

RAFFINAMENTO REQUISITI - Progetto Out !

RAFFINAMENTO REQUISITI

1. REQUISITI sulle Sedi:

- 1.1 Ogni sede è provvista di sale distinte , nella quale si può effettuare uno spettacolo
- 1.2 nome: stringa
- 1.3 indirizzo: indirizzo

2. REQUISITI sulle Sale:

- 2.1 Ogni sala distinta è appartenenete ad una sede, nella quale si può effettuare uno spettacolo
- 2.2 posti: int>=0

3. REQUISITI sui Posti:

- 3.1 Ogni posto p associato ad una sala distinta
- 3.2 numerofila: stringa
- 3.3 colonna: stringa

4. REQUISITI sugli Spettacoli:

- 4.1 Uno o più di uno spettacolo viene tenuto nella stessa sala o in più sale per più volte nella stessa sede
- 4.2 titolo: stringa
- 4.3 tipologia = Tipologia = { concerto, rappresentazione teatrale, film }
- 4.4 genere: stringa
- 4.5 è di interesse conoscere gli artisti che effettuano la performance
- 4.6 conoscere oltre le date, gli orari e le sedi con la sala in cui viene rappresentato
- 4.7. si deve anche modellare la possibilità di avere degli 'SPETTACOLI ITINERANTI'

5. REQUISITI sugli ARTISTI:

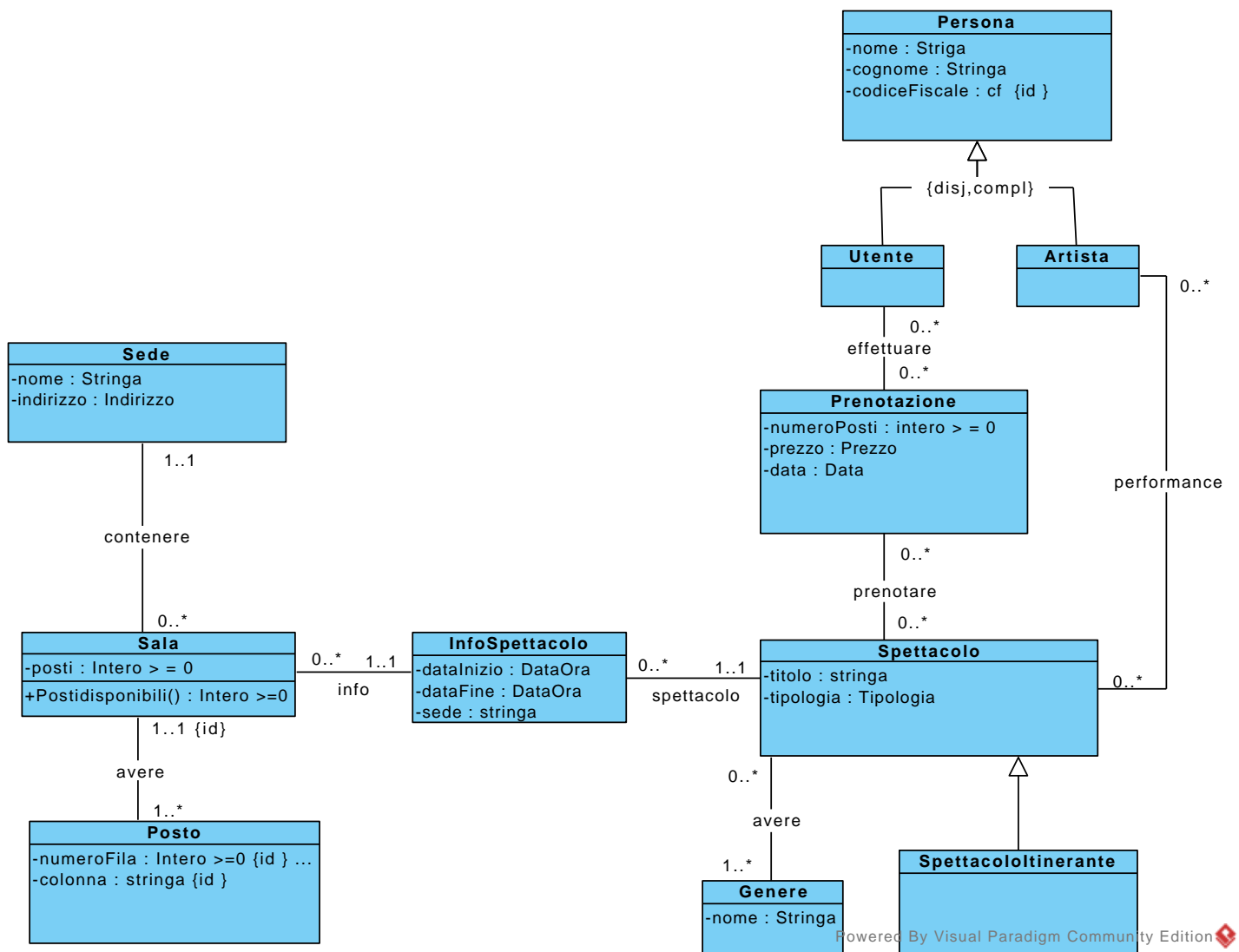
- 5.1

6. REQUISITI sugli UTENTI:

- 6.1 nome : stringa
- 6.2 cognome : stringa
- 6.3 cf: CodiceFiscale {id}

7. REQUISITI sulle PRENOTAZIONI:

- 7.1 una prenotazione deve essere associata ad un dato spettacolo
- 7.2 numeroPosti : int>=0
 - 7.2.1 i posti non sono liberi, quindi il sistema deve assegnarli durante la prenotazione
 - 7.2.1.1 i posti prima di essere assegnati vanno verificati, ovvero se ci sono posti disponibili a sufficienza



SPECIFICA DEI TIPI DATO - Progetto Out !

SPECIFICA DEI TIPI DATO:

1. indirizzo :

stringa alfanumerica che segue il formato standard d'identificazione univoca la posizione secondo la sintassi:

Via NumeroCivico, CAP Città (SiglaProvincia), Nazione

dove:

Via: solo lettere, numeri e caratteri speciali ammessi (' , . , - , spazio)

NumeroCivico: numerico o alfanumerico (es. "12A")

CAP: 5 cifre numeriche (es. 00184)

Città: solo lettere e spazi (iniziale maiuscola)

SiglaProvincia: due lettere maiuscole opzionali racchiuse tra parentesi

Nazione: nome completo della nazione (iniziale maiuscola)

2. cf:

stringa alfanumerica secondo standard che segue la seguente sintassi

3. Prezzo:

tipologia enumerica -> Prezzo = {Prezzo Pieno, Prezzo Ridotto}

4. Data:

stringa secondo standard GG/MM/AAAA -> G = giorno, M = mese, A = anno

5. Tipologia:

tipologia enumerica -> Tipologia = { concerto, rappresentazione teatrale, film }

SPECIFICA DELLE CLASSI - Progetto Out!

Specifica della classe Prenotazione:

[V.Prenotazione.DataPrenotazioneDataSpettacolo]

Sia u:Utente che è sopraclasse p:Persona
Sia pr:Prenotazione tale che (u,pr):effettuare
Sia s:Sala e inf:InfoSpettacolo tale che (s,inf):info
Sia sp:Spettacolo tale che (pr,sp):prenotare e (inf,sp):spettacolo

Sia prenot = (pr,sp):prenotare

Quindi sia pr.data == inf.dataInizio, in caso non sia coerente non è possibile che:

u possa avere un link prenot

Specifica della classe Spettacolo:

spettacoloSovrapposto(d:DataOra,f:DataOra)-1> Boolean:

PreCondizione:

nessuna

PreCondizione:

- Non modifica il livello estensionale dei dati
- l'operazione Result è così definita:

Sia s:Sala e inf:InfoSpettacolo tale che (s,inf):info
Sia spet: Spettacolo tale che (inf,spet):spettacolo

Sia t:DataOra, valore istante di tempo casuale

Nel caso:

1. $d \geq t \leq f \rightarrow$ l'istante t si sovrappone all'intervallo in cui si tiene spet , quindi Result = True

2. $t \leq d$ oppure $t \geq i \rightarrow$ l'istante t non si sovrappone all'intervallo in cui si tiene spet quindi Result = False

Specifica della classe Sala:

Postidispribili(d:DataOra) -> Intero ≥ 0 :

PreCondizione:

nessuna

PostCondizione:

- Non modifica il livello estensionale dei dati
- L'operazione Result è così definita:

Sia u:Utente sopraclasse si p:Persona

```

    Sia pr:Prenotazione tale che (u,pr):effettuare
    Sia sp:Spettacolo tale che (pr,sp):prenotare
    Sia s:Sala e p:Posto tale che (s,p):avere
    Sia inf:InfoSpettacolo tale che (s,inf):info ed (inf,sp):spettacolo

    Sia P l'insieme di tutti i p:Posto tali che:
        (p,s):avere
        (s,inf):info

        info.dataInizio <= d <= info.dataFine

    Result = |P|

```

Specifica della classe Utente:

[V.Utente.NostessoSpettacolo]

```

    Sia u:Utente sopraclasse p:Persona
    Sia pr:Prenotazione tale che (u,pr):effettuare
    Sia sp:Spettacolo tale che (pr,sp):prenotare
    s:Sala e inf:InfoSpettacolo tale che (s,inf):info ed (inf,sp):spettacolo

    efffet = (u,pr):effettuare
    infSpet = (s,inf):info

    Esistano f1:inf.dataInizio, f2:inf.dataFine, d1.pr.data e d2.pr.data.
    Sia d1 != d2, che abbiano i seguenti link:

    (s,inf): info, (inf,f1):spettacolo
    (s,inf):info, (inf,f2):spettacolo
    (u,pr):effettuare, (pr,d1):prenotare
    (u,pr):effettuare, (pr,d2):prenotare

    Non siano quindi sovrapposti (f1.dataInizio, f2.dataFine, d1.data,d2.data)

```