Trabajo Práctico Cuatrimestral: Especificación

Materia:

- Algoritmos y Estructuras de Datos

Profesores:

- Ignacio Cassol
- Alicia Gioia

Alumnos :

- Lucas Manzanelli
- Santiago Hazaña

Exercise 4: Hotel

a. ADT: ClientHotel

Description: Represents a client in a hotel simulation who reserve rooms and rent for it.

i. Constructors:

- ***createClientHotel: String: name X String: surname X int:DNI X double: amount → ClientHotel***
  - Precondition: -
  - Postcondition: Client initialize.

ii. Modifiers:

- ***makeReservation: Client X Receptionist X int:roomNumber X int: days → -***
  - *Reserve a room calling the Receptionist.*
  - Precondition: Client and Receptionist must be initialized.
  - Postcondition: A new Reservation was generated.
- ***increaseWallet: Client X double:amount → -***
  - Precondition: Client must be initialized.
  - Postcondition: Client's wallet increase.

iii. Analyzers:

- ***askForInformation(StaticList: rooms)→ -***
  - *Show rooms information such price and type.*
  - Precondition: Rooms must be initialized.
  - Postcondition: Show in console information.

iv. Destructors:

- ***freeClientHotel: ClientHotel→ -***
  - *Free ClientHotel where it's allocated in the memory.*
  - Precondition: ClientHotel initialize.
  - Postcondition: ClientHotel destroyed.

b. ADT: Invoice

Description: It's a Invoice generate in check in with all information by the rent of a room, such as all client's and room's information .

i. Constructors:

- ***createInvoice : int: invoiceNumber X String: hotelName X String: clientName X String: clientSurname X int: clientDNI X double: amuntToPay → Invoice***
  - Precondition: Client and Room must be initialized which has this information.
  - Postcondition: Invoice initialized.

ii. Modifiers:

- ***payRoom: Invoice X double: clientCash*** → double
    - *Check clientCash and if it's more or equals than the invoice's amount changes payStatement to "PAYMENT".*
    - Precondition: Invoice initialized.
    - Postcondition: payStatement change and return the amount if operation it's successful or zero in the other case.

iii. Destructors:

- ***freeInvoice: Invoice*** → **-**
    - *Free Invoice where it's allocated in the memory.*
    - Precondition: Invoice initialize.
    - Postcondition: Invoice destroyed.

c. ADT: Receptionist

Description: Administrator who it's in charge to take Reservations, generates Invoices and has Rooms information.

i. Constructors:

- ***createReceptionist: String: name X String: surname X int:DNI*** → Receptionist
    - Precondition: unique DNI.
    - Postcondition: Receptionist initialized

ii. Modifiers:

- ***checkIn: Receptionist X String:hotelName X String:clientName X String:clientSurname X int: clientDNI X int: roomNumber*** → void
    - *Generates an Invoice with payStatement in "NO PAYMENT" and added into Receptionist invoices list.*
    - Precondition: -
    - Postcondition: Generate and added of a new Invoice.

iii. Analyzers:

- ***checkRoom: Receptionist X int: number*** → boolean
    - *Check if the room it's initialize and available.*
    - Precondition: Receptionist and Room initialize.
    - Postcondition: -

iv. Destructors:

- ***freeReceptionist: Receptionist*** → **-**
    - *Free Receptionist where it's allocated in the memory.*
    - Precondition: Receptionist initialize.

- Postcondition: Receptionist destroyed.

d. ADT: Reservation

Description: A valid ticket for a previous room to rent. Only save client and rooms information.

i. Constructors:

- ***createReservation: clientName X clietnSurname X clientDni X roomNumber X numbersOfDays*** → Reservation
  - *Generates Reservation with client information, rooms number and number of days.*
  - Precondition: -
  - Postcondition: Initialized Reservation.

ii. Destructors:

- ***freeReservation: Reservation* → -**
  - *Free Reservation where it's allocated in the memory.*
  - Precondition: *Reservation* initialize.
  - Postcondition: *Reservation* destroyed.

e. ADT: Room

Description: Representation of a room information which has price, type, occupied room's information and the guest DNI in case if's occupied.

i. Constructors:

- ***createRoom: roomType X roomNumber X price*** → Room
  - Precondition: -
  - Postcondition: Room initialized.

ii. Modifiers:

- ***enterRoom: Room X clientDNi*** → void
  - *Occupied the room with saving clients DNI and changing it's occupate statement.*
  - Precondition: DNI client inside room must be zero.
  - Postcondition: clients DNI save and statement change to "OCCUPIED"
- ***leaveRoom: Room X*** → void
  - *If guest leaves without stablish day, guest recibe a punishment of amount of money in his wallet.*
  - Precondition: -
  - Postcondition: clients DNI save and statement change to "OCCUPIED"

iii. Destructors:

- ***freeRoom: Room*→ -**
  - *Free Room where it's allocated in the memory.*

- Precondition:*Room* initialize.
- Postcondition: *Room* destroyed.