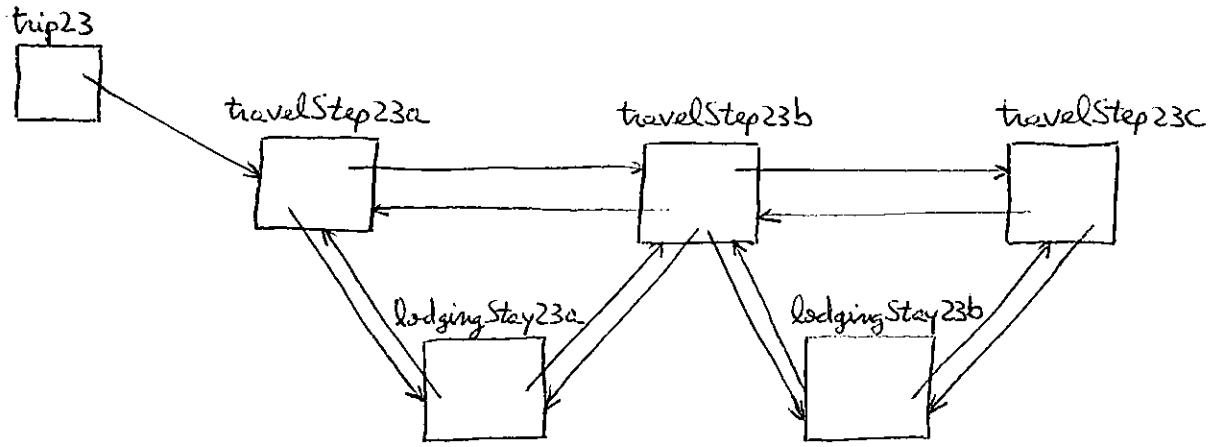
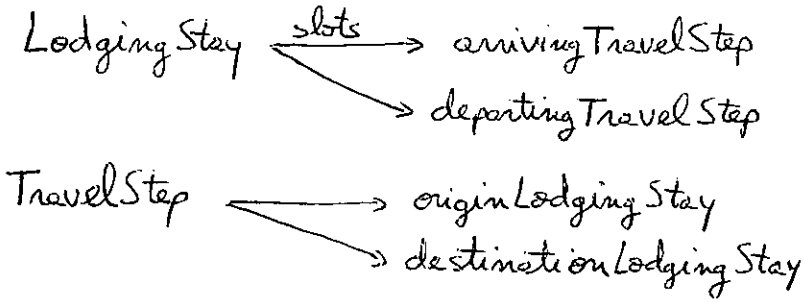
Frames – an example for planning a trip

We start with a general sketch of the basic structure of a trip.



There are two main frames: Trip and TravelStep. A Trip will have a sequence of TravelSteps. A TravelStep usually terminates with a LodgingStay, except when there are two travel steps in a single day or when we reach the last step of the trip.



The generic frame Trip and an instance of it may look as following:

```
(Trip
   <: FirstStep   TravelStep>
   <: Traveler    Person>
   <: BeginDate   Date>
   <: EndDate     Date>
   <: TotalCost   Price>
)
```

```
(trip23
   <: INSTANCE-OF  Trip>
   <: FirstStep    travelStep23a>
   <: Traveler     davidF>
   <: BeginDate    23/11/19>
   <: EndDate      25/11/19>
   <: TotalCost    3400RON>
)
```

The frames TravelStep and LodgingStay share some properties that we group in the more general frame TripPart:

```
(TripPart
    <: BeginDate   Date>
    <: EndDate   Date>
    <: Cost   Price>
    <: PaymentMethod   FormOfPayment>
)
(LodgingStay
    <: IS-A   TripPart>
    <: Place   City>
    <: LodgingPlace   LodgingPlace>
    <: ArrivingTravelStep   TravelStep>
    <: DepartingTravelStep   TravelStep>
)
(TravelStep
    <: IS-A   TripPart>
    <: Origin   City>
    <: Destination   City>
    <: OriginLodgingStay   LodgingStay>
    <: DestinationLodgingStay   LodgingStay>
    <: Means   FormOfTransportation>
    <: DepartureTime   Time>
    <: ArrivalTime   Time>
    <: NextStep   TravelStep>
    <: PreviousStep   TravelStep>
)
```

Default fillers:

```
(TripPart
    <: PaymentMethod   visaCard>
...)
(TravelStep
    <: Means   airplane>
    <: PaymentMethod   masterCard>
...)
```

Notations: - if x is an individual frame and y is a slot, then xy refers to the filler of the slot in that frame;
- SELF is a reference to the current frame.

The filler of the slot : Origin from TravelStep can be calculated as following:

```
(TravelStep
   <: Origin [IF-NEEDED
                { if no SELF : PreviousStep then otopeni;
                     else SELF : PreviousStep : Destination;
                }
              ] > ...)

(Trip
  <: TotalCost [IF-NEEDED
                 { result ← 0
                   x ← SELF : FirstStep;
                  repeat
                   if exists x : NextStep then
                        { result ← result + x : Cost;
                          if exists x : DestinationLodgingStay then
                            { result ← result + x : DestinationLodgingStay : Cost;
                          x ← x : NextStep;
                        }
                   else return result + x : Cost;
                 }
               ] > ...)

(TravelStep
  <: NextStep [IF-ADDED
                { if SELF : EndDate ≠ SELF : NextStep : BeginDate then
                    { SELF : DestinationLodgingStay ←
                      SELF : NextStep : OriginLodgingStay ←
                        new LodgingStay
                          with : BeginDate = SELF : EndDate;
                          with : EndDate = SELF : NextStep : BeginDate;
                          with : ArrivingTravelStep = SELF;
                          with : DepartingTravelStep = SELF : NextStep;
                    }
                }
              ] > ...)
```

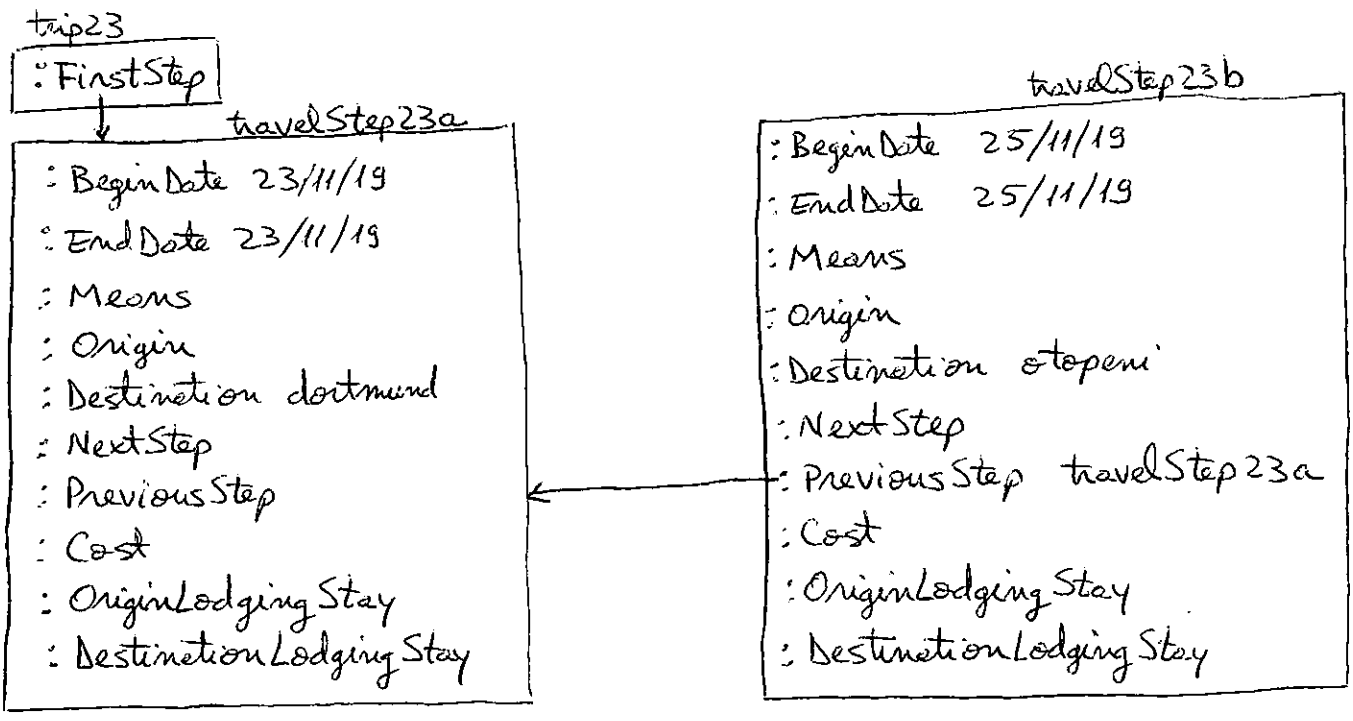
```
(LodgingStay
    <: Place [IF-NEEDED
                {SELF : ArrivingTravelStep : Destination}
              ] > ...)
```

For a certain trip, called trip23, we create an individual frame

```
(trip23
    <: INSTANCE-OF Trip>
    <: FirstStep travelStep23a>
)
```

and two instances of TravelStep

```
(travelStep23a
    <: INSTANCE-OF TravelStep>
    <: Destination dortmund>
    <: BeginDate 23/11/19>
    <: EndDate 23/11/19>
)
(travelStep23b
    <: INSTANCE-OF TravelStep>
    <: Destination otopeni>
    <: BeginDate 25/11/19>
    <: EndDate 25/11/19>
    <: PreviousStep travelStep23a>
)
```



To complete the initial setup, we should have

travelStep23a : NextStep ← travelStep23b

```
(TravelStep
   <: PreviousStep [IF-ADDED
                      {SELF : PreviousStep : NextStep ← SELF ;}
                    ] > ...)
```

As a consequence of this assignment, the IF-ADDED attached procedure to the slot :NextStep is executed:

```
(lodgingStay23a
   <: INSTANCE-OF LodgingStay >
   <: BeginDate 23/11/19 >
   <: EndDate 25/11/19 >
   <: ArrivingTravelStep travelStep23a >
   <: DepartingTravelStep travelStep23b >
)
```