

```
:LuxuryBathroomApartment
```

```
    rdf:type          owl:Class;
    rdfs:subClassOf [ rdf:type          owl:Restriction;
                      owl:onProperty  :hasBathroom ;
                      owl:allValuesFrom :LuxuryBathroom
                    ] .
```

This defines the `:LuxuryBathroomApartment` class as a subclass of the set of individuals that only have instances of `:LuxuryBathroom` as value for the `:hasBathroom` property. Note that an `owl:allValuesFrom` restriction merely states that *if* a member of the restricted class has a value for the property, then that value must be a member of the specified class. The restriction does not require the property to have any value at all: in that case, the restriction is trivially satisfied. In our apartment example, the above definition does not require that a luxury bathroom apartment have a bathroom at all!

Universal restrictions can also be used with datatype properties – for instance, to state that the value of a property must be of a certain type or fall within a certain data range (see below).

Existential Restrictions An existential restriction on a class C and property p states that for every member of C there exists at least *some* value for p that belongs to a certain class. This type of restriction is specified using an `owl:someValuesFrom` keyword:

```
:LuxuryBathroomApartment
```

```
    rdf:type          owl:Class;
    rdfs:subClassOf [ rdf:type          owl:Restriction;
                      owl:onProperty  :hasBathroom ;
                      owl:someValuesFrom :LuxuryBathroom
                    ] .
```

Necessary and Sufficient Conditions Instead of using the `rdfs:subClassOf` property to relate our class to the restriction, we could also have used an `owl:equivalentClass`