This introduces :BaronWayApartment as an instance of the class :Apartment. It has four rooms and is rented by :Paul. Remember that under the direct semantics of OWL2 DL, the rdf:type relations may hold only between two strictly separated levels: that of classes, and that of individuals.[8]

Identity Assertions Because OWL2 has the open world assumption, we can never assume that two individuals with different URIs must be different entities. We might be dealing with a single individual that has multiple names. Although we have seen that in some cases we can infer identity relations automatically, it is often more convenient to state them explicitly:

```
:BaronWayApartment owl:sameAs       :PaulsApartment ;
                   owl:differentFrom :FranksApartment .
```

The list of different individuals can easily grow quite long. For instance, a small city will already contain hundreds of apartments for which we would need to assert pairwise owl:differentFrom relations. Fortunately, we can state this a bit more elegantly using the owl:AllDifferent construct:

```
_:x  rdf:type     owl:AllDifferent ;
     owl:members  ( :FranksApartment :PaulsApartment ) .
```

Negative Assertions Sometimes we know something *not* to be the case. Making this knowledge explicit can be very valuable in an open world: ruling out possibilities often allows us to infer new knowledge. For instance, the knowledge that :BaronWayApartment is not rented by :Frank may allow us to infer that it is not :FranksApartment:

```
_:x  rdf:type              owl:NegativePropertyAssertion ;
     owl:sourceIndividual  :BaronWayApartment ;
```

[8] See also the discussion on punning in section 4.4.5.