Classe **ControlTotem**

*Classe che si occupa di creare le prenotazioni, tenere traccia dei vari contatori delle code e inserire le prenotazioni in un array*

*Attributi*

ArrayContatori 🡪 array di interi che contiene i vari contatori delle code

*Metodi*

CreaPrenotazione 🡪 crea una nuova istanza di Prenotazione con i parametri passati e a seconda della tipologia la aggiunge alla coda giusta mediante il metodo *aggiungiPrenotazione* aggiungiPrenotazione 🡪 contenuto in CreaPrenotazione, aggiunge la prenotazione appena creata nella listaPrenotazioni nella classe Code

Classe **ControlPannello**

*Attributi*

righePannello 🡪 HashMap che contiene i dati da visualizzare su FramePannello

*Metodi*

aggiornaPannello 🡪 si occupa di aggiornare i dati in righePannello e visualizzarle su FramePannello

Classe **ControlSportello**

*Attributi*

ArraySportelli 🡪 contiene le varie istanze della classe Sportello (a seconda di quanti sportelli siano aperti)

*Metodi*

creaSportello 🡪 crea una istanza della classe Sportello e la inserisce in ArraySportelli

Classe **Prenotazione**

*Metodi*

getCodice() 🡪 prende la tipologia e il numero e ritorna una stringa concatenandole (es. tipologia = A, num = 120 🡪 return A120);

Classe **Sportello**

*Attributi*

ID 🡪 identifica lo sportello mediante un intero

Tipologia 🡪 il servizio erogato dallo sportello (per ora singolo, in futuro uno sportello potrà erogare più tipologie di servizi)

Attivo 🡪 indica se l’istanza di Sportello creata è attualmente attiva e quindi in grado di erogare servizi

Libero 🡪 indica se l’istanza di Sportello è libera di accogliere un nuovo cliente o no

*Metodi*

hoFinito() 🡪 l’operatore ha finito di servire

sonoDisponibile() 🡪 l’operatore da disponibilità a ricevere un nuovo cliente

Classe **Coda**

*Attributi*

ArrayPrenotazioni 🡪 contiene le istanze di Prenotazione create da controlTotem

*Metodi*

Next() 🡪 chiama il prossimo numero secondo una certa logica (ancora da definire)

Classe **Gestore**

*Attributi*

Contiene le istanze delle classi “Control” e della classe Coda

*Metodi*

GestisciCode() 🡪 *logica ancora da definire*