unique identifier. For example, the combination of postcode and street address number will provide a unique identifier for any dwelling in the Netherlands:

```
:postcode      rdf:type    owl:DatatypeProperty .
:addressNumber rdf:type    owl:DatatypeProperty .

:Dwelling
        rdf:type    owl:Class ;
        owl:hasKey ( :postcode :addressNumber ) .
```

Note that the key mechanism allows us to define *inverse functional datatype properties* that are local to a class. Any two individuals of type ex:Dwelling that have the same value for the :postcode and :addressNumber must be considered to be the same. Unfortunately OWL2 DL does not allow us to specify global inverse functional datatype properties because of computational consequences.

4.4.7 Individual Facts

Now that we have a general idea of how we define properties and classes in OWL2, we turn our attention to the individual entities governed by our model. In many cases we already have a lot of knowledge about these entities and only need class axioms to infer *extra* information. Statements about individuals are usually called *assertions*.

Class and Property Assertions Class membership and property assertions in OWL2 are stated in the same way as in RDF Schema:

```
:Apartment         rdf:type    owl:Class .

:BaronWayApartment rdf:type         :Apartment ;
          :hasNumberOfRooms "4"^^xsd:integer ;
          :isRentedBy       :Paul .
```