Main uses Hotel
Hotel implements ReservationInterface
Hotel has a composition relationship with all the classes connected with a blue arrow
Customer extends Person
Reservation is associated with the Room and Customer classes
Finally RoomList manages a list of Room objects

Main

```
+ main(args: String[]): void
```

<<interface>>
ReservationInterface

```
+ bookRoom(scanner: Scanner): void
+ cancelReservation(scanner: Scanner): void
+ viewBookingHistory(): void
```

Hotel

```
- name: String
- rooms: ArrayList<Room>
- bookingHistory: Stack<Reservation>
- reservations: Map<Integer, Reservation>
- reservationCounter: static int
```
```
+ Hotel(name: String)
+ bookRoom(scanner: Scanner): void
+ bookRoom(scanner: Scanner, specialRequest: String): void
+ cancelReservation(scanner: Scanner); void
+ viewBookingHistory(): void
- initializeRooms(): void
- viewBookingHistoryRecursive(index: int): void
- isValidPhoneNumber(phoneNumber: String): boolean
- isSameDay(date1; Date, date2: Date): boolean
```

implements

Room

```
- roomNumber: int
- type: String
- rate: double
- isAvailable: boolean
```
```
+ Room(roomNumber: int, type: String, rate: double)
+ getRoomNumber(): int
+ getType(): String
+ getRate(): double
+ isAvailable(): boolean
+ setAvailable(avilable: boolean): void
+ toString(): String
```

Customer

```
- email: String
- phoneNumber: String
```
```
+ Customer(name: String, email: String, phoneNumber: String
+ getEmail(): String
+ getPhoneNumber(): String
+ toString(): String
```

Extends

Person

```
- name: String
```
```
+ Person(name: String)
+ getName(): String
```

RoomList<T>

```
- rooms: List<T>
```
```
+ addRoom(room: T): void
+ getRooms(): List<T>
+ findAvailableRoom(type: String): T
```

<<exception>>
InvalidReservationException

```
+ InvalidReservationException(message: String)
```

Reservation

```
- reservationId: int
- room: Room
- customer: Customer
- startDate: Date
- endDate: Date
- totalRate: Double
```
```
+ Reservation(reseravtionId: int, room: Room, customer: Customer, startDate: Date  endDate: Date, totalRate: double
+ getRoom(): Room
+ getCustomer(): Customer
+ getStartDate(): Date
+ getEndDate(): Date
+ getTotalRate(): Double
+ toString(): String
```