Travel To The Moon Passo A.14

Alessandro Gautieri

April 2024

Classe Ufficio Marketing

Operazioni

calcolaEtaMedia(DataInizio:Data, DataFine:Data): Float

Pre-condizione:

- Esiste almeno una crociera c coinvolta nel link $(this, c)$: *mark_croc*.
- Esiste almeno un link $(c, UfficioPrenotazione)$: *Effettua_Prenotazione*.

Post-condizione:

- Nessun side-effect.
- Sia $InsiemePrenotazioni$ l'insieme dei link $(c, UfficioPrenotazione)$: *Effettua_Prenotazione* tali che:
  - $c.Itinerario.Destubazione.Continente \neq$ "$Europa$"
  - $DataInizio < c.inizio < fine < DataFine$
- Sia $n = |InsiemePrenotazioni|$.
- Per ogni prenotazione in $InsiemePrenotazioni$:
  - Sia $cl = prenotazione.Cliente$.
  - $SommaEta+ = cl.Eta$.
- Ritorna $\frac{SommaEta}{n}$.

calcolaDestinazioniGettonate(DataInizio:Data, DataFine:Data): Float

Pre-condizione:

- Esiste almeno un link $(this, Crociera)$: *mark_croc*.

Post-condizione:

- Nessun side-effect.
- Sia $InsiemeCrociere$ l'insieme dei link $(this, Crociera)$: *mark_croc*.
- Sia $InsiemeLunaDiMiele$ l'insieme formato dai link tali che Crociera è del tipo LunaDiMiele.
- Sia $InsiemeFamiglie$ l'insieme formato dai link tali che Crociera è del tipo PerFamiglie.
- Sia $InsiemeDestinazioni$ l'insieme delle Crociera.Itinerario.Destinazione.
- Sia *Gettonate* un insieme vuoto.
- Per ogni *dest* in $InsiemeDestinazioni$:
  - $cc1 = 0$
  - $cc2 = 0$
  - Per ogni *lunMiel* in $InsiemeLunaDiMiele$:
    - Se $lunMiel.Itinerario.Destinazione = dest$:
      - $cc1 += 1$
  - Per ogni fam in $InsiemeFamiglie$:
    - Se $fam.Itinerario.Destinazuione = dest$:
      - $cc2 += 1$
  - Se $cc1 \geq 10$ oppure $cc2 \geq 15$:
    - $Gettonate.add(dest)$
- $|Gettonate| : |InsiemeDestinazioni| = x : 100$.
- Ritorna x.

Classe Ufficio Prenotazione

Operazioni

effettua_prenotazione(): Booleano

Pre-condizione:

- Sia $DataOrario$ il giorno e l'istante in cui *this* sta chiamando la funzione.
- Esiste almeno una classe di tipo crociera c con $c.inizio > DataOrario$ e $c.fine = Null$.

Post-condizione:

- Nessun side-effect.
- Sia c la crociera che soddisfa le pre-condizioni per la quale l'utente vuole effettuare una prenotazione.
- Sia $PrenotazioneCrociera$ l'oggetto $(this, c)$.
- Ritorna True:
  - Se $this.NumPosti < c.CalcolaPostiRimanenti()$.
- Ritorna False:
  - Se $this.NumPosti > c.CalcolaPostiRimanenti()$.

Classe Crociera

Operazioni

fine() : Data

Pre-condizione: Nessuna.
Post-condizione:

- Sia i l'Itinerario tale che $(this, i) : croc\_itin$.
- Sia $x : \Delta DataOra$ il valore dell'attributo 'istante' dell'unico link di associazione *arrivo_itin* in cui i è coinvolto.
- $result = this.inizio + x.giorno$ "giorni".

CalcolaPostiRimanenti() : int¿0

Pre-condizione: Esiste almeno un collegamento $(this, Ufficio\_Prenotazioni) : Effettua\_Prenotazione$.
Post-condizione:

- Nessun side-effect.
- Sia $NumPostiOccupatiPrenotazioni$ un contatore $= 0$.
- Sia $InsiemePrenotazioni$ l'insieme dei collegamenti $(this, Ufficio\_Prenotazione) : Effettua\_Prenotazione$.
- Per ogni prenotazione i in $InsiemePrenotazioni$:
  - $NumPostiOccupatiPrenotazioni = prenotazione.NumPosti$.
- Ritorna $NumPostiOccupatiPrenotazioni$.

Vincoli esterni

[V.Crociera.date]

Per ogni oggetto $c : Crociera$ deve essere: $c.inizio \leq c.fine$.

Classe Itinerario

Vincoli esterni

[V.Itinerario.arrivo_dopo_ultima_tappa]

Per ogni $i : Itinerario$ siano:

- T l'insieme delle istanze $'t'$ di classe Tappa tali che $(i, t) : itin\_tappa$.
- x il valore dell'attributo 'istante' dell'unico link di associazione *arrivo_itin* in cui $'i'$ è coinvolto.

Per ogni $t : T$, deve essere: $t.ripartenza < x$.

[V.Itinerario.prima_tappa_dopo_partenza]

Per ogni $i : Itinerario$ siano:

- T l'insieme delle istanze $'t'$ di classe Tappa tali che $(i, t) : itin\_tappa$ e $t.arrivo.giorno = 1$.
- x il valore dell'attributo 'istante' dell'unico link di associazione *partenza_itin* in cui $'i'$ è coinvolto.

Per ogni $t : T$, deve essere $t.arrivo.orario > x$.

[V.Itinerario.arrivo_dopo_partenza_se_senza_tappe]

Per ogni $i : Itinerario$ tale siano:

- T l'insieme delle istanze $'t'$ di classe Tappa tali che $(i, t) : itin\_tappa$.
- x il valore dell'attributo 'istante' dell'unico link di associazione *partenza_itin* in cui $'i'$ è coinvolto.
- y il valore dell'attributo 'istante' dell'unico link di associazione *arrivo_itin* in cui $'i'$ è coinvolto.

Se $T = \emptyset$ e $y.giorno = 1$, allora deve essere $x < y.orario$.
Per ogni $t : T$, deve essere $t.arrivo.orario > x$.

Classe Tappa

Operazioni

n_ordine() : Intero ¿ 0

Pre-condizione: Nessuna.
Post-condizione:

- Nessun side-effect.
- Sia i l'itinerario della tappa *this*.
- Sia $TappePrecedenti$ l'insieme degli oggetti $t : Tappa$ tali che:
  - $(i, t) : itin\_tappa$
  - $t.arrivo < this.arrivo$
- $result = |TappePrecedenti| + 1$.

Vincoli esterni

[V.Tappa.date]

Per ogni istanza $t : Tappa$, deve essere:

$$t.arrivo < t.ripartenza$$

Specifiche dei Tipi di Dato

TipoDestinazione

$$TipoDestinazione = \{romantica, divertente\}$$

DeltaDataOra

$$DeltaDataOra = \left(\begin{array}{l} giorno : Intero > 0, \\ orario : Ora \end{array} \right)$$

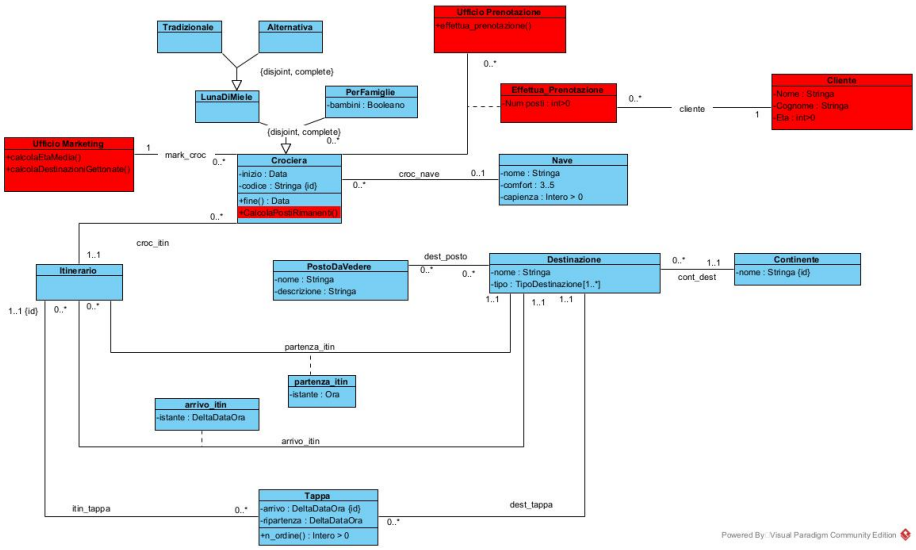
Operazioni del tipo di dato DeltaDataOra:

- $<(x : DeltaDataOra, y : DeltaDataOra) : Booleano$

Pre-condizione: Nessuna.
Post-condizione:

- $result = true$ se e solo se:
  - $(x.giorno < y.giorno)$ oppure
  - $(x.giorno = y.giorno$ e $x.orario < y.orario)$

UML


Tradizionale
Alternativa
{disjoint, complete}
LunaDiMiele
PerFamiglie
-bambini : Booleano
{disjoint, complete}
Ufficio Prenotazione
+effettua_prenotazione()
0..*
Effettua_Prenotazione
-Num.posti : int>0
0..*
cliente
1
Cliente
-Nome : Stringa
-Cognome : Stringa
-Eta : int>0
Ufficio Marketing
+calcolaEtaMedia()
+calcolaDestinazioniGettonate()
1
mark_croc
0..*
Crociera
-inizio : Data
-codice : Stringa {id}
+fine() : Data
#CalcolaPostiRimanenti()
croc_nave
0..1
0..*
Nave
-nome : Stringa
-comfort : 3..5
-capienza : Intero > 0
0..*
croc_itin
1..1
Itinerario
1..1 {id}
0..*
0..*
PostoDaVedere
-nome : Stringa
-descrizione : Stringa
dest_posto
0..*
0..*
Destinazione
-nome : Stringa
-tipo : TipoDestinazione[1..*]
1..1
1..1
1..1
0..*
1..1
cont_dest
Continente
-nome : Stringa {id}
partenza_itin
partenza_itin
-istante : Ora
arrivo_itin
-istante : DeltaDataOra
arrivo_itin
itin_tappa
0..*
Tappa
-arrivo : DeltaDataOra {id}
-ripartenza : DeltaDataOra
+n_ordine() : Intero > 0
dest_tappa
0..*
Powered By: Visual Paradigm Community Edition