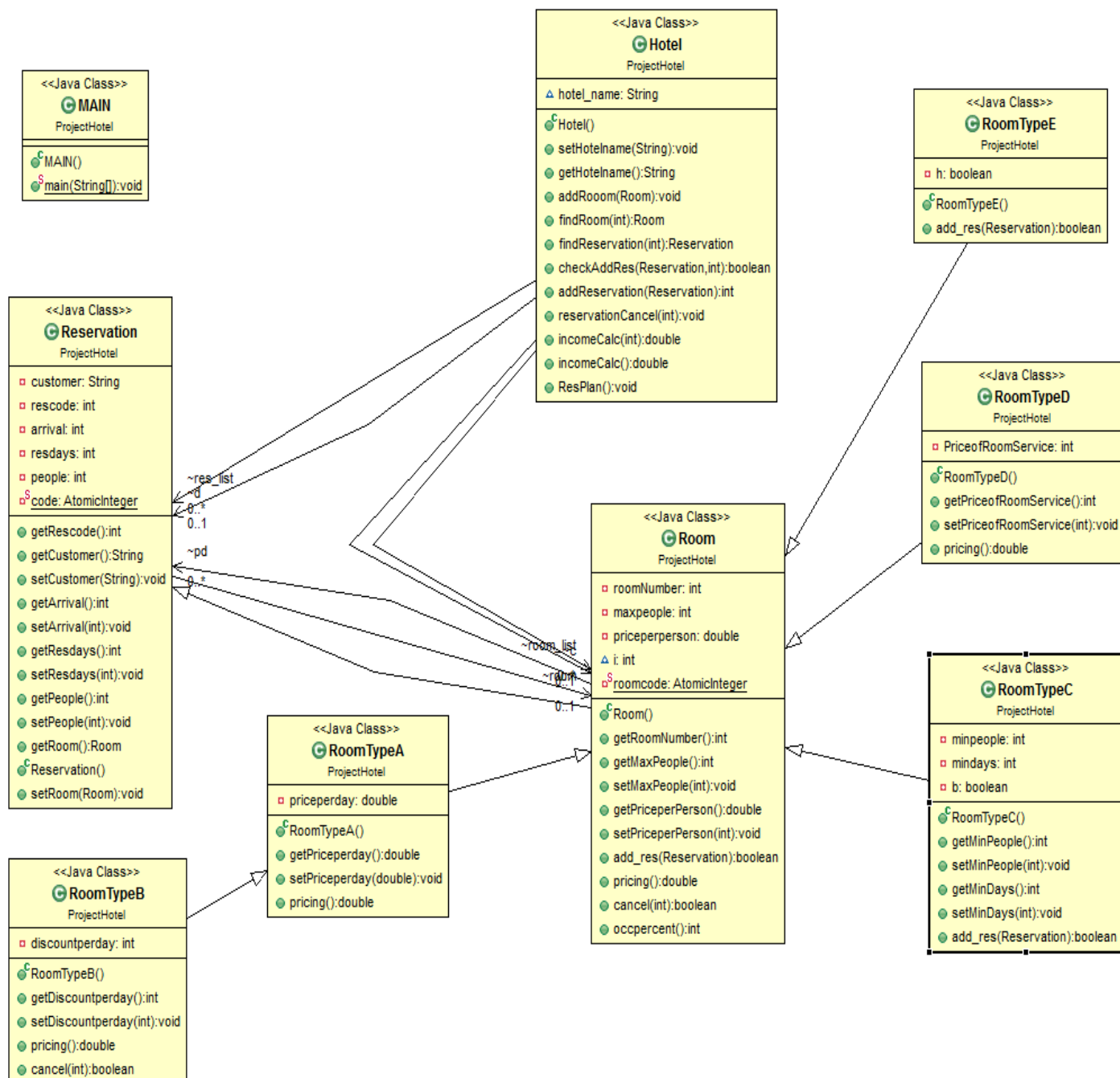


# Αναφορά του Project στην Java

Κάλλιστρος Ανδρέας - 1054351 – [up1054351@upnet.gr](mailto:up1054351@upnet.gr)

Τριανταφυλλόπουλος Παναγιώτης – 1054367 – [up1054367@upnet.gr](mailto:up1054367@upnet.gr)

/\* UML διάγραμμα \*/



## Υλοποίηση του προγράμματος

**Κλάση Κράτηση (Reservation)** : Διαθέτει όλα τα πεδία που ζητούνται ( όνομα πελάτη, κωδικό κράτησης, ημέρα άφιξης, μέρες διαμονής, άτομα και δωμάτιο), καθώς και μεθόδους **get** και **set** για το κάθε ένα από αυτά(εκτός του κωδικού κράτησης που δεν έχει μέθοδο **set**, γιατί φτιάχνεται αυτόματα από το πρόγραμμα). Για να εξασφαλίσουμε ότι κάθε αντικείμενο τύπου κράτηση θα έχει έναν διαφορετικό κωδικό, ο οποίος θα παράγεται αυτόματα, δηλώσαμε μία νέα μεταβλητή ως **AtomicInteger**(η οποία είναι **static**) και στον **constructor** της κλάσης αυξάνουμε την τιμή της κατά 1 καλώντας την μέθοδο **getAndIncrement()** και δίνουμε αυτήν την τιμή στον κωδικό της κράτησης. Αρχικά το πεδίο δωμάτιο γίνεται **null** μέσα στον **constructor**.

**Κλάση Δωμάτιο(Room)** : Διαθέτει αριθμό δωματίου, μέγιστο αριθμό ατόμων, τιμή ανά άτομο και πίνακα διαθεσιμότητας. Έχει επίσης μεθόδους **set** και **get** για τα πεδία αυτά, εκτός από τον κωδικό που έχει μόνο μέθοδο **get** γιατί ο αριθμός φτιάχνεται αυτόματα από το πρόγραμμα και τον πίνακα διαθεσιμότητας. Όπως και στην κλάση κράτηση, χρησιμοποιούμε μία μεταβλητή **AtomicInteger** (**static**) και στον **constructor** της κλάσης της αυξάνουμε κατά 1 την τιμή και την αναθέτουμε στον κωδικό του δωματίου. Στην μέθοδο προσθήκη κράτησης( **boolean add\_res(Reservation res)** ) με μία **for loop** τσεκάρουμε αν οι μέρες διαμονής είναι κενές ( αν είναι **null** στον πίνακα διαθεσιμότητας ) και αν τα άτομα δεν ξεπερνούν τον μέγιστο αριθμό ατόμων. Αν ισχύουν τα παραπάνω, τότε η κράτηση αυτή προστίθεται στον πίνακα διαθεσιμότητας στις μέρες διαμονής( από

ημέρα άφιξης μέχρι “ημέρα άφιξης + μέρες διαμονής” ) και το δωμάτιο της κράτησης του ορίσματος γίνεται το συγκεκριμένο δωμάτιο(αυτό για το οποίο καλείται η μέθοδος). Έπειτα έχουμε μέθοδο τιμολόγησης ( **double pricing()** ) η οποία υπολογίζει και επιστρέφει το συνολικό κέρδος από όλες τις κρατήσεις του δωματίου για το οποίο την καλούμε, έχουμε μέθοδο ακύρωση κράτησης ( **boolean cancel(int cd)** ) η οποία παίρνει ως όρισμα τον κωδικό μιας κράτησης και τσεκάρει αν αυτή υπάρχει στον πίνακα διαθεσιμότητας και αν υπάρχει την αφαιρεί από αυτόν(δηλαδή κάνει τις θέσεις που καταλάμβανε πριν null)και επιστρέφει true, ενώ αν δεν υπάρχει, επιστρέφει false. Τέλος έχουμε μέθοδο πληρότητας ( **int occpercent()** ) η οποία υπολογίζει και επιστρέφει το ποσοστό των ημερών του μήνα που είναι κλεισμένο το δωμάτιο.

**Κλάση Δωμάτιο Τύπου A (RoomTypeA)** :Κληρονομεί την κλάση Δωμάτιο. Έχει όλα τα πεδία που έχει και η κλάση δωμάτιο συν το πεδίο τιμή ανά μέρα και μεθόδους **set** και **get** για αυτό. Στην κλάση αυτή γίνεται υπερκάλυψη της μεθόδου **pricing()** η οποία τώρα υπολογίζει τα έσοδα διαφορετικά : διαπερνά κάθε στοιχείο του πίνακα διαθεσιμότητας και για κάθε μέρα που είναι κρατημένο το δωμάτιο προσθέτει στο συνολικό κέρδος την τιμή ανά μέρα.

**Κλάση Δωμάτιο Τύπου B (RoomTypeB)** :Κληρονομεί την κλάση Δωμάτιο Τύπου A. Έχει όλα τα πεδία που έχει και η κλάση δωμάτιο τύπου A συν το πεδίο έκπτωση ανά μέρα και μεθόδους **set** και **get** για αυτό. Γίνεται υπερκάλυψη της μεθόδου **pricing()** και αυτή τώρα υπολογίζει τα έσοδα του δωματίου διαφορετικά, για κάθε κράτηση, κάθε μέρα αφαιρεί από την αρχική τιμή την έκπτωση ανά μέρα και τελικά υπολογίζει το συνολικό κέρδος. Σε αυτήν την μέθοδο έχουμε προσθέσει ένα try-catch block διότι με τον κώδικα μας, στην επανάληψη που έχουμε φτιάξει μπορεί να πάμε έξω από τα στοιχεία του πίνακα διαθεσιμότητας ( **ArrayIndexOutOfBoundsException** ). Γίνεται επίσης υπερκάλυψη της μεθόδου **cancel(int cd)** η οποία επιστρέφει πάντα false γιατί αυτό το δωμάτιο δεν δέχεται ακυρώσεις κρατήσεων.

**Κλάση Δωμάτιο Τύπου C (RoomTypeC)** : Κληρονομεί την

κλάση Δωμάτιο, έχει όλα τα πεδία της συν τα πεδία ελάχιστες μέρες και ελάχιστα άτομα, καθώς και μεθόδους **set** και **get** για αυτά. Στην κλάση αυτή γίνεται υπερκάλυψη της μεθόδου **add\_res(Reservation res)** , η οποία πλέον ελέγχει καν αν τα άτομα και οι ημερες ξεπερνανε τις ελάχιστες τιμές του δωματίου.

**Κλάση Δωμάτιο Τύπου D (RoomTypeE)** : Κληρονομεί την κλάση Δωμάτιο, έχει όλα τα πεδία της συν το πεδίο τιμή του room service, καθώς και μεθόδους **set** και **get** για αυτό. Γίνεται υπερκάλυψη της μεθόδου **pricing()** και υπολογίζονται τα έσοδα διαφορετικά, καθώς αυτό το δωμάτιο έχει room service.

**Κλάση Δωμάτιο Τύπου D (RoomTypeE)** : Κληρονομεί την κλάση Δωμάτιο και έχει όλα τα πεδία της. Σε αυτή την κλάση γίνεται υπερκάλυψη της μεθόδου **add\_res(Reservation res)** η οποία τώρα τσεκάρει αν η ημέρα άφιξης είναι 15 του μηνός και αν είναι δεν επιτρέπει να γίνει η κράτηση.

**Κλάση Ξενοδοχείο(Hotel)** : Διαθέτει όνομα ξενοδοχείου, λίστα δωματίων και λίστα κρατήσεων. Έχει μεθόδους **set** και **get** για το όνομα. Έχει μία μέθοδο **addRoom(Room r)** η οποία προσθέτει το όρισμα που της δίνουμε στην λίστα δωματίων. Έχει μία μέθοδο **Room findRoom(int roomID)** η οποία παίρνει ως όρισμα έναν κωδικό δωματίου και τσεκάρει αν υπάρχει κάποιο δωμάτιο με αυτόν τον κωδικό, αν ναι το επιστρέφει , αν όχι επιστρέφει null. Υπάρχει επίσης μία μέθοδος **Reservation findReservation(int resID)** η οποία κάνει το ίδιο πράγμα για τις κρατήσεις με όρισμα έναν κωδικό κράτησης. Έχει μία μέθοδο προσθήκης κράτησης σε δωμάτιο ( **boolean checkAddRes ( Reservation res1 , int roomcode)** ) η οποία παίρνει ως όρισμα μία κράτηση και έναν κωδικό δωματίου και αν πληρούνται οι προϋποθέσεις προσθήκης της στο δωμάτιο με τον κωδικό αυτό, την προσθέτει και επιστρέφει true , αλλιώς επιστρέφει false ενώ σε κάθε περίπτωση εμφανίζει στο χρήστη κατάλληλο μήνυμα. Έχουμε μία μέθοδο προσθήκης κράτησης σε τυχαίο δωμάτιο ( **int addReservation(Reservation res2)** ) η οποία παίρνει ως όρισμα μία κράτηση και την προσθέτει σε όποιο δωμάτιο μπορεί να προστεθεί ( τσεκάροντας ένα-ένα από την αρχή προς το τέλος τα δωμάτια ) ,

επιστρέφει τον κωδικό του δωματίου που προστέθηκε αν έγινε η προσθήκη, ενώ αν δεν έγινε, επιστρέφει 0. Η κλάση αυτή έχει επίσης μέθοδο ακύρωσης κράτησης ( **void reservationCancel (int reservationID)** ) η οποία παίρνει ως όρισμα τον κωδικό της κράτησης και αν υπάρχει στη λίστα κρατήσεων, την ακυρώνει. Χρησιμοποιούμε σε αυτή την μέθοδο κάποια try-catch blocks για την αντιμετώπιση εξαιρέσεων. Έχει δύο μεθόδους υπολογισμού εσόδων ( **double incomeCalc(int roomNum1)** και **double incomeCalc()** ), η πρώτη παίρνει ως όρισμα έναν κωδικό δωματίου και επιστρέφει τα συνολικά έσοδα του, ενώ η δεύτερη επιστρέφει τα συνολικά έσοδα όλων των δωματίων. Τέλος έχουμε μία μέθοδο **ResPlan()** η οποία εμφανίζει το πλάνο κρατήσεων.

**Κλάση Main(MAIN)** : Αρχικά φτιάχνουμε ένα αντικείμενο ξενοδοχείου και του δίνουμε ένα όνομα. Ύστερα φτιάχνουμε 10 δωμάτια που να αντιπροσωπεύουν όλους τους τύπους και τα προσθέτουμε στη λίστα δωματίων του ξενοδοχείου. Μετά φτιάχνουμε δύο πίνακες τύπου string με 20 τυχαία ονόματα και επίθετα αντίστοιχα(θα χρειαστούν στην προσθήκη τυχαίας κράτησης). Φτιάξαμε ένα random number generator ( **Random rand = new Random()** ) και ξεκινήσαμε μία for loop η οποία δεν έχει συνθήκη για να τελειώσει αλλά τελειώνει όταν κάνουμε break. Μέσα στην επανάληψη, αρχικά με τη χρήση του rand και των μεθόδων set φτιάχνουμε μία κράτηση με τυχαία ημέρα άφιξης, τυχαίες μέρες διαμονής, τυχαίο αριθμό ατόμων και τυχαίο όνομα(από αυτά των πινάκων)(έχουμε φροντίσει το εύρος των τυχαίων τιμών να είναι επιτρεπτό) και την προσθέτουμε χρησιμοποιώντας την addReservation της Hotel. Μετά με χρήση του rand , με εύρος 4 αριθμούς(0-3), βάζουμε πως αν ο τυχαίος αριθμός είναι ο αριθμός 1 να γίνεται ακύρωση μίας τυχαίας κράτησης από τη λίστα(πάλι με χρήση του rand) και έτσι πετυχαίνουμε την 25% πιθανότητα ακύρωσης μίας κράτησης. Μετά εμφανίζεται στον χρήστη το μενού επιλογών και το πρόγραμμα του ζητά να πατήσει 0-7 για να πραγματοποιηθεί κάποια από αυτές(έχουμε χρησιμοποιήσει try-catch block μέσα σε μία while loop για να εξασφαλίσουμε ότι ο χρήστης θα δώσει σωστά δεδομένα και αν δώσει λανθασμένα, του ζητά καινούρια) κάνοντας Switch την μεταβλητή που δέχεται το πρόγραμμα από το πληκτρολόγιο

επιτυγχάνουμε να υλοποιήσουμε κάθε επιλογή. Σε περίπτωση που ο χρήστης δώσει 0(επόμενη επανάληψη) το πρόγραμμα ξαναδημιουργεί μία τυχαία κράτηση. Αν ο χρήστης δώσει 1(προσθήκη κράτησης) , τότε το πρόγραμμα του ζητά όνομα πελάτη, ημέρα άφιξης, μέρες διαμονής, άτομα , τον ρωτά αν επιθυμεί κάποιο συγκεκριμένο δωμάτιο και αν απαντήσει ναι, ζητά και τον κωδικό δωματίου. Δίνει σαφείς οδηγίες στα μηνύματα που εμφανίζονται. Έχουμε χρησιμοποιήσει try-catch blocks κάθε φορά που ζητά το πρόγραμμα κάτι από τον χρήστη, έτσι ώστε αν δεν δώσει επιτρεπτή τιμή να του την ξαναζητά. Αν ο χρήστης πατήσει 2 στο μενού επιλογών( Ακύρωση κράτησης), το πρόγραμμα του ζητά κωδικό κράτησης και αν αυτή υπάρχει, την ακυρώνει( με try-catch block εξασφαλίζουμε ότι ο χρήστης θα δώσει ακέραιο ). Αν ο χρήστης πατήσει 3(Προβολή κρατήσεων), τότε εμφανίζεται ένας πίνακας με τα εξής στοιχεία: Κωδικός κράτησης | Όνομα πελάτη | Κωδικός δωματίου. Αν ο χρήστης πατήσει 4(Προβολή Δωματίων), τότε εμφανίζεται ένας πίνακας με τα στοιχεία: Κωδικός Δωματίου | Πληρότητα | Έσοδα. Αν ο χρήστης πατήσει 5(Προβολή πλάνου κρατήσεων), τότε χρησιμοποιώντας την **ResPlan()** της Hotel , εμφανίζει το πλάνο κρατήσεων του ξενοδοχείου. Αν ο χρήστης πατήσει 6(Προβολή εσόδων), το πρόγραμμα ζητά αν θέλει συγκεκριμένο δωμάτιο, αν ναι ζητά και τον κωδικό και εμφανίζει τα συνολικά έσοδα του δωματίου και αν όχι εμφανίζει τα συνολικά έσοδα όλων των δωματίων. Αν ο χρήστης πατήσει 7(Τερματισμός), τότε κάνουμε break από την αρχική for loop και τερματίζει το πρόγραμμα.