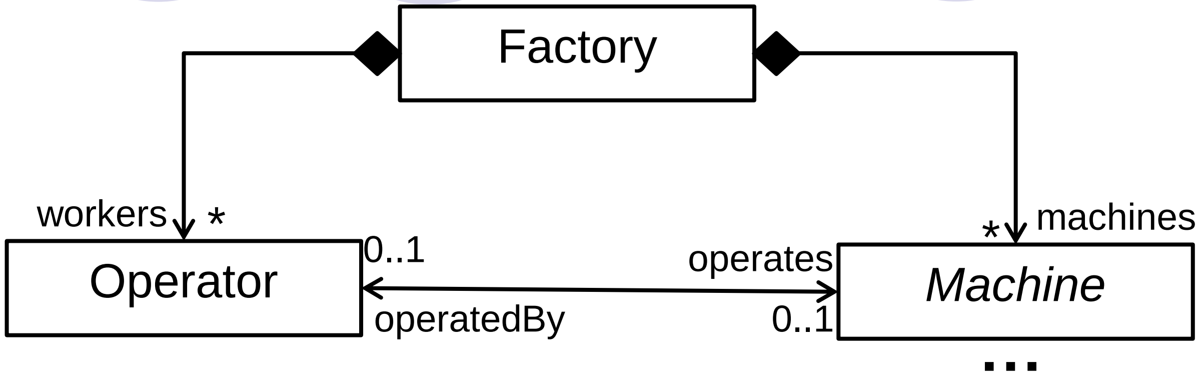
OCL Modelling Quizzes

http://uml.org

Juan de Lara, Elena Gómez, Esther Guerra
{Juan.deLara, MariaElena.Gomez, Esther.Guerra}@uam.es
Computer Science Department
Universidad Autónoma de Madrid

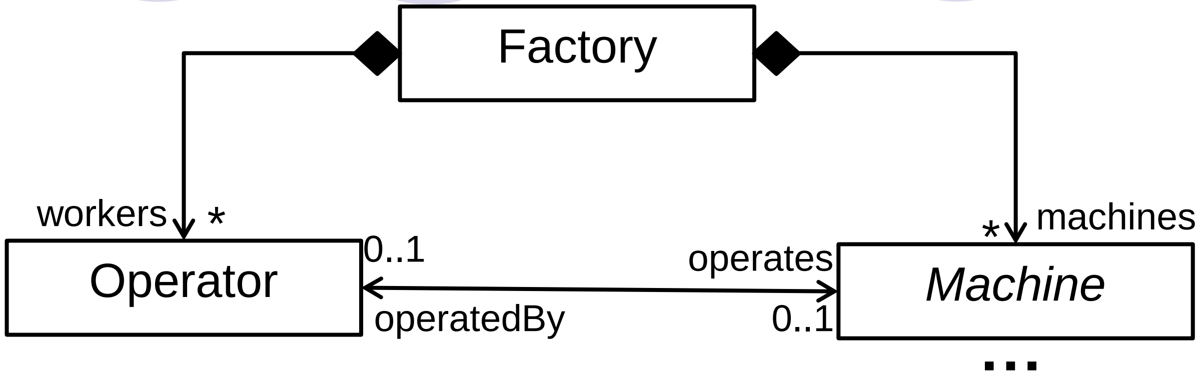
Masters: I2TIC and Formal methods

How do we express that…

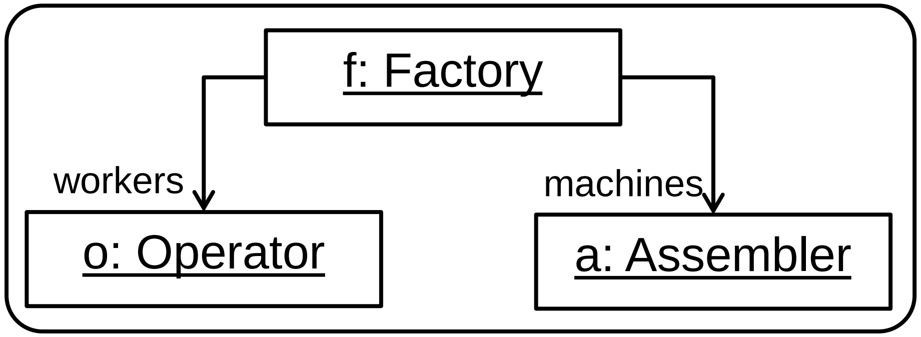


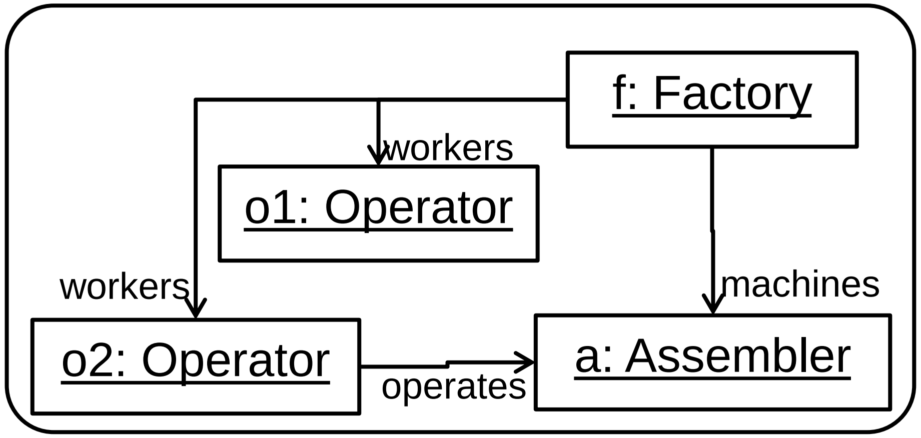
Any machine can be operated by at most one operator.
Each operator operates at most one machine at the same time.
At every moment, in the factory, there should be at least one operator in some machine.

How do we express that…

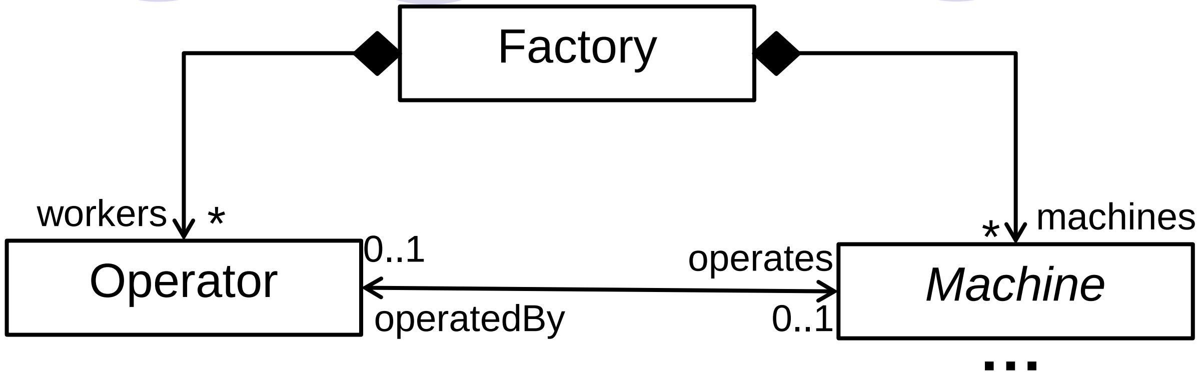


context Factory
noMachineUnattended **inv**:
self.workers->exists(o | o.operates->notEmpty())

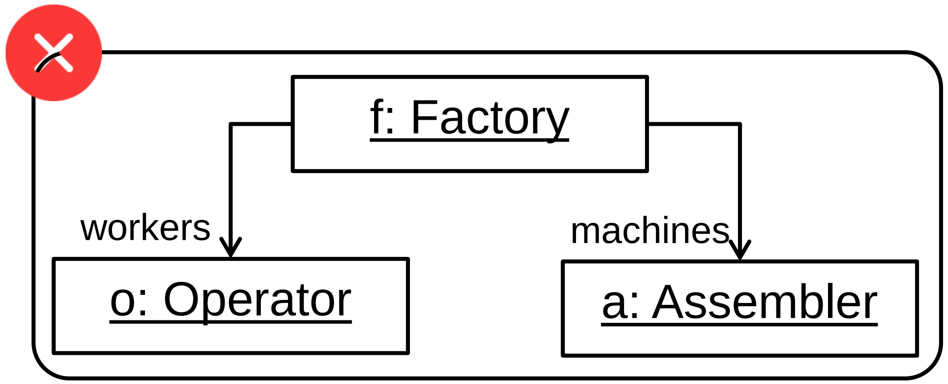


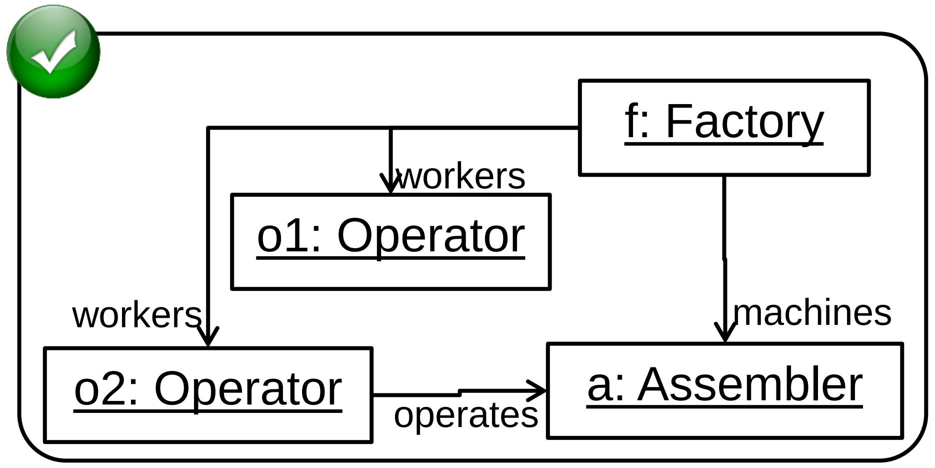


How do we express that…

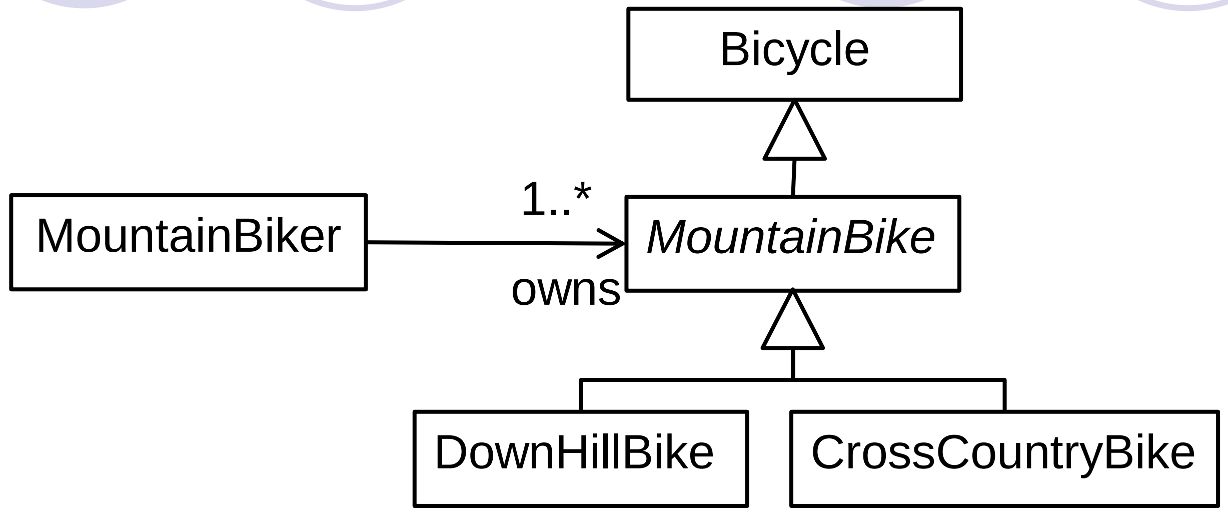


context Factory
noMachineUnattended **inv**:
self.workers->exists(o | o.operates->notEmpty())



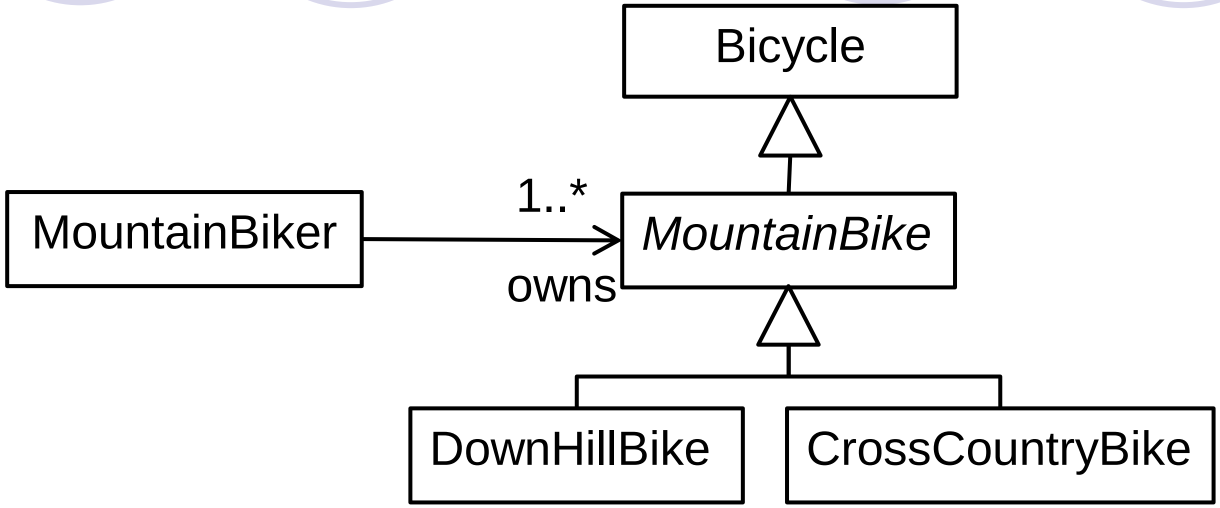


How do we express that…



"Not all MountainBikes owned by any MountainBiker have the same type"

How do we express that…

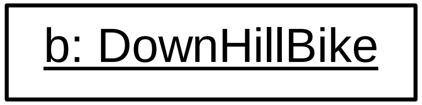


"Not all MountainBikes owned by any MountainBiker have the same type"

Hint:

obj.oclIsKindOf(T) → returns true if obj is of type T, or a supertype
obj.oclIsTypeOf(T) → returns true if obj is of type T

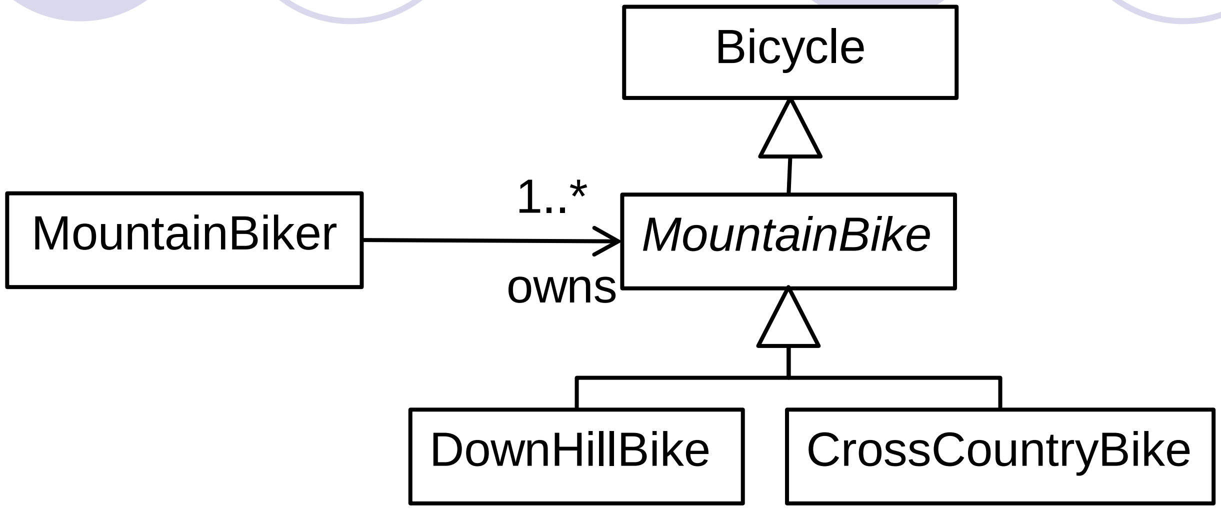
Example:



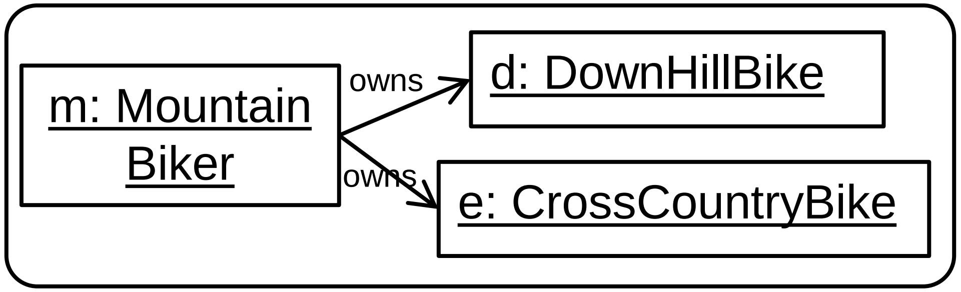
b.oclIsKindOf(DownHillBike) → true
b.oclIsTypeOf(DownHillBike) → true

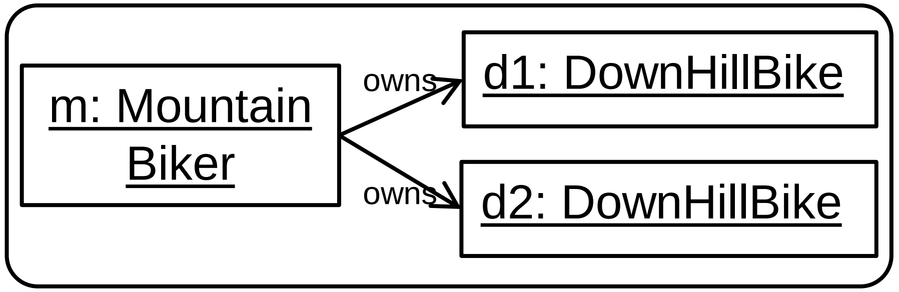
b.oclIsKindOf(MountainBike) → true
b.oclIsTypeOf(MountainBike) → false

How do we express that…

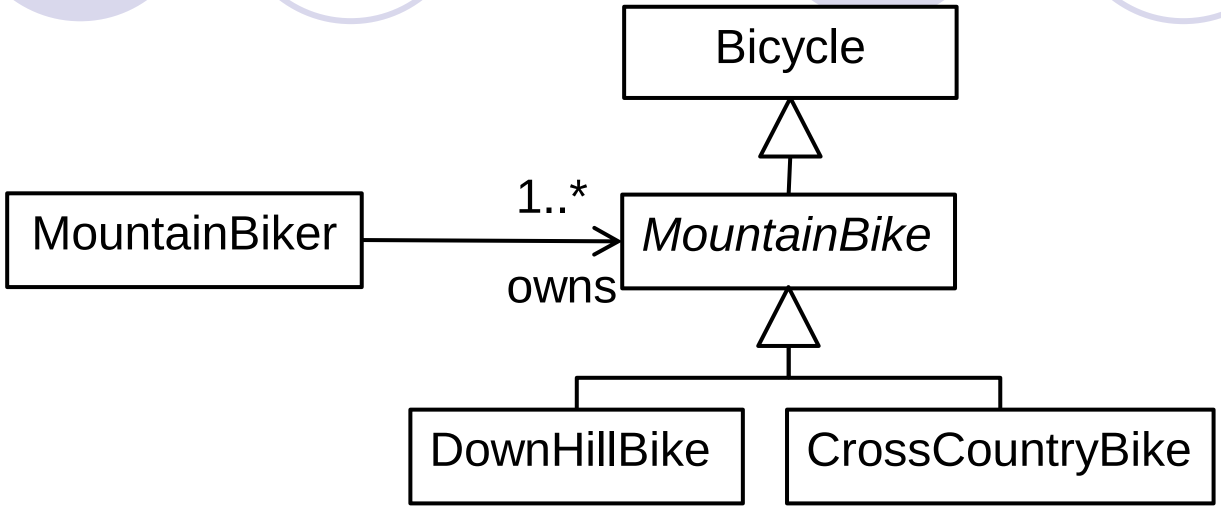


```
context MountainBiker
  notSameBikes inv:
    owns->exists(b | b.oclIsTypeOf(DownHillBike)) and
    owns->exists(b | b.oclIsTypeOf(CrossCountryBike))
```

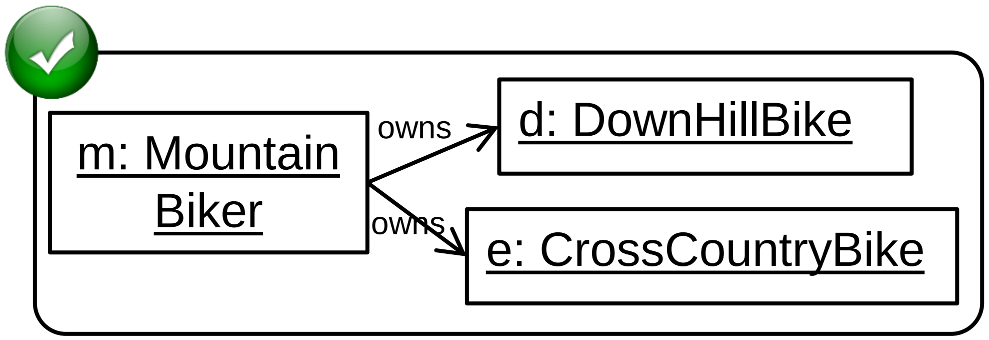


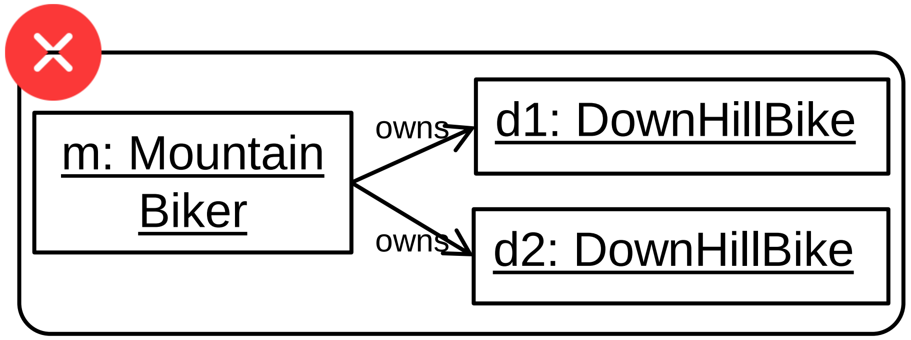


How do we express that…

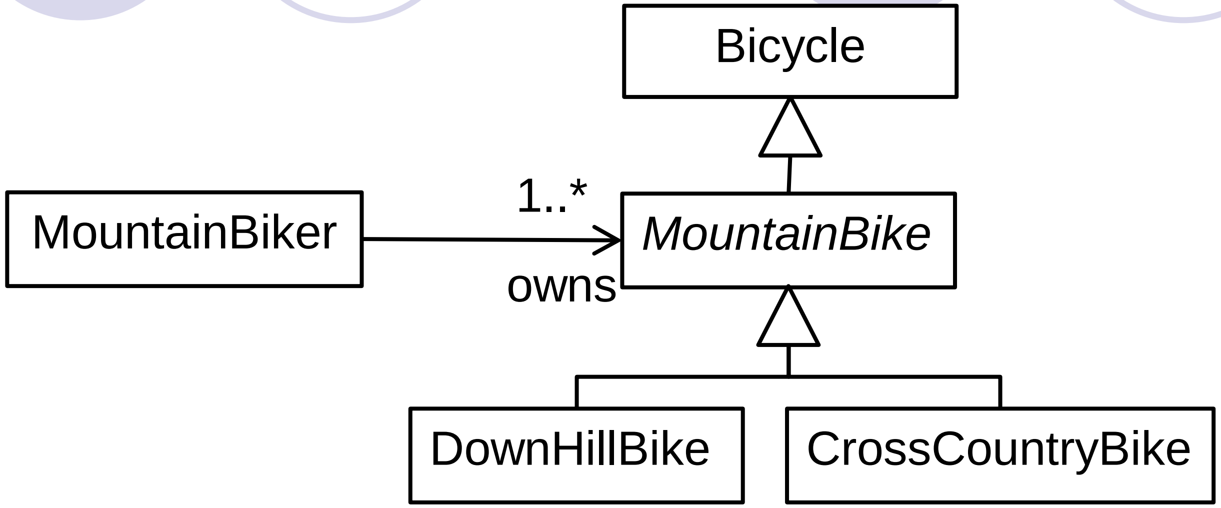


```
context MountainBiker
  notSameBikes inv:
    owns->exists(b | b.oclIsTypeOf(DownHillBike)) and
    owns->exists(b | b.oclIsTypeOf(CrossCountryBike))
```

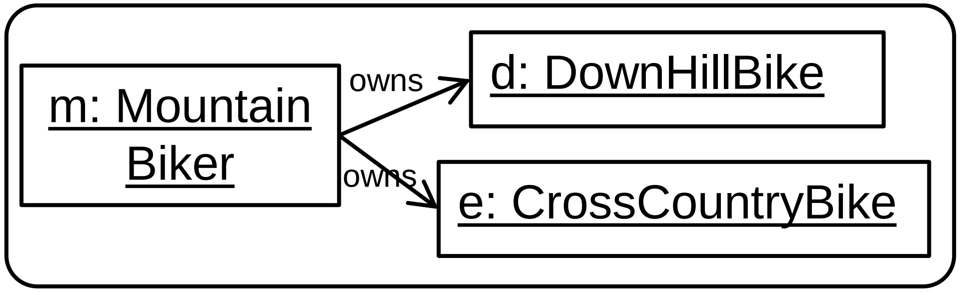


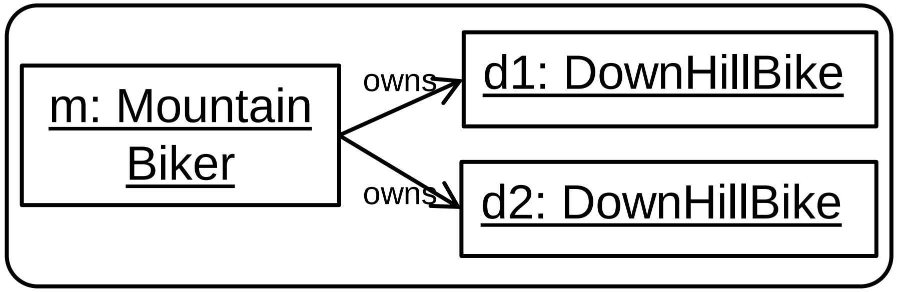


How do we express that…

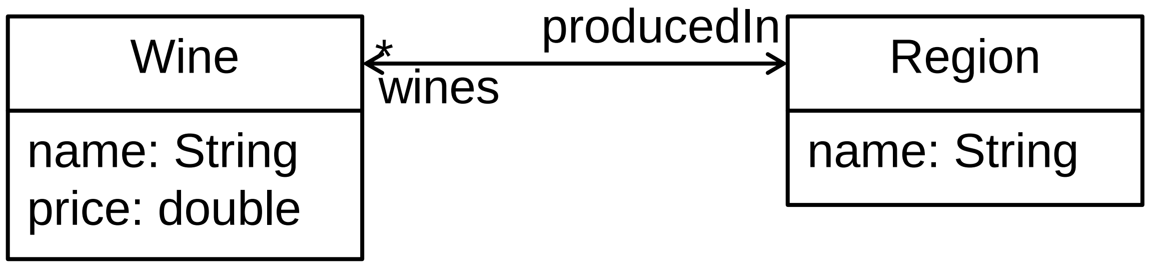


```
context MountainBiker
  notSameBikes inv:
    owns->size()=1 or (  -- to consider the case where the biker has just one bike
          owns->exists(b | b.oclIsTypeOf(DownHillBike)) and
          owns->exists(b | b.oclIsTypeOf(CrossCountryBike)))
```



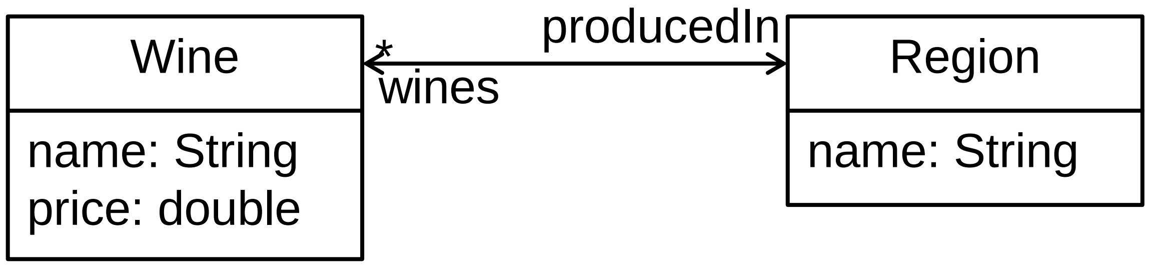


How do we express that…



"The most expensive wine is from the 'Rioja' region"

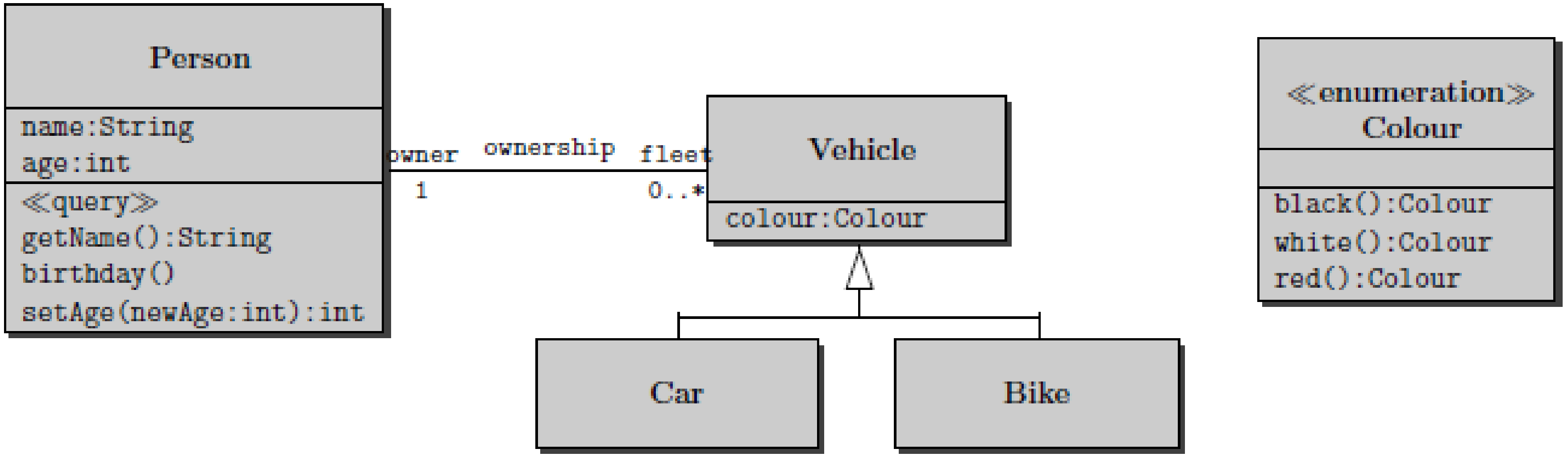
How do we express that…



"The most expensive wine is from the 'Rioja' region"

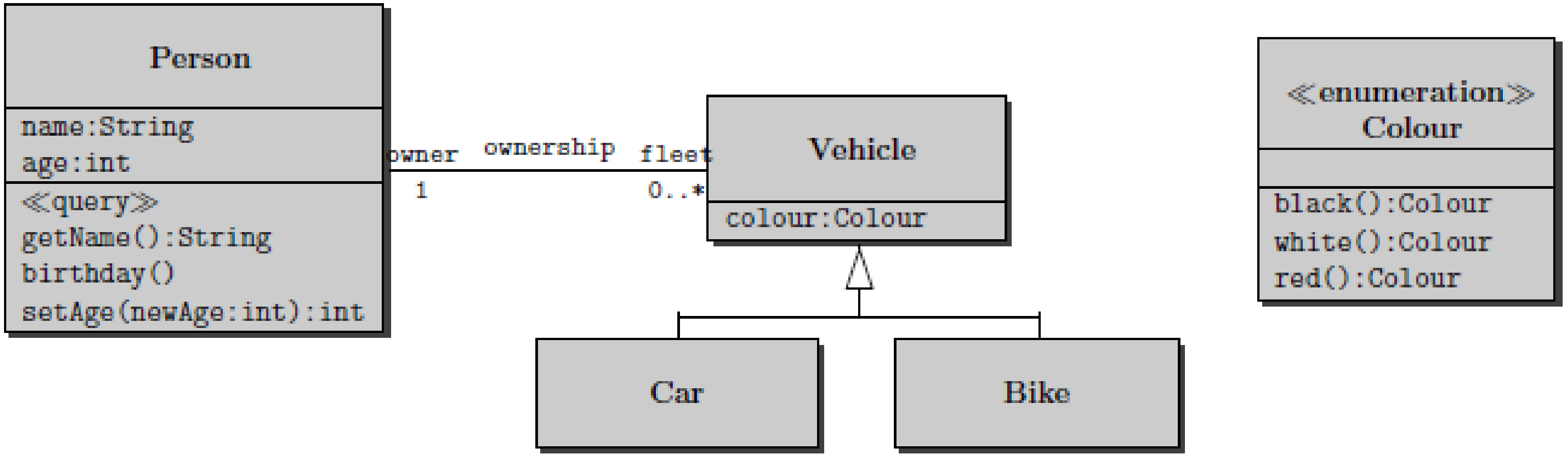
```
Wine.allInstances()->select( ew |
          not Wine.allInstances()->exists ( w | w.price > ew.price ) )
          .producedIn.name->includes('Rioja')
```

How do we express that…



A vehicle owner must be at least 18 years old

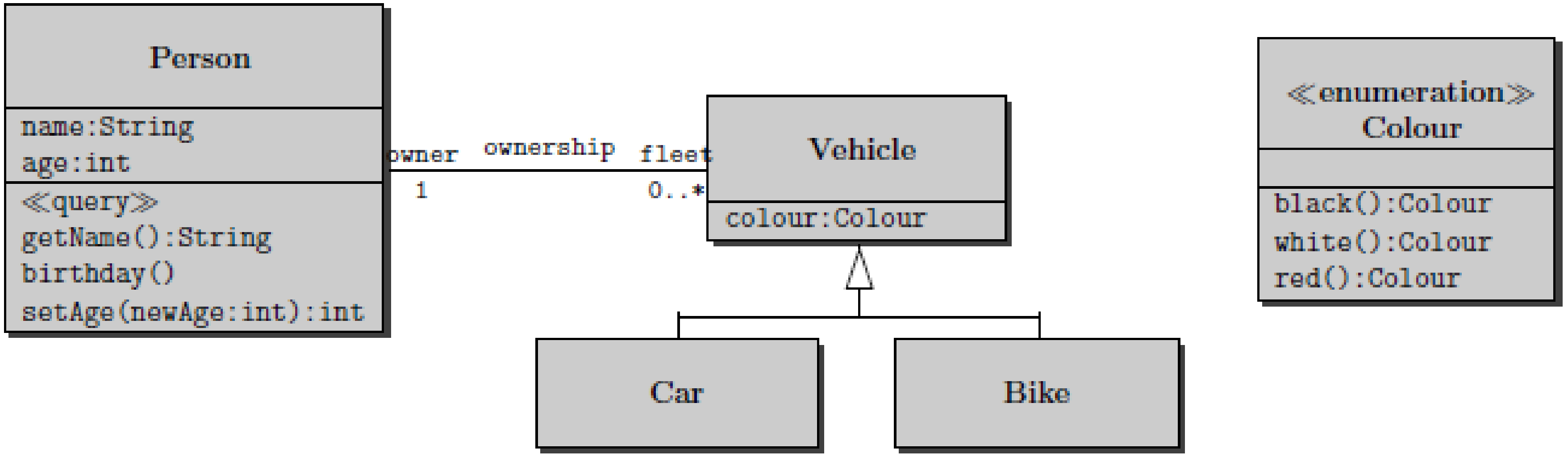
How do we express that…



A vehicle owner must be at least 18 years old

context Vehicle
inv: self.owner.age >= 18

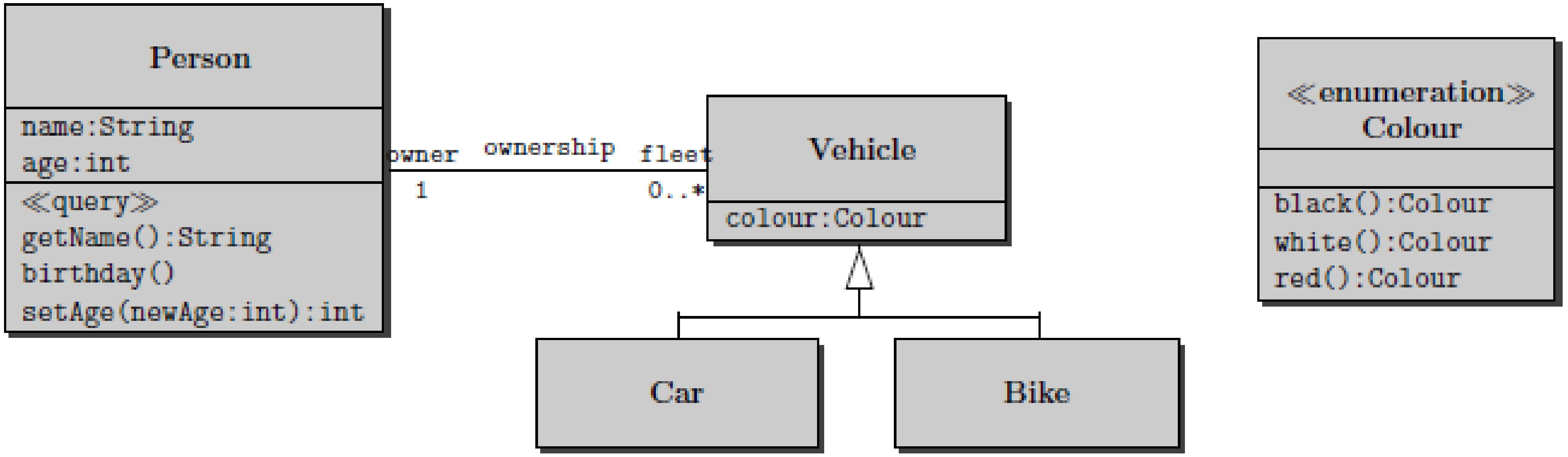
What’s the difference with…?



context Vehicle
inv: self.owner.age >= 18

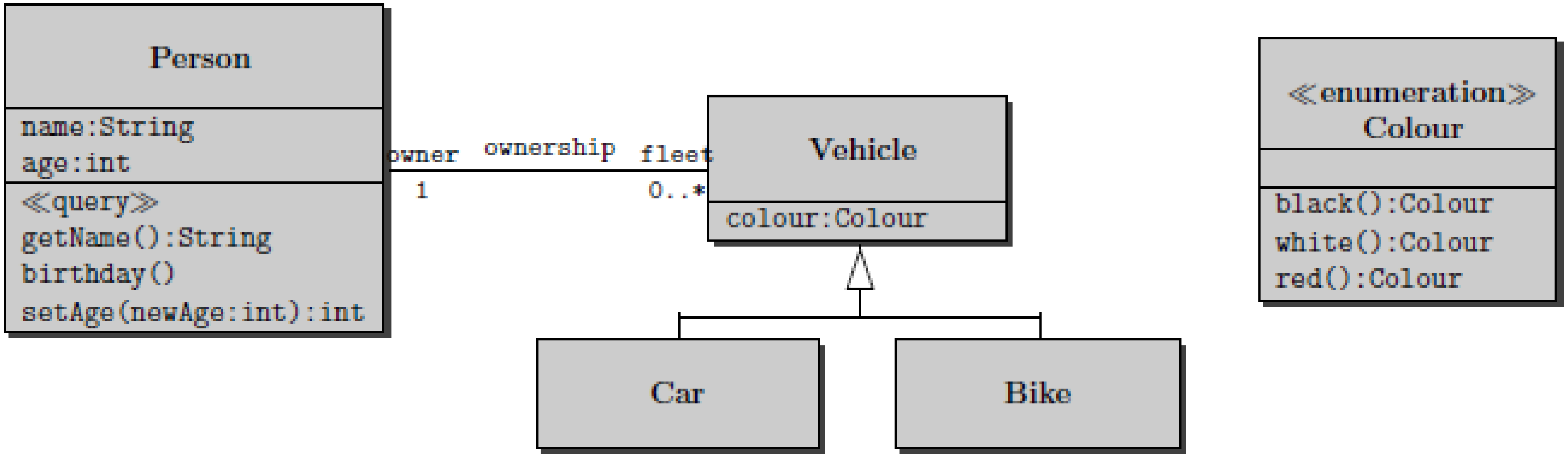
context Person
inv: self.age >= 18

How do we express that…



A car owner must be at least 18 years old

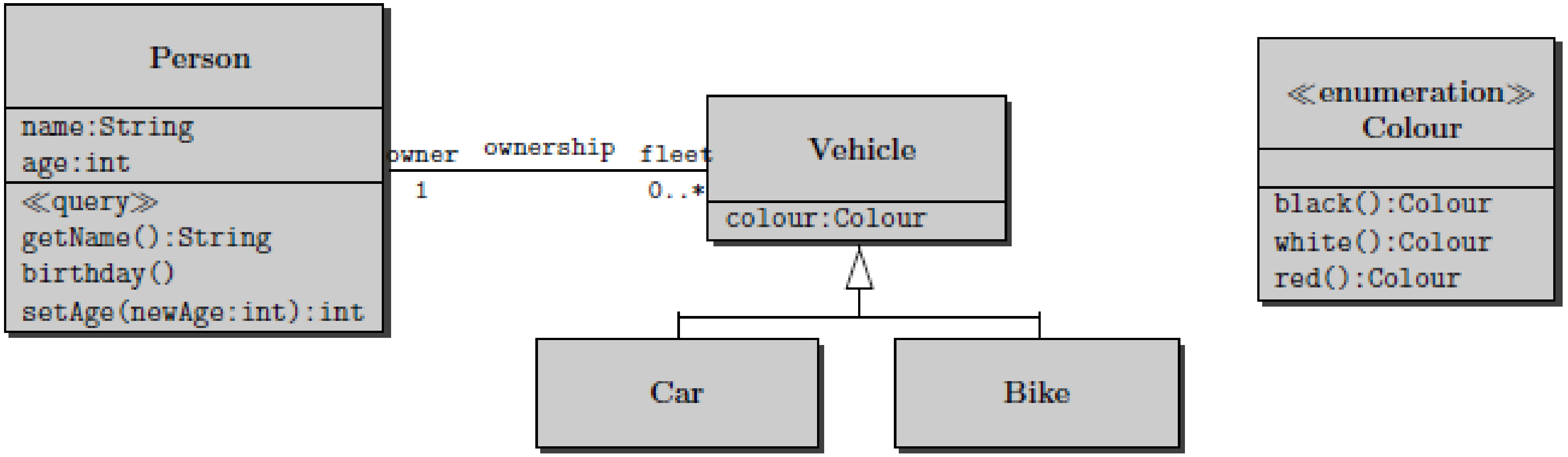
How do we express that…



A car owner must be at least 18 years old.:

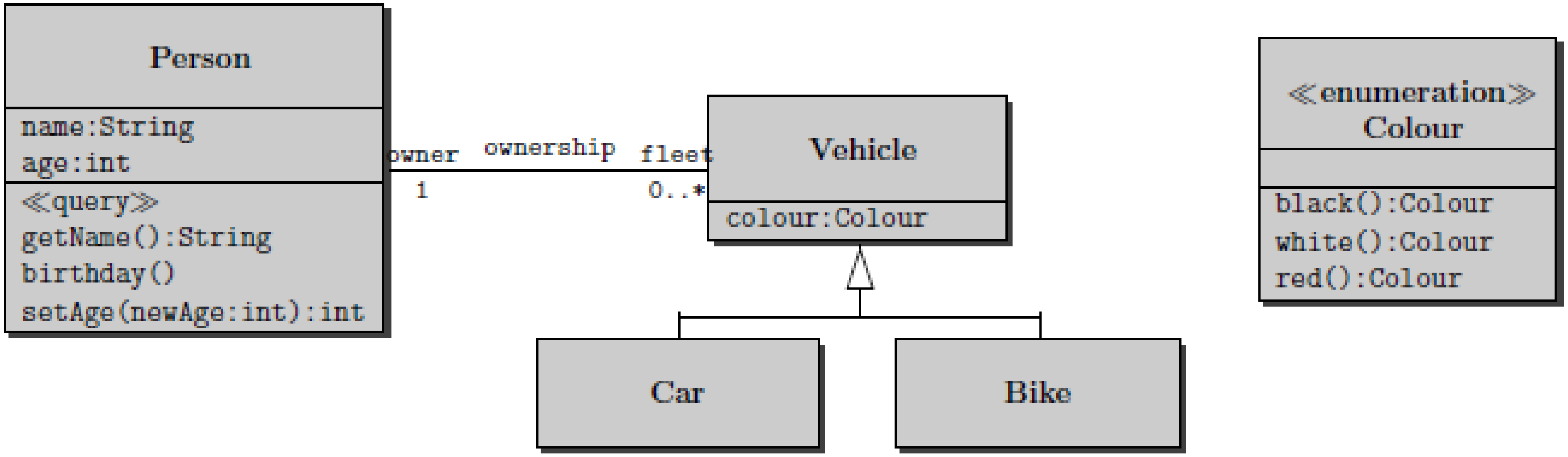
context Car
inv: self.owner.age >= 18

How do we express that…



Nobody has more than three vehicles

How do we express that…

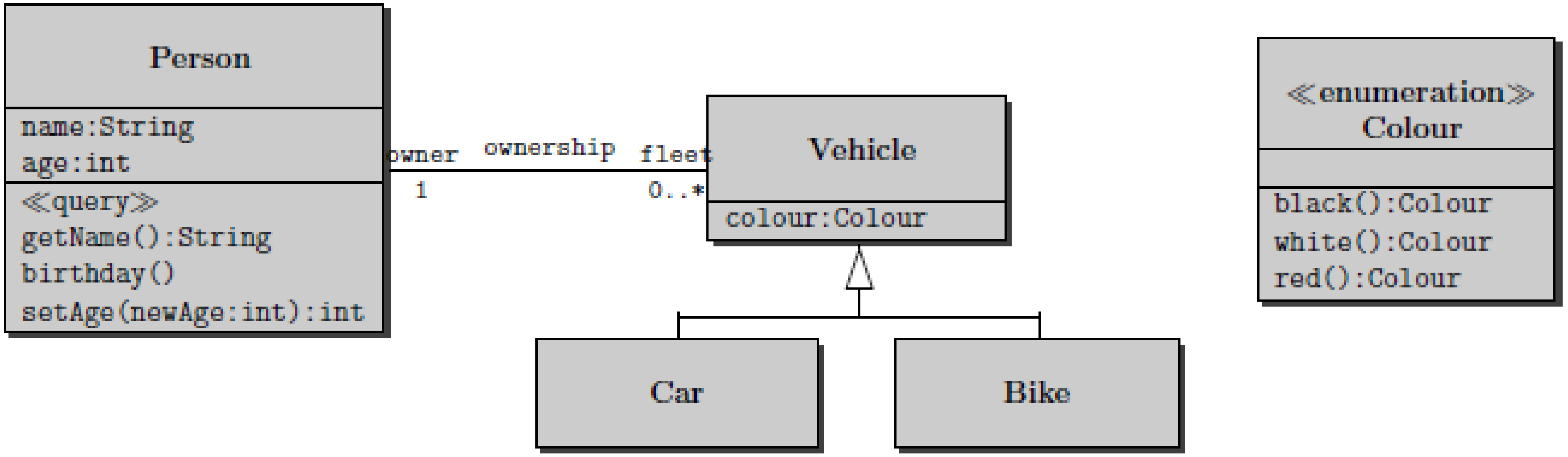


Nobody has more than three vehicles

context Person
inv: self.fleet->size <= 3   -- parentheses for operations w/0 params optional

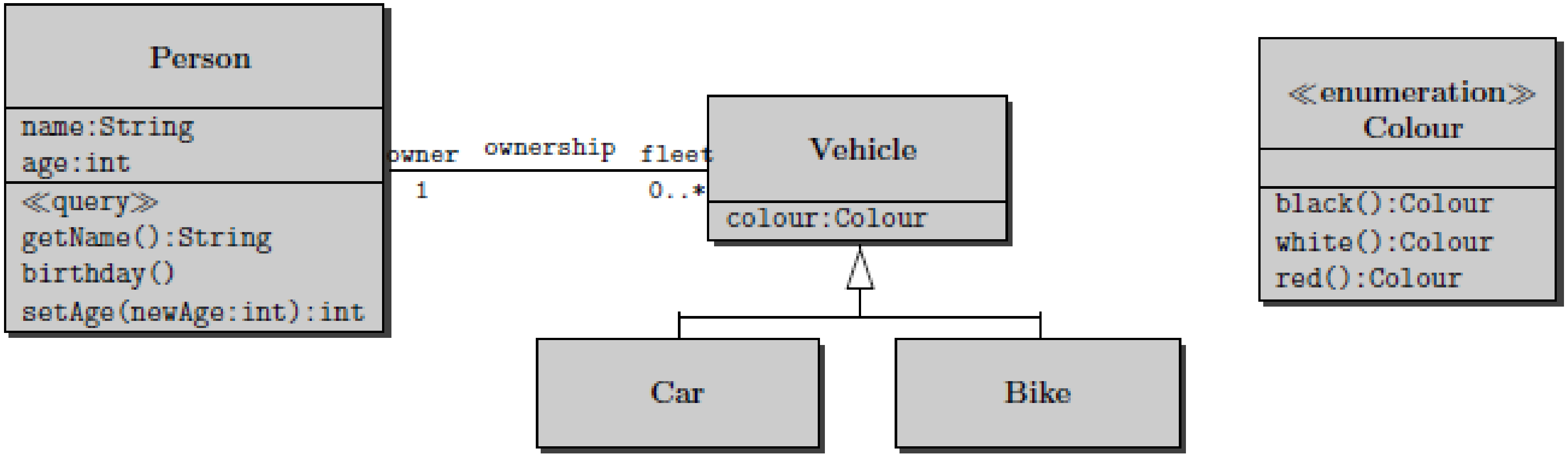
Or change cardinality

How do we express that…



All cars of a person are black

How do we express that…

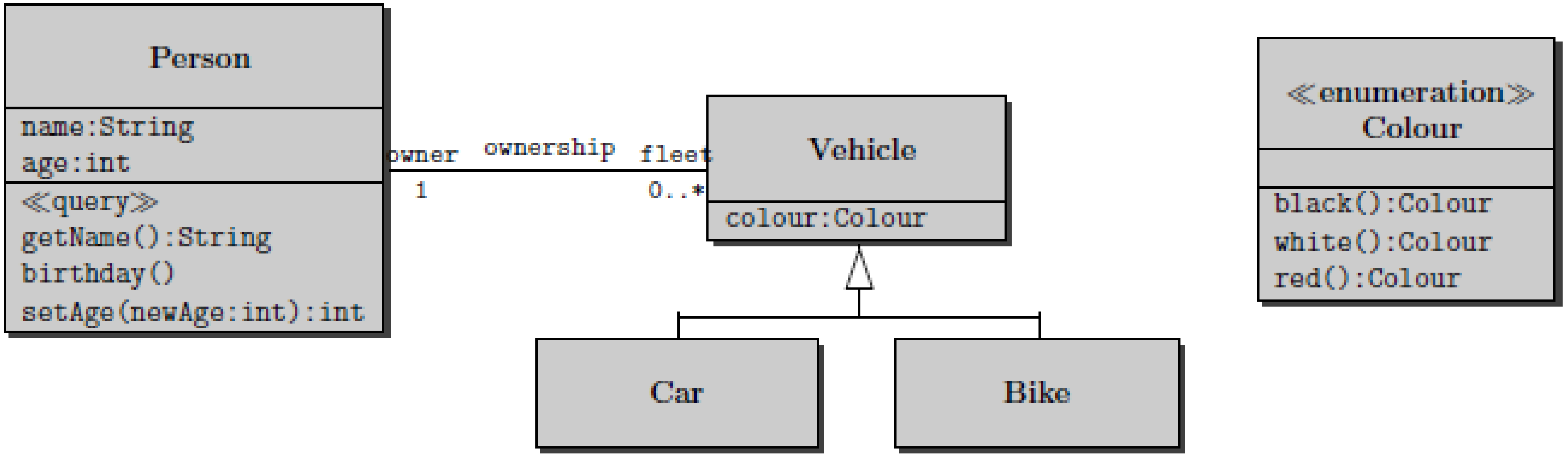


All cars of a person are black

context Person
inv: self.fleet->forAll(v | v.colour = #black)

(we can do better)

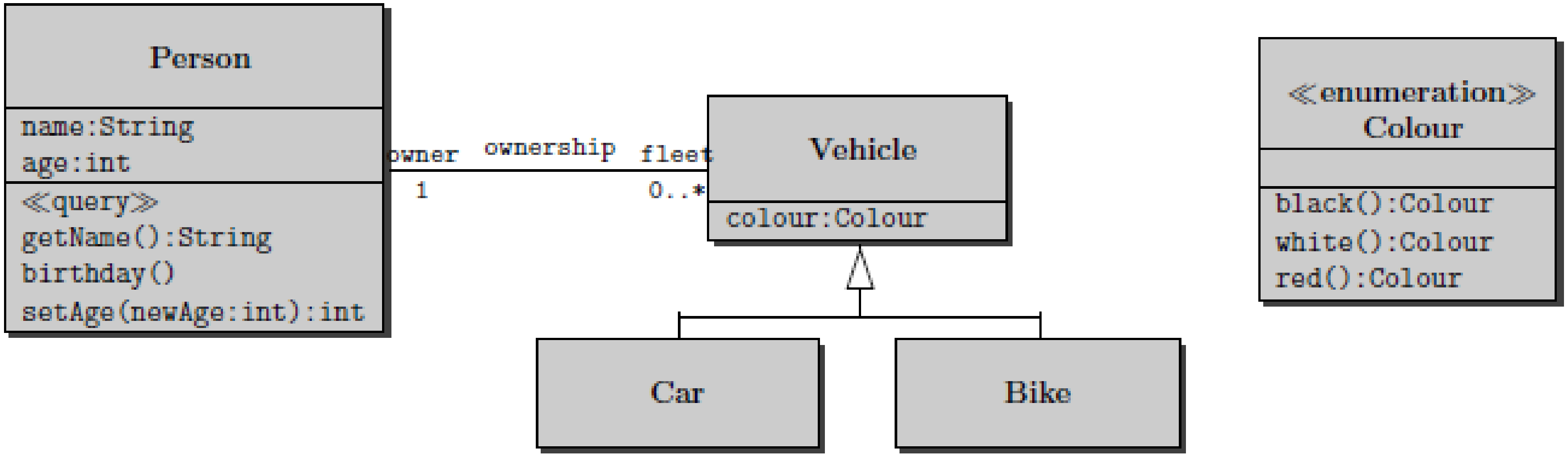
How do we express that…



All cars of a person are black

context Person
inv: self.fleet->forAll(v | v.oclIsKindOf(Car) implies v.colour = #black)

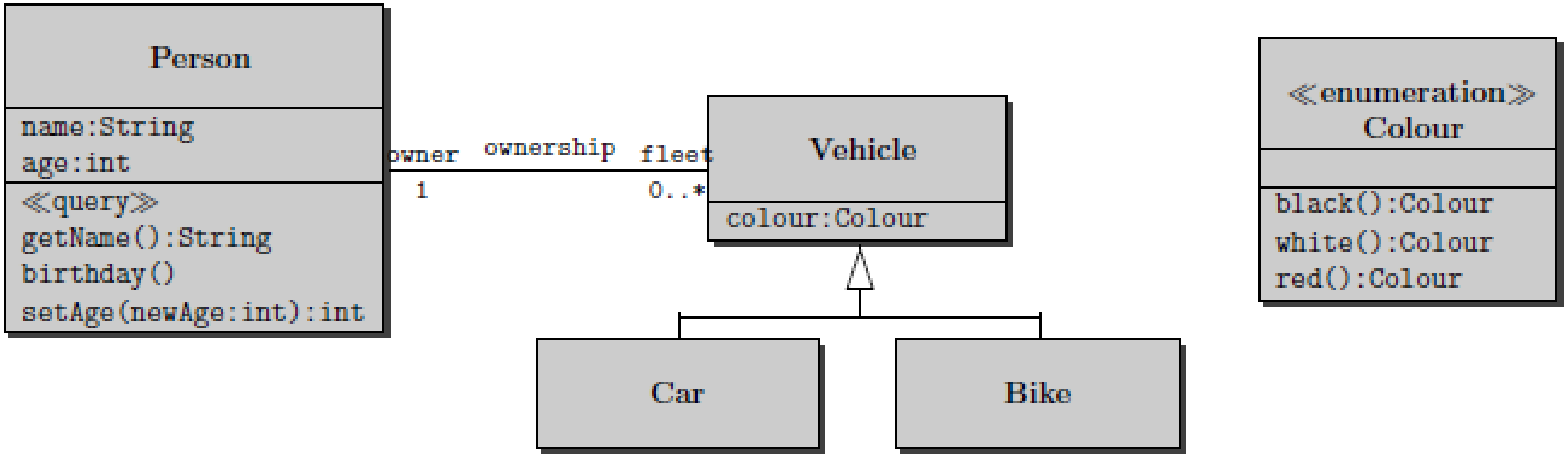
How do we express that…



Nobody has more than 3 black vehicles

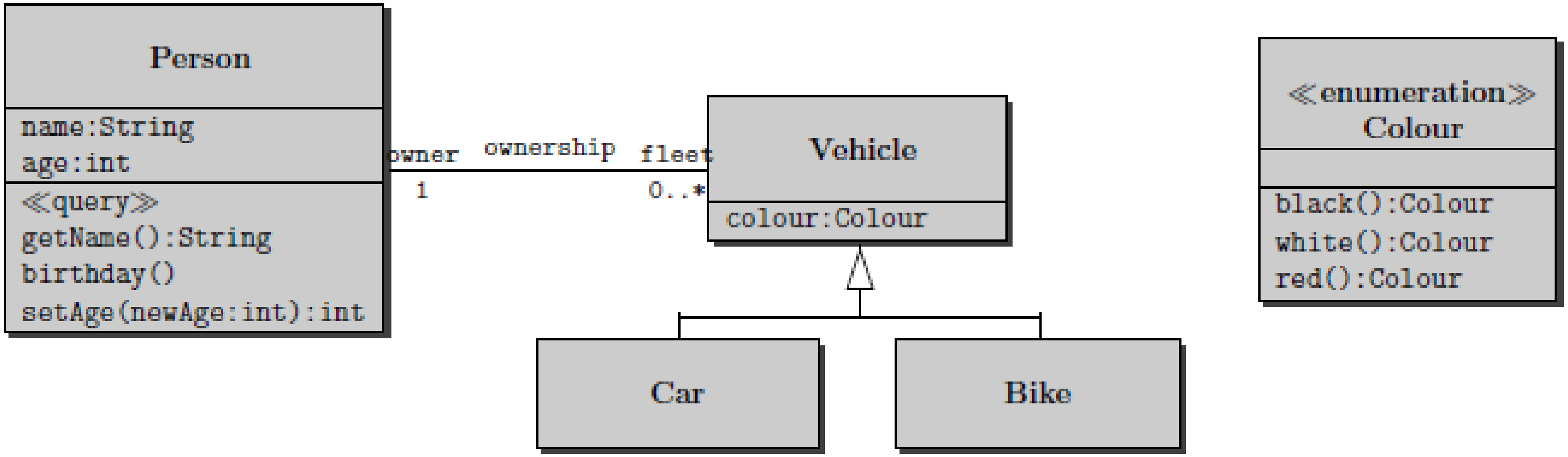
context Person
inv: self.fleet->select(v | v.colour = #black)->size <= 3

What does this expression mean?



context Person
inv: self.fleet->iterate(v; acc:Integer=0 | if (v.colour=#black)
then acc + 1 else acc endif) <=3

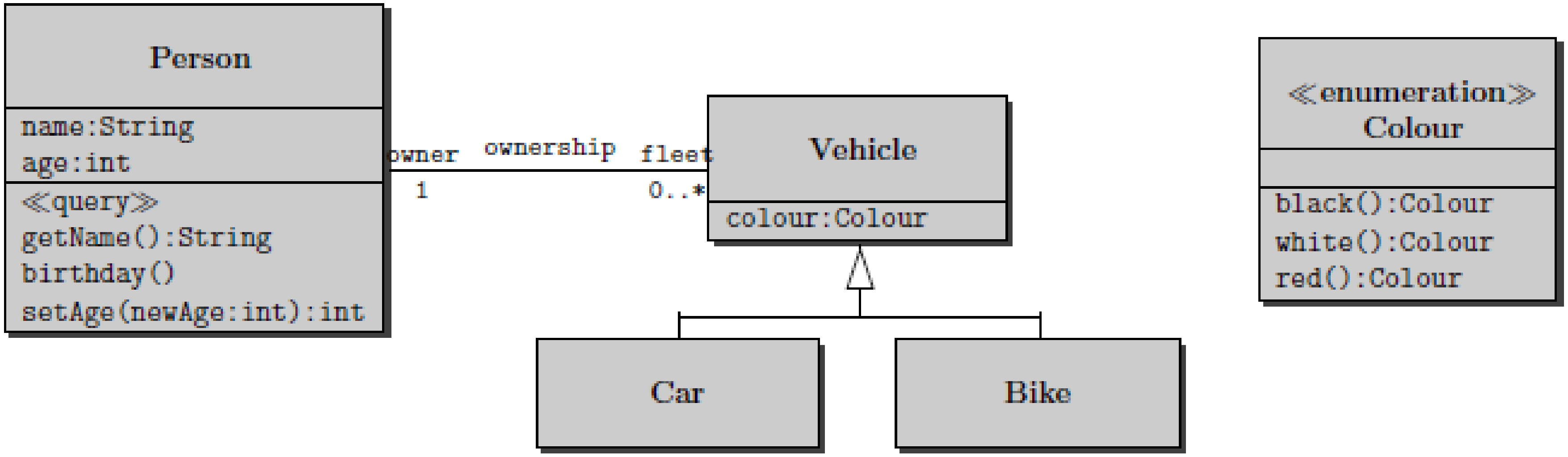
What does this expression mean?



context Person
inv: self.fleet->iterate(v; acc:Integer=0 | if (v.colour=#black)
then acc + 1 else acc endif) <=3

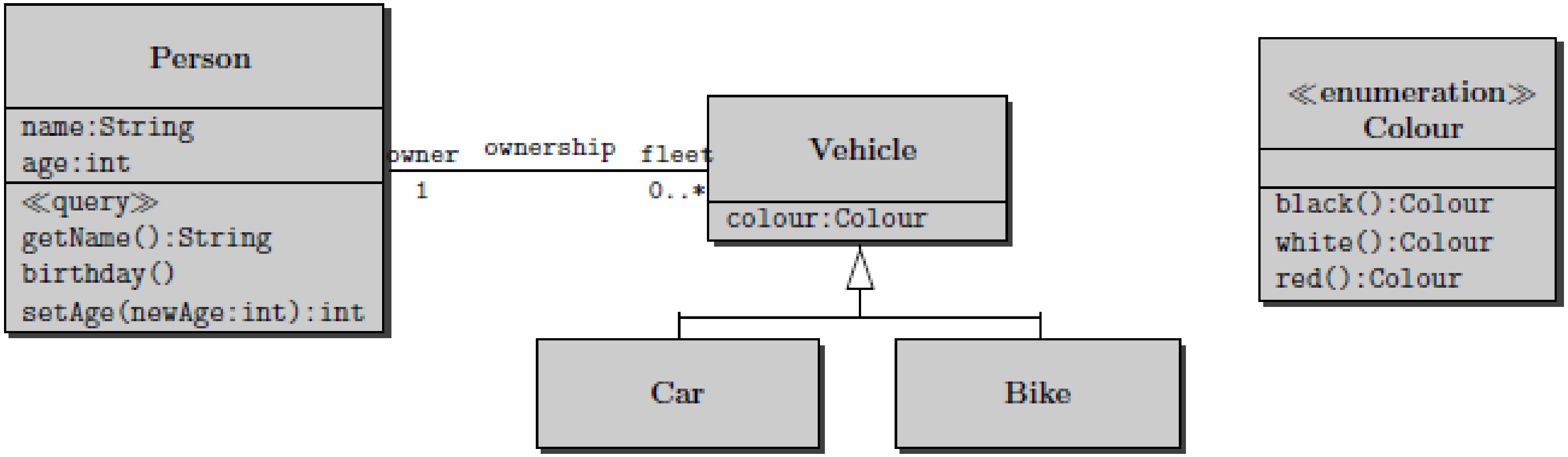
Nobody has more than 3 black vehicles

What does this expression mean?



context Person
inv: age<18 implies self.fleet->forAll(v | not v.oclIsKindOf(Car))

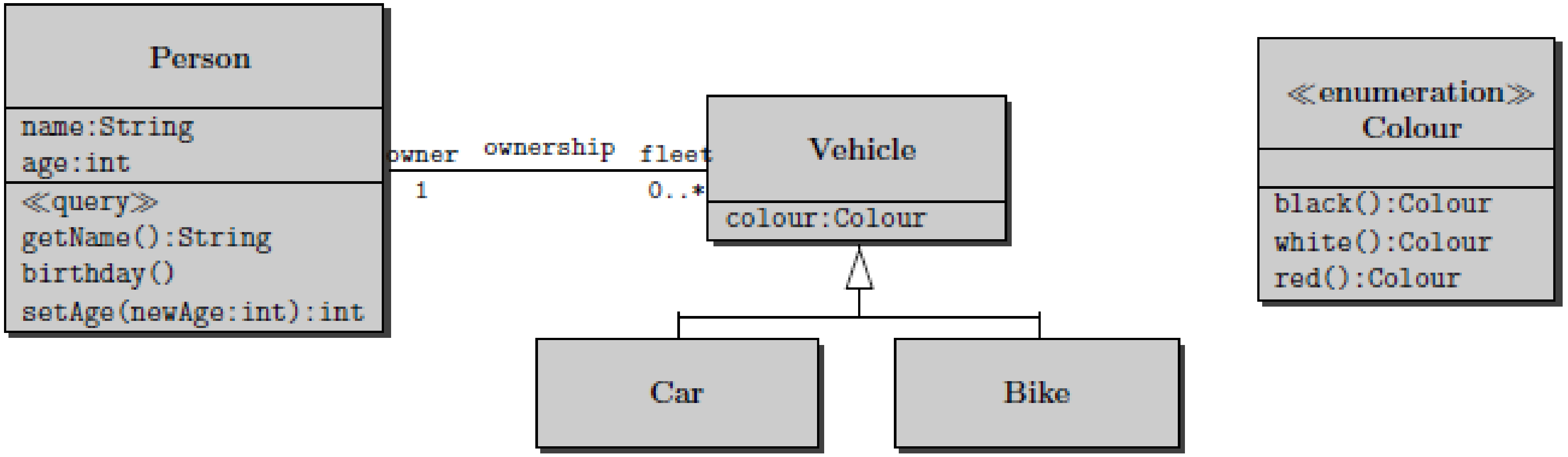
What does this expression mean?



context Person
inv: age<18 implies self.fleet->forAll(v | not v.oclIsKindOf(Car))

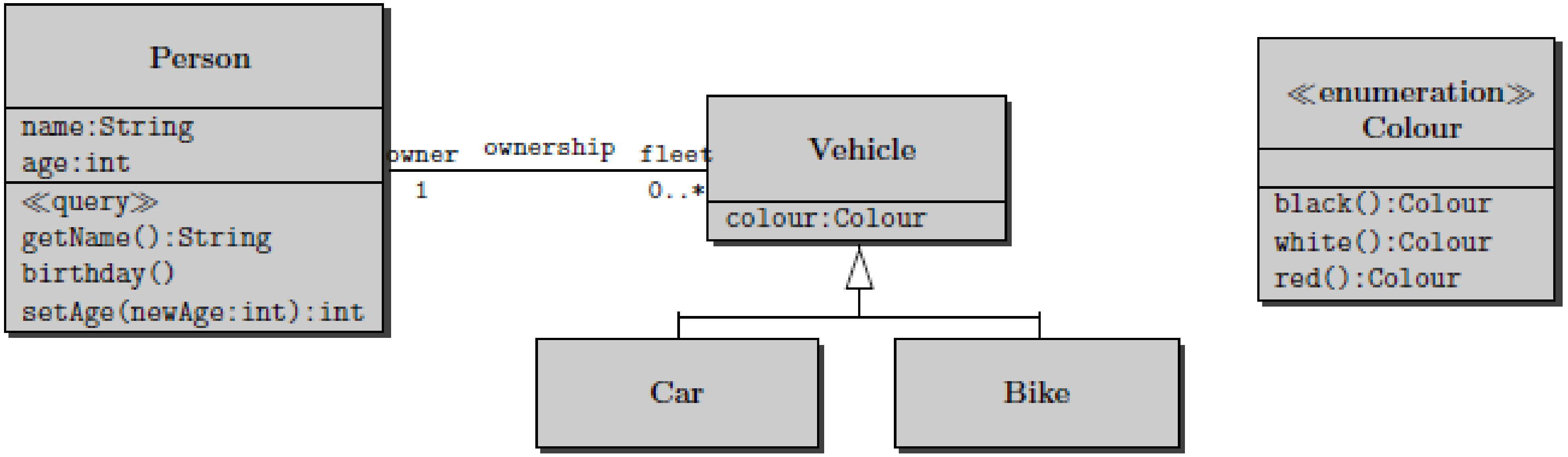
No child owns a car

What does this expression mean?



context Car
inv: Car.allInstances()->exists(c | c.colour=#red)

What does this expression mean?



context Car
inv: Car.allInstances()->exists(c | c.colour=#red)

There is at least one red car