ΠΛΗ24 1η ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2021-2022

ΥΠΟΔΕΙΓΜΑΤΙΚΗ ΛΥΣΗ

ΠΑΠΑΘΕΟΔΩΡΟΥ ΚΩΝΣΤΑΝΤΙΝΟΣ
k.papatheodorou@arnos.gr

Ημερομηνία παράδοσης εργασίας ***Τετάρτη 17/11/2021***

ΣΤΟΙΧΕΙΑ ΠΟΥ ΣΥΜΠΛΗΡΩΝΕΙ Ο ΦΟΙΤΗΤΗΣ / Η ΦΟΙΤΗΤΡΙΑ

Ονοματεπώνυμο Φοιτητή/Φοιτήτριας:	
Αριθμός Μητρώου:	
Κωδικός Θ.Ε.:	**ΠΛΗ24**
Κωδικός Τμήματος:	
Α/Α Γραπτής Εργασίας:	**1η ΕΡΓΑΣΙΑ**
Ονοματεπώνυμο Καθηγητή:	
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.*

	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:

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.Α	Υλοποίηση Μεθόδου showStats	10	
3.Β	Υλοποίηση Μεθόδου showPlayersStats	15	
3.Γ	Επέκταση Κώδικα Εφαρμογής	20	
3.Ε	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, K., 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:

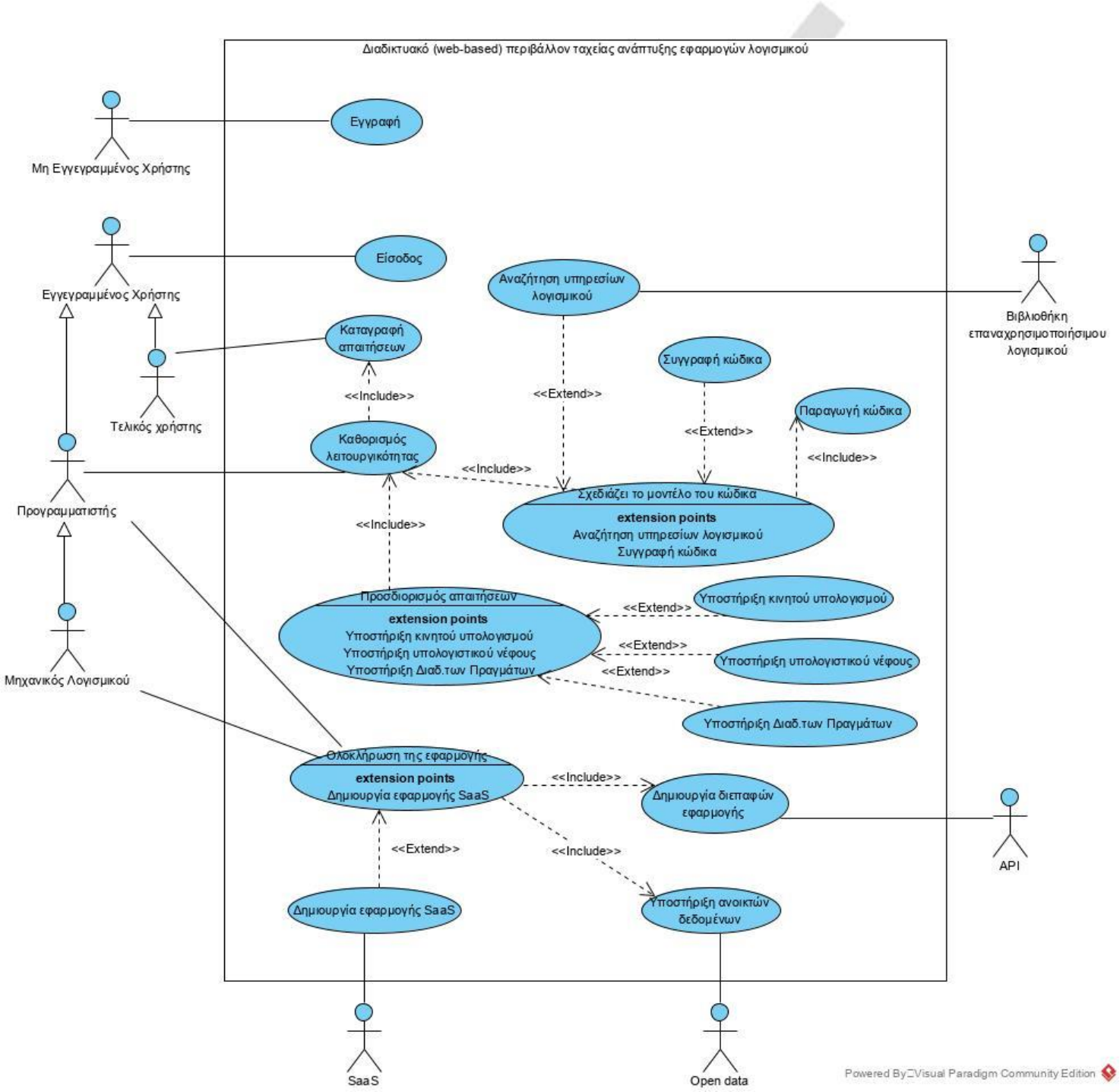
- τον ορισμό των απαιτήσεων και
- τους δύο κύριους τύπους των απαιτήσεων.

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

Διάγραμμα Περιπτώσεων Χρήσης:



Αιτιολόγηση Σχέσεων:

Από την ανάλυση των δεδομένων της εκφώνησης εντοπίζουμε τους εξής Χειριστές του Συστήματος:

1) **Μη Εγγεγραμμένος Χρήστης**
2) **Εγγεγραμμένος Χρήστης**
3) **Τελικός χρήστης**
4) **Προγραμματιστής**
5) **Μηχανικός Λογισμικού**
6) **Βιβλιοθήκη επαναχρησιμοποιήσιμου λογισμικού (Εξωτερικό σύστημα)**
7) **Open data (Εξωτερικό σύστημα)**
8) **API (Εξωτερικό σύστημα)**
9) **SaaS (Εξωτερικό σύστημα)**
   (Το Software as a Service, γνωστό στα ελληνικά ως Λογισμικό ως Υπηρεσία, είναι ένα μοντέλο παροχής λογισμικού κατά το οποίο το λογισμικό και τα σχετικά δεδομένα φιλοξενούνται στο Νέφος)

Σχόλια: Ο Τελικός χρήστης, ο Προγραμματιστής, ο Μηχανικός Λογισμικού είναι **Εγγεγραμμένοι χρήστες Κληρονομικότητα**. Οι **Μηχανικοί λογισμικού** (οι οποίοι **μπορεί να είναι** και **Προγραμματιστές**) **Κληρονομικότητα.**

ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ

Ο **Μη Εγγεγραμμένος** Χρήστης Χρήστης μπορεί:
Να κάνει **Εγγραφή**.
Ο **Εγγεγραμμένος Χρήστης** μπορεί:
Να κάνει **είσοδο** στο σύστημα.
Ο **Τελικός χρήστης** μπορεί:
Να **καταγράψει** τις **απαιτήσεις** του**.**

Ο **Προγραμματιστής** μπορεί:

1) Να **καθορίζει** τη λειτουργικότητα μιας εφαρμογής
2) **Να σχεδιάζει το μοντέλο του κώδικα (πραγματοποιεί κάθε φορά include)**
3) Να **αναζητήσει** έτοιμες στοιχειώδεις υπηρεσίες λογισμικού**(είναι πιθανό extend)**
4) Να **προχωρήσει** στην (περιορισμένη και στοχευμένη) συγγραφή κώδικα **(είναι πιθανό extend)**
5) **Να προσδιορίζει μια σειρά από ειδικές απαιτήσεις (πραγματοποιεί κάθε φορά include)**
6) Να **προσδιορίζει** την υποστήριξη κινητού υπολογισμού **(είναι πιθανό extend)**
7) Να **προσδιορίζει** την υποστήριξη υπολογιστικού νέφους (cloud) **(είναι πιθανό extend)**
8) Να **προσδιορίζει** την υποστήριξη Διαδικτύου των Πραγμάτων (Internet of Things, IoT) **(είναι πιθανό extend)**

Επίσης με την ολοκλήρωση της σχεδίασης του μοντέλου του κώδικα **παράγεται με αυτόματο τρόπο ο κώδικας (include)**

Ο Μηχανικός λογισμικού μπορεί: (σε συνεργασία με τον προγραμματιστή)

1) Να **προχωρήσει** στην ολοκλήρωση (integration) της εφαρμογής.
2) Να **δημιουργήσει** κατάλληλα προγραμματιστικές διεπαφες εφαρμογής**(**είναι αναγκαία **include)**
3) Να **υποστηρίξει** τα ανοικτά δεδομένων (open data) **(**είναι αναγκαία **include)**
4) Να **δημιουργήσει** μια ***εφαρμογή SaaS*** (Software as a Service, SaaS). **(**δεν είναι πάντα εφικτός, **extend)**

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

.

Ιστορία Χρήστη 1:

Όνομα Ιστορίας Χρήστη: Είσοδος ενός εγγεγραμμένου χρήστη στο σύστημα

Περιγραφή: Ως ένας εγγεγραμμένος χρήστης **θέλω να** εισέλθω στο σύστημα ***έτσι ώστε*** να συμμετέχω στην **Ανάπτυξη Χαμηλού Κώδικα**

Κριτήρια Αποδοχής:

ΚΑ#1: Για να μπορεί ένας χρήστης να χρησιμοποιήσει το σύστημα θα πρέπει **πρώτα** να εγγραφεί

ΚΑ#2: Ο χρήστης κατά την **εγγραφή** θα **πρέπει** να δηλώσει τι είναι **προγραμματιστής** ή **μηχανικός λογισμικού** ή **τελικός χρήστης**.

ΚΑ#3: Ο χρήστης κατά την **εγγραφή** εάν είναι **προγραμματιστής** θα πρέπει να δηλώσει τις **γνώσεις** που έχει καθώς και την **εμπειρία**.

ΚΑ#4: Αν ο εγγεγραμμένος χρήστης εισάγει **σωστά** το όνομα χρήστη / κωδικού πρόσβασης τότε θα μπορεί να εισέλθει στο σύστημα. **(σενάριο επιτυχίας)**

ΚΑ#5: Αν ο εγγεγραμμένος χρήστης εισάγει **λάθος** το όνομα χρήστη / κωδικού πρόσβασης τότε το σύστημα θα του εμφανίσει **κατάλληλο μήνυμα** (**σενάριο αποτυχίας**).

Ιστορία Χρήστη 2:

Όνομα Ιστορίας Χρήστη: Αναζήτηση **Προγραμματιστή έτοιμων στοιχειωδών υπηρεσιών λογισμικού** (**components**) σε κατάλληλη εξωτερική του υπό εξέταση συστήματος **βιβλιοθήκη επαναχρησιμοποιήσιμου λογισμικού**.

Περιγραφή: ***Ως ένας*** Προγραμματιστής, ***θέλω να*** αναζητήσω έτοιμες στοιχειώδεις υπηρεσίες λογισμικού (components) ***έτσι ώστε*** να σχεδιάσω το μοντέλο του κώδικα.

Κριτήρια Αποδοχής:

ΚΑ#1: Για να μπορεί ένας **Προγραμματιστής** να αναζητήσει κάποια έτοιμη υπηρεσία λογισμικού θα πρέπει **πρώτα** να πραγματοποιήσει την **είσοδό** του στο σύστημα.

ΚΑ#2: Θα πρέπει οι **τελικοί χρήστες** να έχουν **καταγράψει** τις **απαιτήσεις**.

ΚΑ#3: Το σύστημα θα πρέπει να του παρέχει **αναζήτηση** με χρήση **λέξεων κλειδιών**

ΚΑ#4: Το σύστημα θα πρέπει να μπορεί να **ταξινομήσει** τα **αποτελέσματα** της **αναζήτησης**

ΚΑ#5: Το σύστημα θα πρέπει να μπορεί να **συνδεθεί** σε εξωτερική του συστήματος, κατάλληλη **βιβλιοθήκη επαναχρησιμοποιήσιμου λογισμικού**

ΚΑ#6: Με την ολοκλήρωση της σχεδίασης του μοντέλου του κώδικα θα πρέπει το σύστημα να **παράγει** με **αυτόματο τρόπο** τον κώδικα που υλοποιεί την εφαρμογή

Εκφώνηση 2ης Άσκησης

Σύστημα Ηλεκτρονικού Παιχνιδιού για την Ελληνική Επανάσταση του 1821

Ερώτημα 2.Α – Δημιουργία Εννοιολογικού Μοντέλου

Το 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:

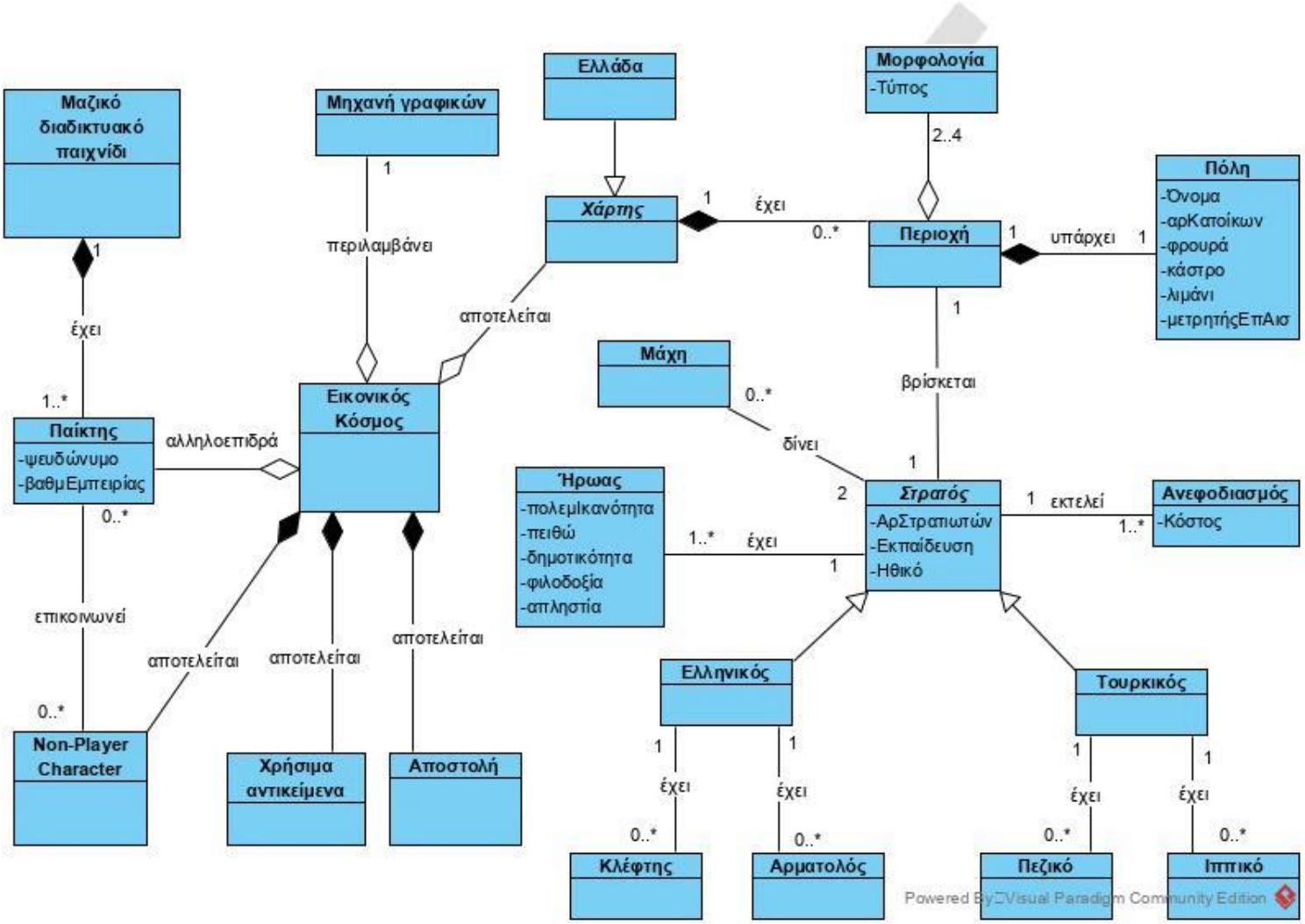
- την κατασκευή του «Εννοιολογικού Μοντέλου» και
- την κατασκευή του «Διαγράμματος Κλάσεων».

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

Εννοιολογικό Μοντέλο (Αρχικό Διάγραμμα Κλάσεων):



Αιτιολόγηση Χρήσης Κληρονομικότητας (inheritance), Συσσώρευσης (aggregation) ή Συγκρότησης (composition):

Το παιχνίδι διεξάγεται σε ένα ***χάρτη*** της ***Ελλάδας***, **Κληρονομικότητα (inheritance) ώστε το παιχνίδι μας να μπορεί στο μέλλον να επεκταθεί και σε άλλο χάρτη.**

Ο χάρτης είναι χωρισμένος σε ***περιοχές*** **Συγκρότηση (composition) ισχυρή συσχέτιση ένας χάρτης έχει συγκεκριμένες περιοχές.**

Σε κάθε περιοχή υπάρχει μία ***πόλη*** **Συγκρότηση (composition) ισχυρή συσχέτιση μία περιοχή έχει συγκεκριμένη πόλη.**

Οι περιοχές αποτελούν **αναπόσπαστο** τμήμα του χάρτη της Ελλάδας και η κάθε πόλη αποτελεί **αναπόσπαστο** τμήμα της κάθε περιοχής. **Συγκρότηση (composition).**

Υπάρχουν δύο (2) ***στρατοί***, ο Ελληνικός και ο Τουρκικός **Κληρονομικότητα (inheritance) ώστε να κληρονομούν** τον αριθμό των στρατιωτών, την εκπαίδευση και το ηθικό.

Οι **παίκτες** του παιχνιδιού με την κλάση **Μαζικό διαδικτυακό παιχνίδι** έχουμε **Συγκρότηση (composition) αφού είναι αναπόσπαστο τμήμα του παιχνιδιού.** Επίσης οι **παίκτες** του παιχνιδιού θεωρούμε ότι έχουμε ανεξαρτησία **Συσσώρευση (aggregation)** με την κλάση **Εικονικός Κόσμος**

Ο **Εικονικός Κόσμος** (Virtual World) του παιχνιδιού, ο οποίος αποτελείται από το χάρτη της **Ελλάδας** έχουμε **Συσσώρευση (aggregation).** Θεωρούμε ότι ο Χάρτης είναι το **παιδί** που μπορεί να υπάρχει **ανεξάρτητα** από τον **γονέα καθώς** μπορεί να τον χρησιμοποιήσουμε σε άλλο **Εικονικό Κόσμο.**

Τα **χρήσιμα αντικείμενα** (που οι παίκτες συλλέγουν), Οι **αποστολές** (που οι παίκτες αναλαμβάνουν), οι 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**. **Συγκρότηση (composition) ισχυρή συσχέτιση**

Αντίθετα η **μηχανή γραφικών** που περιλαμβάνει ο Εικονικός Κόσμος του παιχνιδιού **μπορεί να χρησιμοποιηθεί** για τη δημιουργία και **άλλων παιχνιδιών**. **Συσσώρευση (aggregation).**

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)

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.

Απάντηση 3ης Άσκησης / Ερωτήματα Α και Β

Κώδικας Μεθόδου showStats:

```
public void showStats() {

        // Show the statistics of each team in following format
        // Name
        System.out.println("Team Name :" + name);
```

```
        // Points:
        System.out.println("Points    :" + score);

        // Shoots Attempted:      Shots Made:         Percentage:
        int attempted = 0;
        int made = 0;
        //Διαπερνάω τον πίνακα με τους παίχτες και υπολογίζω τα
        //Shoots Attempted και Shots Made
        for (int i = 0; i < this.players.length; i++) {
            //Shoots Attempted
            attempted = attempted + this.players[i].fg_attempted;
            //Shots Made
            made = made + this.players[i].fg_made;
        }
        //Υπολογισμός ποσοστού
        int percentage = (made * 100) / attempted;
        System.out.println("Shoots Attempted:" + attempted + "   Shots Made:" + made +
"   Percentage:" + percentage + "%");

        // Rebounds
        int defRebounds = 0;
        int offRebounds = 0;

        //Διαπερνάω τον πίνακα με τους παίχτες και υπολογίζω τα Rebounds
        for (int i = 0; i < this.players.length; i++) {
            //Υπολογίζω τα DefRebounds και τα OffRebounds
            defRebounds = defRebounds + this.players[i].getDefRebounds();
            offRebounds = offRebounds + this.players[i].getOffRebounds();
        }
        System.out.println("DefRebounds :" + defRebounds + "  OffRebounds :" +
offRebounds);
}
```

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

        // Name
        System.out.println("Team Name : " + name);

        for (int i = 0; i < this.players.length; i++) {
            //Player name
            System.out.println("Player:" + this.players[i].getName());
            //Rebounds = DefRebounds + OffRebounds
            int defRebounds = this.players[i].getDefRebounds();
            int offRebounds = this.players[i].getOffRebounds();
            //Shoots Attempted
            int attempted = this.players[i].fg_attempted;
            //Shots Made
            int made = this.players[i].fg_made;

            // index = points + rebounds - missed shots
            int missed = attempted - made;
            int index = this.players[i].getPoints() + (defRebounds + offRebounds) -
missed;
            System.out.println("Position Points (" + defRebounds + " DefRebounds, "
+offRebounds + " OffRebounds, "+ made + "/" + attempted + " shoots, index:" + index +
")");
        }
    }
```

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.Β και 3.Γ, 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

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

```
public class Player {

    String pos;
    int points;
    int offRebounds;//Επιθετικά rebounds
    int defRebounds;//Αμυντικά rebounds
    int fg_made;
    int fg_attempted;

    public Player(String p) {
        pos = p;
        points = 0;
        offRebounds = 0; //Επιθετικά rebounds
        defRebounds = 0;//Αμυντικά rebounds
        fg_made = 0;
        fg_attempted = 0;
    }

//Αυξάνω τα Επιθετικά offRebounds
    public void increaseOffRebounds() {
        offRebounds++;
    }

//Αυξάνω τα Αμυντικά defRebounds
    public void increaseDefRebounds() {
        defRebounds++;
    }

//Getter Επιθετικά offRebounds
    public int getOffRebounds() {
        return offRebounds;
    }

//Getter Αμυντικά defRebounds
    public int getDefRebounds() {
        return defRebounds;
    }

//Get Points
    public int getPoints() {
        return points;
    }

}
```

```
public class Team {

public int offensiveRebound() {
        // There is a 20% chance to get an offensive rebound
        Random rand = new Random();
        Random pl_rand = new Random();
        int reb_outcome = rand.nextInt(1000);
        int rebounder = pl_rand.nextInt(5);
        if (reb_outcome > 800) {
            System.out.println(name + "'s " + players[rebounder].getName() + " gets
the offensive rebound");
            //increaseOffRebounds
            players[rebounder].increaseOffRebounds();
            return 1;
        } else {
            return 0;
        }
    }

    public void defensiveRebound() {
        Random pl_rand = new Random();
        int rebounder = pl_rand.nextInt(5);
        System.out.println(name + "'s " + players[rebounder].getName() + " gets the
defensive rebound");
        //increaseDefRebounds
        players[rebounder].increaseDefRebounds();
    }
}
```

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. **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 4.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:

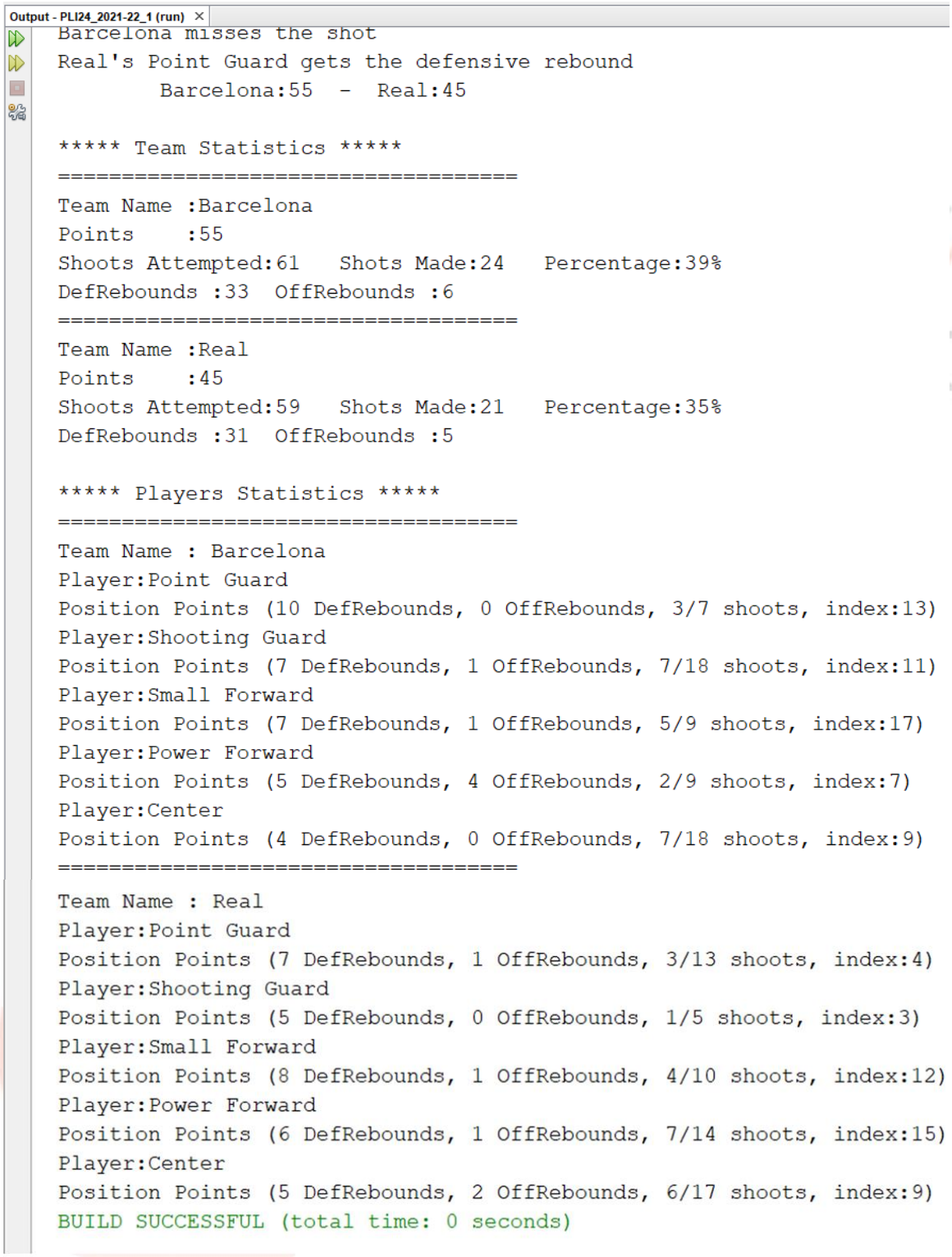
- να ορίσετε με ακρίβεια την έννοια του αντικειμένου,
- να εξηγήσετε τη διαφορά μεταξύ μιας κλάσης και ενός αντικειμένου και
- να κατασκευάσετε αντικείμενα μιας κλάσης με διαφορετικά χαρακτηριστικά σε Java.

Απάντηση 3ης Άσκησης / Ερώτημα Δ

Κώδικας Μεθόδου main:

```
public static void main(String[] args) {

    //Δημιουργήστε δυο ομάδες της αρεσκείας σας
    Team team1 = new Team("Barcelona");
    Team team2 = new Team("Real");

    // Create a Game
    Game game1 = new Game(team1, team2);

    // Simulate Game
    game1.simulateGame();

    // Show Team Statistics
    System.out.println("***** Team Statistics *****");
    game1.showTeamStats();
    System.out.println("");

    // Show Players Statistics
    System.out.println("***** Players Statistics *****");
    game1.showPlayersStats();
}
```

Στιγμιότυπο Οθόνης με Αποτέλεσμα Εκτέλεσης:



```
Output - PLI24_2021-22_1 (run)
Barcelona misses the shot
Real's Point Guard gets the defensive rebound
        Barcelona:55  -  Real:45

***** Team Statistics *****
====================================
Team Name :Barcelona
Points    :55
Shoots Attempted:61   Shots Made:24   Percentage:39%
DefRebounds :33  OffRebounds :6
====================================
Team Name :Real
Points    :45
Shoots Attempted:59   Shots Made:21   Percentage:35%
DefRebounds :31  OffRebounds :5

***** Players Statistics *****
====================================
Team Name : Barcelona
Player:Point Guard
Position Points (10 DefRebounds, 0 OffRebounds, 3/7 shoots, index:13)
Player:Shooting Guard
Position Points (7 DefRebounds, 1 OffRebounds, 7/18 shoots, index:11)
Player:Small Forward
Position Points (7 DefRebounds, 1 OffRebounds, 5/9 shoots, index:17)
Player:Power Forward
Position Points (5 DefRebounds, 4 OffRebounds, 2/9 shoots, index:7)
Player:Center
Position Points (4 DefRebounds, 0 OffRebounds, 7/18 shoots, index:9)
====================================
Team Name : Real
Player:Point Guard
Position Points (7 DefRebounds, 1 OffRebounds, 3/13 shoots, index:4)
Player:Shooting Guard
Position Points (5 DefRebounds, 0 OffRebounds, 1/5 shoots, index:3)
Player:Small Forward
Position Points (8 DefRebounds, 1 OffRebounds, 4/10 shoots, index:12)
Player:Power Forward
Position Points (6 DefRebounds, 1 OffRebounds, 7/14 shoots, index:15)
Player:Center
Position Points (5 DefRebounds, 2 OffRebounds, 6/17 shoots, index:9)
BUILD SUCCESSFUL (total time: 0 seconds)
```

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 NetBeans 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 NetBeans 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.

Η 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!