

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ

ΘΕΜΑ ΑΠΑΛΛΑΚΤΙΚΗΣ ΕΡΓΑΣΙΑΣ 2020-2021

Σύνολο μονάδων: 6

Διδάσκων: Ευθύμιος Αλέπης

Περιγραφή:

Η κάθε ομάδα καλείται να αναπτύξει 3 παραθυρικές εφαρμογές (windows form applications) σε γλώσσα προγραμματισμού C#. Οι εφαρμογές αναλύονται ως ακολούθως:

- **Εφαρμογή εκπαιδευτικού παιχνιδιού μνήμης.** Βασικό επίπεδο υλοποίησης: Στην εφαρμογή αυτή το ζητούμενο είναι να «ταιριάξει» ο χρήστης δύο ίδιες εικόνες, μέσα από ένα σύνολο από ζευγάρια από εικόνες, σε τυχαία σειρά. Οι εικόνες είναι «μπερδεμένες» αρχικά με τυχαίο τρόπο. Επιπλέον είναι όλες γυρισμένες ανάποδα και ο χρήστης δεν τις βλέπει. Επιλέγοντας μία πρώτη εικόνα, η εικόνα γυρίζει. Επιλέγοντας μια δεύτερη εικόνα, εάν είναι η ίδια με την πρώτη, τότε παραμένουν και οι 2 γυρισμένες, αλλιώς γυρίζουν πάλι ανάποδα. Το παιχνίδι ολοκληρώνεται όταν έχουν γυρίσει όλες οι εικόνες. Προχωρημένο επίπεδο υλοποίησης: Το παιχνίδι θα μετρά και χρόνο καθώς και το πόσες προσπάθειες έκανε ο κάθε χρήστης και θα τα εμφανίζει όλα κατά τη λήξη του παιχνιδιού. Το παιχνίδι θα διαθέτει settings, όπου ο χρήστης θα μπορεί να προσδιορίσει τον κατάλογο (directory) από τον οποίο θα φορτωθούν οι εικόνες, δίνοντας στο χρήστη τη δυνατότητα να βάλει δικές του εικόνες. Επιπλέον ο αριθμός των εικόνων θα είναι σταθερός. Το παιχνίδι θα αποθηκεύει όλες τις προσπάθειες, μαζί με κάποιο όνομα χρήστη για στατιστικούς λόγους, καθώς επίσης σε διαφορετική φόρμα θα εμφανίζει τις top 10 καλύτερες προσπάθειες των παικτών.



- **Εφαρμογή Παραθυρικής Ηλεκτρονικής Σκακιέρας.** Βασικό επίπεδο υλοποίησης: Θα δημιουργήσετε μια παραθυρική εφαρμογή η οποία θα λειτουργεί ως σκακιέρα στην οποία μπορούν να παίξουν 2 παίχτες. Η εφαρμογή σας θα διαθέτει τα 8x8, 64 ασπρόμαυρα τετράγωνα που υπάρχουν στο σκάκι, καθώς και όλα τα διαθέσιμα πιόνια. Ο κάθε παίχτης θα μπορεί να μετακινεί μέσα στην σκακιέρα τα πιόνια του. Δεν χρειάζεται να γίνεται κάποιος

έλεγχος από εσάς για τις μετακινήσεις, αυτό είναι ευθύνη των παικτών. Προχωρημένο επίπεδο υλοποίησης: Επιπλέον λειτουργίες: Ύπαρξη χρονομέτρου, τόσο για την ημερομηνία ώρα, όσο και για το πόσο χρόνο έχει ο κάθε παίχτης στη διάθεσή του. Με το που κάνει ένας παίχτης την κίνησή του, ο χρόνος του «παγώνει» και αντίστοιχα, αρχίζει η αντίστροφη μέτρηση του χρόνου του άλλου παίχτη. Η εφαρμογή σας θα πρέπει να διαθέτει και ΒΔ, στην οποία θα καταγράφονται τα παιχνίδια (ποιος παίχτης έχει παίξει με ποιον, καθώς και η χρονική στιγμή του παιχνιδιού).



- **Εφαρμογή Καταγραφής Κρουσμάτων COVID-19.** Βασικό επίπεδο υλοποίησης: Θα υλοποιήσετε μια παραθυρική εφαρμογή στην οποία ένας υπάλληλος του υπουργείου πολιτικής προστασίας θα μπορεί να καταγράφει ένα κρούσμα. Συγκεκριμένα, θα καταγράφει τουλάχιστον τα εξής στοιχεία:

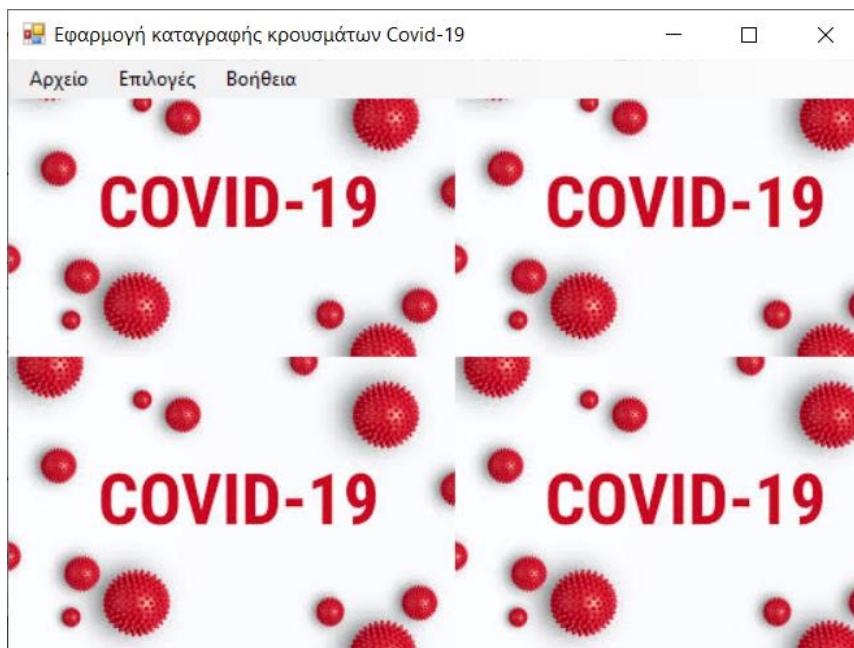
- Ονοματεπώνυμο
- E-mail
- Τηλέφωνο επικοινωνίας
- Φύλλο
- Ηλικία
- Υποκείμενα νοσήματα (αν υπάρχουν)
- Διεύθυνση κατοικίας
- Ημερομηνία/ώρα καταγραφής του κρούσματος

Η εφαρμογή σας θα πρέπει να υποστηρίζει τις εξής βασικές λειτουργίες χειρισμού των δεδομένων της:

- Εισαγωγή κρούσματος
- Προβολή όλων των κρουσμάτων

- Διαγραφή κρούσματος
- Τροποποίηση κρούσματος
- Αναζήτηση κρούσματος (με κριτήρια της επιλογής σας)

Η εφαρμογή σας θα πρέπει να διαθέτει και μενού (menustrip) για κάποιες από τις λειτουργίες της. Επιπλέον, θα πρέπει να βγάξετε κάποια βασικά στατιστικά, όπως πλήθος κρουσμάτων ανά ηλικία και άνδρες/γυναίκες. Προχωρημένο επίπεδο υλοποίησης: Μέρος της παραπάνω υλοποίησης (αν όχι όλη), π.χ. η προβολή των κρουσμάτων να γίνεται μέσω ιστοσελίδας ASP.NET.



**Οι παραπάνω εικόνες είναι ενδεικτικές, δεν σημαίνει ότι πρέπει να κάνετε κάτι αντίστοιχο με αυτό που βλέπετε*

Σημαντικά (ΔΙΑΒΑΣΤΕ ΠΡΟΣΕΚΤΙΚΑ ΤΑ ΠΑΡΑΚΑΤΩ):

- Για να εξεταστεί μια ομάδα θα πρέπει να έχει δηλωθεί στον διδάσκοντα και να έχουν καταγραφεί τα μέλη της. Μη δηλωμένες ομάδες, δεν θα μπορούν να εξεταστούν σε καμία εξεταστική περίοδο.
- Η εξέταση των εργασιών θα γίνει κανονικά κατά τη εξεταστική περίοδο στα εργαστήρια του τμήματος, σε ημερομηνία/ώρα που ορίζει το πρόγραμμα της εξεταστικής σας. Η παρούσα τελική εργασία παραμένει ίδια και για την εξεταστική του Σεπτέμβρη.
- Σε κάθε ομάδα είναι απαραίτητο **όλοι** οι φοιτητές να γνωρίζουν **όλες** τις λεπτομέρειες υλοποίησης και των 3 εφαρμογών. Κατά την εξέταση, εκτός από την επίδειξη των εργασιών στον υπολογιστή, **γίνεται και ταυτόχρονη προφορική** εξέταση, όπου οι εξεταστές ζητούν από κάθε φοιτητή ξεχωριστά να απαντήσει σε συγκεκριμένες ερωτήσεις, οι οποίες πολλές φορές αφορούν τον κώδικα των εφαρμογών, αλλά και εφ'όλης της ύλης γνώσεις προγραμματισμού σχετικά με τη C#. Λανθασμένες απαντήσεις αφαιρούν μονάδες από το συνολικό βαθμό των εργασιών, για τον/τους φοιτητή/φοιτητές που έκαναν τα λάθη.

- Η χρήση της C# ως γλώσσας προγραμματισμού είναι υποχρεωτική.
- Το εργαλείο ανάπτυξης των εφαρμογών είναι της επιλογής σας. Όμως συνίσταται η χρήση κάποιας έκδοσης Visual Studio, όσο πιο νέας και πληρέστερης γίνεται (υπενθυμίζεται ότι όλο το απαραίτητο software είναι διαθέσιμο στους φοιτητές).
- Για κάθε εφαρμογή έχουν οριστεί 2 επίπεδα υλοποίησης. Το 1^ο (βασικό), εφόσον υλοποιηθεί, δίνει τη δυνατότητα σε μια ομάδα να πιάσει τη βάση για την εκάστοτε εφαρμογή. Το 2^ο (προχωρημένο) δίδει στις ομάδες που το έχουν ολοκληρώσει το 100% του βαθμού.
- Για να βαθμολογηθεί μια εργασία με άριστα το 6 πρέπει να έχει υλοποιημένες και τις 3 εφαρμογές. Ωστόσο, μια ομάδα μπορεί να πάρει προβιβασμό βαθμό ακόμα και με 2 (καλά υλοποιημένες) από τις 3 εφαρμογές. Φυσικά συνυπολογίζονται και οι ατομικές εργασίες του κάθε φοιτητή.
- Εντός της κάθε ομάδας τα μέλη καλούνται να συνεργαστούν και να αλληλοβοηθηθούν. Ακόμα και στις περιπτώσεις που μεταξύ των ομάδων γίνει καταμερισμός των εργασιών (ποιος φοιτητής θα υλοποιήσει τι), θα πρέπει σε δεύτερο χρόνο να συναντηθείτε μεταξύ σας και να συζητήσετε τις λεπτομέρειες της υλοποίησης. **Δεν επιτρέπεται** η συνεργασία με άλλες ομάδες. Όμοιες εργασίες διαφορετικών ομάδων μηδενίζονται αυτόματα.
- **ΣΗΜΑΝΤΙΚΟ:** Κάθε εφαρμογή πρέπει να υλοποιηθεί με γνώμονα τον αντικειμενοστρεφή προγραμματισμό. Οι φοιτητές καλούνται να υλοποιήσουν όσο το δυνατόν περισσότερες τάξεις (όπου είναι αυτό εφικτό), οι οποίες να περιλαμβάνουν λειτουργίες και ιδιότητες. Σε κάθε εφαρμογή πρέπει να έχουν υλοποιηθεί **τουλάχιστον 2** τάξεις (και φυσικά και αντικείμενα αυτών) δημιουργημένες από τους φοιτητές –και όχι αυτόματα από το Visual Studio (π.χ. Form Class)–, ακόμα και αν αυτές χρησιμοποιούν ή καλούν τμήματα άλλων τάξεων και οι οποίες να χρησιμοποιούνται σε κάποιο σημείο του κώδικα. Η χρήση «χαρακτηριστικών» του αντικειμενοστρεφούς προγραμματισμού (π.χ. κληρονομικότητα μεταξύ τάξεων) στις δικές σας τάξεις δεν είναι υποχρεωτική, ωστόσο θα εκτιμηθεί ιδιαίτερα εφόσον υλοποιηθεί από εσάς.
- **Απαγορεύεται αυστηρά** η αντιγραφή κώδικα είτε από το διαδίκτυο είτε από συμφοιτητές σας. Ο κώδικας όλων των εφαρμογών θα περάσει από ειδικό λογισμικό εντοπισμού λογοκλοπής κώδικα και στις περιπτώσεις αντιγραφής, οι εν λόγω εργασίες θα μηδενιστούν.
- Τυχόν απορίες που προκύπτουν κατά την εκπόνηση των εργασιών μπορούν να συζητηθούν **ΜΟΝΟ** κατά τη διάρκεια των μαθημάτων ή/και στο γραφείο του διδάσκοντα και ΟΧΙ με την αποστολή e-mail στον διδάσκοντα!

Καλή επιτυχία!!