Object Oriented Programming Course (CEID_NNY106)

Εργαστηριακή Άσκηση

WindowsApp

1. Στόχος

Η Εργαστηριακή Άσκηση (ΕΑ) έχει στόχο την εξοικείωση με:

- Την ανάπτυξη εφαρμογής με βάση την περιγραφή ενός **σεναρίου χρήσης** της
- Την αναγνώριση αντικειμένων που απαρτίζουν μια εφαρμογή
- Την ανάθεση αρμοδιοτήτων (responsibilities) σε αυτά
- Τον ορισμό της συνεργασίας (collaboration) των αντικειμένων για να έχουμε την απαιτούμενη συμπεριφορά σε επίπεδο εφαρμογής
- Την χρήση του τύπου αναφοράς πίνακα και των κλάσεων **ArrayList**, **LinkedList** και **Vector** της βασικής βιβλιοθήκης που είναι βασικές της κατηγορίας **Collections** της Java.
- την Incremental development τεχνική στην ανάπτυξη αντικειμενοστρεφούς εφαρμογής

WindowsApp

Η Εργαστηριακή Άσκηση βασίζεται στη Δραστηριότητα 5.9 του κεφαλαίου 5 (Ενότητα 5.14) του βιβλίου "Από τις Υπολογιστικές στις Κυβερνο-Φυσικές Διεργασίες και το IoT: Αντικείμενα και Υπηρεσίες", Κ. Θραμπουλίδης, ISBN 978-960-418-961-8, 2022, ΕΚΔΟΣΕΙΣ Α. ΤΖΙΟΛΑ & ΥΙΟΙ Α.Ε. https://sites.google.com/view/fromcomputationaltocyber-physi/home

2. Αναμενόμενα αποτελέσματα

Με την ολοκλήρωση αυτής της ΕΑ θα έχετε εξοικειωθεί με την αναφορά τύπου πίνακα της Java καθώς και με τις βασικές κλάσεις ArrayList, LinkedList και Vector του πακέτου util της βασικής βιβλιοθήκης και την αξιοποίηση τους στην ανάπτυξη εφαρμογών καθώς και των διαφορών τους από τον τύπο του πίνακα. Θα έχετε εφαρμόσει την top-down προσέγγιση στην διαδικασία ανάπτυξης αντικειμενοστρεφούς εφαρμογής.

3. Δομή ΕΑ

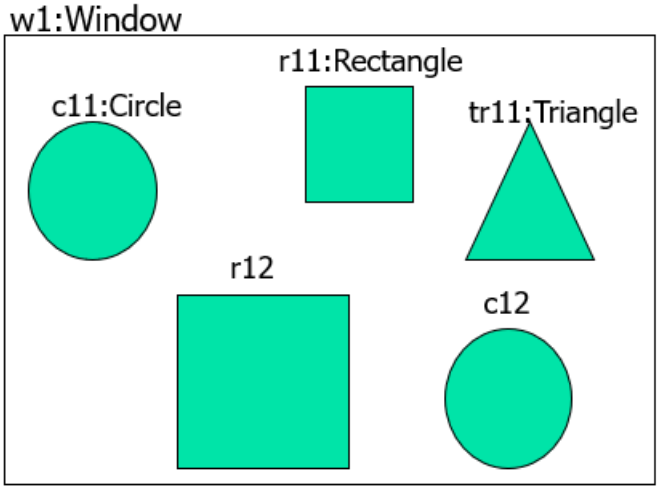
Η ΕΑ αποτελείται από τρεις ενότητες. Στην ενότητα 4 περιγράφεται το πρόβλημα. Στην ενότητα 5 σας δίνονται οδηγίες για την ανάπτυξη μιας πρώτης έκδοσης του προγράμματος. Στην έκδοση αυτή θα εντοπίσουμε προβλήματα τα οποία θα αντιμετωπίσουμε σε άλλη ΕΑ.

4. Περιγραφή Σεναρίου

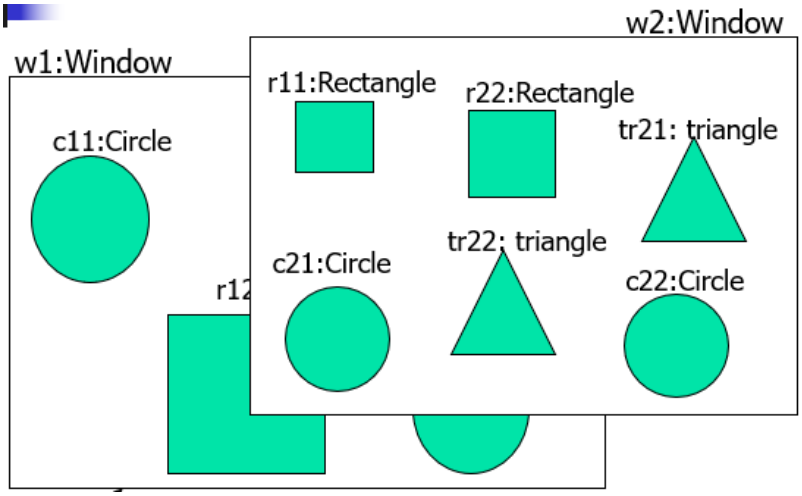
Σε εφαρμογές στις οποίες ο χρήστης έχει τη δυνατότητα να δημιουργεί παράθυρα και το κάθε παράθυρο έχει το δικό του περιεχόμενο συναντάμε το παρακάτω σενάριο. Ο χρήστης εκτελεί τις παρακάτω ενέργειες:

1. δημιουργεί ένα παράθυρο w1
2. δημιουργεί μέσα σε αυτό δύο τετράγωνα (r11, r12), δύο κύκλους (c11,c12), και ένα τρίγωνο (tr11) (**Σχήμα 4-1**).
3. Στη συνέχεια δημιουργεί ένα άλλο παράθυρο w2 που επικαλύπτει το w1.
4. Δημιουργεί μέσα σε αυτό δύο τετράγωνα (r21,r22), δύο κύκλους (c21,c22) και δυο τρίγωνα (tr21,tr22) (**Σχήμα 4-2**),

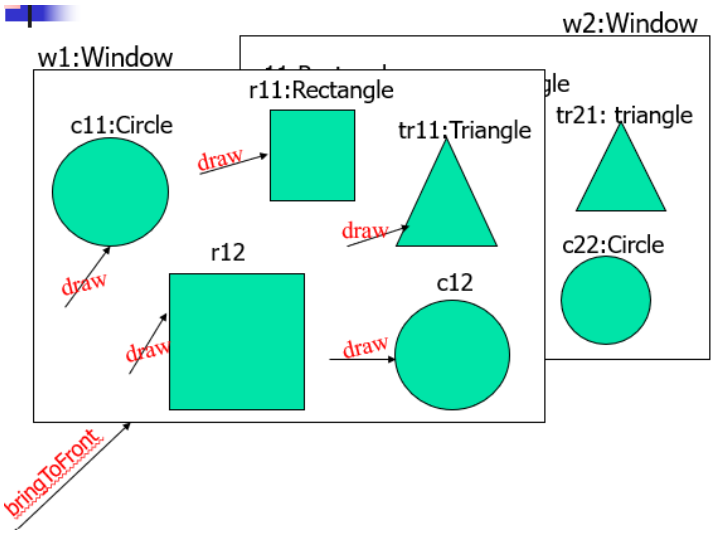
5. Στη συνέχεια επιλέγει το w1 και με δεξί κλικ επιλέγει BringToFront. Η εφαρμογή φέρνει το παράθυρο w1 μπροστά από το w2 (**Σχήμα 4-3**).



Σχήμα 4-1. *Δημιουργία παραθύρου w1 και των περιεχομένων του*



Σχήμα 4-2 *Δημιουργία παραθύρου w2 και των περιεχομένων του*



Σχήμα 4-3 *Αποστολή μηνύματος bringToFront στο παράθυρο w1*

5. Η Εφαρμογή WindowsApp – 1η Έκδοση

Σας ζητείται να αναπτύξετε μια εφαρμογή WindowsApp. Η WindowsApp θα έχει στόχο να επιδείξει την βασική δομή των κλάσεων που εντοπίζονται στην περιγραφή του σεναρίου που δόθηκε στην περιγραφή του Προβλήματος, ώστε να είναι δυνατή η υλοποίηση της λειτουργικότητας bringToFront στην κλάση Window της εφαρμογής.

Δεν θα ασχοληθείτε με τη γραφική αναπαράσταση του παραθύρου και των περιεχομένων του, καθώς αφενός δεν είστε σε θέση να το κάνετε, αφετέρου δεν είναι αυτός ο στόχος της ΕΑ. Στην θέση του κώδικα που υλοποιεί την εμφάνιση στην οθόνη του πλαισίου του παραθύρου μπορείτε να εμφανίζετε κατάλληλο μήνυμα στην οθόνη που απλά θα ενημερώνει για την λειτουργία αυτή, π.χ. "drawing w1 on screen". Το ίδιο θα κάνετε και για τον κύκλο, το ορθογώνιο και το τρίγωνο.

5.1 Οδηγίες εκτέλεσης

5.1.1 Δράση 1 – Δημιουργία διαγράμματος κλάσεων (class diagram)

Σχεδιάστε ένα διάγραμμα κλάσεων (class diagram) που θα καταγράφει τις βασικές κλάσεις του προγράμματος και τις μεταξύ τους σχέσεις. Καταγράψτε τα βασικά αντικείμενα και αναθέστε αρμοδιότητες σε αυτά.

Κατά τη διάρκεια της διάλεξης δώσαμε ένα διάγραμμα κλάσεων που καταγράφει και τις αναθέσεις αρμοδιοτήτων στα επιμέρους αντικείμενα που απαρτίζουν την εφαρμογή.

5.1.2 Δράση 2- Περιγραφή συμπεριφοράς της εφαρμογής με Διάγραμμα Ακολουθίας

Σχεδιάστε ένα διάγραμμα ακολουθίας που θα περιγράφει τη συνεργασία των αντικειμένων της εφαρμογής για να υλοποιήσει αυτή την απαιτούμενη λειτουργικότητα.

Κατά τη διάρκεια της διάλεξης δώσαμε μέρος του διαγράμματος ακολουθίας το οποίο περιγράφει τη λειτουργικότητα που πρέπει να υλοποιεί η main της WindowApp.

5.1.3 Δράση 3 – Συγγραφή πηγαίου κώδικα

Για την συγγραφή του πηγαίου κώδικα έχουμε εναλλακτικές υλοποιήσεις. Μπορείτε να αξιοποιήσετε:

- πίνακα, για να εξοικειωθείτε με τη διαχείριση πινάκων σε ένα αντικειμενοστρεφές περιβάλλον, ή
- μία από τις κλάσεις Vector, ArrayList, και LinkedList της βασικής βιβλιοθήκης της Java για να εξοικειωθείτε με τις Συλλογές (Collections) που θα βρείτε σε αντικειμενοστρεφή περιβάλλοντα.

Οι δράσεις 4-7 είναι προς την κατεύθυνση αυτή. **Θα πρέπει να υλοποιήσετε οπωσδήποτε την έκδοση 1.1.**

5.1.4 Δράση 4 – Έκδοση 1.1 (java arrays)

Για να απλοποιήσετε την πρώτη αυτή έκδοση ενσωματώστε στον πηγαίο κώδικα τη δημιουργία των κύκλων, τετραγώνων και τριγώνων χωρίς να δίνει τα στοιχεία τους ο χρήστης. Ενσωματώστε

επίσης και την "τοποθέτηση" τους μέσα στα παράθυρα w1 και w2. Ακολουθήστε δηλαδή την **hardcoding τεχνική**[1].

Δώστε έμφαση στο πως θα κατανεμηθούν οι αρμοδιότητες (**responsibilities**) στα διάφορα αντικείμενα που θα απαρτίζουν την εφαρμογή στο χρόνο εκτέλεσης και διαμορφώστε κατάλληλα τους ορισμούς των αντίστοιχων κλάσεων, ενημερώνοντας κατάλληλα το διάγραμμα κλάσεων. Αποφασίστε δηλαδή, για παράδειγμα, σε ποια κλάση θα βάλετε τη λειτουργικότητα της εμφάνισης του κύκλου στην οθόνη. Η ανάθεση αρμοδιοτήτων επηρεάζει άμεσα φυσικά και το πως θα πρέπει να ορίσετε τον τρόπο συνεργασίας (**collaboration**) των αντικειμένων για να έχετε σε επίπεδο συστήματος την λειτουργικότητα που θέλετε από την εφαρμογή σας. Αξιοποιήστε το διάγραμμα ακολουθίας για να περιγράψετε την συμπεριφορά της εφαρμογής σας.

Μετά την δημιουργία των διαγραμμάτων κλάσεων και ακολουθίας έχετε την βασική δομή της εφαρμογής σας, δηλαδή τις κλάσεις από τις οποίες αποτελείται με την δομή και συμπεριφορά αυτών. Η συγγραφή του κώδικα της main ανάγεται σε μετατροπή του διαγράμματος ακολουθίας που περιγράφει το σενάριο σε κώδικα.

Στην ενότητα Αναλυτικές Οδηγίες θα βρείτε περισσότερες Οδηγίες. Μελετήστε τις μόνο μετά από την δική σας προσπάθεια να δώσετε την βασική δομή της εφαρμογής σας. Αξιοποιήστε τις οδηγίες για να διαμορφώσετε την πρώτη λειτουργούσα έκδοση της WindowApp.

Q1: Κάντε μια προσπάθεια να εντοπίσετε προβλήματα που έχει η υλοποίηση αυτή. Καταγράψτε τα.

Μετά την ολοκλήρωση της έκδοσης 1.1 **έχετε δύο επιλογές**. Η μία είναι να περάσετε στην έκδοση 1.2 για να δείτε τα πλεονεκτήματα που σας δίνει η αξιοποίηση της κλάσης ArrayList σε σύγκριση με τον πίνακα. Η δεύτερη είναι να πάτε στην έκδοση 2.1 που αξιοποιεί τον μηχανισμό της κληρονομικότητας.
Προτιμότερη είναι η δεύτερη επιλογή. Και όταν ολοκληρώσετε την 2.1 μπορείτε να δείτε την αξιοποίηση

5.1.5 Δράση 5 – Έκδοση 1.2 (ArrayList)

Αξιοποιήστε την έκδοση 1.1 για να διαμορφώσετε μια νέα έκδοση, την έκδοση 1.2, η οποία θα χρησιμοποιεί την **java.util. ArrayList<E>** της βασικής βιβλιοθήκης αντί για τον δικό σας πίνακα για να βάζετε τις αναφορές των κύκλων στο παράθυρο στο οποίο αυτοί ανήκουν. Το ίδιο θα κάνετε και για τα Τρίγωνα και Τετράγωνα.

Q2: Ποια είναι τα πλεονεκτήματα από την αξιοποίηση της `ArrayList` στην θέση του δικού σας πίνακα;

5.1.6 Δράση 6 – Έκδοση 1.3 (LinkedList)

Αξιοποιήστε την έκδοση 1.1 ή την 1.2 για να διαμορφώσετε μια νέα έκδοση, την έκδοση 1.3, η οποία θα χρησιμοποιεί την **java.util. LinkedList<E>** της βασικής βιβλιοθήκης αντί για τον δικό σας πίνακα για να βάζετε τις αναφορές των κύκλων στο παράθυρο στο οποίο αυτοί ανήκουν. Το ίδιο θα κάνετε και για τα Τρίγωνα και Τετράγωνα.

Q3: Ποια είναι τα πλεονεκτήματα από την αξιοποίηση της `LinkedList` στην θέση του δικού σας πίνακα;

5.1.7 Δράση 7 – Έκδοση 1.4 (Vector)

Αξιοποιήστε την έκδοση 1.1 για να διαμορφώσετε μια νέα έκδοση, την έκδοση 1.4, η οποία θα χρησιμοποιεί την **java.util. Vector<E>** της βασικής βιβλιοθήκης αντί για τον δικό σας πίνακα για

[1] "Hard coding (also hard-coding or hardcoding) is the software development practice of embedding data directly into the source code of a program or other executable object, as opposed to obtaining the data from external sources or generating it at runtime." Wikipedia

να βάζετε τις αναφορές των κύκλων στο παράθυρο στο οποίο αυτοί ανήκουν. Το ίδιο θα κάνετε και για τα Τρίγωνα και Τετράγωνα.

Q4: Ποια είναι τα πλεονεκτήματα από την αξιοποίηση της `Vector` στην θέση του δικού σας πίνακα;

Q5: Ποια από τις τρείς παραπάνω εκδόσεις θα προτιμούσατε και γιατί;

Q6: Κάντε μια προσπάθεια να εντοπίσετε προβλήματα που υπάρχουν και στις τέσσερεις παραπάνω υλοποιήσεις.

5.2 Αναλυτικές Οδηγίες

5.2.1 Έκδοση 1.1 (java arrays)

Σας συνιστούμε στην πρώτη σας υλοποίηση να ορίσετε στην κλάση Window ένα πίνακα circles όπου θα βάζετε τις αναφορές των κύκλων που ο χρήστης θα βάζει στο παράθυρο.

```
Circle [] circles;
```

Στον πίνακα αυτό θα τοποθετείτε την αναφορά του κύκλου που τοποθετείται στο παράθυρο με μία μέθοδο που μπορεί να έχει υπογραφή

```
addCircle(Circle c);
```

Στην έκδοση αυτή δεν είναι απαραίτητο να ορίσετε την θέση στην οποία τοποθετείται ο κύκλος μέσα στο παράθυρο.

Αντίστοιχα θα πράξετε για τα Τρίγωνα και τα Ορθογώνια οπότε θα προκύψουν οι μέθοδοι `addRectangle(Rectangle r)` και `addTriangle(Triangle tr)`.

Στη συνέχεια δοκιμάστε να αξιοποιήσετε το **method overloading** στον ορισμό των παραπάνω μεθόδων.

Ορίστε όλες τις κλάσεις που απαιτούνται για την υλοποίηση του σεναρίου που περιγράφεται στην περιγραφή του προβλήματος.

Διαμορφώστε στη συνέχεια την εφαρμογή σας ώστε να επιδεικνύει την σωστή λειτουργία του σεναρίου.

5.3 Ενδεικτικές Απαντήσεις

Δεν δίνονται.

6. Η Εφαρμογή WindowsApp – 2η Έκδοση

Έχοντας υλοποιήσει μια πλήρως λειτουργική 1η έκδοση της εφαρμογής και έχοντας εξοικειωθεί με τη διαδικασία ανάπτυξης ήμαστε σε θέση να αξιοποιήσουμε την κληρονομικότητα.

Η αξιοποίηση του μηχανισμού της κληρονομικότητας της γλώσσας θα μας επιτρέψει να δούμε τα πλεονεκτήματα που προσφέρει στην διαμόρφωση της εφαρμογής μας.

6.1 Οδηγίες εκτέλεσης

6.1.1 Δράση 8 – Αναγνώριση σχέσεων γενίκευσης/εξειδίκευσης

Ξεκινήστε με την αναγνώριση σχέσεων γενίκευσης/εξειδίκευσης μεταξύ των εννοιών που εμπλέκονται στην εφαρμογή. Διαμορφώστε κατάλληλα το διάγραμμα κλάσεων που είχατε κατασκευάσει για την 1η έκδοση της εφαρμογής σας. Εντοπίστε τα κοινά χαρακτηριστικά των κλάσεων Rectangle, Circle και Triangle και μεταφέρτε τα στην κλάση Shape.

Κατά τη διάρκεια της διάλεξης δώσαμε ένα διάγραμμα κλάσεων που καταγράφει τις σχέσεις αυτές και τις αναθέσεις αρμοδιοτήτων στα επιμέρους αντικείμενα που απαρτίζουν την εφαρμογή.

6.1.2 Δράση 9 – Συγγραφή πηγαίου κώδικα

Με βάση το νέο διάγραμμα κλάσεων διαμορφώστε τον πηγαίο κώδικα της εφαρμογής σας ώστε να αξιοποιεί τον μηχανισμό της κληρονομικότητας (extends). Όσον αφορά την υλοποίηση έχετε στην διάθεση όλες τις επιλογές που είδαμε στην έκδοση 1.

Για κάθε μία από τις εκδόσεις 1.2-1.4 μπορείτε να διαμορφώσετε την αντίστοιχη 2.1-2.4 η οποία θα αξιοποιεί την κληρονομικότητα.

Θα πρέπει να υλοποιήσετε μία τουλάχιστο από τις παρακάτω εκδόσεις.

6.1.3 Δράση 10 – Έκδοση 2.1 (java arrays)

Έκδοση με βάση πίνακες και κληρονομικότητα.

6.1.4 Δράση 11 – Έκδοση 2.1 (ArrayList)

Έκδοση με βάση ArrayList και κληρονομικότητα.

6.1.5 Δράση 12 – Έκδοση 2.1 (LinkedList)

Έκδοση με βάση LinkedList και κληρονομικότητα.

6.1.6 Δράση 13 – Έκδοση 2.1 (Vector)

Έκδοση με βάση Vector και κληρονομικότητα.