Pre-condizioni generali : L'attore è identificato con un'istanza *o* di Organizzatore

1. consultaTabelloneTurni()

Pre-condizioni : assenti, potrebbe essere eseguita in qualsiasi momento / UC
Post-condizioni : assenti, è un'interrogazione al sistema

2. creaEvento(nome: testo, luogo : testo, dataInizio : Data, partecipanti : intero, elegante : sì/no, privato : sì/no)

Pre-condizioni:
Post-condizioni:

- è stata creata un'istanza *ev* di Evento
- *o* **organizza** *ev*
- *ev*.nome = nome
- *ev*.luogo = luogo
- *ev*.data inizio = dataInizio
- *ev*.partecipanti attesi = partecipanti
- *ev*.elegante = elegante
- *ev*.privato = privato

2a.1 creaEventoRicorrente(nome: testo, luogo : testo, partecipanti : intero, elegante : sì/no, privato : sì/no, dataInizio: Data, tipoRicorrenza: enum, ripetizioni?: intero, dataFine?: Data)

Pre-condizioni:

- Almeno un parametro tra ripetizioni e dataFine è specificato

Post-condizioni:

- per ogni ricorrenza *data* dettata dal tipoRicorrenza, a partire da dataInizio e o fino a dataFine, o per un certo numero di ripetizioni
  - è stata creata un'istanza *ev* di Evento
  - *o* **organizza** *ev*
  - *ev*.nome = nome
  - *ev*.luogo = luogo
  - *ev*.data inizio = *data*
  - *ev*.partecipanti attesi = partecipanti
  - *ev*.elegante = elegante
  - *ev*.privato = privato

2b.1 scegliEventoPerModifica(evento: Evento)

Pre-condizioni: --
Post-condizioni: --

3. aggiungiServizio(tipoServizio : testo, data : Data, partecipanti : intero)

Pre-condizioni: è in corso la definizione di un Evento *ev*
Post-condizioni: se partecipanti <= *ev*.partecipanti attesi e *ev*.terminato = no

- è stata creata un'istanza *serv* di Servizio
- *serv*.tipo servizio = tipoServizio
- *serv*.data = data
- *serv*.partecipanti = partecipanti
- è stata creata un'istanza *turno* di TurnoDiServizio
- *serv* **è effettuato in** *turno*
- *ev* **prevede** *serv*

4. scegliChefPerEvento(chef : Chef)

Pre-condizioni: è in corso la definizione di un Evento *ev*
Post-condizioni: se *ev*.terminato = no

- [se esiste uno Chef *ch* tale che *ev* **è assegnato a** *ch*] la relazione **è assegnato a** tra *ev* e *ch* è eliminata
- *ev* **è assegnato a** chef

5. modificaPartecipantiAttesi(numeroPartecipanti : intero)

Pre-condizioni: è in corso la definizione di un Evento *ev*
Post-condizioni: se per ogni Servizio *serv* tale che *ev* **prevede** *serv* si ha *serv*.partecipanti <= numeroPartecipanti e *ev*.terminato = no

- *ev*.partecipanti attesi = numeroPartecipanti

5a.1 proponiAggiuntaAMenù(servizio: Servizio, menu : Menu, ricetta: Ricetta)

Pre-condizioni:

- è in corso la definizione di un Evento *ev*
- *ev* **prevede** servizio
- menu **è in uso in** servizio

Post-condizioni: se *ev*.in corso = no

- [se non esiste una Sezione *sez* tale che menu **contiene** *sez* e *sez*.nome = "Proposte dell'organizzatore"]
  - è stata creata un'istanza *sez* di Sezione
  - menu **contiene** *sez*
  - *sez*.nome = "Proposte dell'organizzatore"
- è stata creata un'istanza *v* di Voce
- *v* **fa riferimento a** ricetta
- *v* **appartiene** a *sez*

5b.1 proponiRimozioneAMenù(servizio: Servizio, menu: Menu, sezione?: Sezione, voce: Voce)

Pre-condizioni:
- è in corso la definizione di un Evento *ev*
- *ev* **prevede** servizio
- menu è **in uso in** servizio
- voce **appartiene a** menu oppure, se sezione è specificata, menu **contiene** sezione e voce **appartiene a** sezione

Post-condizioni: se ev.in corso = no
- voce.proposta rimozione = sì

5c.1 approvaMenùPerServizio(servizio: Servizio)

Pre-condizioni:
- è in corso la definizione di un Evento *ev*
- *ev* **prevede** servizio

Post-condizioni:
- servizio.menù approvato = sì
- *ev*.in corso = sì

5d.1 modificaServizio(servizio: Servizio, tipoServizio?: enum, data?: Data, partecipanti?: intero)

Pre-condizioni:
- è in corso la definizione di un Evento *ev*
- *ev* **prevede** servizio

Post-condizioni: se non sono specificati i partecipanti oppure partecipanti <= *ev*.partecipanti attesi, e se *ev*.terminato = no
- [se è specificato un tipoServizio] servizio.tipo servizio = tipoServizio
- [se è specificata una data] servizio.data = data
- [se sono specificati i partecipanti] servizio.partecipanti = partecipanti

5e.1 terminaEvento(evento: Evento)

Pre-condizioni: *o* **organizza** evento
Post-condizioni: se evento.in corso = sì
- evento.terminato = sì

5f.1 aggiungiNote(note: testo)

Pre-condizioni: è in corso la definizione di un Evento *ev*
Post-condizioni:
- ev.note = note

5g.1 modificaLocation(luogo: testo)

Pre-condizioni: è in corso la definizione di un Evento *ev*
Post-condizioni: se *ev*.terminato = no

- ev.luogo = luogo

6. assegnaPersonale(personale : PersonaleDiServizio, servizio: Servizio, turno : TurnoDiServizio, ruolo? : testo)

Pre-condizioni:

- è in corso la definizione di un Evento *ev*
- *ev* **prevede** servizio
- servizio **è effettuato in** turno

Post-condizioni:
se personale **è impiegabile per** turno e *ev*.terminato = no

- personale **è chiamato per** turno
- [se è specificato un ruolo]
  - [se non esiste un RuoloPersonale per la relazione **è chiamato per** tra personale e turno]
    - è stata creata una istanza di RuoloPersonale *r*
  - *r*.ruolo = ruolo

6a.1 rimuoviPersonale(servizio: Servizio, turno: TurnoDiServizio, personale: PersonaleDiServizio)

Pre-condizioni:

- è in corso la definizione di un Evento *ev*
- *ev* **prevede** servizio
- servizio **è effettuato in** turno
- personale **è chiamato per** turno

Post-condizioni:

- l'istanza *rp* di RuoloPersonale relativa alla relazione **è chiamato per** è eliminata
- la relazione **è chiamato per** tra personale e turno è eliminata

6b.1 aggiungiTurnoDiServizio(servizio: Servizio, oraInizio?: Ora, oraFine?: Ora)

Pre-condizioni: è in corso la definizione di un Evento *ev*
Post-condizioni: se *ev*.terminato = no

- è stata creata un'istanza *turno* di TurnoDiServizio
- servizio **è effettuato in** *turno*
- [se è specificata una oraInizio] *turno*.ora inizio = oraInizio
- [se è specificata una oraFine] *turno*.ora inizio = oraFine

6c.1 rimuoviTurnoDiServizio(servizio: Servizio, turno: TurnoDiServizio)

Pre-condizioni:

- è in corso la definizione di un Evento *ev*
- servizio **è effettuato in** turno

Post-condizioni: se *ev*.terminato = no

- per ogni Personale *pers* tale che *pers* **è chiamato per** turno
  - l'istanza *rp* di RuoloPersonale relativa alla relazione **è chiamato per** è eliminata
  - la relazione **è chiamato per** è eliminata
- per ogni Personale di Servizio *ps* tale che *ps* **è impiegabile per** turno, la relazione **è impiegabile per** è eliminata

7. cancellaEvento(evento : Evento)

Pre-condizioni: --
Post-condizioni: Se evento.in corso = no

- per ogni Servizio *serv* tale che evento **prevede** *serv*
  - per ogni TurnoDiServizio *t* tale che *serv* **è effettuato in** *t* , *t* è eliminato
  - *serv* è eliminato
- evento è eliminato

7a.1 annullaEvento(evento: Evento)

Pre-condizioni: --
Post-condizioni:

- per ogni Servizio *serv* tale che evento **prevede** *serv*, *serv.*annullato = sì
- evento.annullato = sì

7b.1 annullaServizio(servizio: Servizio)

Pre-condizioni:

- è in corso la definizione di un Evento *ev*
- *ev* **prevede** servizio

Post-condizioni:

- servizio.annullato = sì

7c.1 cancellaServizio(servizio: Servizio)

Pre-condizioni:

- è in corso la definizione di un Evento *ev*
- *ev* **prevede** servizio

Post-condizioni: Se *ev*.in corso = no

- per ogni TurnoDiServizio *t* tale che servizio **è effettuato in** *t*, *t* è eliminato
- servizio è eliminato