1 Java RMI Hotel Reservation Service

Το RMI (Remote Method Invocation) είναι ένα πλήρως κατανεμημένο Interface της Java, το οποίο υποστηρίζει μόνο Java to Java επικοινωνία και χρησιμοποιείται για την εύκολη πρόσβαση σε αντικείμενα που βρίσκονται σε απομακρυσμένα συστήματα. Τα αντικείμενα αυτά, χειρίζονται όπως και τα τοπικά αντικείμενα της εφαρμογής μας.

Το κομμάτι για την ανταλλαγή δεδομένων μεταξύ client και server είναι έτοιμο, δεν χρειάζεται να το υλοποιήσουμε ή να επέμβουμε εμείς, μόνο να ακολουθήσουμε κάποια βήματα και κανόνες. Για παράδειγμα, όλες οι κλάσεις που παράγουν αντικείμενα τα οποία μπορούν να καλέσουν και οι clients πρέπει να κάνουν implement την κλάση "Serializable". Η επικοινωνία αυτή δεν γίνεται απευθείας, αλλά παρεμβάλλεται το RMI Registry στο οποίο ο server καταχωρεί τις υπηρεσίες/μεθόδους που προσφέρει, με ένα όνομα (π.χ HotelService), και οι clients αναζητούν με βάση το όνομα/ip του server, το port και το service που θέλουν, πληροφορίες για το που θα βρουν τις υπηρεσίες αυτές.

1.1 Hotel_Server.java

Η κλάση Hotel_Server χρησιμοποιείται για την έναρξη του registry, την αρχικοποίηση του interface (hInterface) με το αντικείμενο που υλοποιεί τις μεθόδους του (Hotel_Implementation) και την καταχώρηση του στο registry.

1.2 Hotel_Interface.java

Το interface πρέπει να είναι γνωστό και στους δυο (client & server) για να μπορέσουν να αλληλεπιδράσουν μεταξύ τους. Για να χρησιμοποιήσουμε ένα απομακρυσμένο αντικείμενο πρέπει να ξέρουμε τις μεθόδους που υλοποιεί, χωρίς να χρειάζεται να ξέρουμε πώς τις υλοποιεί. Αυτό τον σκοπό εξυπηρετεί το interface που καταχώρησε ο server στο registry.

1.3 Hotel_Implementation.java

Η αναφερόμενη κλάση υλοποιεί τις μεθόδους που υπάρχουν στο Hotel_Interface.java και αποτελεί το κύριο στοιχείο της λειτουργίας του server. Για την υποστήριξη των απαιτούμενων λειτουργιών δημιουργήθηκαν δύο ακόμα κλάσεις, η κλάση Room και η κλάση Reservation. Αυτές οι κλάσεις παριστούν και χρησιμοποιούνται για την υλοποίηση των οντοτήτων "δωμάτιο" και "κράτηση" αντίστοιχα.

Υλοποιημένες μέθοδοι:

list()

Δεν έχει παραμέτρους και πιστρέφει στον client την λίστα με τα διαθέσιμα δωμάτια.

book(String roomType, int reservedRooms, String reservationName)

Η book πραγματοποιεί την κράτηση του client. Δηλαδή, κάθε φορά που καλείται παίρνει την παράμετρο roomType και ψάχνει στην λίστα των δωματίων για αντίστοιχο τύπο δωματίου, αν το βρει αφαιρεί από τον αριθμό των διαθέσιμων δωματίων την παράμετρο reservedRooms και ελέγχει αν το αποτέλεσμα είναι μεγαλύτερο ή ίσο του 0 και ταυτόχρονα αν είναι διάφορο του αριθμού των διαθέσιμων δωματίων. Αν αυτά ισχύουν τότε αποθηκεύει την κράτηση στην λίστα των κρατήσεων, ανανεώνει τον αριθμό των διαθέσιμων δωματίων, βρίσκει το συνολικό κόστος και επιστρέφει μήνυμα στον client πως η κράτηση πραγματοποιήθηκε. Σε διαφορετική περίπτωση στέλνει ανάλογο μήνυμα.

guests()

Η μέθοδος guests δεν έχει παραμέτρους. Όταν κληθεί ψάχνει την λίστα των κρατήσεων και για κάθε όνομα μέσα σε αυτή δημιουργεί μια καινούργια λίστα κρατήσεων στην οποία αποθηκεύει όλες τις κρατήσεις που έχουν γίνει στο συγκεκριμένο όνομα και τελικά επιστρέφει στον client ένα Map που κρατάει/αντιστοιχεί ονόματα πελατών με τις κρατήσεις που έχει πραγματοποιήσει ο κάθε ένας.

cancel(String roomType, int canceledRooms, String reservationName)

Η cancel υλοποιεί την ακύρωση μιας κράτησης. Αρχικά ελέγχει αν υπάρχει κράτηση στο όνομα ”reservationName” και αν δεν υπάρχει επιστρέφει κατάλληλο μήνυμα, αλλιώς συνεχίζει. Κοιτάει αν μέσα στην λίστα με τις κρατήσεις υπάρχει κράτηση με αριθμό δωματίων ίσο με ”canceledRooms” και τύπο δωματίου ίδιο με ”roomType”, αν υπάρχει τότε διαγράφει την κράτηση από την λίστα και θέτει την μεταβλητή ”ReservationFound” ίση με 1. Αυτό σημαίνει ότι για το υπόλοιπο της επανάληψης δεν θα πραγματοποιηθεί άλλη διαγραφή, αλλά στο string ”msg” θα αποθηκευτούν όλες οι υπόλοιπες κρατήσεις του client και θα επιστραφούν σε αυτόν για ενημέρωση.

> [Η μεταβλητή ”ReservationFound” λέει στο πρόγραμμα αν βρέθηκε κάποια κράτηση για ακύρωση. Όταν η αυτή είναι ”0” τότε επιστρέφεται μήνυμα στον client ότι δεν βρέθηκε κάποια εγγραφή και σταματάει η διαγραφή κρατήσεων από την λίστα.]

Στη συνέχεια ελέγχει αν η ”ReservationFound” είναι 0, αν δεν είναι τότε ψάχνει στην λίστα των δωματίων, βρίσκει το δωμάτιο με τύπο ”roomType” και ενημερώνει την μεταβλητή η οποία είναι υπεύθυνη για την διαθεσιμότητα των δωματίων αυτού του τύπου. Επίσης, ψάχνει στο Map ”waitingLists” με βάση το όνομα ”reservationName” και καλεί με το αντίστοιχο callback αντικείμενο την μέθοδο ”notify” έτσι ώστε να ενημερωθούν οι clients που ενδιαφέρονται γι αυτού του τύπου τα δωμάτια.

getRoomPrice(String roomType)

Η μέθοδος αυτή ψάχνει στην λίστα των δωματίων τον τύπο δωματίου roomType και επιστρέφει το κόστος του.

addToWaitingList(String roomType, CallbackClientInterface clientInterface)

Η μέθοδος addToWaitingList προσθέτει στην λίστα αναμονής, για τα δωμάτια τύπου roomType, το αντικείμενο clientInterface. Το clientInterface είναι ένα αντικείμενο που λειτουργεί σαν αναγνωριστικό για τον συγκεκριμένο client. Επομένως όταν το αντικείμενο αυτό καλέσει την μέθοδο του, τα αποτελέσματα θα εμφανιστούν σε αυτόν.

1.4 Hotel_Client.java

Ξεκινώντας, ο client ελέγχει αν του έχει δοθεί σωστός αριθμός ορισμάτων, αν ναι, κοιτάει στο RMI registry αν υπάρχει το service/interface που ψάχνει, αν δεν υπάρχει τότε "πετάει" exception, αλλιώς συνεχίζει την εκτέλεση του και "κοιτάει" το πρώτο όρισμα αν είναι κάποιο από τα αναμενόμενα, αν είναι ελέγχει αν ο αριθμός των υπόλοιπων ορισμάτων που συνοδεύουν την κύρια εντολή (πρώτο όρισμα) είναι σωστός, διαφορετικά τυπώνει μήνυμα λάθους και τερματίζει. Στην συνέχεια υλοποιεί τέσσερις λειτουργίες οι οποίες αναλύονται παρακάτω.

list()

Η list καλεί, μέσω του hInterface την μέθοδο list του server, παίρνοντας πίσω μια λίστα με τα διαθέσιμα δωμάτια του ξενοδοχείου την οποία στην συνέχεια τυπώνει.

book()

Η book καλεί, μέσω του hInterface την μέθοδο book του server, παίρνοντας πίσω ένα string το οποίο τυπώνει και έπειτα ελέγχει για λέξεις κλειδιά. Ανάλογα με αυτές εκτελεί τα κατάλληλα τμήματα κώδικα.

Λέξεις κλειδιά:

- **waiting**: Αν υπάρχει αυτή η λέξη μέσα στο επιστρεφόμενο string, σημαίνει ότι ο server ρωτάει αν ο χρήστης θέλει να εγγραφεί σε κάποια από τις λίστες αναμονής του ξενοδοχείου, έτσι ώστε αν ελευθερωθεί κάποιο δωμάτιο να ενημερωθεί αμέσως. Αν ο χρήστης ανταποκριθεί θετικά, τότε δημιουργείται ένα αντικείμενο τύπου CallbackClientInterface και καλείται η μέθοδος, του hInterface, addToWaitingList που με την σειρά της αποθηκεύει το CallbackClientInterface αντικείμενο στην λίστα αναμονής.
- **reserve**: Αν υπάρχει αυτή η λέξη μέσα στο επιστρεφόμενο string, σημαίνει ότι ο server δεν μπόρεσε να πραγματοποιήσει την κράτηση του πελάτη λόγω έλλειψης δωματίων και ρωτάει αν ο χρήστης θέλει να δεσμεύσει τα διαθέσιμα δωμάτια που βρέθηκαν. Αν απαντήσει ναι, τότε πραγματοποιείται κράτηση γι αυτά τα δωμάτια.

guests()

Η guests καλεί, μέσω του hInterface την μέθοδο guests του server, παίρνοντας πίσω ένα Map. Αν το Map είναι άδειο τυπώνει κατάλληλο μήνυμα, αλλιώς για κάθε όνομα τυπώνει τις κρατήσεις που του αντιστοιχούν και με την βοήθεια της συνάρτησης getRoomPrice() το τελικό τους κόστος.

cancel()

Η cancel καλεί, μέσω του hInterface την μέθοδο cancel του server, παίρνοντας πίσω ένα string το οποίο προκειμένου να ενημερώσει τον πελάτη τυπώνει το αν πραγματοποιήθηκε η ακύρωση των δωματίων ή όχι και τις υπόλοιπες ενεργές κρατήσεις που αυτός έχει, αν έχει.