# **Contratti per lo UC “Gestire i turni”**

**Pre-condizione generale:** l’attore è identificato come un’istanza *or* di Organizzatore, inoltre deve essergli già assegnato almeno un’istanza *ev* Evento.  
  
**1.ConsultaEventi()  
  
Pre-condizione:  
Post-condizione:**  
   
**2.CreaBlocco(evento: Evento)  
Pre-condizione:**   
**Post-condizione:** -è stata creata un’ istanza *b* di Blocco   
 -*or* è proprietario di *b*

**2a.1 EliminaBlocco(blocco: Blocco)  
Pre-condizione:**   
**Post-condizione:**se *or* è il proprietario di blocco, elimina l’istanza blocco   
**2a.2 EliminaBlocco(evento: Evento)  
Pre-condizione:**   
**Post-condizione:**-Ogni istanza Turno appartenente ad un’istanza Blocco in cui *or* è proprietario tali per cui Blocco.evento == evento, vengono eliminate.  
-Tutte le istanze Blocco in cui *or* è proprietario tali per cui Blocco.evento == evento vengono eliminate  
**2b.1 ModificaBlocco(blocco: Blocco)  
  
Pre-condizione:**   
**Post-condizione:** - è in corso la definizione dell’istanza blocco.  
 **3.CreaTurnoPreparatore(data: Date, luogo: testo, fasciaOraria: testo, numeroPersonale: numero, scadenza?: Date, stato?: bool)  
Pre-condizione:**   
**Post-condizione:**- è stata creata un’istanza t di TurnoPreparatore  
-*or* è proprietario di *t***3.1CreaTurnoServizio(servizio: Servizio, data: Date, luogo: testo?, fasciaOraria: testo, numeroPersonale: numero, scadenza?: Date, stato?: bool)  
Pre-condizione:**   
**Post-condizione:**- è stata creata un’istanza t di TurnoServizio  
-*or* è proprietario di *t*  
**3a.1 CercaTurni(tipo: testo, dataInizio?: Date, dataFine?: Date, giornoSettimana?: testo, luogo?: testo, fasciaOraria?: testo, ricorrenza?: Ricorrenza)  
Pre-condizione:**esiste almeno un’ istanza Turno tale per cui Turno.tipo == tipo && (dataInizio == null || Turno.data >= dataInizio) && (dataFine == null || Turno.data <= dataFine) && (giornoSettimana == null || Turno.data contenga giornoSettimana) && (luogo == null || Turno.luogo == luogo) && (fasciaOraria == null || Turno.fasciaOraria == fasciaOraria) && (ricorrenza == null || Turno.ricorrenza ==ricorrenza)  
  
**Post-condizione:****3b.1 EliminaTurno(turno: Turno)  
Pre-condizione:**  
**Post-condizione:**se *or* è il proprietario dell’istanza turno, l’istanza turno viene eliminata**3b.2 EliminaTurni(turni: Turno[])  
Pre-condizione:   
Post-condizione:**per ogni istanza Turno appartenente a turni:  
 -se *or* è il proprietario dell’istanza Turno, l’istanza Turno viene eliminata  
 **3c.1ModificaTurno(turno: Turno, tipo?: testo, data?: Date, luogo?: testo, fasciaOraria?: testo, numeroPersonale?: numero, scadenza?: Date, stato?: bool, servizio?: Servizio)  
Pre-condizione:**  
**Post-condizione:**-se *or* è il proprietario dell’istanza turno:-[se specificato tipo] turno.tipo = tipo  
 -[se specificato data] turno.data = data  
 -[se specificato luogo] turno.luogo = luogo  
 -[se specificato fasciaOraria] turno.fasciaOraria = fasciaOraria  
 -[se specificato numeroPersonale] turno.numeroPersonale = numeroPersonale  
 -[se specificato scadenza] turno.scadenza = scadenza  
 -[se specificato stato] turno.stato = stato  
 -[se specificato servizio && turno è un’istanza TurnoServizio] turno.servizio = servizio  
  
**3c.2 ModificaTurno(turni: Turni[], tipo?: testo, data?: Date, luogo?: testo, fasciaOraria?: testo, numeroPersonale?: numero, scadenza?: Date, stato?: bool, servizio?: Servizio)  
  
Pre-condizione:**   
**Post-condizione:**   
per ogni istanza Turno appartenente a turni:  
 -se *or* è il proprietario dell’istanza Turno   
 -[se specificato tipo] Turno.tipo = tipo  
 -[se specificato data] Turno.data = data  
 -[se specificato luogo] Turno.luogo = luogo  
 -[se specificato fasciaOraria] Turno.fasciaOraria = fasciaOraria  
 -[se specificato numeroPersonale] Turno.numeroPersonale = numeroPersonale  
 -[se specificato scadenza] Turno.scadenza = scadenza  
 -[se specificato stato] Turno.stato = stato  
 -[se specificato servizio && Turno è un’istanza TurnoServizio] Turno.servizio = servizio  
   
  
**3d.1 AggiungiTurno(turno: TurnoPreparatore)**  
**Pre-condizione:** è in corso la definizione di un’ istanza Blocco  
**Post-condizione:** -l’istanza turno viene aggiunta all’istanza Blocco  
  
**3e.1 RimuoviTurno(turno: TurnoPreparatore)**  
**Pre-condizione:** è in corso la definizione di un’ istanza Blocco  
**Post-condizione:** - l’istanza Turno viene rimossa dall’istanza Blocco  
**3f.1 EsciBlocco()**  
**Pre-condizione:** è in corso la definizione di un’ istanza Blocco  
**Post-condizione:**non è piu in corso la definizione di un’istanza Blocco.  
  
**4.ApplicaRicorrenza(turno: Turno, start: Date, end: Date, day: String)  
Pre-condizione:**   
**Post-condizione:**-Viene creata un’istanza *r* di Ricorrenza:  
 r.start = start  
 r.end = end  
 r.day = day-Per un n calcolato sulla base dell’istanza r:  
 se *or* è il proprietario dell’istanza turno crea un’istanza *t* di Turno tale che:  
 -t.tipo = turno.tipo  
 -t.data = una data calcolata sulla base di n e dell’istanza ricorrenza  
 -t.luogo = turno.luogo  
 -t.fasciaOraria = turno.fasciaOraria  
 -t.numeroPersonale = turno.numeroPersonale

-t.scadenza = turno.scadenza  
 -t.stato = turno.stato

-t.ricorrenza = r

-turno.ricorrenza = r

**4.1ApplicaRicorrenza(turni: Turno[],Turno, start: Date, end: Date, day: String)**-Viene creata un’istanza *r* di Ricorrenza:  
 r.start = start  
 r.end = end  
 r.day = day-Per ogni istanza Turno appartenente a turni:  
 -Turno.ricorrenza = r-Per un n calcolato sulla base dell’istanza ricorrenza:  
 -se *or* è il proprietario dell’istanza Turno crea un’istanza *t* di Turno tale che:  
 -t.tipo = Turno.tipo  
 -t.data = una data calcolata sulla base di n e dell’istanza ricorrenza  
 -t.luogo = Turno.luogo  
 -t.fasciaOraria = Turno.fasciaOraria  
 -t.numeroPersonale = Turno.numeroPersonale

-t.scadenza = Turno.scadenza  
 -t.stato = Turno.stato

-t.ricorrenza = r

**5.InformaPersonale()  
Pre-condizione:  
Post-condizione:**