

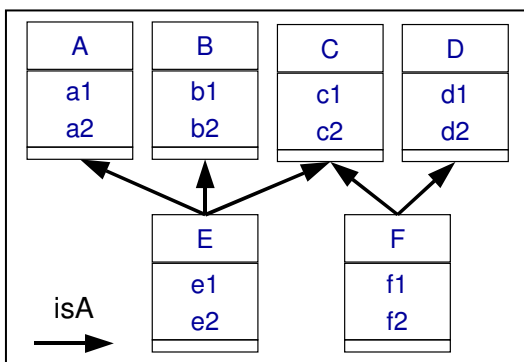
5^η Σειρά Ασκήσεων
 (Σχεδιασμός Κλάσεων και Μεθόδων - Object Constraint Language)
 Ημερομηνίες: 30/11-9/12

Άσκηση 1

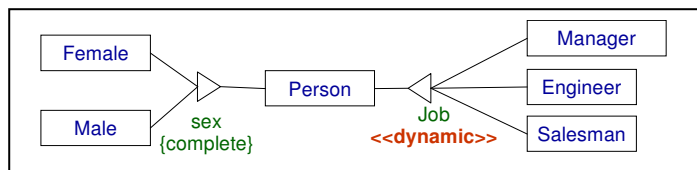
Μετασχηματίστε το διάγραμμα της Εικόνας 1 ώστε να μπορεί να παρασταθεί σε μια γλώσσα προγραμματισμού που δεν υποστηρίζει πολλαπλή γενίκευση. Δώστε το μετασχηματισμένο διάγραμμα για την περίπτωση που από τις κλάσεις του ανώτερου επιπέδου : (α) μόνο οι κλάσεις A και D δεν είναι αφηρημένες, και (β) μόνο οι κλάσεις A, B, C δεν είναι αφηρημένες.

Άσκηση 2

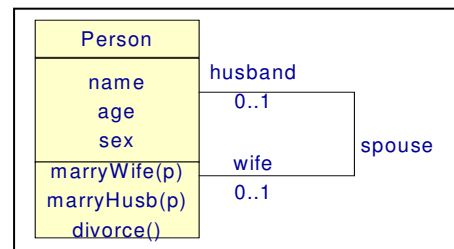
Μετασχηματίστε το διάγραμμα της Εικόνας 2 ώστε να μπορεί να παρασταθεί σε μια γλώσσα προγραμματισμού που δεν υποστηρίζει γενίκευση.



Εικόνα 1



Εικόνα 2



Εικόνα 3

Άσκηση 3

Έστω ότι τα διαγράμματα κλάσεων της UML δεν είχαν συμβολισμούς για την έκφραση περιορισμών αντιστοίχισης (multiplicity constraints) στις συσχετίσεις. Δείξτε πως θα μπορούσαμε να εκφράσουμε αυτούς τους περιορισμούς χρησιμοποιώντας την OCL.

Προσπαθήστε να δώσετε παραπάνω από έναν τρόπο έκφρασης για κάθε έναν από τους 4 τύπους αντιστοίχισης.

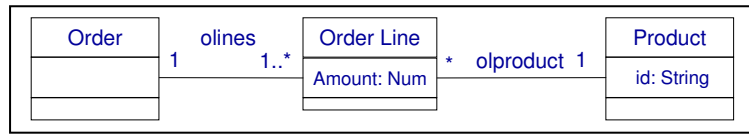
Άσκηση 4

Θεωρώντας το διάγραμμα της Εικόνας 3, εκφράστε σε OCL τους ακόλουθους περιορισμούς:

- (α) «A person cannot be married with himself (herself).»
- (β) «If x is the wife of y, then x is female.»
- (γ) «If x is the husband of y, then x is male.»
- (δ) «A person can be married only if its age is greater than 18. »
- (ε) Γράψτε τις προϋποθέσεις και τις «μετασυνθήκες» (pre/post-conditions) για τις marryWife(p:Person) και marryHusband (p:Person) και divorce(). Κάνετε την υπόθεση ότι δεν έχουμε πάντα εξ' αρχής πλήρη γνώση των στοιχείων του κάθε προσώπου. Για παράδειγμα, αν υπάρχουν δύο πρόσωπα p1 και p2 και δεν γνωρίζουμε το φύλο τους, τότε η p1.marryWife(p2) συν τοις άλλοις θα συμπληρώσει κατάλληλα τις πληροφορίες φύλλου. Υπάρχει όμως η περίπτωση να ξέρουμε το φύλο του ενός ή και των δύο. Σε αυτήν την άσκηση θα χρησιμοποιήσετε κάποιο CASE Tool που υποστηρίζει την OCL. Ένα τέτοιο εργαλείο π.χ είναι το ArgoUML, το οποίο προσφέρει έλεγχο σύνταξης και τύπων της OCL.

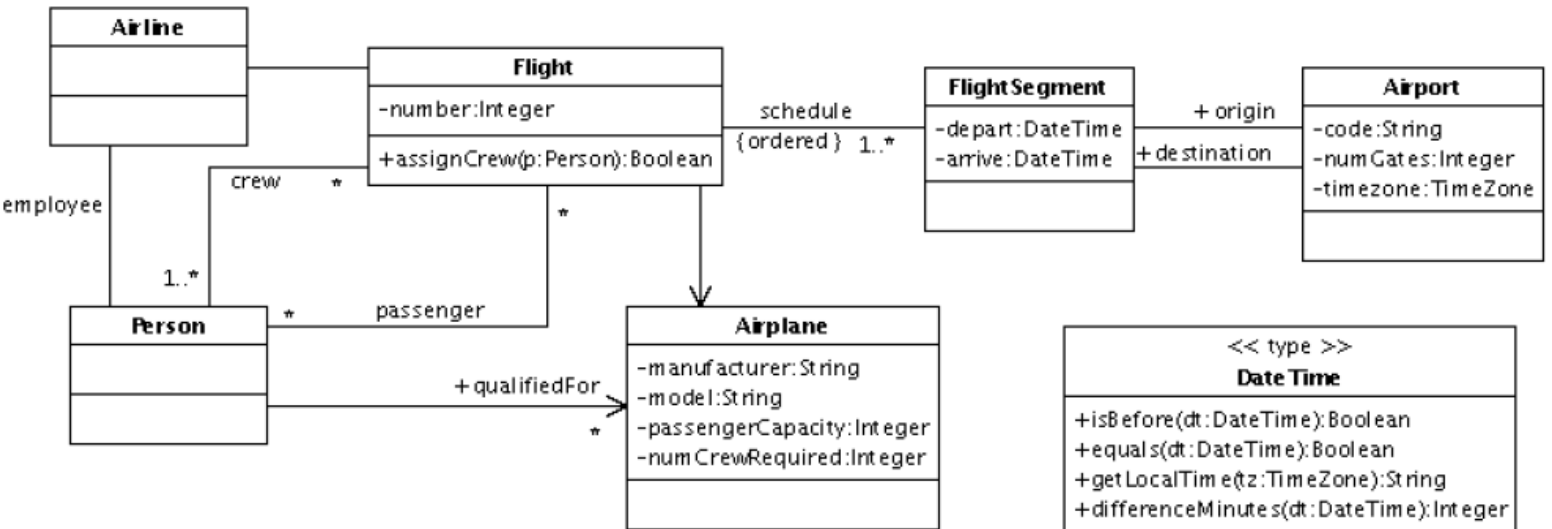
Άσκηση 5

Μια παραγγελία δεν πρέπει να έχει δύο ή περισσότερες παραγγελιογραμμές που να αναφέρουν το ίδιο προϊόν. Εκφράστε αυτόν τον περιορισμό σε OCL θεωρώντας το διάγραμμα της Εικόνας 4.



Εικόνα 4

Άσκηση 6



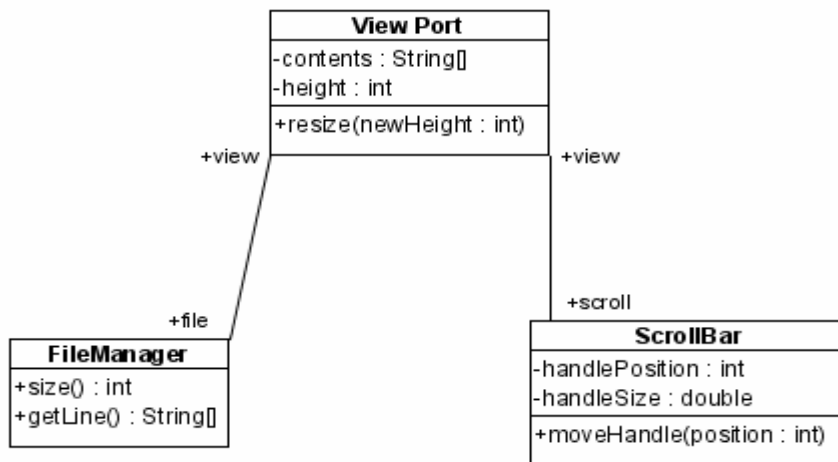
Εικόνα 5

Θεωρώντας το διάγραμμα κλάσεων της εικόνας 5 (για ένα σύστημα διαχείρισης αερογραμμών), εκφράστε τα ακόλουθα σε OCL:

1. Ο αριθμός των επιβατών σε μία πτήση δεν πρέπει ποτέ να υπερβαίνει την χωρητικότητα σε επιβάτες του αεροπλάνου.
2. Όλα τα μέλη του πληρώματος μιας πτήσης πρέπει να είναι εργαζόμενοι της εταιρείας στην οποία ανήκει η πτήση.
3. Το πλήθος του πληρώματος σε μία πτήση πρέπει να είναι ίσο με το ακέραιο γνώρισμα numCrewRequired για το αεροπλάνο.
4. Ένα τμήμα (FlightSegment) μιας πτήσης δεν μπορεί να φτάσει πριν αναχωρήσει.
5. Το πρόγραμμα μιας πτήσης πρέπει να επιτρέπει τουλάχιστον 30 λεπτά στάση σε κάθε αεροδρόμιο.
6. Τις προϋποθέσεις και τις μετασυνθήκες (pre/post conditions) της μεθόδου Flight::assignCrew.

Άσκηση 7 (BONUS)

Ο *Text Browser* είναι ένα εργαλείο για την εξερεύνηση ενός εγγράφου με κείμενο. Θεωρούμε πως ο *Text Browser* αποτελείται από τρία τμήματα, το *File Manager*, το *ViewPort* και το *ScrollBar*. Στην παρακάτω εικόνα έχουμε ένα UML class διάγραμμα που περιγράφει την εξωτερικά ορατή συμπεριφορά του *Text Browser*:



Ο File Manager είναι υπεύθυνος στο να παρέχει το κείμενο που θα δείξει ο ViewPort. Έχει δύο δημόσιες μεθόδους, την *getLine* και τη *size*. Θεωρούμε ότι το αρχείο αποτελείται μόνο από χαρακτήρες, χωρίζεται σε γραμμές και ότι μόνο ένα αρχείο μπορεί να εμφανιστεί τη φορά. Η *getLine* λαμβάνει ένα ακέραιο ως όρισμα και επιστρέφει τη γραμμή του αρχείου, με τους δείκτες να ξεκινάνε από τη γραμμή 0. Η *size* επιστρέφει ένα ακέραιο που δηλώνει τον αριθμό των γραμμών του αρχείου.

Το ViewPort είναι υπεύθυνο για την εμφάνιση των γραμμών του κειμένου. Έχει δύο γνώρισμα. Το πρώτο είναι ο ακέραιος *height* που δηλώνει τον αριθμό των γραμμών του κειμένου που αυτή τη στιγμή εμφανίζονται στο ViewPort. Το δεύτερο γνώρισμα είναι το *contents*, που είναι μια σειρά από γραμμές που αυτή τη στιγμή εμφανίζονται. Το ViewPort έχει και μία μέθοδο, τη *resize*. Η *resize* παίρνει ένα ακέραιο ως όρισμα και αλλάζει το μέγεθος του ViewPort έτσι ώστε να μπορεί να εμφανίσει τόσες γραμμές κειμένου.

Το ScrollBar είναι υπεύθυνο για τον έλεγχο των γραμμών του αρχείου που αυτή τη στιγμή εμφανίζονται. Το ScrollBar έχει δύο γνώρισμα και μία μέθοδο. Το πρώτο γνώρισμα ονομάζεται *handlePosition* και είναι ένας ακέραιος, οι τιμές του οποίου δείχνουν ποια γραμμή του αρχείου αυτή τη στιγμή εμφανίζεται στην κορυφή του ViewPort. Επομένως, οι τιμές του είναι μεταξύ του μηδενός και ενός λιγότερου του αριθμού γραμμών του αρχείου. Το δεύτερο γνώρισμα είναι το *handleSize* και είναι ένας πραγματικός αριθμός, μικρότερος ή ίσος του ένα, που δηλώνει το ποσοστό του αρχείου που αυτή τη στιγμή εμφανίζεται στο ViewPort. Η μέθοδος του ScrollBar ονομάζεται *moveHandle*. Λαμβάνει ένα μόνο ακέραιο όρισμα και μοντελοποιεί την κατάσταση στην οποία ο χρήστης έχει μετακινήσει το ScrollBar για να δει ένα διαφορετικό μέρος του αρχείου.

Για να μοντελοποιηθούν πλήρως οι εργασίες ενός πραγματικού Text Browser, πρέπει να ισχύουν οι εξής πέντε ιδιότητες :

1. Όταν αλλάζει το μέγεθος του ViewPort, η νέα τιμή του *height* είναι ίδια με τη τιμή του ορίσματος της μεθόδου *resize*.
2. Όταν το ScrollBar μετακινείται, η νέα τιμή του *handlePosition* είναι ίδια με τη τιμή του ορίσματος της μεθόδου *moveHandle*.
3. Οι γραμμές που εμφανίζονται στο ViewPort που όλες προέρχονται από το αρχείο κειμένου, και το ViewPort εμφανίζει όσες γραμμές από το αρχείο μπορεί.
4. Η τιμή του *handlePosition* στο ScrollBar αντιστοιχεί στον αριθμό της γραμμής που εμφανίζεται στην κορυφή του ViewPort.
5. Η τιμή του *handleSize* αντανακλά το ποσοστό των γραμμών από το αρχείο που εμφανίζονται στο ViewPort.

Εκφράστε όλους αυτούς τους περιορισμούς των μεθόδων και κλάσεων σε OCL.

Παραδοτέα

Γραπτή αναφορά σε ηλεκτρονική μορφή.