραμματιστής χρησιμοποίησα τη σχέση **καταγραφή απαιτήσεων** την οποία την επισήμανα ως <<include>> διότι είναι απαιτούμενο για την υλοποίηση αυτής της σχέσεις να έχει συνδεθεί με την καταγραφή των απαιτήσεων του τελικού χρήστη. Παραδοχή: θεωρώ ότι αν δεν υπάρχει καταγραφή των απαιτήσεων δεν θα μπορέσει να υλοποιηθεί η σχέση **καθορισμός λειτουργικότητας**.

Όπως και οι σχέσεις **σχεδίαση μοντέλου κώδικα** και **προσδιορισμός απαιτήσεων** βάση της εκφώνησης της άσκησης θεωρώ ότι η συμβολή τους είναι απαιτούμενο στο να υλοποιηθούν αυτές οι σχέσεις.

Δηλαδή για να υλοποιηθεί οποιαδήποτε από αυτές τις σχέσεις πρέπει να περιλαμβάνει τον **καθορισμό λειτουργικότητας** όπως και **την καταγραφή απαιτήσεων.**

* Για τη σχέση **Σχεδίαση μοντέλου κώδικα:**

Χρησιμοποίησα τη σχέση extends ως επιλογή βάση της εκφώνησης για το εάν θα κάνει ο προγραμματιστής συγγραφή κώδικα ή για το εάν χρησιμοποιήσει έτοιμη βιβλιοθήκη λογισμικού η οποία είναι εξωτερική του συστήματος.

* Για τη σχέση **παραγωγή κώδικα:**

Χρησιμοποίησα γενίκευση διότι η σχέση **παραγωγή κώδικα** έχουν σίγουρα κοινά χαρακτηριστικά τα οποία κληρονομούνται από τη σχέση συγγραφής κώδικα.

* Για τη σχέση **προσδιορισμός απαιτήσεων:**

Βάση της εκφώνησης για τη σχέση αυτή δίνονται τρείς επιλογές (Υποστήριξη λογισμικού νέφους, υποστήριξη κινητού υπολογισμού, υποστήριξη διαδικτύου πραγμάτων). Οι τρείς σχέσεις **Υποστήριξη λογισμικού νέφους, υποστήριξη κινητού υπολογισμού, υποστήριξη διαδικτύου πραγμάτων** επεκτείνειτην περίπτωση χρήσης **προσδιορισμός απαιτήσεων.**

* Για τη σχέση **ολοκλήρωση εφαρμογής:**

Η σχέση αυτή συνδέεται με τον Χρήστη προγραμματιστή ή μηχανικό λογισμικού διότι είναι σχέση η οποία συνδέεται δια της συνεργασίας μεταξύ χρηστών. Για να μπορέσει να υλοποιηθεί η σχέση **ολοκλήρωση εφαρμογής** περιλαμβάνει τις σχέσεις «δηλαδή είναι απαραίτητο» η σχέση **δημιουργία κατάλληλων προγραμματιστικών εφαρμογών** όπως και η σχέση **υποστήριξη ανοιχτών δεδομένων.** Τέλος η σχέση **δημιουργία μιας εφαρμογής SaaS** επεκτείνει την περίπτωση χρήσης **ολοκλήρωση εφαρμογής.**

**Ερώτημα 1.Β – Ορισμός Ιστοριών Χρηστών**

Κάθε αντικειμενοστρεφής μεθοδολογία ανάπτυξης λογισμικού έχει ως σημείο αφετηρίας τον προσδιορισμό των (λειτουργικών) απαιτήσεων που θα πρέπει να ικανοποιεί η υπό εξέταση εφαρμογή λογισμικού. Η βασική δραστηριότητα που χρησιμοποιείται για το σκοπό αυτό είναι ο προσδιορισμός των ***περιπτώσεων χρήσης*** (use cases) και η δημιουργία ενός κατάλληλου ***διαγράμματος περιπτώσεων χρήσης*** (use case diagram). Στη συνέχεια, κάθε περίπτωση χρήσης που περιέχεται στο διάγραμμα περιπτώσεων χρήσης περιγράφεται λεκτικά με κείμενο, προσδιορίζοντας τη βασική ροή και την ή τις πιθανές εναλλακτικές ροές που τη χαρακτηρίζουν (δείτε το Ερώτημα 1.Β της 1ης Γραπτής Εργασίας της ΠΛΗ24 το ακαδημαϊκό έτος 2020-2021).

Η ***ευέλικτη ανάπτυξη λογισμικού*** (Agile Software Development, ASD) είναι μια (σχετικά) νέα και ιδιαίτερα διαδεδομένη ομάδα μεθόδων ανάπτυξης λογισμικού που προωθεί την εξελικτική ανάπτυξη μιας εφαρμογής και ενθαρρύνει την ταχεία και ευέλικτη ανταπόκριση στις αλλαγές. Μια ευέλικτη μεθοδολογία ανάπτυξης λογισμικού (όπως η μέθοδος SCRUM που θα χρησιμοποιηθεί στην εκπόνηση της 3ης Γραπτής Εργασίας της ΠΛΗ24) έχει και αυτή ως σημείο αφετηρίας την εύρεση των περιπτώσεων χρήσης, μόνο που, μετά τη δημιουργία του διαγράμματος περιπτώσεων χρήσης, προσδιορίζεται μια σειρά από ***ιστορίες χρηστών*** (user stories) που περιγράφουν τη λειτουργικότητα των περιπτώσεων χρήσης σε μεγαλύτερο επίπεδο λεπτομέρειας.

Οι ιστορίες χρηστών περιγράφουν συνοπτικά με τη μορφή κειμένου την αλληλεπίδραση ενός χρήστη με το σύστημα (τί κάνει ο χρήστης ή τί πρέπει να κάνει στο πλαίσιο αυτής της αλληλεπίδρασης και όχι το πώς το σύστημα θα υλοποιήσει την απόκρισή του). Ο πιο διαδεδομένος τρόπος για τη σύνταξη μιας ιστορίας χρήστη ακολουθεί το εξής πρότυπο (template):

**Όνομα Ιστορίας Χρήστη:** Κατάλληλο όνομα

**Περιγραφή:**

***Ως ένας*** <τύπος χρήστη>, ***θέλω να*** < πραγματοποιήσω κάποιους στόχους>

***έτσι ώστε*** <να επιτύχω κάποιο σκοπό>

**Κριτήρια Αποδοχής:**

***ΚΑ#1:*** …

***ΚΑ#2:*** …

κ.λπ.

Συνοπτικά, σε μια ιστορία χρήστη απαντάμε στις ακόλουθες ερωτήσεις:

* ***Ποιος:*** Προς όφελος ποιου ατόμου εκτελούμε τις συγκεκριμένες ενέργειες;
* ***Τί:*** Τί ενέργειες κάνουμε;
* ***Γιατί:*** Γιατί κάνουμε αυτές τις ενέργειες;

Τα ***κριτήρια αποδοχής*** περιγράφουν τί πρέπει να γίνει ώστε μια ιστορία χρήστη να θεωρηθεί ολοκληρωμένη. Αποτελούν σενάρια (επιτυχίας, αλλά και αποτυχίας) ή επεξηγήσεις αναφορικά με το τί πρέπει να γίνει. Είναι δυνατό να υπάρχουν πολλαπλά κριτήρια αποδοχής ανά ιστορία χρήστη.

Ακολουθεί ένα παράδειγμα μιας ιστορίας χρήστη:

**Όνομα Ιστορίας Χρήστη:**

Οργάνωση μιας δια ζώσης ΟΣΣ για την ΠΛΗ24 (με μέτρα Covid19) σε μια αίθουσα στην πανεπιστημιούπολη του ΕΚΠΑ.

**Περιγραφή:**

***Ως ένας*** ΣΕΠ της ΠΛΗ24, ***θέλω να*** οργανώσω μια εποικοδομητική δια ζώσης ΟΣΣ για την ΠΛΗ24

***έτσι ώστε*** όλοι οι συμμετέχοντες στην ΠΛΗ24 να γνωριστούμε καλύτερα μεταξύ μας και να κατανοήσουμε τους μαθησιακούς στόχους της ΠΛΗ24.

***Κριτήρια* Αποδοχής:**

***ΚΑ#1:*** Η δια ζώσης ΟΣΣ πρέπει να αφορά όλους τους φοιτητές της ΠΛΗ24 (νέους και παλαιούς) για το ακαδημαϊκό έτος 2021-2022.

***ΚΑ#2:*** Η δια ζώσης ΟΣΣ πρέπει να ακολουθεί τα ισχύοντα μέτρα για την αντιμετώπιση της πανδημίας Covid19.

***ΚΑ#3:*** Η δια ζώσης ΟΣΣ πρέπει να γίνει σε αίθουσα κατάλληλου μεγέθους και διαμόρφωσης που να επιτρέπει την ενεργή συμμετοχή και παρακολούθηση όλων των φοιτητών / φοιτητριών.

***ΚΑ#4:*** Η δια ζώσης ΟΣΣ πρέπει να ενθαρρύνει την αλληλεπίδραση όλων των συμμετεχόντων.

***ΚΑ#5:*** Η δια ζώσης ΟΣΣ πρέπει να πραγματοποιηθεί σε τρία χρονικά τμήματα της μίας ώρας με δύο μισάωρα διαλείμματα ενδιάμεσα.

***ΚΑ#6:*** Επιβεβαίωση ότι μετά το τέλος της ΟΣΣ όλοι οι συμμετέχοντες της δια ζώσης ΟΣΣ θα λάβουν ένα έντυπο με τους μαθησιακούς στόχους της ΠΛΗ24 και οδηγίες για την επιτυχή παρακολούθησή της.

***ΚΑ#7:*** Αν το λεωφορείο που εκτελεί δρομολόγιο στο εσωτερικό της πανεπιστημιούπολης είναι εκτός λειτουργίας, οι συμμετέχοντες θα πρέπει να περπατήσουν ως την αίθουσα της δια ζώσης ΟΣΣ (σενάριο αποτυχίας).

***ΚΑ#8:*** Αν δεν λειτουργεί ο υπολογιστής της αίθουσας που θα γίνει η δια ζώσης ΟΣΣ, θα χρησιμοποιηθεί ο φορητός υπολογιστής του ΣΕΠ (σενάριο αποτυχίας).

Χρησιμοποιώντας το πρότυπο που αναφέρεται πιο πάνω περιγράψτε δύο (2) ιστορίες χρηστών που αντιστοιχούν σε απαιτήσεις της πλατφόρμας ανάπτυξης χαμηλού κώδικα που περιγράφεται στο Ερώτημα 1.Α και πιο συγκεκριμένα:

**α)** Στην είσοδο ενός εγγεγραμμένου χρήστη στο υπό εξέταση σύστημα.

**β)** Στην αναζήτηση από έναν προγραμματιστή έτοιμων στοιχειωδών υπηρεσιών λογισμικού (components) σε κατάλληλη εξωτερική του υπό εξέταση συστήματος βιβλιοθήκη επαναχρησιμοποιήσιμου λογισμικού.

***Υπόδειξη:***

Στην περιγραφή των ζητούμενων ιστοριών χρήστη μπορείτε να λάβετε υπόψη σας (μαζί με άλλες ιδέες που έχετε):

* Την περίπτωση εισαγωγής λάθους ονόματος χρήστη / κωδικού πρόσβασης (για την ιστορία χρήστη της εισόδου).
* Την αναζήτηση με χρήση λέξεων κλειδιών (για την ιστορία χρήστη της αναζήτησης).
* Την ταξινόμηση των αποτελεσμάτων της αναζήτησης (για την ιστορία χρήστη της αναζήτησης).

***Προαιρετική Δραστηριότητα***

Όσοι/όσες ενδιαφέρονται να γνωρίσουν καλύτερα τον τρόπο που οι ευέλικτες μεθοδολογίες αντιμετωπίζουν τον προσδιορισμό των απαιτήσεων μέσα από τη δημιουργία ιστοριών χρηστών μπορούν να ακούσουν την ακόλουθη διαδικτυακή ραδιοφωνική μετάδοση (podcast), η οποία ηχογραφήθηκε το 2017 στο πλαίσιο του παγκόσμιου συνεδρίου του Project Management Institute (PMI):

“*How to Write Excellent User Stories (Episode 408)*”

<https://www.project-management-podcast.com/podcast-episodes/episode-details/775-episode-408-how-to-write-excellent-user-stories>

***(Σύνολο: 10 μονάδες)***

|  |
| --- |
| ***Μαθησιακά Αποτελέσματα***  Στην Άσκηση 1.Β θα σας δοθεί η δυνατότητα να κατανοήσετε τη χρήση των ιστοριών χρηστών (user stories). Πιο συγκεκριμένα θα μπορέσετε:   * να ορίσετε την έννοια μιας ιστορίας χρήστη, * να διατυπώσετε τις ερωτήσεις που απαντά μια ιστορία χρήστη, * να περιγράψετε, ακολουθώντας κατάλληλο πρότυπο, μια ιστορία χρήστη και * να εξηγήσετε το ρόλο των κριτηρίων αποδοχής σε μια ιστορία χρήστη. |

**Απάντηση 1ης Άσκησης – Ερώτημα Β**

*Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα,* ***ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ****.*

*Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα,* ***ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ****.*

*Εξηγήστε σε ποιο σημείο θεωρείτε την απάντηση ελλιπή και για ποιους λόγους.*

**Ιστορία Χρήστη 1:**

**Όνομα Ιστορίας Χρήστη:** Για την είσοδο ενός εγγεγραμμένου χρήστη (ταυτοποιημένου χρήστη) στο σύστημα.

**Περιγραφή:**

***Ως ένας*** τελικός χρήστης, ***θέλω να*** εισέλθω στο σύστημα πλατφόρμα ανάπτυξης χαμηλού κώδικα

***έτσι ώστε*** να κάνω καταγραφή απαιτήσεων

**Κριτήρια Αποδοχής:**

***ΚΑ#1:*** Επιλογή εγγραφής στο σύστημα, αν δεν είναι εγγεγραμμένος χρήστης (σενάριο αποτυχίας).

***ΚΑ#2:*** Επιλογή εισόδου στο σύστημα, αν είναι ήδη εγγεγραμμένος (σενάριο επιτυχίας).

***ΚΑ#3:*** *Επιλογή τύπου χρήστη Τελικός, Προγραμματιστής ή μηχανικός λογισμικού.*

***ΚΑ#4:*** *Επιλογή καταγραφής απαιτήσεων.*

***ΚΑ#5:*** *Ενημέρωση του συστήματος «καθορισμού λειτουργικότητας» με τις καταγραφές απαιτήσεων του χρήστη.*

***ΚΑ#6:*** *Αποστολή email επιβεβαίωσης αλλαγών του χρήστη.*

**Ιστορία Χρήστη 2:**

**Όνομα Ιστορίας Χρήστη:** Αναζήτηση ενός προγραμματιστή από τις έτοιμες στοιχειώδης υπηρεσίας λογισμικού στην εξωτερική υπό εξέταση βιβλιοθήκη επαναχρησιμοποιήσιμου λογισμικού

**Περιγραφή:**

***Ως ένας*** χρήστης προγραμματιστής, ***θέλω να*** χρησιμοποιήσω την αναζήτηση έτοιμων στοιχειωδών υπηρεσιών λογισμικού ***έτσι ώστε*** να σχεδιάσω το μοντέλο του κώδικα.

**Κριτήρια Αποδοχής:**

***ΚΑ#1:*** *Επιλογή εγγραφής στο σύστημα, αν δεν είναι εγγεγραμμένος χρήστης (σενάριο αποτυχίας).*

***ΚΑ#2:*** Επιλογή εισόδου στο σύστημα, αν είναι ήδη εγγεγραμμένος (σενάριο επιτυχίας).

***ΚΑ#3:*** *Επιβεβαίωση τύπου χρήστη προγραμματιστή και αποστολή email ταυτοποίησης.*

***ΚΑ#4:*** *Επιλογή**ελέγχου καταγραφής απαιτήσεων του τελικού χρήστη.*

***ΚΑ#5:*** *Επιλογή φόρμας αναζήτησης στις έτοιμες στοιχειώδεις υπηρεσίες λογισμικού.*

***ΚΑ#6:*** *Συμπλήρωση φόρμας για αναζήτηση από το σύστημα διασύνδεσης στην βιβλιοθήκη επαναχρησιμοποιούμενου λογισμικού.*

***ΚΑ#7:*** *Διασύνδεση του συστήματος με το εξωτερικό σύστημα βιβλιοθήκης επαναχρησιμοποιούμενου κώδικα.*

***ΚΑ#8:*** *Επικοινωνία του συστήματος έτοιμες στοιχειώδεις υπηρεσίες λογισμικού με την βιβλιοθήκη επαναχρησιμοποιούμενου λογισμικού βάση των λέξεων κλειδιών αναζήτησης που εισήγαγε στην φόρμα από τον χρήστη προγραμματιστή*

***ΚΑ#9:*** *Επιστροφή μηνύματος αποτυχημένης αναζήτησης των λέξεων κλειδιών από το εξωτερικό σύστημα βιβλιοθήκης επαναχρησιμοποιούμενου κώδικα(σενάριο αποτυχίας).*

***ΚΑ#10:*** *Επιστροφή μηνύματος επιτυχημένης αναζήτησης των λέξεων κλειδιών από το εξωτερικό σύστημα βιβλιοθήκης επαναχρησιμοποιούμενου κώδικα(σενάριο επιτυχίας).*

***ΚΑ#11:*** *Επιλογή των components από την εξωτερική βιβλιοθήκη επαναχρησιμοποιούμενου λογισμικού.*

***ΚΑ#12:*** *Διασύνδεση και μεταφορά του/των component/s από την εξωτερική βιβλιοθήκη επαναχρησιμοποιούμενου λογισμικού στο σύστημα καθορισμού λειτουργικότητας του χρήστη.*

***ΚΑ#13:***  *Αποσύνδεση του συστήματος από την εξωτερική βιβλιοθήκη επαναχρησιμοποιούμενου λογισμικού.*

***ΚΑ#14:*** *Επιλογή**σχεδίασης μοντέλου του κώδικα.*

***ΚΑ#15:*** *Επιλογή**κατοχύρωσης (αποθήκευση) του κώδικα στο σύστημα στη βάση δεδομένων.*

***ΚΑ#16:*** *Αποστολή**email και ενημέρωση**τους τελικούς και μηχανικούς χρήστες του συστήματος.*

***ΚΑ#17:*** *Επιλογή νέας σχεδίασης μοντέλου κώδικα από τον καθορισμό λειτουργικότητας (loop ξανά από την αρχή)*

***ΚΑ#18:*** *Επιλογή αποσύνδεσης του χρήστη προγραμματιστή από το σύστημα.*

|  |
| --- |
| **Εκφώνηση 2ης Άσκησης**  **Σύστημα Ηλεκτρονικού Παιχνιδιού για την Ελληνική Επανάσταση του 1821** |

**Ερώτημα 2.Α – Δημιουργία Εννοιολογικού Μοντέλου**

To 2021 συμπληρώθηκαν 200 χρόνια από την Ελληνική Επανάσταση του 1821. Για το λόγο αυτό η εταιρεία πληροφορικής στην οποία εργάζεστε αποφάσισε να υλοποιήσει ένα ηλεκτρονικό παιχνίδι που να διαδραματίζεται τη χρονική περίοδο της επανάστασης. Πιο συγκεκριμένα, πρόκειται για ένα μαζικό διαδικτυακό παιχνίδι ρόλων πολλαπλών παικτών (Massively Multiplayer Online Role-Playing Game, MMORPG) που θα έχει τα ακόλουθα βασικά χαρακτηριστικά (απαιτήσεις):

Το παιχνίδι διεξάγεται σε ένα ***χάρτη*** της ***Ελλάδας***, ο οποίος είναι χωρισμένος σε ***περιοχές*** στις οποίες μπορεί να κινείται ο ***στρατός***. Σε κάθε περιοχή υπάρχει μία ***πόλη***. Κάθε πόλη διαθέτει ως ιδιότητες έναν αριθμό από κατοίκους, φρουρά, κάστρο ή / και λιμάνι και ένα μετρητή επαναστατικού αισθήματος, που όσο υψηλότερος είναι, τόσο ευκολότερη γίνεται η κατάληψή της. Οι περιοχές αποτελούν αναπόσπαστο τμήμα του χάρτη της Ελλάδας και η κάθε πόλη αποτελεί αναπόσπαστο τμήμα της κάθε περιοχής.

Υπάρχουν δύο (2) ***στρατοί***, ο Ελληνικός και ο Τουρκικός, που έχουν ως ιδιότητες τον αριθμό των στρατιωτών, την εκπαίδευση και το ηθικό. Ο ***Ελληνικός στρατός*** αποτελείται από τους ***Κλέφτες*** και τους ***Αρματολούς***, ενώ ο ***Τουρκικός στρατό***ς αποτελείται από το ***πεζικό*** και το ***ιππικό***. Η κάθε κατηγορία στρατού διαθέτει ιδιότητες σχετικές με τις δυνατότητές του και τον τρόπο δράσης του, οι οποίες όμως δεν ενδιαφέρουν στο πλαίσιο της συγκεκριμένης άσκησης. Αυτό που ενδιαφέρει όμως είναι ότι ο κάθε στρατός διαθέτει τουλάχιστον έναν ***ήρωα*** για να ηγείται, ο οποίος χαρακτηρίζεται από πολεμική ικανότητα, πειθώ, δημοτικότητα, φιλοδοξία και απληστία. Επίσης, είναι αναγκαίος ο τακτικός ***ανεφοδιασμός*** ενός στρατού, ο οποίος έχει κόστος σε χρήματα.

Ανάμεσα στους δύο αντίπαλους στρατούς διεξάγονται μάχες, όταν αυτοί βρεθούν στην ίδια περιοχή. Σημαντικό ρόλο στην έκβαση των μαχών έχει η μορφολογία του εδάφους της κάθε περιοχής. Υπάρχουν συνολικά τέσσερα είδη μορφολογίας εδάφους σε κάθε περιοχή, τα οποία χαρακτηρίζονται από την ιδιότητα του τύπου του εδάφους (βουνό, δάσος, λόφος και κάμπος) και κάθε περιοχή έχει από δύο έως και τέσσερα διαφορετικά από αυτά τα είδη.

Οι παίκτες του παιχνιδιού (που χαρακτηρίζονται από ψευδώνυμο και βαθμούς εμπειρίας), συνεργάζονται μεταξύ τους διαδικτυακά και αλληλοεπιδρούν με τον Εικονικό Κόσμο (Virtual World) του παιχνιδιού, ο οποίος αποτελείται από το χάρτη της Ελλάδας (που οι παίκτες εξερευνούν), από χρήσιμα αντικείμενα (που οι παίκτες συλλέγουν), από αποστολές (που οι παίκτες αναλαμβάνουν) και από NPCs (Non-Player Characters, δηλαδή από χαρακτήρες που ελέγχονται από το παιχνίδι, με τους οποίους οι παίκτες επικοινωνούν). Τα συστατικά αυτά μέρη έχουν λόγο ύπαρξης μόνο στο πλαίσιο του συγκεκριμένου παιχνιδιού. Αντίθετα η μηχανή γραφικών που περιλαμβάνει ο Εικονικός Κόσμος του παιχνιδιού μπορεί να χρησιμοποιηθεί για τη δημιουργία και άλλων παιχνιδιών.

Με βάση τα πιο πάνω βασικά χαρακτηριστικά του παιχνιδιού να σχεδιάσετε το εννοιολογικό μοντέλο (αρχικό διάγραμμα κλάσεων) και να τεκμηριώσετε σύντομα τις επιλογές σας αιτιολογώντας το λόγο που χρησιμοποιήσατε (αν φυσικά το κάνατε) κληρονομικότητα (inheritance), συσσώρευση (aggregation) ή συγκρότηση (composition) ανάμεσα σε κλάσεις.

***Υποδείξεις:***

* Δε χρειάζεται να προσδιορίσετε τον τύπο των πεδίων των κλάσεων που θα χρησιμοποιήσετε.
* Με ***έντονα και πλάγια γράμματα*** σημειώνονται, στο μισό του κειμένου των απαιτήσεων, οι κλάσεις (οντότητες/έννοιες) που είναι σημαντικές στο πεδίο του προβλήματος και έχουν προκύψει κάνοντας ανάλυση του κειμένου των βασικών χαρακτηριστικών του παιχνιδιού. Συνεχίστε εξετάζοντας το σύνολο των απαιτήσεων.

***(Σύνολο: 15 μονάδες)***

***(Σχεδιασμός Εννοιολογικού Μοντέλου: 13 μονάδες)***

***(Αιτιολόγηση Χρήσης Κληρονομικότητας, Συσσώρευσης ή Συγκρότησης: 2 μονάδες)***

|  |
| --- |
| ***Μαθησιακά Αποτελέσματα***  Στην Άσκηση 2.Α θα σας δοθεί η δυνατότητα να κατανοήσετε:   * την κατασκευή του «Εννοιολογικού Μοντέλου» και * την κατασκευή του «Διαγράμματος Κλάσεων».   Σχετικά με την κατασκευή του «Εννοιολογικού Μοντέλου» θα μπορέσετε:   * να κατανοήσετε την έννοια του μοντέλου, * να κατανοήσετε την έννοια του εννοιολογικού μοντέλου, * να περιγράψετε τα 2 είδη μελών που το εννοιολογικό μοντέλο περιλαμβάνει και * να περιγράψετε τη σχεδίαση των μελών που περιλαμβάνει το εννοιολογικό μοντέλο. |

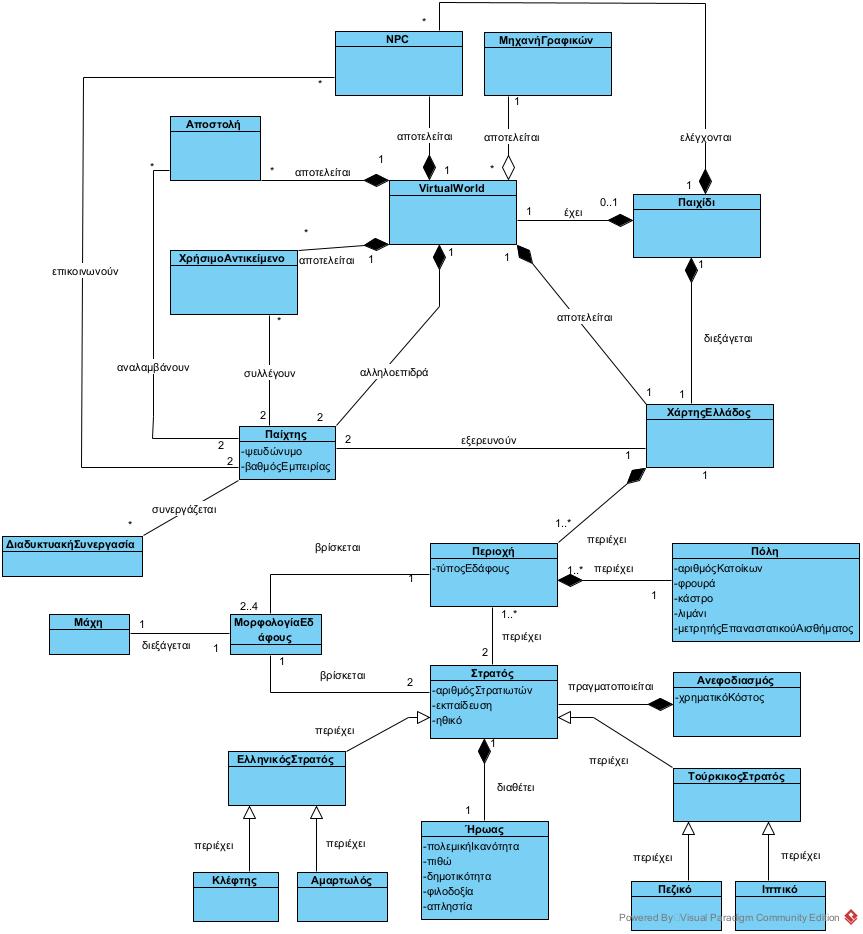
**Απάντηση 2ης Άσκησης / Ερώτημα Α**

*Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα,* ***ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ****.*

*Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα,* ***ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ****.*

*Εξηγήστε σε ποιο σημείο θεωρείτε την απάντηση ελλιπή και για ποιους λόγους.*

**Εννοιολογικό Μοντέλο (Αρχικό Διάγραμμα Κλάσεων):**



**Αιτιολόγηση Χρήσης Κληρονομικότητας, Συσσώρευσης ή Συγκρότησης:**

Αρχικά χώρισα το διάγραμμα βάση των ζητούμενων της άσκησης σε τρία μέρη.

Το πρώτο αποτελείται από την εικονική μηχανή. Το δεύτερο αποτελείται από το παιχνίδι δηλαδή την περιοχή, τις πόλεις και την μάχη. Τρίτο τον στρατό

Επειδή ο η κλάση στρατός σε σχέση με την κλάση Ελληνικό, Τούρκικο στρατό χρησιμοποίησα την σχέση κληρονομικότητας ώστε να κληρονομήσει όλα τα χαρακτηριστικά και τις ιδιότητες της κλάσης **Στρατός.** Για τις κλάσεις (Κλέφτης, Αμαρτωλός, Πεζικό και Ιππικό) χρησιμοποίησα εξίσου κληρονομικότητα ώστε να κληρονομήσει όλα τα χαρακτηριστικά της της κλάσης **Ελληνικός, Τούρκικος στρατός.** Διότι θεωρώ ότι τα χαρακτηριστικά όπως αριθμό στρατιωτών και ηθικό του κάθε στρατιώτη πρέπει να κληρονομηθούν και στην υπό κλάση (Κλέφτης, Αμαρτωλός, Πεζικό και Ιππικό).

Χρησιμοποίησα σύνθεση για την κλάση **Ήρωας** διότι βάση ζητούμενων της άσκησης πρέπει τουλάχιστον να υπάρχει ένας ήρωας ώστε να ηγείται ένας στρατός. Ένας στρατός μπορεί να υπάρχει αλλά να μην ηγείται όπως και σίγουρα δεν μπορεί να υπάρχει ένας ήρωας αν υπάρχει στρατός μιας και ο ήρωας αποτελείται από την κλάση Στρατός.

Για την κλάση **Ανεφοδιασμός** χρησιμοποίησασύνθεσηδιότι ένας Στρατός ανεφοδιάζεται από αυτή την κλάση αυτή αλλά αν δεν ανεφοδιαστεί ο στρατός τότε δεν θα μπορεί να υπάρχει.

Κλάση **Πόλη:** χρησιμοποίησα σύνθεση διότι όπως και αναφέρει η άσκηση η πόλη αποτελεί αναπόσπαστο κομμάτι της περιοχής. Άρα η κλάση **Πόλη** αποτελείταικαι δεν μπορεί να υπάρξει μόνη της αν δεν υπάρχει η κλάση **Περιοχή.**

Κλάση **Περιοχή:** χρησιμοποίησα σύνθεση διότι όπως και αναφέρει η άσκηση η περιοχή αποτελεί αναπόσπαστο κομμάτι του χάρτη της Ελλάδος. Άρα η κλάση **Περιοχή** αποτελείταικαι δεν μπορεί να υπάρξει μόνη της αν δεν υπάρχει η κλάση **ΧάρτηςΕλλάδος.**

Κλάση **ΧάρτηςΕλλάδος** χρησιμοποίησα σύνθεση διότι η κλάση αυτή διεξάγεται από την Κλάση **Παιχνίδι** και δεν μπορεί να υπάρχει η Κλάση **ΧάρτηςΕλλάδος** αν δεν υπάρχει η κλάση **Παιχνίδι.** Όπως και αναφέρει η άσκηση η κλάση Κλάση **ΧάρτηςΕλλάδος** αποτελείται από την κλάση VirtualWorld.

Η κλάση **Παιχνίδι** χρησιμοποίησα σύνθεση διότι η κλάση **NPC** ελέγχεται από το παιχνίδι όπως και η κλάση **VirtualWorld** εξίσου και η κλάση **ΧάρτηςΕλλάδος** και όπως αναφέρει και η άσκηση δεν μπορούν να υπάρχουν και αποτελούνται τα συστατικά αυτά και έχουν λόγω ύπαρξης στο πλαίσιο του συγκεκριμένου παιχνιδιού.

Κλάση **VirtualWorld** χρησιμοποίησα σύνθεση διότι όπως αναφέρει η εκφώνηση της άσκησης αποτελείται από τις κλάσεις **Παίχτης, ΧρήσιμοΑντικείμενο, Αποστολή, NPC, ΧάρτηςΕλλάδος** και **Παιχνίδι**  και για τις κλάσεις αυτές δεν υπάρχει λόγος ύπαρξης εάν δεν υπάρχει η κλάση **VirtualWorld.**

Κλάση **ΜηχανήΓραφικών** χρησιμοποίησα συναρμολόγηση διότι όπως και αναφέρει η εκφώνηση της άσκησης η μηχανή γραφικών που περιλαμβάνει η κλάση **VirtualWorld** μπορεί να χρησιμοποιηθεί και για την κατασκευή άλλων παιχνιδιών. Άρα ακόμα και αν δεν υπάρχει η κλάση **VirtualWorld** τότε η κλάση **ΜηχανήΓραφικών** θα υπάρχει ώστε να χρησιμοποιηθεί και για την δημιουργία άλλων παιχνιδιών/κόσμων.

|  |
| --- |
| **Εκφώνηση 3ης Άσκησης**  **Σύστημα Προσομοίωσης Αγώνων Μπάσκετ** |

Στο συμπιεσμένο αρχείο ***start.rar****,* που συνοδεύει την εκφώνηση της εργασίας, δίνονται ορισμένες κλάσεις που υλοποιούν μερικώς έναν προσομοιωτή αγώνων μπάσκετ. Εισάγετε τις κλάσεις στο περιβάλλον NetBeans δημιουργώντας ένα σχετικό project.

**Ερωτήματα 3.Α και 3.Β – Υλοποίηση Μεθόδων**

Υλοποιήστε τις μεθόδους showStats και showPlayerStats, ώστε να εκτυπώνουν στην οθόνη τα στατιστικά των ομάδων και των παιχτών, όπως υποδηλώνεται από τα σχετικά σχόλια.

Φροντίστε ο κώδικας να είναι μορφοποιημένος κατάλληλα και να είναι ευανάγνωστος. Θα πρέπει να υπάρχει **τεκμηρίωση σε μορφή σχολίων**.

***(Σύνολο: 25 μονάδες)***

**Απάντηση 3ης Άσκησης / Ερωτήματα Α και Β**

*Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα,* ***ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ****.*

*Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα,* ***ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ****.*

*Εξηγήστε σε ποιο σημείο θεωρείτε την απάντηση ελλιπή και για ποιους λόγους.*

**Κώδικας Μεθόδου showStats:**

public void showStats() {

// Show the statistics of each team in following format

// Name

// Points:

// Shoots Attempted: Shots Made: Percentage:

// Rebounds

/\*αρχικοποίηση μεταβλητής και χρήση for..loop για να μπω στα στοιχεία

του πίνακα του αντικειμένου players ώστε

να κάνω την χρήση των getters και setters όπως και μεθόδων, fields για τα ζητούμενα της άσκησης\*/

int i;

//αρχικοποίηση των μεταβλητών ώστε να βγει το αποτέλεσμα ως άθροισμα των στοιχείων του πίνακα

int shootsAttempted=0, shotsMade=0, rebounds=0;

//χρησιμοποίησα float λόγω της φόρμουλας της διαίρεσης και του ποσοστού για να μπορέσω να έχω πράξεις στα δεκαδικά μέρη

float percentage;

//δεν είναι private η μεταβλητές της κλάσης team παρά ταύτα χρησιμοποίησα τους getters αυτών αντί των μεταβλητών

for(i=0;i<players.length;i++){ //χρησιμοποίησα .length ώστε η for..loop να διατρέξει τα στοιχεία ώς το μέγιστο μήκος του μονοδιάστατου πίνακα players

shootsAttempted += players[i].getFGattempted();

shotsMade += players[i].getFGmade();

rebounds += players[i].getReb();

}

//percentage calculation χρησιμοποίησα την φόρμουλα: (Shots Made / Attempted shoots \* 100)

percentage = (( (float) shotsMade / (float)shootsAttempted )\*100);

//εκτύπωση των στατιστικών βάση των ζητούμενων της άσκησης. επέλεξα String.format ώστε να έχω δύο δεκαδικά ψηφία στο αποτέλεσμα

System.out.println("Name: "+this.getName()+"\n"+

"Points: "+this.getScore()+"\n"+

//επέλεξα String.format ώστε να έχω δύο δεκαδικά ψηφία στο αποτέλεσμα

"Shoots Attempted: "+shootsAttempted+"\tShots Made: "+shotsMade+"\tPercentage: "+String.format("%.2f", percentage)+"%\n"+

"Rebounds: "+rebounds);

}//end of showStats() method

**Κώδικας Μεθόδου showPlayerStats:**

public void showPlayersStats() {

// Show the statistics of each player in following format

// Position Points (... rebounds, ... / ... shoots, index)

// index = points + rebounds - missed shots

//Χρήση της for..loop για να εισέλθω στα στοιχεία του πίνακα

int i; //αρχικοποίηση μεταβλητής για να την χρησιμοποιήσω ως μετρητή της for..loop

for(i=0;i<players.length;i++){

/\*αρχικοποίηση μεταβλητής index μέσα στην for..loop διότι δεν με απσχολεί η διάρκειά της με τό πέρας της for..loop

σε κάθε θέση του πίνακα του αντικειμένου players της κλάσης Player χρησιμοποίησα τον τελεστή . ώστε να έχω πρόσβαση στα στοιχεία της κλάσης.

Για το ζητούμενο shoots αφαίρεσα από το σύνολο των προσπαθειών FGattemted το σύνολο των επιτυχημένων getFGmade\*/

int index = players[i].points + players[i].getReb() - (players[i].getFGattempted() - players[i].getFGmade());

/\*χρησιμοποιώντας sout για τα ζητούμενα της άσκησης την τοποθέτησα μέσα στην λούπα ώστε να έχω το counter του μετρητή

για όλες θέσεις του πίνακα του αντικειμένου players[] της κλάσης Players \*/

System.out.println(players[i].getName()+" ( "+players[i].getReb()+" rebounds, "+

players[i].getFGattempted()+" / "+players[i].getFGmade()+" shoots, "+"index: "+index);}

}//end of showPlayersStats() method

**Ερώτημα 3.Γ – Επέκταση Κώδικα Εφαρμογής**

Τροποποιείστε τα σημεία του κώδικα που χρειάζονται, ώστε ο προσομοιωτής να διαχειρίζεται χωριστά, τα επιθετικά από τα αμυντικά rebounds. Πέραν του τρόπου αποθήκευσης στις οντότητες, τροποποιείστε και τις μεθόδους εμφάνισης των ερωτημάτων 3.Α και 3.Β, ώστε να εμφανίζονται και οι δυο τύποι από rebound.

Φροντίστε ο κώδικας να είναι μορφοποιημένος κατάλληλα και να είναι ευανάγνωστος. Θα πρέπει να υπάρχει **τεκμηρίωση σε μορφή σχολίων**.

***(Σύνολο: 20 μονάδες)***

**Απάντηση 3ης Άσκησης / Ερώτημα Γ**

*Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα,* ***ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ****.*

*Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα,* ***ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ****.*

*Εξηγήστε σε ποιο σημείο θεωρείτε την απάντηση ελλιπή και για ποιους λόγους.*

*Να μπει ΜΟΝΟ ο τροποποιημένος κώδικας σε επίπεδο μεθόδου ή πεδίου και όχι όλη η κλάση.*

**Τροποποιημένος Κώδικας:**

**Τροποποίηση της Κλάσης Player:**

public class Player {

String pos;

int points;

//Δημιουργία μεταβλητής για τα επιθετικά rebounds

int offensiveRebounds;

////Δημιουργία μεταβλητής για τα αμυντικά rebounds

int defensiveRebounds;

//int rebounds;

int fg\_made;

int fg\_attempted;

public Player(String p) {

pos = p;

points = 0;

//Αρχικοποίηση της μεταβλητής των επιθετικών rebounds στην τιμή 0 μέσα στον κατασκευαστή

offensiveRebounds = 0;

//Αρχικοποίηση της μεταβλητής των αμυντικών rebounds στην τιμή 0 μέσα στον κατασκευαστή

defensiveRebounds = 0;

fg\_made = 0;

fg\_attempted = 0;

}

public void increasePoints(int p) {points += p;}

//Δημιουργία μεθόδου η οποία αυξάνει κατά 1 χρησιμοποιώντας τον τελεστή ++ όποτε καλείται η μέθοδος increaseOffensiveRebounds()

public void increaseOffensiveRebounds() {offensiveRebounds++;}

//Δημιουργία μεθόδου η οποία αυξάνει κατά 1 χρησιμοποιώντας τον τελεστή ++ όποτε καλείται η μέθοδος increaseDefensiveRebounds()

public void increaseDefensiveRebounds() {defensiveRebounds++;}

public void increaseFG\_made() {fg\_made++;}

public void increaseFG\_attempted() {fg\_attempted++;}

public String getName() { return pos; }

public int getFGattempted() { return fg\_attempted; }

public int getFGmade() { return fg\_made; }

//Δημιουργία getter η οποία επιστρέφει την μεταβλητή offensiveRebounds

public int getOffReb() { return offensiveRebounds; }

//Δημιουργία μεθόδου getter η οποία επιστρέφει την μεταβλητή defensiveRebounds

public int getDefReb() { return defensiveRebounds; }

}

**Τροποποίηση της Κλάσης Team για τη μέθοδο showStats():**

public void showStats() {

// Show the statistics of each team in following format

// Name

// Points:

// Shoots Attempted: Shots Made: Percentage:

// Rebounds

/\*αρχικοποίηση μεταβλητής και χρήση for..loop για να μπω στα στοιχεία

του πίνακα του αντικειμένου players ώστε

να κάνω την χρήση των getters και setter όπως και μεθόδων, fields για τα ζητούμενα της άσκησης\*/

int i;

//αρχικοποίηση των μεταβλητών ώστε να βγει το αποτέλεσμα ως άθροισμα των στοιχείων του πίνακα

int shootsAttempted=0, shotsMade=0, offensiveRebounds=0, defensiveRebounds=0;

float percentage;

//χρησιμοποίηση των getter από τον πίνακα αντικειμένων players

for(i=0;i<players.length;i++){

//δημιουργία μεταβλητών αθροίσματος

shootsAttempted += players[i].getFGattempted();

shotsMade += players[i].getFGmade();

offensiveRebounds += players[i].getOffReb();

defensiveRebounds += players[i].getDefReb();

}

//percentage calculation χρησιμοποίησα την φόρμουλα: (Shots Made / Attempted shoots \* 100)

percentage = (( (float) shotsMade / (float)shootsAttempted )\*100);

//εκτύπωση των στατιστικών βάση των ζητούμενων της άσκησης. επέλεξα String.format ώστε να έχω δύο δεκαδικά ψηφία στο αποτέλεσμα

System.out.println("Name: "+this.getName()+"\n"+

"Points: "+this.getScore()+"\n"+

//επέλεξα String.format ώστε να έχω δύο δεκαδικά ψηφία στο αποτέλεσμα

"Shoots Attempted: "+shootsAttempted+"\tShots Made: "+shotsMade+"\tPercentage: "+String.format("%.2f", percentage)+"%\n"+

"Offensive Rebounds: "+offensiveRebounds+

"\tDefensive Rebounds: "+defensiveRebounds);

}//end of showStats() method

**Τροποποίηση της Κλάσης Team για τη μέθοδο showPlayersStats():**

public void showPlayersStats() {

// Show the statistics of each player in following format

// Position Points (... rebounds, ... / ... shoots, index)

// index = points + rebounds - missed shots

//Χρήση της for..loop για να εισέλθω στα στοιχεία του πίνακα

int i; //αρχικοποίηση μεταβλητής για να την χρησιμοποιήσω ως μετρητή της for..loop

for(i=0;i<players.length;i++){

/\*αρχικοποίηση μεταβλητής index μέσα στην for..loop διότι δεν με απσχολεί η διάρκειά της με τό πέρας της for..loop

σε κάθε θέση του πίνακα του αντικειμένου players της κλάσης Player χρησιμοποίησα τον τελεστή . ώστε να έχω πρόσβαση στα στοιχεία της κλάσης.

Για το ζητούμενο shoots αφαίρεσα από το σύνολο των προσπαθειών FGattemted το σύνολο των επιτυχημένων getFGmade\*/

int index = players[i].points + (players[i].getDefReb()+players[i].getOffReb()) - (players[i].getFGattempted() - players[i].getFGmade());

/\*χρησιμοποιώντας sout για τα ζητούμενα της άσκησης την τοποθέτησα μέσα στην λούπα ώστε να έχω το counter του μετρητή

για όλες θέσεις του πίνακα του αντικειμένου players[] της κλάσης Players \*/

System.out.println(players[i].getName()+" ( "+players[i].getOffReb()+" offensive rebounds, "+players[i].getDefReb()+" defensive rebounds, "+

players[i].getFGattempted()+" / "+players[i].getFGmade()+" shoots, "+"index: "+index);}

}//end of showPlayersStats() method

}//end of Team class

|  |
| --- |
| ***Μαθησιακά Αποτελέσματα***  Στις Ασκήσεις 3.Α-3.Γ θα σας δοθεί αρχικά η δυνατότητα να:   * αναλύσετε τον κώδικα μιας κλάσης Java ως προς τη λειτουργικότητά της και να * διαπιστώσετε την ορθότητα του κώδικα μιας κλάσης Java ως προς το συντακτικό της γλώσσας.   Επιπλέον, η άσκηση αυτή θα σας επιτρέψει να μελετήσετε και να κατανοήσετε την έννοια της «Κλάσης».  Πιο συγκεκριμένα, θα μπορέσετε:   * να ορίσετε με ακρίβεια την έννοια της κλάσης, * να κατανοήσετε τα βασικά δομικά στοιχεία μιας κλάσης, * να κατανοήσετε την έννοια της μεθόδου, * να εξηγήσετε τις βασικές διαφορές ενός κατασκευαστή από μια απλή μέθοδο, * να κατασκευάσετε σε Java μέθοδο με ορίσματα και τύπο επιστροφής, * να κατανοήσετε τη δημιουργία και επεξεργασία πινάκων και * να επεξηγήσετε τη συμπεριφορά μιας άγνωστης κλάσης. |

**Ερώτημα 3.Δ – Δημιουργία της main και Εκτέλεσή της**

Ακολουθώντας το σχόλια που υπάρχουν στο αρχείο MainClass.java δοκιμάστε την λειτουργία της εφαρμογής. Δημιουργήστε δυο ομάδες της αρεσκείας σας, εκκινήστε τον προσομοιωτή και εκτυπώστε τα ομαδικά και ατομικά στατιστικά.

Δώστε τον κώδικα της μεθόδου main, **εκτελέστε το πρόγραμμά σας** και συμπεριλάβετε σχετικό **στιγμιότυπο οθόνης (screenshot)**, όπου θα φαίνεται το **αποτέλεσμα** της εκτέλεσης. Αν έχετε απαντήσει το ερώτημα 3.Γ θα πρέπει να συμπεριλάβετε και **δεύτερο στιγμιότυπο οθόνης (screenshot)**, όπου θα φαίνεται το αποτέλεσμα της εκτέλεσης του τροποποιημένου κώδικά σας. Φροντίστε ο κώδικας να είναι μορφοποιημένος κατάλληλα και να είναι ευανάγνωστος. Θα πρέπει να υπάρχει **τεκμηρίωση σε μορφή σχολίων**.

***(Σύνολο: 5 μονάδες)***

***Υπόδειξη:***

Κατά τον έλεγχο της ορθής εκτέλεσης του προγράμματός σας μπορείτε να αξιοποιήσετε τον αποσφαλματωτή (debugger) που διαθέτει το NetBeans:

- <https://www.youtube.com/watch?v=2Z9B8wYhKWw>

- <https://www.youtube.com/watch?v=06B9tsOKtZE>

- <https://netbeans.org/kb/docs/java/debug-visual.html>

|  |
| --- |
| ***Μαθησιακά Αποτελέσματα***  Στην Άσκηση 3.Δ θα σας δοθεί αρχικά η δυνατότητα σε σχέση με την έννοια της «Κλάσης»:   * να αναπτύξετε μια απλή κλάση σε Java, * να ορίσετε με ακρίβεια την έννοια της κλάσης, * να αναφέρετε τα βασικά δομικά στοιχεία μιας κλάσης, * να ορίσετε με ακρίβεια την έννοια της μεθόδου, * να κατασκευάσετε σε Java μέθοδο με ορίσματα και τύπο επιστροφής και * να δημιουργήσετε τις κλάσεις για την ανάπτυξη μιας εφαρμογής.   Επίσης, σχετικά με την έννοια του «Αντικειμένου» θα μπορέσετε:   * να ορίσετε με ακρίβεια την έννοια του αντικειμένου, * να εξηγήσετε τη διαφορά μεταξύ μιας κλάσης και ενός αντικειμένου και * να κατασκευάσετε αντικείμενα μιας κλάσης με διαφορετικά χαρακτηριστικά σε Java. |

**Απάντηση 3ης Άσκησης / Ερώτημα Δ**

*Εάν δεν έχετε δώσει απάντηση γράψτε με κεφαλαία γράμματα,* ***ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ****.*

*Εάν εν γνώση σας δίνετε ελλιπή απάντηση γράψτε με κεφαλαία γράμματα,* ***ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ****.*

*Εξηγήστε σε ποιο σημείο θεωρείται την απάντηση ελλιπή και γιατί.*

**Κώδικας Μεθόδου main:**

public class MainClass {

public static void main(String[] args) {

// Create a Game

/\*Team object δεύσμευση μνήμης και δημιουργίας αντικειμένων ομάδων hometeam και awayteam με τα ονόματα των αντικειμένων

OmadaraA και OmadaraB\*/

Team team1 = new Team("OmadaraA");

Team team2 = new Team("OmadaraB");

/\*αρχικοποίηση αντικειμένου παιχνιδιού με το όνομα game. Βλέπωντας την κλάση Game παρατηρώ ότι υπάρχει constructor ο οποίος παίρνει δύο ορίσματα αντικειμένων

την Team hT και την Team aT όπου στέλνω τα αντικείμενα που δημιούργησα OmadaraA και OmadaraB\*/

Game game = new Game(team1,team2);

// Simulate Game

//παρατηρώ ότι η κλάση Game έχει σαν κύρια μέθοδο την simulateGame() την οποία την εκτελώ μέσω του αντικειμένου Game οπου δημιούργησα και τον τελεστή "."

game.simulateGame();

// Show Team Statistics

/\*Βλέπω ότι η Κλάση Game έχει δύο μεθόδους όπου και τυπώνει με sout στην οθονη του χρήστη μέσω με τιμές που παίρνει από τις μεθόδους που δημιούργησα της κλάσης Team

η συγκεκριμένη τυπώνει στην οθόνη του χρήστη τα στατιστικά της ομάδας και τοποθετεί επιεπλέον και ένα διακριτικό διαχωριστικό\*/

game.showTeamStats();

// Show Players Statistics

/\*Βλέπω ότι η Κλάση Game έχει δύο μεθόδους όπου και τυπώνει με sout στην οθονη του χρήστη μέσω με τιμές που παίρνει από τις μεθόδους που δημιούργησα της κλάσης Team

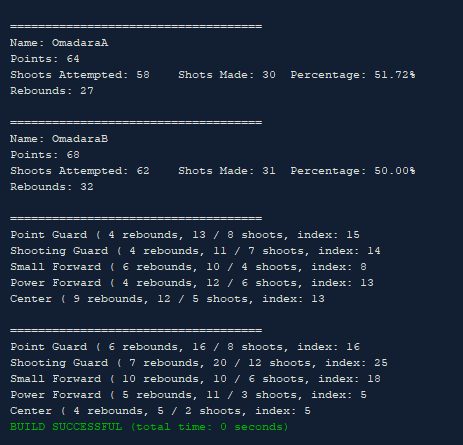
η συγκεκριμένη τυπώνει στην οθόνη του χρήστη τα στατιστικά της των παιχτών και τοποθετεί επιεπλέον και ένα διακριτικό διαχωριστικό\*/

game.showPlayersStats();

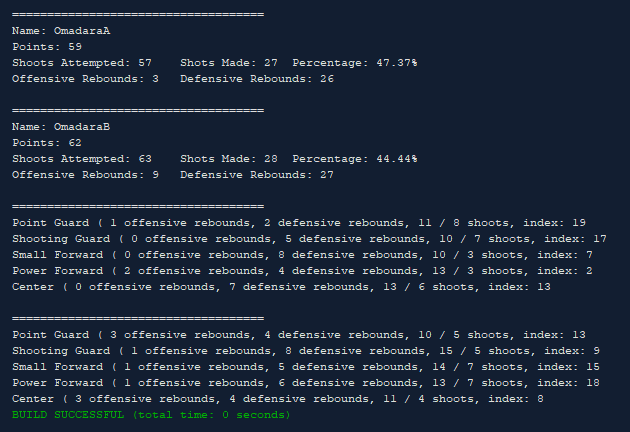
}//end of main metho

}//end of Main Class

**Στιγμιότυπο Οθόνης με Αποτέλεσμα Εκτέλεσης:**



**Στιγμιότυπο Οθόνης με Αποτέλεσμα Εκτέλεσης (αν έχετε απαντήσει το Ερώτημα 3.Γ):**



|  |
| --- |
| **Υποδείξεις για τη συγγραφή της εργασίας**  **1)** Για την απάντηση της εργασίας θα πρέπει να χρησιμοποιηθεί το κείμενο της εκφώνησης της εργασίας. Στο κείμενο αυτό:  - Συμπληρώστε, στο χώρο των απαντήσεων, όλα τα στοιχεία με κίτρινο χρώμα.  - Μην ξεχάσετε να δηλώσετε εάν η εργασία αποτελεί προϊόν αποκλειστικά δικής σας εργασίας.  - Ενσωματώστε τις απαντήσεις (διαγράμματα ή/και κώδικα) στην κατάλληλη θέση. Δεν θα πρέπει να κάνετε παραπομπές της μορφής «βλέπε αρχείο…».  - Αν δεν έχετε απαντήσει σε ένα ερώτημα γράψτε «**ΔΕΝ ΑΠΑΝΤΗΘΗΚΕ**».  - Αν απαντήσατε με ελλείψεις σε ένα ερώτημα γράψτε «**ΕΛΛΙΠΗΣ ΑΠΑΝΤΗΣΗ**».  **2)** Η συνεργασία στην ανάλυση της εργασίας επιτρέπεται, αλλά καλό είναι να αναφερθεί στον ειδικό χώρο στην πρώτη σελίδα της εργασίας. Η συνεργασία δεν πρέπει να οδηγεί σε από κοινού επίλυση και συγγραφή της εργασίας. Η υποβολή κοινών απαντήσεων από διαφορετικούς φοιτητές που συνεργάστηκαν δεν επιτρέπεται και θεωρείται ως **ΑΝΤΙΓΡΑΦΗ**. Οι απαντήσεις ελέγχονται, τόσο μεταξύ των φοιτητών του ιδίου τμήματος, όσο και μεταξύ φοιτητών διαφορετικών τμημάτων. 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.  **3)** Η εκπόνηση της εργασίας θα πρέπει να γίνει **αποκλειστικά και υποχρεωτικά** με το εργαλείο Visual Paradigm για τη UML και NetΒeans για τη Java.  **4)** Η εργασία θα υποβληθεί στο σύστημα υποβολή εργασιών του study.eap.gr.  **5)** Ο φοιτητής θα πρέπει να υποβάλει την εργασία του σε δύο αρχεία:   * **Το 1ο αρχείο θα έχει όνομα PLH24\_1ERG\_EPITHETO\_ONOMA.doc** και είναι το κείμενο της εκφώνησης της εργασίας με συμπληρωμένες τις απαντήσεις. * **Το 2ο αρχείο είναι ένα συμπιεσμένο αρχείου zip ή rar** με όνομα PLH24\_1ERG\_EPITHETO\_ONOMA.<rar|zip>. Το συμπιεσμένο αρχείο θα πρέπει να αποτελείται από:   i. Το αρχείο Visual Paradigm με το project που θα περιέχει τα διαγράμματα των ασκήσεων.  ii. Τον κατάλογο με τον κώδικα Java που θα πρέπει να περιλαμβάνει το project όπως αυτό δημιουργείται από το εργαλείο NetΒeans και το οποίο θα πρέπει να μπορεί να εκτελείται χωρίς αλλαγές από τον καθηγητή.  iii. Να γίνει χρήση λατινικών χαρακτήρων **ΑΠΟΚΛΕΙΣΤΙΚΑ** για την αποφυγή προβλημάτων με το moodle.  H εφαρμογή των παραπάνω κανόνων είναι **ΥΠΟΧΡΕΩΤΙΚΗ** και βαθμολογείται σύμφωνα με το αντίστοιχο κριτήριο αξιολόγησης. Η μη εφαρμογή του πρώτου κανόνα μπορεί να οδηγήσει σε συνολική απόρριψη της εργασίας.    **ΚΑΛΗ ΕΠΙΤΥΧΙΑ!** |