

# Contratti per lo UC “Gestire I Turni”

## Pre-condizione generale:

L’attore è identificato con un’istanza *Org di Organizzatore*

Esiste un Tabellone *tab*

## 1. creaTurno(data: Data, oraInizio: Numero, oraFine: Numero, luogo?:Testo,oraInizioAgg?:numero,oraFineAgg?:numero, isTurnoServizio: si/no)

### Pre-Condizioni:

--

### Post-Condizioni:

[se isTurnoServizio= no]

per ogni turno *tr* in *tab*:

tr.data , tr.oraInizio , tr.oraFine e tr.luogo è stato controllato che non si sovrappongono con data, oraInizio, oraFine, luogo

allora:

è stata creata un’istanza *tr* di Turno:

tr.data = data.

tr.oraInizio = oraInizio.

tr.oraFine = oraFine.

[se specificato un luogo] tr.luogo=luogo.

[se non è specificato un luogo] tr.luogo = cucina.

*tr* è inserito in *tab*.

altrimenti:

[se isTurnoServizio= si]

è stata creata un’istanza di *tr* di Turno

tr.data = data.

tr.oraInizio = oraInizio.

tr.oraFine = oraFine.

[se specificato un luogo] tr.luogo=luogo.

[se non è specificato un luogo] tr.luogo = cucina.

[se è stato specificato oraInizioAgg]tr.oraInizioAgg=oraInizioAgg.

[se è stato specificato oraFineAgg] tr.oraFineAgg=oraFineAgg.

*tr* è inserito in *tab*.

## 1a.1 modificaTurno(data: Data, oraInizio: Numero, oraFine:Numero,luogo?:Testo,oraInizioAgg?:numero, oraFineAgg?:numero)

### Pre-Condizioni:

Esiste turno *tr*  
è in corso la modifica di un turno *tr*

**Post-Condizioni:**

[se *tr.isTurnoServizio*=no]

per ogni turno *tr* in *tab*:

*tr.data* , *tr.oraInizio*,*tr.oraFine* e *tr.luogo* è stato controllato che diverso con *data*, *oraInizio*,  
*oraFine*, *luogo*

allora:

[se è specificato una *data*] *tr.data* = *data*.

[se è specificato un *oraInizio*] *tr.oraInizio* = *OraInizio*.

[se è specificato un *oraFine*] *tr.oraFine* = *OraFine*.

[se è specificato un *luogo*] *tr.luogo* = *luogo*.

*tr* aggiornato in *tab*

[se *tr.isTurnoServizio*=si]

[ se specificato *oraInizioAgg*] *tr.oraInizioAgg*=*oraInizioAgg*.

[se specificato *oraFineAgg*] *tr.oraFineAgg*=*oraFineAgg*.

[se è specificato una *data*] *tr.data* = *data*.

[se è specificato un *oraInizio*] *tr.oraInizio* = *OraInizio*.

[se è specificato un *oraFine*] *tr.oraFine* = *OraFine*.

[se è specificato un *luogo*] *tr.luogo* = *luogo*.

*tr* aggiornato in *tab*.

## 1b.1 cancellaTurno(*tr*:Turno)

**Pre-Condizioni:**

*tr* non ha disponibilità

**Post-Condizioni:**

*tr* è stato rimosso da *tab*

l'istanza *tr* è eliminata

## 1c.1 creaRipetizioneTurno(*tr*:Turno, *cadenza*: testo, *dataFineRipetizione*: numero)

**Pre-Condizioni:**

Esiste turno *tr* definito.

**Post-Condizioni:**

[se *tr.isTurnoServizio*=si]

è stata creata un'istanza *ripet* di Ripetizione.

per ogni *cadenza* fino a *dataFineRipetizione*:

viene creata un'istanza *trNuovo* di Turno

*trNuovo.orarioInizio*=*tr.oraInizio*

*trNuovo.oraFine*=*tr.oraFin*

trNuovo.luogo = tr.luogo  
 trNuovo.data viene modificata in accordo a cadenza  
 [se specificato tr.oraInizioAgg]trNuovo.oraInizioAgg= tr.oraInizioAgg  
 [se specificato tr.oraInizioAgg]trNuovo.oraInizioAgg= tr.oraInizioAgg  
 trNuovo **appartiene** a *ripet*  
*ripet* è stato inserito in *tab*  
 [se IsTurnoServizio=no]  
 per ogni *tr* **appartenente** a *tab*:  
     è stato controllato che per ogni *trNuovo* siano diversi data , oraInizio, oraFine , luogo  
     (viene verificato che per ogni trNuovo non si sovrappone con altri turni in tab)  
 allora:  
     è creata un'istanza *ripet* di Ripetizione.  
     per ogni cadenza fino a dataFineRipetizione:  
         viene creata un'istanza *trNuovo* di Turno  
         trNuovo.orarioInizio=tr.oraInizio  
         trNuovo.oraFine=tr.oraFine  
         trNuovo.luogo = tr.luogo  
         trNuovo.data viene modificata in accordo a cadenza  
         trNuovo **appartiene** a *ripet*  
         *ripet* è stato inserito in *tab*

## **1d.1 modificaRipetizioneTurno(tr:Turno, NuovadataFineRipetizione?: Data, nuovaCadenza?: Testo)**

### **Pre-Condizioni:**

esiste turno *tr* in *tab*  
 esistono *tr* che **appartiene** a un'istanza di *ripet*

### **Post-Condizioni:**

[se tr.isTurnoServizio=si]  
     [se è stata specificata una nuovaCadenza:]  
         [se è stata specificata una NuovaDataFineripetizione]  
             per ogni turno ripetuto *trp* **appartenente** a *ripet* con trp.data > tr.data  
             è stato aggiornato trp.data in base alla nuovaCadenza fino alla  
             NuovaDataFineRipetizione  
             sono state eliminate tutte le istanze di *trp* con trp.data >  
             NuovaDataFineripetizione  
         [se non è stata specificata una NuovaDataFineripetizione ]  
             per ogni turno ripetuto *trp* **appartenente** a *ripet* con trp.data > tr.data  
             è stato aggiornato trp.data in base alla nuovaCadenza  
     [se non è stata specificata una NuovaCadenza]  
         [se è stata specificata una NuovaDataFineRipetizione]

[se NuovaDataFineRipetizione < tr.DataFineRipetizione]  
sono state eliminate tutte le istanze di *trp* con *trp.data* >  
NuovaDataFineripetizione  
[se NuovaDataFineRipetizione > tr.DataFineRipetizione]  
sono state create istanze di *trNuovo* fino all NuovaDataFineRipetizione  
rispettando la cadenza  
i turni nuovi **appartengono** a *ripet*

[se *tr.isTurnoServizio*=no]

[se è stata specificata una nuovaCadenza:]

per ogni turno *tr* in *tab* per ogni turno ripetuto *trp* *trp.data* calcolato  
con *nuovaCadenza* deve essere diverso da *tr.data*

[se è stata specificata una NuovaDataFineripetizione]

per ogni turno ripetuto *trp* **appartenente** a *ripet* con *trp.data* > *tr.data*  
è stato aggiornato *trp.data* in base alla *nuovaCadenza* fino alla  
NuovaDataFineRipetizione  
sono state eliminate tutte le istanze di *trp* da *ripet* con *trp.data* >  
*NuovaDataFineripetizione*

[se non è stata specificata una NuovaDataFineripetizione]

per ogni turno ripetuto *trp* **appartenente** a *ripet* con *trp.data* > *tr.data*  
è stato aggiornato *trp.data* in base alla *nuovaCadenza*

[se non è stata specificata una *NuovaCadenza*]

[se è stata specificata una NuovaDataFineRipetizione]

[ se NuovaDataFineRipetizione < tr.DataFineRipetizione]

sono state eliminate tutte le istanze di *trp* con *trp.data* >  
NuovaDataFineripetizione

[se NuovaDataFineRipetizione > tr.DataFineRipetizione]

sono state create istanze di *trNuovo* fino all NuovaDataFineRipetizione  
rispettando la cadenza  
i turni nuovi **appartengono** a *ripet*

## 1e.1 cancellaRipetizioneTurno(tr:Turno, dataFineElim?: numero)

### **Pre-Condizioni:**

*tr* **appartiene** ad una ripetizione *ripet*

### **Post-Condizioni:**

[se è stata specificata dataFineElim]

per ogni istanza turno ripetuto *trp* in *ripet*, con *tr.data*≤*trp.data*

per ogni *trp* **appartenente** a *ripet* *trp.data*≤*dataFineElim*

*trp* non **appartiene** più a *ripet*.

*tr* è stato eliminato .

[se non è stata specificata dataFineElim]

per ogni istanza turno ripetuto *trp* in *ripet*, con *tr.data*≤*trp.data*

*tr* è stato eliminato

*trp* non **appartiene** più a *ripet*.

## 2. verificaDisponibilità()

**Pre-Condizioni:** --

**Post-Condizioni:** --

## 3. creaRaggruppamento(tr:Turno, altroTr:Turno)

### **Pre-condizioni:**

Esiste turno *tr* e altro turno *altroTr*

*tr* diverso da *altroTr*

*tr.isTurnoServizio*=no e *altroTr.isTurnoServizio*=no

### **Post-condizioni:**

[se *tr.isTurnoServizio*=no e *altroTr.isTurnoservizio*=no]

è stata creata un'istanza *rag* di Raggruppamento

*tr* **appartiene** a *rag*

*altroTr* **appartiene** a *rag*

*tr* e *altroTr* vengono rimossi da *tab*

*ragg* è stato inserito in *tab*

### 3a.1 modificaRaggruppamento(ragg:Raggruppamento,nuovaDataTurno:data)

**Pre-condizioni:**

esiste *ragg* appartenente a *tab*

**Post-condizioni:**

per ogni Turno *t* **appartenente** a *ragg*

per ogni turno *tr* in *tab* *tr.data* è stato controllato non sia uguale a nuovaDataTurno  
*t.data*= *nuovaDataTurno*

### 3b.1 EliminaRaggruppamento(rag:Raggruppamento)

**Pre-condizioni**

esiste *ragg* appartenente a *tab*

**Post-condizioni:**

per ogni *tr* **appartenente** a *rag*:

*tr* non appartiene più a *rag*

*tr* è stato eliminato

*rag* è stato rimosso da *tab*

*rag* è eliminato

### 3c.1 creaRipetizioneRaggruppamento(ragg: Raggruppamento, cadenza: testo, dataFineRipetizione: numero)

**Pre-condizioni:**

esiste *ragg* appartenente a *tab*

**Post-condizioni:**

per ogni *tr* **appartenente** a *tab* e per ogni turno raggruppato *trg* **appartenente** a *ragg*:

di cui si devono creare le istanze in *ripet*, *trg.data* calcolato per ogni cadenza fino alla *dataFineRipetizione* non è stata trovata corrispondenza con *tr.data*

allora:

è stata creata un'istanza *ripet* di Ripetizione.

sono state create istanze turno ripetuto *trp* in *ripet* che seguono la cadenza e la *dataFineRipetizione*.

per ogni *trp* in *ripet*:

*trp.OraInizio* = *trg.oraInizio*

*trp.OraFine* = *trg.oraFine*

*trp.location* = *trg.location*

*ragg* **appartiene** a *ripet*

*ripet* **appartiene** a *tab*

### **3d.1 modificaRipetizioneRaggruppamento(ragg: Raggruppamento, cadenza: testo, dataFineRipet: numero)**

***Pre-condizioni:***

Esiste turno *tr*

*tr* appartiene a *ragg*

***Post-condizioni:***

Per ogni istanza *tr* **appartenente** a *ripet*:

*tr* viene modificato in accordo alla nuova cadenza e dataFineRipetizione.

### **3e.1 eliminaRipetizioneRaggruppamento(ragg: Raggruppamento, dataInizioElim:numero, dataFineElim: numero)**

***Pre-condizioni:***

Esiste Turno *tr*

*tr* appartiene a *ragg*

*ragg* appartiene a *ripet* di Ripetizione

***Post-condizioni:***

per ogni istanza *ragg* in *ripet*

per ogni istanza *tr* **appartenente** a *ragg*, con *tr.data* ≥ *dataInizioElim* e *tr.data* ≤ *dataFineElim*:

*tr* non appartiene più a *ragg*

*tr* è stato eliminato

*ragg* è eliminato

*ripet* è stato rimosso da tab

*ripet* è eliminato

