**Pre-condizioni generali:**

* L’attore è identificato con un’istanza di *org* di Organizzatore
* E’ stata resa nota l’istanza *calendario* di Calendario

**1 LavoraSullaCucina(cucina: Cucina)**

**Pre-condizioni:** ---

**Post-condizioni:** ---

**2 CreaTurniCucinaRicorrentemente(dalleOre: Orario, alleOre: Orario, dalGiornoSettimana: Giorno, alGiornoSettimana: Giorno, dallaData: Data, allaData: Data, scadenzaDisponibilità?: Data)**

**Pre-condizioni:**

* Si sta lavorando sulla *cucina*
* alleOre > dalleOre
* alGiornoDellaSettimana >= dalGiornoDellaSettimana
* dallaData <= allaData

**Post-condizioni:**

* E’ stata creata creata un’istanza di Ricorrenza cucina *r* tale che:
  + r.dalle\_ore = dalleOre
  + r.alle\_ore = alleOre
  + r.dal\_giorno = dalGiorno
  + r.al\_giorno = alGiorno
  + r.dalla\_data = dallaData
  + r.alla\_data = allaData
  + *r* **riferita a** *cucina*
* Per ogni Data d tale che:
  + dallaData <= d <= allaData
  + dalGiornoSettimana <= g <=alGiornoSettimana

(Dove il giorno della data d coincide con g)

* + Non esiste un **Turno di cucina** *t* tale che:
    - *t* **è tenuto nella** *cucina*
    - *t*.data\_svolgimento =*d*
    - dalleOre<= *t*.inizio<=alleOre oppure dalleOre<= *t*.fine<=alleOre

E’ stato creato un Turno di cucina *t* tale che:

* + *t*.data\_svolgimento =*d*
  + *t*.inizio *=* dalleOre
  + *t*.fine = alleOre
  + [Se specificata scadenzaDisponibilità]:

*t*.data\_scadenza =*scadenzaDisponibilità*

* + *org* **organizza** *t*
  + *t* **è tenuto nella** *cucina*
  + *calendario* **contiene** *t*
  + *t* **ricorre nella** *r*

**2a.1 CreaTurnoCucina(dalleOre: Orario, alleOre: Orario, data: Data, scadenzaDisponibilità?**: **Data)**

**Pre-condizioni:**

* Si sta lavorando sulla *cucina*
* alleOre > dalleOre
* Non esiste un Turno di cucina *t* tale che:
  + *t* **è tenuto nella** *cucina*
  + *t*.data\_svolgimento =data
  + dalleOre<= *t*.inizio<=alleOre oppure dalleOre<= *t*.fine<=alleOre

**Post-condizioni:**

* E’ stata creata un’istanza *t* di Turno di cucina
* *t*.data\_svolgimento =*d*
* *t*.inizio = dalleOre
* *t*.fine = alleOre
* [Se specificata scadenza]:*t*.data\_scadenza = scadenzaDisponibilità
* *org* **organizza** *t*
* *t* **è tenuto nella** *cucina*
* *calendario* **contiene** *t*

**2b.1.EliminaTurnoCucina(turnoCucina: Turno di cucina, eliminaTurniRicorrenti: si/no)**

**Pre-condizioni:**

* Si sta lavorando sulla *cucina*
* turnoCucina **è tenuto nella** *cucina*
* *org* **organizza** turnoCucina
* Non esiste un Cuoco *c* tale che *c* **è disponibile** nel turnoCucina

**Post-condizioni:**

[se **eliminaTurniRicorrenti** = si]:

[se esiste una Ricorrenza di cucina r taleche turnoCucina **ricorre nella** *r*]:

* Per ogni Turno di cucina t tale che *t* **ricorre nella** *r:*

[Se non esiste un Cuoco *c* tale che *c* **è disponibile** nel t]: *t* è stato eliminato

* *r* è stata eliminata

Altrimenti: turnoCucina è stato eliminato

Altrimenti turnoCucina è stato eliminato

**2d.1.ModificaTurnoCucina(turnoCucina: Turno di cucina, dalleOre?: Orario, alleOre?: Orario, data?: Data, scadenzaDisponibilità?: Data, èValidaScadenza: si/no)**

**Pre-condizioni:**

* Si sta lavorando sulla *cucina*
* turnoCucina **è tenuto nella** *cucina*
* *org* **organizza** turnoCucina
* Non esiste un Cuoco *c* tale che *c* **è disponibile** nel turnoCucina
* Esiste una Data *dataTurnoCucina* tale che turnoCucina **è tenuto nella** *dataTurnoCucina*
* [se specificato alleOre]:
  + [se specificato dalleOre]: alleOre>dalleOre

altrimenti: alleOre> turnoCucina.inizio

altrimenti:

* [se specificato dalleOre]: turnoCucina.fine > dalleOre
* Non esiste un Turno di cucina *t* tale che:
  + *t* non è turnoCucina
  + *t* **è tenuto nella** *cucina*
  + [se specificata data]:t.data\_svolgimento = data

altrimenti*:* t.data\_svolgimento *= dataTurnoCucina*

* + [se specificato dalleOre]:
    - [se specificato alleOre]:
      * dalleOre<= *t*.inizio<=alleOre oppure dalleOre<= *t*.fine<=alleOre
    - Altrimenti:
      * dalleOre<= *t*.inizio<=turnoCucina.fine oppure dalleOre<= *t*.fine<=turnoCucina.fine

*altrimenti:*

* + - [se specificato alleOre]:
      * turnoCucina.inizio<= *t*.inizio<=alleOre oppure turnoCucina.inizio<= *t*.fine<=alleOre
    - Altrimenti:
      * turnoCucina.inizio<= *t*.inizio<=turnoCucina.fine oppure turnoCucina.inizio<= *t*.fine<=turnoCucina.fine

**Post-condizioni:**

* [se specificato dalleOre]: turnoCucina.inizio = dalleOre
* [se specificato alleOre]: turnoCucina.fine = alleOre
* [se specificata data]:
  + *t*.data\_svolgimento = data
* [se èValidaScadenza = si]
  + [se specificata scadenzaDisponibilità]:
    - turnoCucina.data\_scadenza = scadenzaDisponibilità

altrimenti:

*t*.data\_svolgimento = \*Data nulla\*

**3.RaggruppaTurniCucinaCumulativamente(dalleOre: Orario, alleOre: Orario, dalGiornoSettimana: Giorno, alGiornoSettimana: Giorno, dallaData: Data, allaData: Data)**

**Pre-condizioni:**

* Si sta lavorando sulla *cucina*
* alleOre > dalleOre
* alGiornoDellaSettimana >= dalGiornoDellaSettimana
* dallaData <= allaData

**Post-condizioni:**

* E’ stata creata un’istanza Raggruppamento cucina *r* tale che:

*r* **è riferito a** *cucina*

* Per Turno di cucina *t tale* che:
  + *t* **è tenuto nella** *cucina*
  + *org* **organizza** *t*
  + *t*.inizio >= dalleOre
  + *t*.fine <= alleOre

dalGiornoSettimana <= g <=alGiornoSettimana

(Dove il giorno della data *t.data\_svolgimento* coincide con g)

* + dallaData <= *t.data\_svolgimento* <= allaData
  + Non esiste un Raggruppamento cucina *r* tale che *r* **contiene** *t*

allora *r* **contiene** *t*

**3a.1.RaggruppaTurniCucinaEsplicitamente(insieme turniCucina di Turni di cucina)**

**Pre-condizioni:**

* Si sta lavorando sulla *cucina*
* La cardinalità di turniCucina è maggiore di 1
* Per ogni Turno di cucina *t* appartenente a turniCucina:
  + *t* **è tenuto nella** *cucina*
  + *org* **organizza** *t*
  + Non esiste un Raggruppamento cucina *r* tale che *r* **contiene** *t*

**Post-condizioni:**

* E’ stata creata un’istanza Raggruppamento cucina *r* tale che:
  + Per ogni *t* appartenente a turniCucina: *r* **contiene** *t*
  + *r* **è riferito a** *cucina*

**3b.1.EliminaRaggruppamentoTurnoCucina(turnoCucina: Turno di cucina, eliminaGruppo: si/no)**

**Pre-condizioni:**

* Si sta lavorando sulla *cucina*
* *t* **è tenuto nella** *cucina*
* *org* **organizza** *t*
* Esiste un Raggruppamento cucina *r* tale che *r* **contiene** turnoCucina

**Post-condizioni:**

[se eliminaGruppo = si]

*r* è stato eliminato

altrimenti: E’ stato eliminato il collegamento d’associazione che *r* **contiene** turnoCucina

**(1-3)a.1.LavoraSuEvento(evento: Evento)**

**Pre-condizioni:** *org* **organizza** evento

**Post-condizioni:** ---

**(1-3)a.2.GeneraTurniServiziRicorrenti(minutiAnticipo:Numero , minutiRitardo:Numero, scadenzaDisponibilità?:Data)**

**Pre-condizioni:**

* Si sta lavorando su *evento*
* Esiste un’istanza di Ricorrenza servizio *r* tale che *r* **è riferito a** *evento*

**Post-condizioni:**

* Per ogni Servizio *s* tale che s **ricorre nella** *r*:
* [Se non esiste un Turno di servizio *t* tale che *t* **è riferito a** *s*]:

E’ stata creata una nuova istanza di Turno di servizio t tale che:

* + - t.data\_svolgimento = s.data\_svolgimento
    - *t* **è riferito a** *servizio*

*t*.inizio=o1

(Dove o1 è l’orario con minutiAnticipo minuti di anticipo rispetto a s.inizio)

* + - *t*.fine =o2

(Dove o2 è l’orario con minutiRitardo minuti di ritardo rispetto a s.fine)

* + - [Se specificata scadenzaDisponibilità]:

*t*.data\_scadenza =*scadenzaDisponibilità*

* + - *org* **organizza** *t*
    - *calendario* **contiene** *t*
    - *t* **è riferito a** *servizio*

**(1-3)a.2a.1.EliminaTurniServiziRicorrenti()**

**Pre-condizioni:**

* Si sta lavorando su *evento*
* Esiste un’istanza di Ricorrenza servizio *r* tale che *r* **è riferito a** *evento*

**Post-condizioni:**

Per ogni Servizio *s* tale che s **ricorre nella** *r*:

Per ogni Turno di servizio *t* tale che *t* **è riferito a** *r*:

[Se non esiste un’istanza di Personale di servizio *p* tale che *p* è disponibile nel *t* e non esiste un’istanza di Cuoco c tale che *c* è disponibile nel *t*]:

* *t* è stato eliminato
* Per ogni istanza di Mansione di servizio *m* tale che m **è svolta nel** *t*: *m* è stata eliminata

**(1-3)a.3.LavoraSulServizio(servizio: Servizio)**

**Pre-condizioni:** *evento* **prevede** servizio

**Post-condizioni:** ---

**(1-3)a.4.CreaTurnoServizio(minutiAnticipo:Numero, minutiRitardo:Numero, scadenzaDisponibilità?:Data)**

**Pre-condizioni:**

* Si sta lavorando sul *servizio*
* Non esiste un Turno di servizio *t* tale che *t* **è riferito a** *servizio*

**Post-condizioni:**

* E’ stata creata una nuova istanza di Turno di servizio t
* *t* **è riferito a** *servizio*
* t.data\_svolgimento = *servizio*.data\_svolgimento
* *t*.inizio=o1

(Dove o1 è l’orario con minutiAnticipo minuti di anticipo rispetto a *servizio*.inizio)

* *t*.fine =o2

(Dove o2 è l’orario con minutiRitardo minuti di ritardo rispetto a *servizio*.fine)

* [Se specificata scadenzaDisponibilità]:

*t*.data\_scadenza =*scadenzaDisponibilità*

* *org* **organizza** *t*
* *calendario* **contiene** *t*

**(1-3)a.4a.1.EliminaTurnoServizio()**

**Pre-condizioni:**

* Si sta lavorando sul *servizio*
* Esiste un Turno di servizio *t* è riferito a *servizio*
* Non esiste un’istanza di Personale di servizio *p* tale che *p* è disponibile nel *t*
* Non esiste un’istanza di Cuoco c tale che *c* è disponibile nel *t*

**Post-condizioni:**

* Per ogni Mansione di servizio *m* tale che m è **svolta nel** *t:*

*m* è stata eliminata

* *t* è stato eliminato

**(1-3)a.5.AssegnaMansioneServizio(personaleServizio: Personale)**

**Pre-condizioni:**

* Si sta lavorando sul *servizio*
* personaleServizio è un’istanza di Personale di servizio oppure è un’istanza di Cuoco
* Esiste un Turno di servizio *t* tale che *t* **è riferito a** *servizio*

**Post-condizioni:**

* E’ stata creata una nuova istanza di Mansione di servizio *m*
* *m* **è svolta da**personaleServizio
* *m* **è svolta nel** *t*

**(1-3)a.5a.1.EliminaMansioneServizio(mansioneServizio: Mansione di servizio)**

**Pre-condizioni:**

* Si sta lavorando sul *servizio*
* Esiste un Turno di servizio *t* tale che mansioneServizio **è svolta nel** *t*
* *t* **è riferito a** *servizio*

**Post-condizioni:** mansioneServizio è stata eliminata

**(1-3)a.6.RaggruppaTurniServizioCumulativamente(dalleOre: Orario, alleOre: Orario, dalGiornoSettimana: Giorno, alGiornoSettimana: Giorno, dallaData: Data, allaData: Data)**

**Pre-condizioni:**

* Si sta lavorando su *evento*
* alleOre > dalleOre
* alGiornoDellaSettimana >= dalGiornoDellaSettimana
* dallaData <= allaData

**Post-condizioni:**

* E’ stata creata un’istanza Raggruppamento di servizio *r* tale che: *r* **è riferito a** *evento*
* Per ogni Turno di servizio *t tale* che:
  + Esiste un Servizio *s* tale che *t* **è riferito a** *s*
  + *evento* **prevede** *s*
  + *org* **organizza** *t*
  + *t*.inizio >= dalleOre
  + *t*.fine <= alleOre
  + dalGiornoSettimana <= g <=alGiornoSettimana

(Dove il giorno della data *t.data\_svolgimento* coincide con g)

* + dallaData <= *t.data\_svolgimento* <= allaData
  + Non esiste un Raggruppamento di servizio *r* tale che *r* **contiene** *s*

allora: *r* **contiene** *t*

**(1-3)a.6a.1.RaggruppaTurniServizioEsplicitamente(insieme turniServizio di Turni di servizio)**

**Pre-condizioni:**

* Si sta lavorando su *evento*
* La cardinalità di turniServizio è maggiore di 1
* Per ogni Turno di servizio *t* appartenente a turniServizio:
  + Esiste un Servizio *s* tale che *t* **è riferito a** *s*
  + *evento* **prevede** *s*
  + *org* **organizza** *t*
  + Non esiste un Raggruppamento di servizio *r* tale che *r* **contiene** *t* e *r* **riferito a** *evento*

**Post-condizioni:**

* E’ stata creata un’istanza Raggruppamento di servizio *r* tale che:
  + Per ogni *t* appartenente a turniServizio: *r* **contiene** *t*
  + *r* **è riferito a** *evento*

**(1\_3)a.6b.1.EliminaRaggruppamentoTurnoServizio(turnoServizio: Turno di Servizio, eliminaGruppo: si/no)**

**Pre-condizioni:**

* Si sta lavorando su *evento*
* *org* **organizza** *t*
* Esiste un Raggruppamento di servizio *r* tale che *r* **contiene** turnoServizio e *r* **riferito a** *evento*

**Post-condizioni:**

[se eliminaGruppo = si]

*r* è stato eliminato

altrimenti: E’ stato eliminato il collegamento d’associazione che *r* **contiene** turnoServizio