ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

# ΟΝΤΟΚΕΝΤΡΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

2016-2017

## C++

ΔΙΔΑΣΚΩΝΤΕΣ : Ι. ΧΑΤΖΗΛΥΓΕΡΟΥΔΗΣ,Χ. ΜΑΚΡΗΣ,Μ. ΡΗΓΚΟΥ
ΕΤΟΣ : 1ο
ΕΞΑΜΗΝΟ : 2ο
ΠΑΡΑΔΟΣΗ :

# ΕΡΓΑΣΙΑ ΟΝΤΟΚΕΝΤΡΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ 2016-2017

# C++

💻 Τα μέλη της ομάδας :


| ΟΜΟΜΑΤΕΠΩΝΥΜΟ | ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ | EMAIL |
|---|---|---|
| Βασδάρης Όμηρος | 1054429 | vasdaris@ceid.upatras.gr |
| Καλαματιανού Δήμητρα | 1054406 | kalamatianou@ceid.upatras.gr |
| Κωστορρίζος Δημήτρης | 1054419 | kostorrizos@ceid.upatras.gr |

- Μια σύντομη περιγραφή του προγράμματος :

  - Η κλάση Booking έχει συναρτήσεις οι οποίες ορίζουν το όνομα πελάτη (set/getClientName),τον κωδικός κράτησης (set/getBookingID),την ημέρα της άφιξης (set/getArrivalDate),τις μέρες διαμονής (set/getAccomodationDays), το πλήθος των ατόμων (set/getPeopleNum) και το δωμάτιο (set/get Room). Συγκεκριμένα η συνάρτηση setRoom αναθέτει τον δείκτη τύπου Room στην μεταβλητή room της κλάσης Booking.
  - Η κλάση Room έχει συναρτήσεις οι οποίες ορίζουν τον κωδικό δωματίου (set/getRoomID), την μέγιστη χωρητικότητα (set/getMaxPeople) και την τιμή ανά άτομο (set/getPricePerPerson).Περιέχει ακόμα και τις συναρτήσεις που ζητήθηκαν (προσθήκης κράτησης : addBooking, Τιμολόγησης : getPriceOfRoom, Ακύρωσης : cancelBooking, Πληρότητας : checkFullness). Αυτή η κλάση έχει δύο μεταβλητές στις οποίες αξίζει να αναφερθούμε. Η roomInstanceCounter, που είναι static και μετράει το πόσα αντικείμενα δημιουργήθηκαν. Δεν μπορούμε όμως να την χρησιμοποιήσουμε ως μεταβλητή για το ID γιατί κάθε φορά που αλλάζει

τιμή θα αλλάζει για όλα τα αντικείμενα. Έτσι δημιουργούμε και άλλη μία μεταβλητή την roomID, που δεν είναι static και κρατά το ID για το δωμάτιο.

- Η κλάση RoomTypeA είναι υποκλάση της Room και υπερκαλύπτει την συνάρτηση getPriceOfRoom ώστε η τιμή του δωματίου να υπολογίζεται ανά μέρα διαμονής και όχι με βάση πόσα είναι τα άτομα. Για αυτό έχει και μία επιπλέον μεταβλητή η οποία είναι private την pricePerDay για την οποία έχουν κατασκευαστεί οι getPricePerDay και setPricePerDay για να μπορέσουμε να διαχειριστούμε την μεταβλητή.
- Η κλάση RoomTypeB είναι υποκλάση της RoomTypeA και υπερκαλύπτει την συνάρτηση getPriceOfRoom ώστε να λειτουργεί όπως στην κλάση RoomTypeA αλλά για κάθε επιπλέον μέρα διαμονής μιας κράτησης η τιμή της μέρας μειώνεται διαδοχικά με το ποσό discountPerDay (έκπτωση ανά μέρα). Η τιμή της μέρας μετά την έκπτωση δεν πρέπει να πέσει κάτω από το 50% της αρχικής τιμής ανά μέρα. Επίσης υπερκαλύπτει την συνάρτηση cancelBooking ώστε να μην μπορεί να γίνει η ακύρωση του δωματίου επιστρέφοντας μήνυμα "Can’ t cancel this kind of room".
- Η κλάση RoomTypeC είναι υποκλάση της Room και υπερκαλύπτει την συνάρτηση addBooking ώστε να συγκρίνει τα άτομα της κράτησης με τον

ελάχιστο αριθμό ατόμων τις μέρες διαμονής με τις ελάχιστες μέρες διαμονής.

- Η κλάση RoomTypeD και η κλάση RoomTypeE είναι δύο ακόμα δικές μας κλάσεις που κληρονομούν άμεσα ή έμμεσα την Room και υλοποιούν διαφορετικά κάποιες μεθόδους της. Συγκεκριμένα η RoomTypeD είναι υποκλάση της Room και υπερκαλύπτει την μέθοδο getPriceOfRoom και αφορά κρατήσεις μια μόνο μέρας διαμονής. Επιπρόσθετα γίνεται υπερκάλυψη της μεθόδου cancelBooking ώστε να μην μπορεί να γίνει η ακύρωση του δωματίου επιστρέφοντας μήνυμα "Can' t cancel this kind of room". Από την άλλη η κλάση RoomTypeE είναι υποκλάση της RoomTypeC και υπερκαλύπτει την μέθοδο getPriceOfRoom με την τιμή να μειώνεται αφού η τιμή ανά άτομο είναι πλέον η μίση για τις 3 πρώτες μέρες.
- Η κλάση Hotel έχει συνάρτηση η οποία ορίζει το όνομα του ξενοδοχείου (getHotelName) καθώς και τις συναρτήσεις που ζητήθηκαν (προσθήκη δωματίου : addHotelRoom, ανάκτηση δωματίου από κωδικό : recoverRoomOutOfID, ανάκτηση κράτησης από κωδικό : recoverBookingOutOfID, προσθήκη κράτησης σε δωμάτιο : addBookingToRoom, προσθήκη κράτησης : addBookingNoID , ακύρωση

κράτησης : deleteBooking, υπολογισμός εσόδων : income, πλάνο κρατήσεων : bookingPlan). Στην κλάση hotel έχουν δημιουργηθεί δύο vector ως μια δομή δεδομένων για να αποθηκευτούν τα δωμάτια και οι κρατήσεις. Με τον κατασκευαστή της κλάσης Hotel δημιουργούμε ένα αντικείμενο της κλάσης με όνομα που δίνεται ως όρισμα. Πιο συγκεκριμένα η συνάρτηση AddHotelRoom προσθέτει τα αντικείμενα τύπου δωμάτιο και της υποκλάσης τους. Η συνάρτηση recoverRoomOutOfID υπάρχει για την ανάκτηση δωματίου από το vector δίνοντας ως όρισμα το ID ενός δωματίου. Η recoverBookingOutOfID συνάρτηση ομοίως κάνει ανάκτηση δωματίου από το vector δίνοντας όρισμα το ID μιας κράτησης. Με την συνάρτηση addBookingToRoom ουσιαστικά δίνοντας ως όρισμα το ID από ένα δωμάτιο και μια κράτηση το πρόγραμμα ελέγχει άμα μπορεί να κάνει αυτήν την κράτηση. Είναι σημαντικό να τονίσουμε ότι αν ο χρήστης επιλέξει να μείνει πάνω από μια μέρα στο ξενοδοχείο και κάνει κράτηση δίνοντας το ID δωματίου που θέλει να μείνει τότε η κράτηση θα δημιουργηθεί ασχέτως του αν υπάρχει δωμάτιο τύπου D,το οποίο και είναι δωμάτιο για μια μέρα. Ομοίως αν ο χρήστης κάνει κράτηση επιλέγοντας να μείνει περισσότερες από τις ελάχιστες μέρες καθώς και τα άτομα είναι περισσότερα από το

ελάχιστα άτομα που υπάρχουν ως παράμετροι στα δωμάτια τύπου C και διαλέξει δωμάτιο διαφορετικού του τύπου από το C η κράτηση θα δημιουργηθεί ανεξαρτήτως του εάν υπάρχουν διαθέσιμα δωμάτια τύπου C.Η addBookingNoID συνάρτηση αντιστοιχίζει μια κράτηση σε ένα δωμάτιο αν ο χρήστης δεν επιλέξει συγκεκριμένο δωμάτιο. Η λογική με την οποία δουλεύει η συγκεκριμένη συνάρτηση είναι ότι σκανάρει τον πίνακα δωμάτιων του ξενοδοχείου και εάν βρει δωμάτιο στο οποίο μπορεί να γίνει κράτηση την προσθέτει σε αυτό. Για παράδειγμα αν έχουμε μια κράτηση για παραμονή στο ξενοδοχείο πάνω από 1 μέρα και καθώς σκαναρουμε τον πίνακα βρούμε δωμάτιο τύπου D τότε το κράτηση δεν θα γίνει .Η συνάρτηση deleteBooking διαγραφεί μια κράτηση από τις κρατήσεις του ξενοδοχείου. Η income(int ID) συνάρτηση εκτυπώνει τα έσοδα που έχει το ξενοδοχείο από το συγκεκριμένο δωμάτιο του οποίου το ID δόθηκε. Η income() εκτυπώνει τα έσοδα που έχει το ξενοδοχείο από όλα τα δωμάτια συνολικά. Τέλος η συνάρτηση bookingPlan εκτυπώνει τον πίνακα διαθεσιμότητας.

Η κλάση ConsoleApplication έχει την main συνάρτηση καθώς και την συνάρτηση setCancelationChance η οποία θέτει την πιθανότητα

ακύρωσης(πιθανοτηταΑκυρωσης = ((rand() % cancelationChance) == chance)) * 100%). Ο πίνακας names περιέχει strings με διάφορα ονόματα που θα χρησιμοποιηθούν για τυχαία ακύρωση. Μέσα στην do-while έχουμε την προσομοίωση. Κάθε loop της do-while είναι και μια προσομοίωση κρατήσεων. Στην συνέχεια με την switch case ο χρήστης επιλέγει τι θέλει να κάνει από τις παρακάτω επιλογές:

- Επόμενη Επανάληψη (το πρόγραμμα συνεχίζει την λειτουργία του)
- Προσθήκη Κράτησης (ο χρήστης δίνει: Όνομα, Άφιξη, μέρες, άτομα και προαιρετικά συγκεκριμένο κωδικό δωματίου)
- Ακύρωση Κράτησης (ο χρήστης δίνει τον κωδικό της κράτησης)
- Προβολή Κρατήσεων (εκτυπώνεται πίνακας με όλες τις κρατήσεις του Ξενοδοχείου): Κωδικός Κράτησης | Όνομα Πελάτη| Κωδικός Δωματίου
- Προβολή Δωματίων (εκτυπώνεται πίνακας με τα δωμάτια του Ξενοδοχείου) Κωδικός Δωματίου | Πληρότητα | Έσοδα - Προβολή Πλάνου Κρατήσεων (καλείται η αντίστοιχη μέθοδος του Ξενοδοχείου)
- Προβολή Εσόδων (ο χρήστης δίνει προαιρετικά συγκεκριμένο κωδικό δωματίου)
- Τερματισμός (το πρόγραμμα τερματίζει την λειτουργία του).

Στην αρχή του loop δημιουργούμε μια τυχαία κράτηση δίνοντας τυχαίες τιμές στις μεταβλητές του booking. Αμέσως μετά υπάρχει κώδικας για τυχαία ακύρωση κράτησης. Τέλος μετά την δημιουργία της τυχαίας κράτησης και της ακύρωσης τυχαίας κράτησης δίνεται η επιλογή στον χρήστη για ακύρωση κράτησης.

- Στο πρόγραμμα έχει υλοποιηθεί και υπερφόρτωση 3 τελεστών για απλοποίηση του κώδικα. Οι τελεστές που υπερφορτώθηκαν είναι “<<” , “<” και “>”.

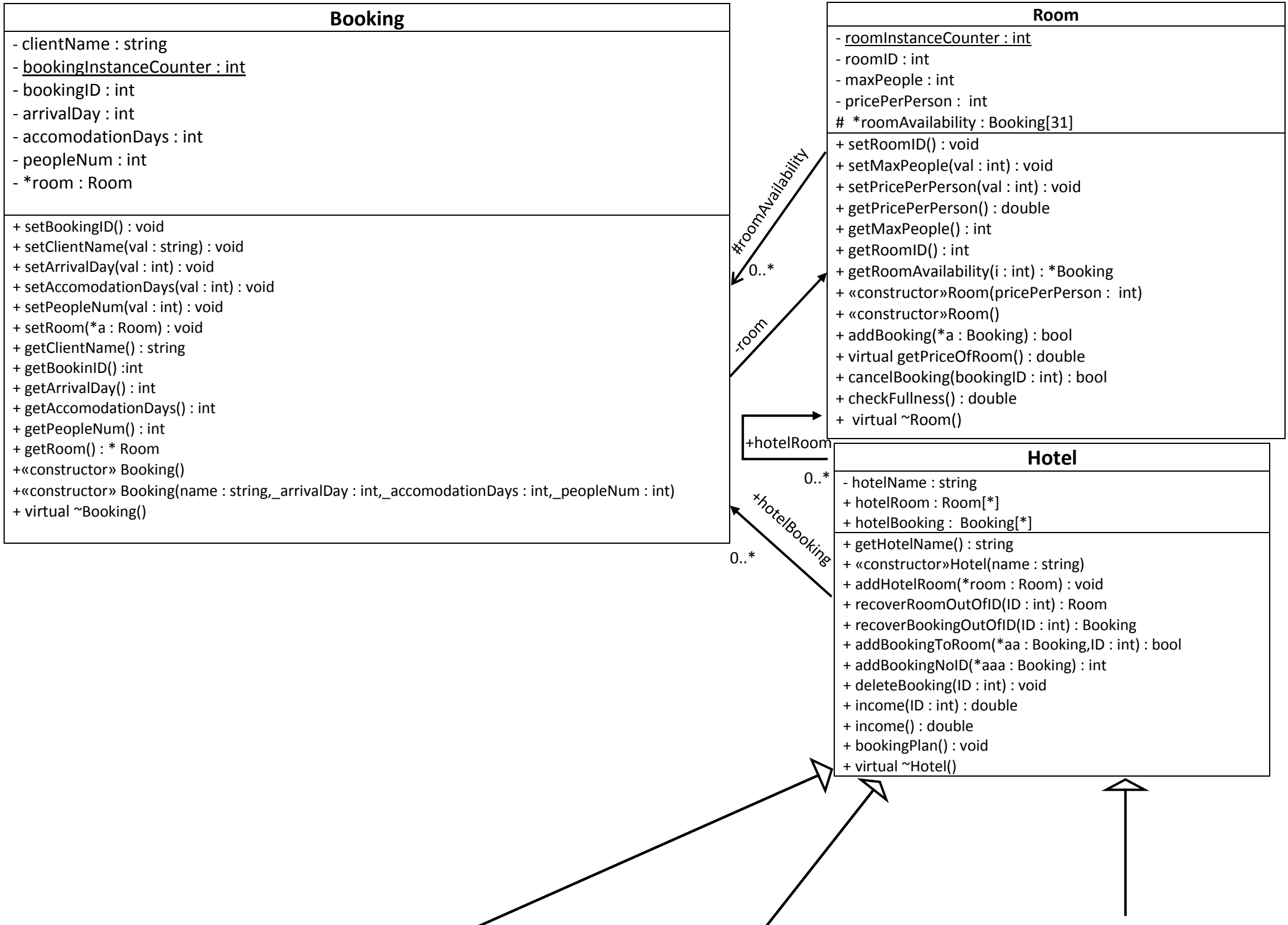
- Ο τελεστής “<<”, δέχεται ως όρισμα ένα αντικείμενο τύπου Booking, εκτυπώνει το όνομα πελάτη, τον κωδικό κράτησης, την ημέρα άφιξης και τις μέρες διαμονής. Τον έχουμε υπερφορτώσει για να βλέπουμε ποια είναι τα στοιχεία της τυχαίας κράτησης.

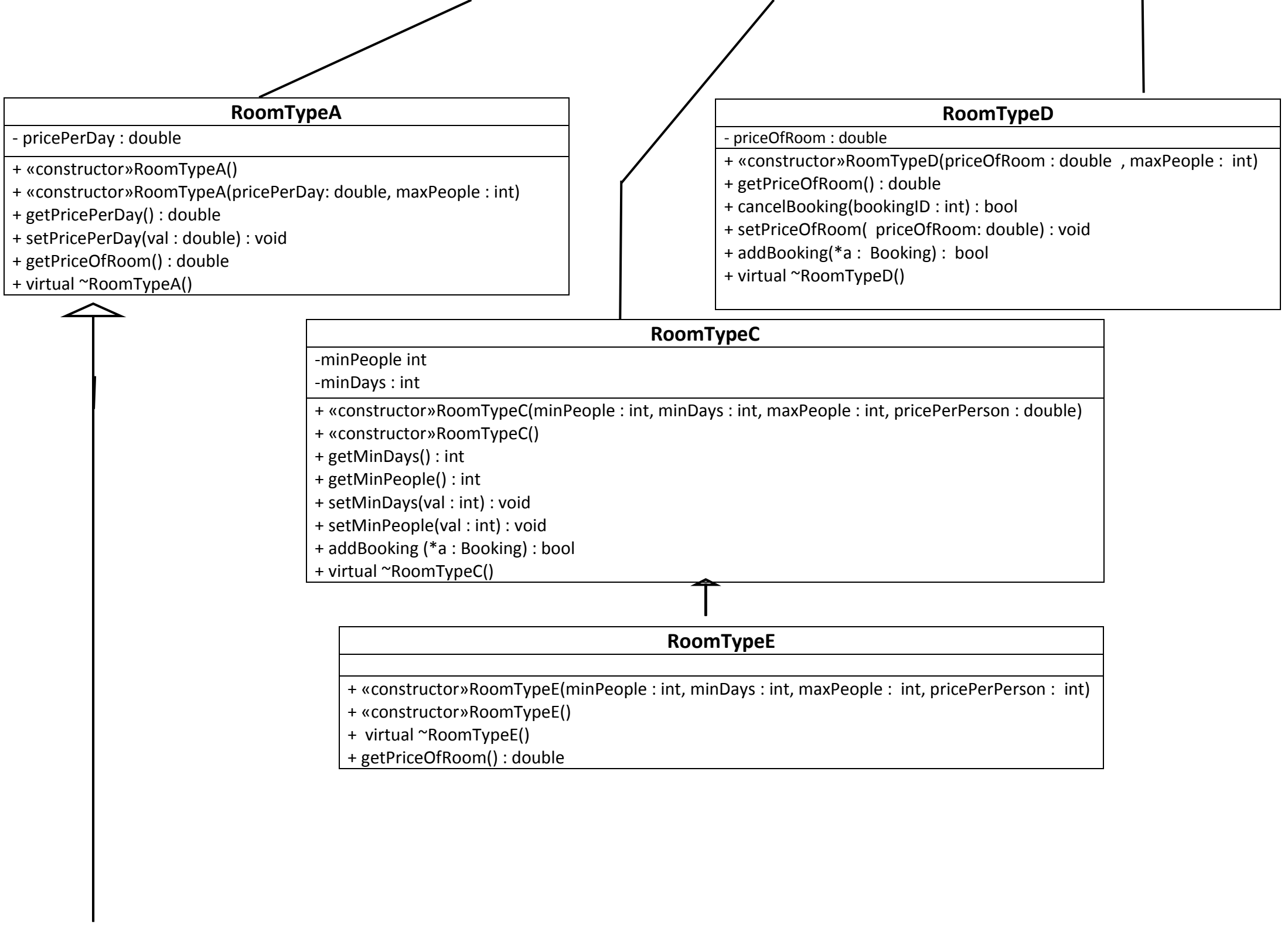
- Ο τελεστής “<”, δέχεται ως όρισμα ένα αντικείμενο τύπου Booking και έναν ακέραιο, επιστρέφει true αν η τιμή της μεταβλητής accomodationDays του

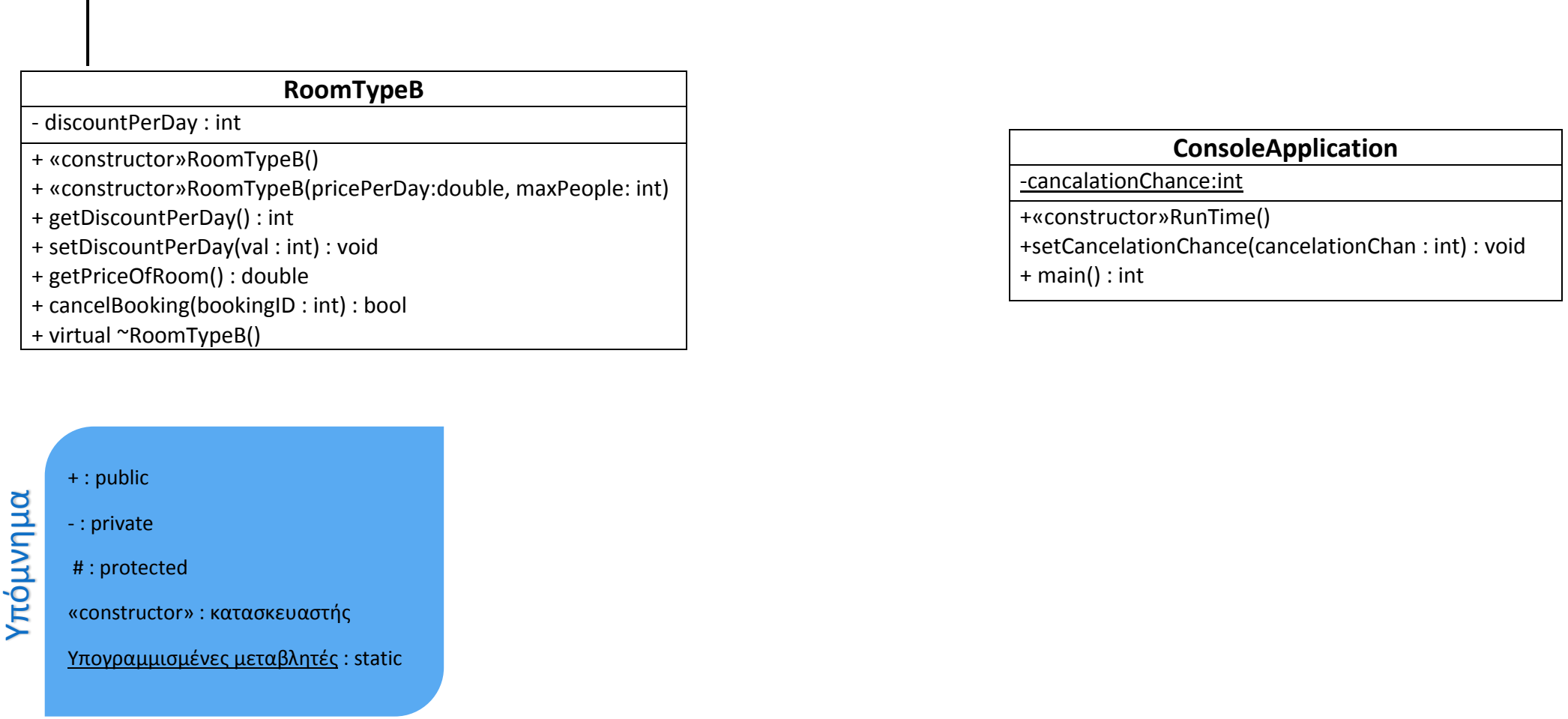
αντικειμένου τύπου Booking είναι μεγαλύτερη από 1. Τον έχουμε υπερφορτώσει για να ελέγχουμε εάν μπορεί να γίνει κράτηση σε ένα δωμάτιο τύπου D.

- Ο τελεστής “>”, δέχεται ως όρισμα ένα αντικείμενο τύπου Booking και ένα αντικειμενο τυπου RoomTypeC, επιστρέφει true αν η τιμή της μεταβλητής minPeople του αντικειμένου τύπου Booking είναι μεγαλύτερη από την τιμή της μεταβλητής minPeople του αντικειμένου τύπου RoomTypeC και αν η τιμή της μεταβλητής minDays του αντικειμένου τύπου Booking είναι μεγαλύτερη από την τιμή της μεταβλητής minDays του αντικειμένου τύπου RoomTypeC. Τον έχουμε υπερφορτώσει για να ελέγχουμε εάν μπορεί να γίνει κράτηση σε ένα δωμάτιο τύπου C.
- Παρακάτω παρατίθεται και το UML διάγραμμα κλάσεων. Κάθε πίνακας περιέχει το όνομα της κλάσης, τις μεταβλητές και τις μεθόδους της. Κάθε βέλος δείχνει την σχέση μεταξύ των κλάσεων και των υποκλάσεων τους. Επίσης παρατίθεται ένα υπόμνημα που εξηγεί βασικά σύμβολα του διαγράμματος :

ΝΑ ΔΩ ΠΩΣ ΔΗΛΩΝΩ VIRTUAL ΚΑΙ POINTER


Booking
- clientName : string
- bookingInstanceCounter : int
- bookingID : int
- arrivalDay : int
- accomodationDays : int
- peopleNum : int
- *room : Room
+ setBookingID() : void
+ setClientName(val : string) : void
+ setArrivalDay(val : int) : void
+ setAccomodationDays(val : int) : void
+ setPeopleNum(val : int) : void
+ setRoom(*a : Room) : void
+ getClientName() : string
+ getBookinID() :int
+ getArrivalDay() : int
+ getAccomodationDays() : int
+ getPeopleNum() : int
+ getRoom() : * Room
+«constructor» Booking()
+«constructor» Booking(name : string,_arrivalDay : int,_accomodationDays : int,_peopleNum : int)
+ virtual ~Booking()
Room
- roomInstanceCounter : int
- roomID : int
- maxPeople : int
- pricePerPerson : int
# *roomAvailability : Booking[31]
+ setRoomID() : void
+ setMaxPeople(val : int) : void
+ setPricePerPerson(val : int) : void
+ getPricePerPerson() : double
+ getMaxPeople() : int
+ getRoomID() : int
+ getRoomAvailability(i : int) : *Booking
+ «constructor»Room(pricePerPerson : int)
+ «constructor»Room()
+ addBooking(*a : Booking) : bool
+ virtual getPriceOfRoom() : double
+ cancelBooking(bookingID : int) : bool
+ checkFullness() : double
+ virtual ~Room()
#roomAvailability
0..*
-room
+hotelRoom
0..*
+hotelBooking
0..*
Hotel
- hotelName : string
+ hotelRoom : Room[*]
+ hotelBooking : Booking[*]
+ getHotelName() : string
+ «constructor»Hotel(name : string)
+ addHotelRoom(*room : Room) : void
+ recoverRoomOutOfID(ID : int) : Room
+ recoverBookingOutOfID(ID : int) : Booking
+ addBookingToRoom(*aa : Booking,ID : int) : bool
+ addBookingNoID(*aaa : Booking) : int
+ deleteBooking(ID : int) : void
+ income(ID : int) : double
+ income() : double
+ bookingPlan() : void
+ virtual ~Hotel()


RoomTypeA
- pricePerDay : double
+ «constructor»RoomTypeA()
+ «constructor»RoomTypeA(pricePerDay: double, maxPeople : int)
+ getPricePerDay() : double
+ setPricePerDay(val : double) : void
+ getPriceOfRoom() : double
+ virtual ~RoomTypeA()
RoomTypeD
- priceOfRoom : double
+ «constructor»RoomTypeD(priceOfRoom : double , maxPeople : int)
+ getPriceOfRoom() : double
+ cancelBooking(bookingID : int) : bool
+ setPriceOfRoom( priceOfRoom: double) : void
+ addBooking(*a : Booking) : bool
+ virtual ~RoomTypeD()
RoomTypeC
-minPeople int
-minDays : int
+ «constructor»RoomTypeC(minPeople : int, minDays : int, maxPeople : int, pricePerPerson : double)
+ «constructor»RoomTypeC()
+ getMinDays() : int
+ getMinPeople() : int
+ setMinDays(val : int) : void
+ setMinPeople(val : int) : void
+ addBooking (*a : Booking) : bool
+ virtual ~RoomTypeC()
RoomTypeE
+ «constructor»RoomTypeE(minPeople : int, minDays : int, maxPeople : int, pricePerPerson : int)
+ «constructor»RoomTypeE()
+ virtual ~RoomTypeE()
+ getPriceOfRoom() : double


RoomTypeB
- discountPerDay : int
+ «constructor»RoomTypeB()
+ «constructor»RoomTypeB(pricePerDay:double, maxPeople: int)
+ getDiscountPerDay() : int
+ setDiscountPerDay(val : int) : void
+ getPriceOfRoom() : double
+ cancelBooking(bookingID : int) : bool
+ virtual ~RoomTypeB()
ConsoleApplication
-cancalationChance:int
+«constructor»RunTime()
+setCancelationChance(cancelationChan : int) : void
+ main() : int
Υπόμνημα
+ : public
- : private
# : protected
«constructor» : κατασκευαστής
Υπογραμμισμένες μεταβλητές : static