



Ημερομηνία Παράδοσης: 21-12-2016

Π. Βασιλειάδης

Η προγραμματιστική άσκηση για το μάθημα είναι **υποχρεωτική** και αφορά τη σχεδίαση, υλοποίηση και ρύθμιση ενός συστήματος λογισμικού. Η εργαστηριακή άσκηση προσφέρει **3 μονάδες** στον τελικό βαθμό του μαθήματος και εκπονείται σε ομάδες των **2 προσώπων**. Φυσικά, πρέπει να πιάσετε τουλάχιστον τη βάση στην εργασία, όπως και στο διαγώνισμα. Σε περιπτώσεις εξαιρετικών εργασιών, η επίδοση επιβραβεύεται με bonus στον τελικό βαθμό.

Το σύστημα πρέπει να υλοποιηθεί σε όλα τα επί μέρους στάδια.

Κατασκευάζετε μια εφαρμογή διαχείρισης φωτογραφικών άλμπουμ. Ένας χρήστης μπορεί να έχει πολλά άλμπουμ και να κατασκευάζει δυναμικά νέα άλμπουμ με βάση κάποιες αναζητήσεις. Λέμε ένα άλμπουμ «βασικό» όταν προέρχεται από χειροκίνητη δημιουργία από ένα χρήστη και «παράγωγο» αν είναι αποτέλεσμα αναζήτησης. Ένα φωτογραφικό άλμπουμ έχει ένα όνομα, και ένα σύνολο από «σελίδες». Οι σελίδες είναι πολλών ειδών. Υπάρχουν κατ' αρχήν οι άδειες σελίδες, που μπορούν να λειτουργούν διακοσμητικά (για παράδειγμα, ως κλείσιμο σε κάποιες «υποενότητες» ενός άλμπουμ). Υπάρχουν οι σελίδες με τίτλο μόνο που λειτουργούν ως αρχή της κάθε υποενότητας. Υπάρχουν σελίδες κειμένου, που έχουν (υποχρεωτικά) τίτλο, ημερομηνία, κάποιο κείμενο, και ένα σύνολο από λέξεις κλειδιά. Επίσης, αφού είναι άλμπουμ φωτογραφιών, υπάρχουν και σελίδες φωτογραφιών, που είναι σαν τις σελίδες κειμένου, μόνο που αντί για κείμενο, η σελίδα εμφανίζει φωτογραφικό υλικό (υπόμνηση: εσείς χρειάζεται να κρατάτε το path μιας υπαρκτής εικόνας).

Με όλα τα άλμπουμ, ο χρήστης πρέπει να μπορεί να **δει τα άλμπουμ σε μορφή Powerpoint**. Οδηγίες για το πώς θα γίνει αυτό, έπονται. Η λίστα των πράξεων που μπορεί να κάνει ο χρήστης με τα άλμπουμ είναι η εξής:

- Να δει τη λίστα από τα άλμπουμ του: με αίτηση του χρήστη παρατίθεται η λίστα των άλμπουμ στην κονσόλα, ή σε κάποια γραφική διεπαφή, όχι σε μορφή pptx)
- Να τα δει, με τα περιεχόμενά τους με τη σειρά που φτιάχτηκαν: ο χρήστης επιλέγει ένα συγκεκριμένο άλμπουμ με βάση το όνομά του, το ζητούμενο είναι να παρατεθούν από το σύστημα τα περιεχόμενα του άλμπουμ, με τη σειρά που αυτά προστέθηκαν στο άλμπουμ.
- Να τα δει, με τα περιεχόμενά τους ταξινομημένα χρονολογικά: αντίστοιχα με το παραπάνω, αλλά οι σελίδες του άλμπουμ πρέπει να ταξινομηθούν ως προς την ημερομηνία τους (αρχίζοντας από την παλαιότερη και καταλήγοντας στη νεότερη).

Με τα βασικά άλμπουμ, ο χρήστης πρέπει να μπορεί:

- Να προσθέσει ένα νέο, άδειο άλμπουμ (όπου απλά δημιουργείται ένα νέο άλμπουμ που προστίθεται στην λίστα των άλμπουμ).
- Να προσθέσει μια νέα σελίδα σε ένα άλμπουμ (όπου ο χρήστης προσδιορίζει για ποιο άλμπουμ ενδιαφέρεται, το είδος της σελίδας, και προσθέτει όλα τα στοιχεία που χρειάζεται η νέα σελίδα). Το σύστημα εκδίδει κωδικό λάθους αν δοθεί μη υπάρχον όνομα για το άλμπουμ ή δοθούν λάθος είδους στοιχεία.
- Να αλλάξει στοιχεία σε μια υπάρχουσα σελίδα: *καταγράψτε εσείς με ποια διαδικασία θα αλλάζουν τα περιεχόμενα μιας σελίδας.*

Με τα σύνθετα άλμπουμ, ο χρήστης πρέπει να μπορεί να φτιάξει ένα νέο σύνθετο άλμπουμ ως αποτέλεσμα μιας αναζήτησης με λέξη κλειδί (δλδ., ρωτά, φτιάχνεται ένα άλμπουμ και εντάσσεται στη λίστα με τα άλμπουμ).

Προφανώς, ο χρήστης θα πρέπει επίσης να μπορεί να αποχωρήσει ομαλά από το πρόγραμμα.

Όπως προειπώθηκε, ο στόχος είναι να μπορεί να εξαχθεί ένα άλμπουμ σε μορφή Powerpoint. Εσείς θα πρέπει (α) να διαχειριστείτε άλμπουμ και σελίδες σαν απλά αντικείμενα Java και (β) να κουμπώσετε την υλοποίησή σας σε ένα σύνολο μετατροπών που μετατρέπουν αντικείμενα java σε σελίδες μιας pptx παρουσίασης και παρουσιάσεις pptx. Για το σκοπό θα χρησιμοποιήσουμε τη βιβλιοθήκη Apache POI η οποία χρησιμοποιείται για να μπορεί κανείς να κατασκευάσει προγραμματιστικά αντικείμενα του MS Office. Σας δίδεται λοιπόν ένα πακέτο με έτοιμες, υλοποιημένες κλάσεις και διεπαφές, οι οποίες περιέχουν τον κώδικα της μετατροπής, και τις οποίες θα πρέπει να κουμπώσετε στον κώδικά σας για να μπορέσετε να επιτύχετε την μετατροπή από plain java objects σε pptx.

http://www.cs.uoi.gr/~pvassil/courses/sw_dev/exercises/PhotoAlbumWithPOI_public.zip

Είναι **ανελαστικό** να χρησιμοποιήσετε το πακέτο poiExtractor που σας δίδεται χωρίς απολύτως καμία αλλαγή! Πρέπει να επεκτείνετε την εφαρμογή με δικό σας επιπλέον κώδικα, χωρίς όμως να επέμβετε στο εν λόγω πακέτο. Αντιμετωπίστε το σαν συμβόλαιο: καλείστε να υλοποιήσετε κώδικα που θα χρησιμοποιήσει την έτοιμη «βιβλιοθήκη» που σας δίδεται, χωρίς ουσιαστικά να μπορείτε να την πειράξετε!

Οδηγίες

Το σύστημα πρέπει οπωσδήποτε να έχει packages, με ένα χωριστό package για την αλληλεπίδραση με το χρήστη το οποίο, το μόνο που επιτρέπεται να κάνει είναι: για κάθε use case, (α) να ζητά τα κατάλληλα στοιχεία (input) από το χρήστη, (β) με βάση αυτό το input να ζητά από το υπόλοιπο σύστημα να εκτελέσει την ουσία της use case, και (γ) να παρουσιάζει το αποτέλεσμα. Τα (β) και (γ) πρακτικά μεταφράζονται στην κλήση μιας μεθόδου και τη λήψη του αποτελέσματός της. Κανένα κομμάτι υπολογισμού, ελέγχου, ..., δεν πρέπει να εκτελείται από το υποσύστημα αυτό.

Εκτός από το να υλοποιήσετε το όλο σύστημα, θα χρειαστεί να σχεδιάσετε από πιο πριν:

- Όλα τα use cases αναλυτικά
- Όλα τα packages
- Όλες τις κλάσεις (σε επίπεδο (α) ανάλυσης στην αρχή και (β) σχεδίασης έπειτα)
- Ελέγχους: (α) και unit tests για κάθε κλάση χωριστά, με ελέγχους για τις βασικές μεθόδους, και (β) integration test όπου τα βασικά use cases της εφαρμογής θα ελεγχθούν συνολικά

Επιβάλλεται να ακολουθήσετε τις βασικές αρχές ενθυλάκωσης (υποχρεωτικά), χαμηλής σύζευξης, DIP, OCP, abstract coupling, factories κλπ (όσο αυτό είναι εφικτό και εύλογο).

Υπόδειγμα αναφοράς: για τα επιμέρους στάδια, μπορείτε να συμπληρώνετε / αναθεωρείτε σταδιακά την αναφορά σας. Για διευκόλυνσή σας, υπάρχει ένα υπόδειγμα στο http://www.cs.uoi.gr/~pvassil/courses/sw_dev/exercises/TemplateFinalReport.zip

Χρονοδιάγραμμα

Στη συνέχεια παρατίθενται στάδια της ανάπτυξης, ενδιάμεσες προθεσμίες (milestones) και καταληκτικές ημερομηνίες ολοκλήρωσης (deadlines).

[10/10]	Εκφώνηση	
4 weeks		Καταγραφή των Use Cases
[06/11:: 23.59]	Turnin: DLV1.1: pdf file with all the use cases	
3 weeks		design of classes & first implementations
[27/11:: 23.59]	Turnin: DLV2.1: pdf file with the class diagram (προαιρετικά: any other diagrams, too) DLV2.2: ascii file with a list of tests you will perform per class DLV2.3: pdf file with traceability matrix of tests	
3.5 weeks		Complete implementation [ΕΝΔΕΧΟΜΕΝΩΣ ΕΞΕΤΑΣΗ ΕΝΔΙΑΜΕΣΩΣ]
[21/12:: 23.59]	Turnin : ΑΝΕΛΑΣΤΙΚΑ DLV 3.1: tar file with the code for all classes DLV 3.2: FINAL pdf file with all the design and the documentation of the project	
[~09/01]	ΕΞΕΤΑΣΗ	Ακριβής ημερομηνία και λεπτομέρειες εξέτασης θα ανακοινωθούν αργότερα

Δεν θα ξεπεράσουμε το όριο των Χριστουγέννων, εκτός κι αν προκύψουν ανυπέρβλητες αναποδιές. Η πράξη έχει αποδείξει ότι στις γιορτές οι ομάδες αποσυντονίζονται σε πολύ μεγάλο βαθμό. Έτσι, το deadline της παράδοσης είναι ανελαστικό. Θα πιεστείτε περισσότερο πριν τις γιορτές, αλλά θα φύγετε για τις γιορτές χωρίς το φορτίο του project.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!