CONTRATTI

GESTIRE I TURNI

**Pre-condizione generale:** l’attore è identificato con un'istanza  *o*  di Organizzatore

1. **creoTabellaCucina(Tipo: testo, foglioPreparazioni: FoglioPreparazioni)**

**pre-condizioni:**

**post-condizioni:**

* è stata creata un’istanza *tc* di TabellaDeiTurni per la cucina
* [se è specificato un Tipo] *tc.Tipo* = Tipo
* *o* è proprietario di *tc*
* tc.ordine = no

**1a. eliminoTabellaCucina(tabellaDeiTurni: TabellaDeiTurni, evento: Evento)**

**pre-condizioni:**

* esiste un’istanza tc di TabellaDeiTurni
* tc è **assegnato** a Evento

**post-condizioni:**

**1b.1 aggiungoTabellaDeiTurni(tabellaDeiTurni: TabellaDeiTurni, evento :Evento)**

**pre-condzioni:**

* è in corso la definizione di una TabellaDeiTutuni tt

**post-condizioni:**

* è stata creata un’istanza *t*  di TabellaDeiTurni
* *t*  fa riferimento a tabellaDeiTurni

**1b.2 impostoCriterioRicorrenza(tabellaDeiTurni: TabellaDeiTurni, evento: Evento)**

**pre-condizioni:**

* esiste un'istanza di Evento

**post-condizioni:**

* lista eventi ricorrenti aggiornata

**1b.1a.1 modificoDettagliIstanza(tabellaDeiTurni: TabellaDeiTurni, evento: Evento, ricorrente si/no)**

**pre-condizioni:**

* si ha in corso un evento ricorrente

**post-condizioni: sali su discord**

**1b.1b.1 eliminoIstanza()**

**pre-condizioni:**

**post-condizioni:**

**1b.1c.1 modificoDettagliIstanze()**

**pre-condizioni:**

**post-condizioni:**

**1b.1d.1 eliminoEventoRicorrente(tabellaDeiTurni: TabellaDeiTurni, evento: Evento, ricorrente si/no)**

**pre-condzioni:**

**post-condizioni**

**1b.2 impostazioneCriterioRicorrenza(tabellaDeiTurni: TabellaDeiTurn)**

**pre-condizioni:**

**post-condizioni:**

1. **creoTabellaServizio(Tipo: testo)**

**pre-condizioni:**

**post-condizioni:**

* è stata creata un’istanza *ts* di TabellaDeiTurni
* [se è specificato un Tipo] *ts.Tipo* = Tipo
* *o*  è proprietario di *ts*
* tc.ordine = no

**2a.1 eliminoTabella(tabellaDeiTurni: TabellaDeiTurni)**

**pre-condizioni:**

* tabellaDeiTurni non è in uso e non è vuota
* *o* è il **proprietario** di tabellaDeiTurni

**post-condizioni:**

1. **aggiungoTurno( TipoT: testo, pre-estensione?: ora, post-estensione?: ora, deadline?: data, data?: data, oraInizio?: ora, oraFine?: ora)**

**pre-condizioni:**

* è in corso la definizione di una TabellaDeiTurni *tc*
* è in corso la definizione di una TabellaDeiTurni *ts*

**post-condizioni:**

* è stata creata un’istanza *t* di Turno
* [se TipoT = “cucina”] *t* appartiene a *tc*
* [se TipoT = “servizio”] *t* appartiene a *ts*
* [se è specificata una oraInizio] *t.oraInizio* = oraInizio
* [se è specificata una oraFine] *t.oraFine* = oraFine
* [se è specificata una pre-estensione] *t.oraInizio*  = pre-estensione
* [se è specificata una post-estensione] *t.oraFine*  = post-estensione
* [se è specificata una deadline] *t.deadline* = deadline
* [se è specificata una data] *t.data* = data

**3a.1. turnoInGruppo(turno: Turno, tipoT: testo, gruppo?: si/no, nomeGruppo?: testo)**

**pre-condizioni:**

* turno non è ancora iniziato
* è in corso la definizione di TabellaDeiTurni *tc*
* è in corso la definizione di TabellaDeiTurni *ts*
* turno.*tipoT*  deve essere uguale a tipoT

**post-condizioni:**

* [se è specificato un gruppo] turno.*gruppo* = gruppo
* [se è specificato un nomeGruppo] turno.*nomeGruppo* = nomeGruppo

**3a.2. aggiornoTurnoInGruppo(turno: Turno, gruppo?: si/no, nomeGruppo?: testo)**

**pre-condizioni:**

* turno non è ancora iniziato

**post-condizioni:**

* [se gruppo = no] turno.*gruppo* = gruppo , turno.*nomeGruppo* = “”
* [se è specificato un nomeGruppo] turno.*nomeGruppo*  = nomeGruppo

**3a.3. aggiornaGruppo( nomeGruppo: testo, gruppo?: si/no, nuovoNomeGruppo?: testo)**

**pre-condizioni:**

* i Turni che *turno.nomeGruppo*  = nomeGruppo non sono ancora iniziati

**post-condizioni:**

* [se gruppo = no] per i Turni con *turno.nomeGruppo* = nomeGruppo: *turno.gruppo =* gruppo, *turno.nomeGruppo* = “”
* [se è specificato nuovoNomeGruppo] per i Turni con *turno.nomeGruppo* = nomeGruppo: *turno.nomeGruppo* = nuovoNomeGruppo

**3b.1.** **modificaTurno( turno: Turno, pre-estensione?: ora, post-estensione?: ora, deadline?: data, data?: data, oraInizio?: ora, oraFine?: ora)**

**pre-condizioni:**

* turno non è ancora iniziato
* turno.*deadline* non è stata raggiunta

**post-condizioni:**

* [se è specificata una oraInizio] *t.oraInizio* = oraInizio
* [se è specificata una oraFine] *t.oraFine* = oraFine
* [se è specificata una pre-estensione] *t.oraInizio*  = pre-estensione
* [se è specificata una post-estensione] *t.oraFine*  = post-estensione
* [se è specificata una deadline] *t.deadline* = deadline
* [se è specificata una data] *t.data* = data

1. **modificaDeadline(deadline: data/ora)**

**pre-condizioni:**

* è in corso la definizione di una Disponibilità *d*

**post-condizioni:**

* [se è stata specificata una deadline] *d.deadline* = deadline

1. **mailConferma( personale: Personale, turno: Turno)**

**pre-condizioni:**

* turno non è ancora iniziato

**post-condizioni:**

Operazione di sistema, non sono necessarie post condizioni

**5a.1. creoGruppoPersonale(turno: Turno)**

**//TODO** da sistemare su mod dom nuovo concetto

**pre-condizioni:**

* turno.*deadline* non è stata raggiunta

**post-condizioni:**

* ListaGruppo *lg* creata
* *lg.turno* = turno

**5a.2. personaleInGruppo(personale: Personale, listaGruppo: ListaGruppo, capogruppo?: si/no)**

**pre-condizioni:**

* listaGruppo.*turno.deadline*  non è stata raggiunta

**post-condizioni:**

* *lg* **contiene** personale
* personale **è assegnato** a *lg.turno*
* [se è specificato capogruppo] *lg.capogruppo* = personale

**5a.3. mailAlGruppo(listaGruppo: ListaGruppo)**

**pre-condizioni:**

* listaGruppo.*turno.deadline*  non è stata raggiunta
* listaGruppo.*capogruppo* non è vuoto

**post-condizioni:**

* viene inviata una mail a listaGruppo.*capogruppo*