

Αντικειμενοστραφής προγραμματισμός II – Java

1η φάση ομαδικής εργασίας

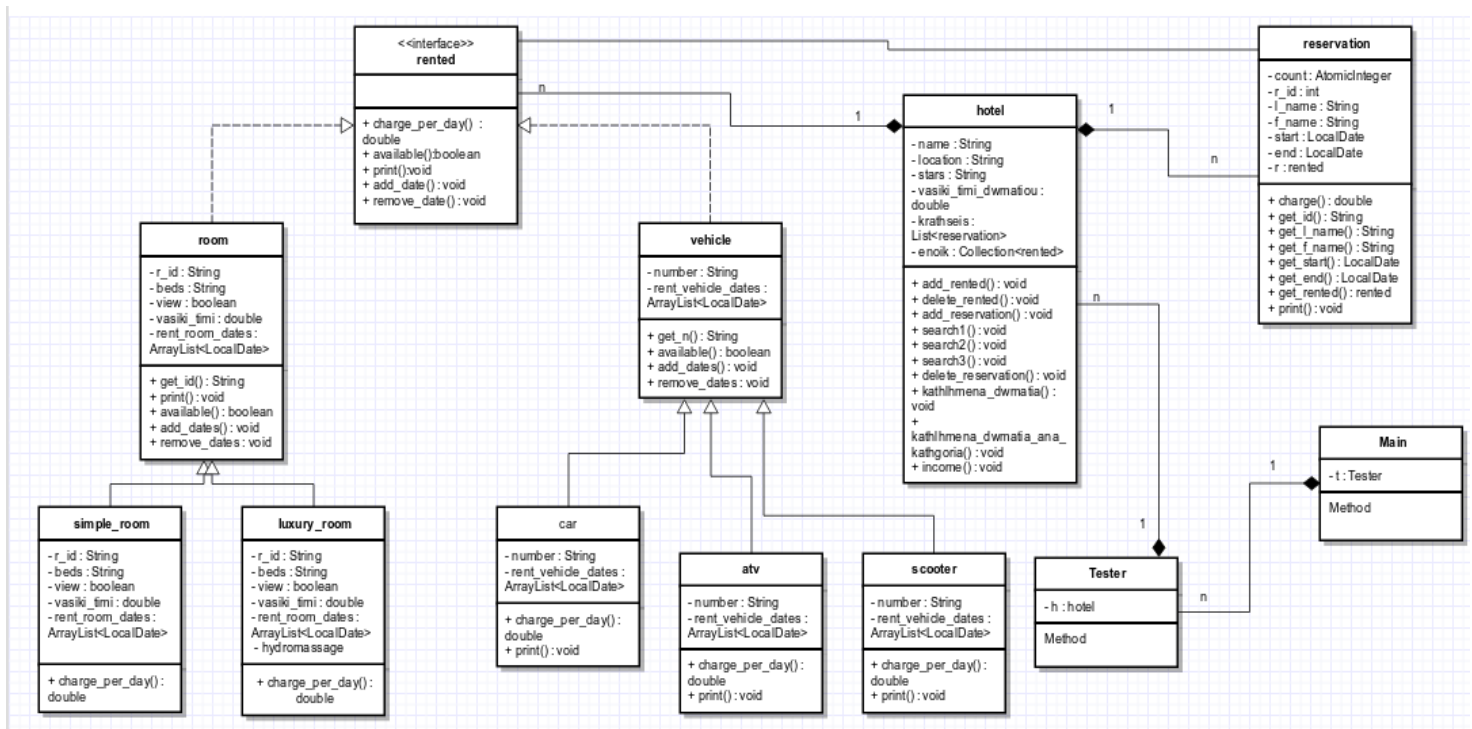


Ομάδα Εργασίας :

Μπινιάκου Θεοφάνης icsd13126

Αφαλωνιατης Στρατης icsd13018

Βασικό UML διάγραμμα



(επειδή το screenshot του διαγράμματος είναι μεγάλο και δεν φαίνεται καλά, υπάρχει στο zip της εργασίας και η εικόνα ξεχωριστά)

Τεκμηρίωση Σχεδιαστικών επιλογών

- Αρχικά, ο τύπος δεδομένων που χρησιμοποίησα για τις ημερομηνίες είναι `LocalDate`. Ο λόγος είναι ότι βρήκα συναρτήσεις στο `JAVA API` που βάση αυτών μπορούσα πολύ πιο εύκολα να επεξεργαστώ τις ημερομηνίες από ότι αν χρησιμοποιούσα απλά `Date`.
- Ήθελα να χρησιμοποιήσω exceptions αλλά αποφάσισα να τα προσθέσω κατευθείαν στην 2η φάση της εργασίας όπου θα προσθέσουμε το γραφικό περιβάλλον.
- Στις κλάσεις `room` και `vehicle` έχω προσθέσει από μία λίστα η οποία κρατάει τις ημερομηνίες για τις οποίες το αντίστοιχο ενοικιαζόμενο είναι κατειλημμένο.
- Την βασική τιμή ενός δωματίου, επειδή όπως λέει και η εκφώνηση, μπορεί να αλλάζει ανά σαιζόν, την προσθέτω μόλις δημιουργείτε το αντικείμενο `hotel` (το ξενοδοχείο δηλαδή). Την έχω δηλώσει σαν `protected` όμως και στην κλάση `room` για να μπορώ να την επεξεργαστώ κάθε φορά που θέλω να υπολογίζω το κόστος του κάθε δωματίου.

Γενικά, όπως φαίνεται και στο UML διάγραμμα, υπάρχει ένα Interface “rented” το οποίο αντιπροσωπεύει ένα ενοικιαζόμενο. Στην συνέχεια υπάρχουν 2 abstract κλάσεις οι οποίες

κάνουν implement το interface “rented”, οι room και vehicle. Το room αντιπροσωπεύει ένα δωμάτιο και το vehicle ένα όχημα. Στην συνέχεια, έχω δημιουργήσει για κάθε κατηγορία δωματίου ή οχήματος από μία κλάση. Οι κλάσεις που κάνουν extend την κλάση room είναι οι simple_room (ένα κοινό δωμάτιο) και luxury_room (ένα πολυτελές δωμάτιο) και οι κλάσεις που κάνουν extend την κλάση vehicle είναι οι car (ένα αυτοκίνητο), atv (μία γουρούνα), και scooter (ένα σκούτερ). Ακόμη υπάρχει η κλάση reservation η οποία αντιπροσωπεύει μία κράτηση. Στην συνέχεια, δημιούργησα και την κλάση hotel, η οποία είναι ένα ξενοδοχείο και μέσα στην οποία γίνονται όλες οι λειτουργίες ενός ξενοδοχείου. Επίσης υπάρχει η κλάση Tester η οποία υλοποιεί όλες τις λειτουργίες ενός ξενοδοχείου. Τέλος, υπάρχει και η κλάση Main η οποία απλά δημιουργεί ένα αντικείμενο τύπου Tester.

- Όσον αφορά την κλάση hotel και την λειτουργία της να προσθέτει ένα ενοικιαζόμενο (add_rented()), αρχικά προσθέτω μόνος μου κάποια ενοικιαζόμενα δημιουργώντας τα κατάλληλα αντικείμενα, και στην συνέχεια δίνω στον χρήστη να προσθέσει και αυτός όσα ενοικιαζόμενα θέλει δίνοντας τα απαραίτητα στοιχεία από το πληκτρολόγιο. Την ίδια ακριβώς λογική ακολουθώ και στην συνάρτηση delete_rented(), όπου αφαιρείτε κάποιο ενοικιαζόμενο. Αφαιρώ δηλαδή κάποια ενοικιαζόμενα μόνος μου δημιουργώντας κατάλληλα αντικείμενα, και στην συνέχεια δίνω στον χρήστη την επιλογή να αφαιρέσει και αυτός ενοικιαζόμενα δίνοντας τις απαραίτητες πληροφορίες από το πληκτρολόγιο.
- Ο τρόπος που λειτουργεί η add_rented() είναι ο εξής : Αρχικά ζητάω από τον χρήστη τις απαραίτητες πληροφορίες, για το τι ενοικιαζόμενο θέλει(π.χ. κοινό δωμάτιο, αυτοκίνητο) να προσθέσει και στην συνέχεια προσθέτει επιπλέον πληροφορίες ανάλογα τον τύπο ενοικιαζόμενου. Στο τέλος προσθέτω στο collection με τα ενοικιαζόμενα και αυτό που πρόσθεσε ο χρήστης.
- Ο τρόπος λειτουργίας της delete_rented() είναι παρόμοιος. Αρχικά, ο χρήστης δίνει από το πληκτρολόγιο τον μοναδικό κωδικό που χαρακτηρίζει κάθε ενοικιαζόμενο, και ψάχνω στην collection που είναι όλα τα ενοικιαζόμενα, το ενοικιαζόμενο που έδωσε ο χρήστης. Αν το βρω το βάζω σε μία λίστα και τέλος, αφαιρώ αυτή την λίστα από το collection με τα ενοικιαζόμενα (enoik.removeAll(pros_diagraphh)), όπου “enoik” είναι το collection με όλα τα ενοικιαζόμενα του ξενοδοχείου και “pros_diagraphh” η λίστα με τα ενοικιαζόμενα που είναι για διαγραφή.
- Η διαδικασία για την δημιουργία μίας κράτησης είναι η εξής : Αρχικά ζητάω από τον χρήστη να μου “πει” τι είδους ενοικιαζόμενο θέλει και ψάχνω στην συλλογή με τα ενοικιαζόμενα να το βρω. Όταν το βρω, ζητάω και τις ημερομηνίες που θέλει για την κράτηση (από – μέχρι) και κάνω τον έλεγχο αν για εκείνες τις ημερομηνίες είναι διαθέσιμο το ενοικιαζόμενο που ενδιαφέρεται ο χρήστης να κάνει κράτηση. Αν είναι διαθέσιμο, δημιουργώ το αντικείμενο τύπου reservation (γίνεται δηλαδή η κράτηση) και προσθέτω τις ημερομηνίες στην λίστα στην οποία υπάρχουν οι ημερομηνίες που το αντίστοιχο ενοικιαζόμενο είναι κατειλημμένο.
- Όσον αφορά την διαγραφή μιας κράτησης (delete_reservation), εμφανίζω στον χρήστη τους κωδικούς των κρατήσεων και αυτός από το πληκτρολόγιο επιλέγει ποια κράτηση θέλει να αφαιρέσει. Στην συνέχεια όμως, βρίσκω ποιο ενοικιαζόμενο αφορούσε η διαγράφουσα κράτηση και αφαιρώ τις ημερομηνίες της κράτησης από

την λίστα στην οποία υπάρχουν οι ημερομηνίες που το αντίστοιχο ενοικιαζόμενο είναι κατειλημμένο.

- Στις τρεις αναζητήσεις (με βάση τον κωδικό της κράτησης ή το όνομα του πελάτη ή μία συγκεκριμένη ημερομηνία), ζητάω από τον χρήστη να δώσει την αντίστοιχη πληροφορία και ψάχνω στην λίστα με τις κρατήσεις να βρω την αντίστοιχη κράτηση.
- Στην συνέχεια η συνάρτηση `kathlhmena_dwmatia()` ζητάει από τον χρήστη να του δώσει μία ημερομηνία και αυτή εμφανίζει όλα τα δωμάτια τα οποία είναι κατειλημμένα εκείνη την ημερομηνία.
- Η συνάρτηση `kathlhmena_dwmatia_ana_kathgoria()` κάνει ακριβώς την ίδια διαδικασία με την προηγούμενη συνάρτηση (`kathlhmena_dwmatia()`) με την διαφορά ότι εμφανίζει τα δωμάτια χωρισμένα στις δύο κατηγορίες : κοινό και πολυτελές δωμάτιο.
- Τέλος, η συνάρτηση `income()` υπολογίζει τα έσοδα του ξενοδοχείου για τον μήνα που επιλέγει ο χρήστης. Η διαδικασία που ακολουθεί είναι η αναζήτηση στην λίστα με τις κρατήσεις, για τις κρατήσεις που είναι καταχωρημένες για τον μήνα που ενδιαφέρει τον χρήστη. Όταν βρίσκει μία κράτηση, αν η κράτηση είναι ολόκληρη μέσα στον μήνα που έδωσε ο χρήστης, τότε καλεί την συνάρτηση (`charge()`) που υπάρχει στην κλάση `reservation` (κράτηση) η οποία υπολογίζει την χρέωση πελάτη για την κράτηση και την προσθέτει στα συνολικά έσοδα του ξενοδοχείου. Αν μία κράτηση όμως είναι χωρισμένη σε δύο μήνες (πχ αρχίζει 28 Ιουλίου και τελειώνει 5 Αυγούστου), ψάχνω να βρω μόνο τις μέρες που είναι μέσα στον μήνα που έχει δώσει ο χρήστης, και καλώ την συνάρτηση `charge_per_day()` η οποία για κάθε ενοικιαζόμενο επιστρέφει την χρέωση του πελάτη για το ενοικιαζόμενο για μια μόνο μέρα, και προσθέτω το σύνολο στα συνολικά έσοδα του ξενοδοχείου.

Αναμενόμενη Απόδοση Προγράμματος

Μετά από αρκετές δοκιμές που κάναμε, το πρόγραμμα λειτουργούσε κανονικά χωρίς προβλήματα.