

Contratti per lo UC “Gestire turni”

Pre-condizione generale:

- l'attore è identificato con un'istanza *o* di Organizzatore
- esiste un'unica istanza *calendario* di CalendarioTurni
- Per ogni istanza *i* di Turno, GruppoTurni, RicorrenzaGruppi, RicorrenzaTurni, *calendario* **contiene** *i*

1. consultaCalendario()

Pre-condizioni: ---

Post-condizioni: ---

2. creaTurnoPreparatorio(data: data, oraInizio: numero, oraFine: numero, sede?: SedeCucina)

Pre-condizioni:

- esiste un'unica istanza *cat* di CatalogoSedi
- esiste un'istanza *sedePrinc* di SedeCucina tale che *cat* **contiene** *s* e *sedePrinc* è **sede principale di** *cat*
- [se sede è definito] *cat* **contiene** sede

Post-condizioni:

- Se data non è nel passato e oraInizio è antecedente a oraFine:
 - Se sede è definito e non esiste un'istanza *t2* di TurnoPreparatorio tale che *calendario* **contiene** *t2*, *t2* è **svolto presso** sede e *t2.oraFine* precede oraInizio oppure oraFine precede *t2.oraInizio*:
 - è stata creata un'istanza *t* di TurnoPreparatorio
 - *t.data* = data
 - *t.oraInizio* = oraInizio
 - *t.oraFine* = oraFine
 - *t.completo* = no
 - *t* è **svolto presso** sede
 - *calendario* **contiene** *t*
 - Se sede non è definito e non esiste un'istanza *t2* di TurnoPreparatorio tale che *calendario* **contiene** *t2*, *t2* è **svolto presso** *sedePrinc* e *t2.oraFine* precede t.oraInizio:
 - è stata creata un'istanza *t* di TurnoPreparatorio
 - *t.data* = data
 - *t.oraInizio* = oraInizio
 - *t.oraFine* = oraFine
 - *t.completo* = no
 - *t* è **svolto presso** *sedePrinc*
 - *calendario* **contiene** *t*

2a.1. consultaListaEventi()

Pre-condizioni: ---

Post-condizioni: ---

2a.2. creaTurnoServizio(data: data, oraInizio: numero, oraFine: numero, evento: Evento)

Pre-condizioni: ---

Post-condizioni:

- Se data non è nel passato e oraInizio è antecedente a oraFine:
 - è stata creata un'istanza *t* di TurnoServizio
 - *t.data* = data
 - *t.oraInizio* = oraInizio
 - *t.oraFine* = oraFine
 - *t.completo* = no
 - *t* è riferito a evento
 - *calendario* **contiene** *t*

2a.2a.1. precompilaTurniServizio(evento: Evento)

Pre-condizioni:

- evento non è nel passato
- Per ogni Servizio *s* tale che evento **contiene** *s*, non esiste alcun TurnoServizio *ts* tale che *ts* è riferito a *s*

Post-condizioni:

- Per ogni Servizio *ser* tale che evento **contiene** *ser*:
 - è stata creata un'istanza *t* di TurnoServizio
 - *t.data* = *ser.data*
 - *t.oraInizio* = *ser.oraInizio*
 - *t.oraFine* = *ser.oraFine*
 - *t.completo* = no
 - *t* è riferito a *ser*
 - *t* è riferito a evento
 - *calendario* **contiene** *t*

3. **impostaRicorrenza(frequenza, dataInizio: data, dataFine: data)**

Pre-condizioni:

- è in corso la definizione di un Turno *t*
- *t.data* = dataInizio

Post-condizioni:

- Se dataInizio è antecedente a dataFine:
 - è stata creata un'istanza *ric* di RicorrenzaTurni
 - *ric.dataInizio* = dataInizio
 - *ric.dataFine* = dataFine
 - *ric.oraInizio* = *t.oraInizio*
 - *ric.oraFine* = *t.oraFine*
 - Per ogni data *data* compresa tra dataInizio e dataFine e che rispetti frequenza:
 - Se *t* è un TurnoPreparatorio e non vengono a crearsi sovrapposizioni di turni preparatori in *data*, oppure se *t* è un TurnoServizio:
 - [se *t* è un TurnoPreparatorio] è stata creata un'istanza *t_i* di TurnoPreparatorio
 - [se *t* è un TurnoServizio] è stata creata un'istanza *t_i* di TurnoServizio
 - *t_i.data* = *data*
 - *t_i.oraInizio* = *ric.oraInizio*
 - *t_i.oraFine* = *ric.oraFine*
 - *t_i.completo* = no
 - [se *t_i* è un TurnoPreparatorio] *t_i* è **svolto presso** la SedeCucina *sede* tale che *t* è **svolto presso sede**
 - [se *t_i* è un TurnoServizio] *t_i* è **referito all'Evento** e tale che *t* è **referito a e**
 - *calendario* **contiene** *t_i*
 - *t_i* **ricorre in** *ric*
 - *calendario* **contiene** *ric*

4. **impostaDataScadenza(data: data)**

Pre-condizioni:

- è in corso la definizione di un Turno *t*

Post-condizioni:

- Se data non è nel passato ed è antecedente a *t.data*:
 - *t.scadenzaModificaDisponibilita* = data

5. creaGruppo(insieme \underline{T} di TurnoPreparatorio)

Pre-condizioni:

- Per ogni TurnoPreparatorio t in \underline{T} :
 - non esiste alcun GruppoTurni g tale che g contiene t
 - non esiste alcun Cuoco c tale che c **ha indicato disponibilità per** t

Post-condizioni:

- è stata creata un'istanza *gruppo* di GruppoTurni
- Per ogni Turno t di \underline{T} :
 - *gruppo* **contiene** t
- *calendario* **contiene** *gruppo*

5a.1. creaGruppoRicorrente(lista ordinata \underline{T} di TurnoPreparatorio, frequenza, dataInizio: data, dataFine: data)

Pre-condizioni:

Post-condizioni:

- Se dataInizio e dataFine non sono nel passato, dataInizio è antecedente a dataFine, per ogni TurnoPreparatorio t in \underline{T} $t.data$ non è nel passato e non esiste alcun Cuoco c tale che c **ha indicato disponibilità per** t :
 - è stata creata un'istanza $ricGruppi$ di RicorrenzaGruppi
 - $ricGruppi.dataInizio = \underline{dataInizio}$
 - $ricGruppi.dataFine = \underline{dataFine}$
 - Per ogni TurnoPreparatorio t contenuto in \underline{T} :
 - È stata creata un'istanza ric di RicorrenzaTurni
 - $ric.dataInizio = t.data$
 - $ric.dataFine = \underline{dataFine}$
 - $ric.oraInizio = t.oraInizio$
 - $ric.oraFine = t.oraFine$
 - Per ogni data $data$ compresa tra $t.data$ e dataFine e che rispetti frequenza:
 - Se non vengono a crearsi sovrapposizioni di turni preparatori in $data$:
 - è stata creata un'istanza t_i di TurnoPreparatorio
 - $t_i.data = data$
 - $t_i.oraInizio = ric.oraInizio$
 - $t_i.oraFine = ric.oraFine$
 - $t_i.completo = no$
 - t_i è **svolto presso** la SedeCucina $sede$ tale che t è **svolto presso** $sede$
 - $calendario$ **contiene** t_i
 - t_i **ricorre in** ric
 - $calendario$ **contiene** ric
 - Per ogni lista di TurnoPreparatorio $\underline{T_i}$ tale che ogni turno t_i contenuto in $\underline{T_i}$ è un'occorrenza ordinata di ciascuna ricorrenza ric_i creata:
 - È stata creata un'istanza g_i di GruppoTurni
 - Per ogni TurnoPreparatorio t_i contenuto in $\underline{T_i}$:
 - g_i **contiene** t_i
 - $calendario$ **contiene** g_i
 - g_i **ricorre in** $ricGruppi$
 - $calendario$ **contiene** $ricGruppi$

6. modificaTurnoPreparatorio(turnoP: TurnoPreparatorio, data?: data, oraInizio?: numero, oraFine?: numero, sede?: SedeCucina, dataScadenzaDisponibilita?: data)

Pre-condizioni:

- Non esiste alcun Gruppo Turni *g* tale che *g* **contiene** turnoP
- Non esiste alcun Cuoco *c* tale che *c* **lavora in** turnoP
- [se oraInizio è definito] oraFine è definito
- [se oraFine è definito] oraInizio è definito

Post-condizioni:

- Se non esiste alcun Cuoco *c2* tale che *c2* **ha indicato disponibilità per** turnoP, e le modifiche sono tali che non esistano un TurnoPreparatorio *t2* e una SedeCucina *s2* tali che *t2* **è svolto presso** *s2*, turnoP **è svolto presso** *s2*, *t2.data* sia uguale a turnoP.data e *t2.oraFine* sia successivo a turnoP.oraInizio (cioè tali da non creare una sovrapposizione di turni preparatori):
 - [se data è definito e data non è nel passato] turnoP.data = data
 - [se oraInizio e oraFine sono definiti e oraInizio < oraFine]:
 - turnoP.oraInizio = oraInizio
 - turnoP.oraFine = oraFine
 - [se sede è definito ed esiste una SedeCucina *sc* tale che turnoP **è svolto presso** *sc*]:
 - l'associazione **svolto presso** tra turnoP e *sc* è eliminata
 - turnoP **è svolto presso** sede
 - [se dataScadenzaDisponibilita è definito e dataScadenzaDisponibilita non è nel passato]: turnoP.scadenzaModificaDisponibilita = dataScadenzaDisponibilita
 - [se esiste una RicorrenzaTurni *ric* tale che turnoP **ricorre in** *ric*] l'istanza *ric* viene eliminata

6a.1. modificaTurnoServizio(turnoS: TurnoServizio, tempoAggiuntivoPrima?: numero, tempoAggiuntivoDopo?: numero, dataScadenzaDisponibilita?: data)

Pre-condizioni: ---

Post-condizioni:

- Se non esistono alcun PersonaleServizio *ps* e alcun Cuoco *c* tali per cui *ps* **ha indicato disponibilità per** turnoS o *c* **ha indicato disponibilità per** turnoS:
 - [se tempoAggiuntivoPrima è definito] turnoS.oraInizio = turnoS.oraInizio – tempoAggiuntivoPrima
 - [se tempoAggiuntivoDopo è definito] turnoS.oraFine = turnoS.oraFine + tempoAggiuntivoDopo
 - [se dataScadenzaDisponibilita è definito e dataScadenzaDisponibilita non è né nel passato né successiva a turnoS.data] turnoS.scadenzaModificaDisponibilita = dataScadenzaDisponibilita
 - [se esiste una RicorrenzaTurni *ric* tale che turnoS **ricorre in** *ric*] l'istanza *ric* viene eliminata

6b.1. **modificaRicorrenzaTurni(ricT: RicorrenzaTurni, oraInizio: numero, oraFine: numero)**

Pre-condizioni: ---

Post-condizioni:

- Se non esistono alcun Cuoco *c*, PersonaleServizio *ps* e TurnoServizio *ts* tali che *c* **ha indicato la disponibilità per** *ts* oppure *ps* **ha indicato la disponibilità per** *ts*, e *ts* **ricorre in** ricT, oppure se non esistono alcun Cuoco *c* e TurnoPreparatorio *tp* tali che *c* **ha indicato la disponibilità per** *tp* e *tp* **ricorre in** ricT:
 - Se le modifiche non comportano una sovrapposizione di turni preparatori (cioè se non esistono due istanze di TurnoPreparatorio *t1*, *t2* tali che *t1.data* = *t2.data*, *t1.oraFine* > *t2.oraInizio* e tali che esista una SedeCucina *sede* tale che *t1* è **svolto presso** *sede* e *t2* è **svolto presso** *sede*):
 - Se oraInizio < oraFine:
 - ricT.oraInizio = oraInizio
 - ricT.oraFine = oraFine
 - Per ogni Turno *t* tale che *t* **ricorre in** ricT:
 - *t.oraInizio* = oraInizio
 - *t.oraFine* = oraFine

7. **eliminaTurno(turno: Turno)**

Pre-condizioni: --

Post-condizioni:

- [se esiste una RicorrenzaTurni *ric* tale che turno **ricorre in** *ric*] l'istanza *ric* viene eliminata
- l'istanza turno viene eliminata

7a.1 **eliminaTurnoRicorrente(ricT: RicorrenzaTurni)**

Pre-condizioni: ---

Post-condizioni:

- per ogni Turno *t* tale che *t* **ricorre in** ricT:
 - l'istanza *t* viene eliminata
 - [se esiste un GruppoTurno *gruppo* tale che *gruppo* **contiene** *t*] l'istanza *gruppo* viene eliminata
- L'istanza ricT viene eliminata

7b.1 eliminaGruppo(gruppo: GruppoTurni)

Pre-condizioni: ---

Post-condizioni:

- [se esiste una RicorrenzaGruppo *ricG* tale che gruppo **ricorre in** *ricG*] l'istanza *ricG* viene eliminata
- L'istanza gruppo viene eliminata

7c.1 eliminaGruppoRicorrente(ricG: RicorrenzaGruppi)

Pre-condizioni: ---

Post-condizioni:

- per ogni Gruppo *g* tale che *g* **ricorre in** ricG:
 - l'istanza *g* viene eliminata
- L'istanza ricG viene eliminata

7d.1 eliminaRicorrenzaTurni(ricT: RicorrenzaTurni)

Pre-condizioni: ---

Post-condizioni:

- L'istanza ricT viene eliminata

7e.1 eliminaRicorrenzaGruppi(ricG: RicorrenzaGruppi)

Pre-condizioni: ---

Post-condizioni:

- L'istanza ricG viene eliminata