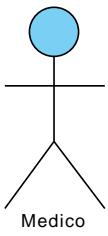


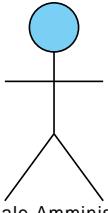
Powered By Visual Paradigm Community Edition



Medico



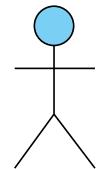
Calcolo Itinerario



Personale Amministrativo



Calcolo specializzazioni



Personale ufficio prenotazioni



GEstione Pazienti e ricoveri

Powered By Visual Paradigm Community Edition

Vincoli esterni

Funzioni ausiliarie

```
sovraposti(i1 : DataOra, i2 : DataOra, f1 : DataOra, f2 : DataOra) : Booleano
    precondizioni:
        Deve essere vero che i1 <= f1 e che i2 <= f2.
    postcondizioni:
        result = true se gli intervalli si sovrappongono, cioÃ‡ se i1 <=
f2 e i2 <= f1;
        altrimenti result = false.
```

[V_Medico_non_Paziente_di_se_stesso]

Un medico non puo essere paziente di se stesso

Per ogni p:Persona tale che p:Medico e p:PazienteInterno non deve esistere r: Ricovero

(p, r):med_ric e (p, r):paz_ric

[V_PazienteInterno_PazienteEsterno_non_contemporanei]

Lo stesso paziente non deve essere Paziente Intero e Paziente Esterno nello stesso momento

Per ogni p:Paziente tali che p:PazienteInterno e p:PazienteEsterno non devono esistere pe:PrestazioneMedica, r:Ricovero, r:Dimissione,

(p, pe):PM_PE e (p, r):paz_ric con sovrapposti(r.data_ricovero, pe.data_richiesta, r.data_dimissione, pe.data_richiesta)

[V.Data_Ricovero_prima_Data_Dimissione]

La data del ricovero deve essere prima della data di dimissione

Per ogni r:Ricovero tale che r:Dimissione non deve essere che r.data_ricovero >= r.data_dimissione

[V_Due_ricoveri_non_Sovraposti_stesso_Letto]

Due ricoveri diversi non possono avere assegnato lo stesso letto se sono nello stesso arco di tempo

Per ogni l:Letto, sia r1:Ricovero e r2:Ricovero tali che r1:Dimissione e r2:Dimissione dove r1 != r2,

(l, r1):ric_let e (l, r2):ric_let non deve essere vero che sovrapposti(r1.data_ricovero, r1.data_dimissione, r2.data_ricovero, r2.data_dimissione)

disponibili(d : Data) : Letto[0..*]:

- precondizioni:
- Nessuna

- postcondizioni:
- L'operazione non modifica il livello estensionale
- Sia L l'insieme di tutti i l:Letto, tali che:
- Sia R l'insieme di tutti i r:Ricovero tali che (r, l):ric_let

```
- Non esista un r in R tale che sovrapposti(r.data_recupero,
r.data_dimissione, d, d)
- result = L
```

Calcolo Itinerario

```
calcolo_itinerario(m: Medico) : {Intero >= 0, Intero >= 0} :

    precondizioni:
        - nessuna

    postcondizioni:
        - Sia R l'insieme di tutti i r:Ricovero, tali che (r,
m):med_ric
            - Per ogni r in R, sia L l'insieme di tutti i l:Letto, tali
che (r, l):ric_let
                - Sia S l'insieme di tutte le s:Stanza, tali che st partecipa
al link dell'associazione con un l in L
                - Sia Res l'insieme delle coppie (p, s), dove per ogni s:Stanza
in S:
                    - p è l'attributo s.piano
                    - s è l'attributo s.settore
                    - result = Res
```

Calcolo Specializzazione

```
calcola_specializzazione(s:Specializzazione): Medico [0..*]():

    - precondizioni:
        - Nessuna

    - postcondizioni
        - Sia M_primaria l'insieme di tutti i m:Medico, tale che (s,
m):primaria
        - Sia M_secondaria l'insieme di tutti i med:Medico, tale che (s,
m):secondaria
        - Se M_primaria è un insieme vuoto allora result = M_secondaria
altrimenti result = M_primaria
```

Gestione pazienti e ricoveri:

```
registrazione_ricovero(data_ricovero : Data, int : Interno, med :
Medico, l : Letto):
    - precondizioni:
        - l deve essere presente nell'insieme
l.disponibili(data_ricovero)

    - postcondizioni:
        - viene creati e restituiti i seguenti link/oggetti:
            - r:Ricovero, tale che:
                - r.data_ricovero = data_ricovero
            - (r, pl):ric_let
            - (int, r):paz_ric
            - (r, m):med_ric
```

```

registra_dimissione(r : Ricovero, data_dimissione : Data) : Data:
    - precondizioni:
        - Nessuna

    - postcondizioni:
        - viene creato e restituito l'attributo r.data_dimissione =
data_dimissione


    registra_paziente(nome : Stringa, cognome : Stringa, data_nascita :
Data, email : Email [0..*], telefono : NumTel[0..*], indirizzo :
Indirizzo, prest : Prestazione[0..1]) :
    - precondizioni:
        - Vengono creati e restituiti i seguenti link/oggetti:
            - p:Paziente, tale che:
                - p.nome = nome
                - p.cognome = cognome
                - p.data_nascita = data_nascita
                - p.email = email
                - p.telefono = telefono
                - p.indirizzo = indirizzo
            - (p, pr):p_pr

```