

Classe Organizzatore

```
organizza_viaggio(a:Attivita_Composta,v:Viaggio):viag_comp
    precondizioni:
        nessuna
    postcondizioni:
        dati a:Attivita_Composta e v:Viaggio
        result = (a,v):viag_comp
```

Classe Viaggio

```
[v.minMinoreMax]
    Per ogni this sia min_part < max_part
```

Classe Semplice

```
    Per ogni this sia inizio < fine
```

Classe Spostamento

```
    Per ogni this siano l1:Luogo e l2:Luogo dati i link
    (this,l1):l_partenza
    e (this,l2):l_arrivo diversi tra loro
```

Classe att_comp

```
[v.viaggiNonSovrapposti]
    Dati v:Viaggio, ac:Attivita_Composta e a:Attivita sia
    (ac,v):viag_comp
        per ogni link (this,u):partecipanti siano (u,v):partecipa e
        (ac,a):this
```

Classe Attivita_Composta

```
    Per ogni link (this,a):att_comp siano a.inizio e a.fine non
    sovrapposti con gli altri a appartenenti ai
```

Operazione ausiliaria:

```
sovraposti...
```