

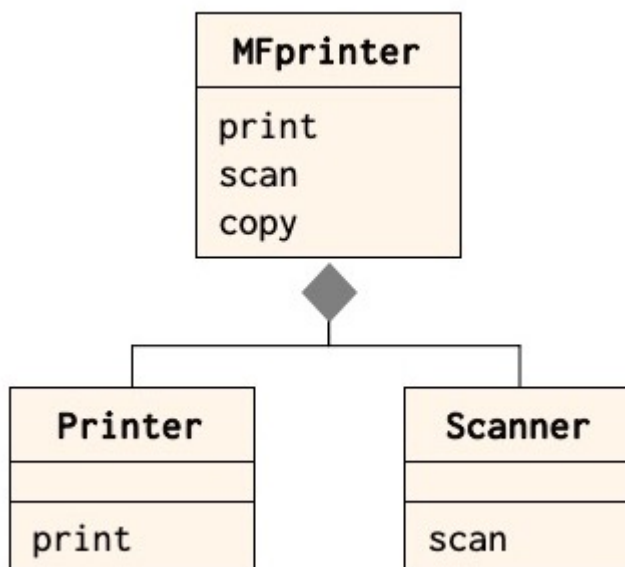
## 17. Composite objects

### Description

In this chapter, we show how to define *composite objects*. A composite object consists of other objects. For material objects, we use intrinsic objects (declared using the keyword `obj` (section )) to represent possible parts. We will use the term *part object* for an intrinsic object that represents a part of a composite object. For value objects, composite objects are declared by using `var` or `val`.

A part-object may itself have part-objects and in this way be structured as a composition-hierarchy. Composition is analogous to classification and contextualization, that all are important conceptual means for understating and structuring our knowledge about the real world and thus of a given application domain.

A first very simple example of a composite object is a model of a multi-function printer, that is a printer that in addition to printing also can scan and copy. It consists of a `Printer` part-object and a `Scanner` part-object.

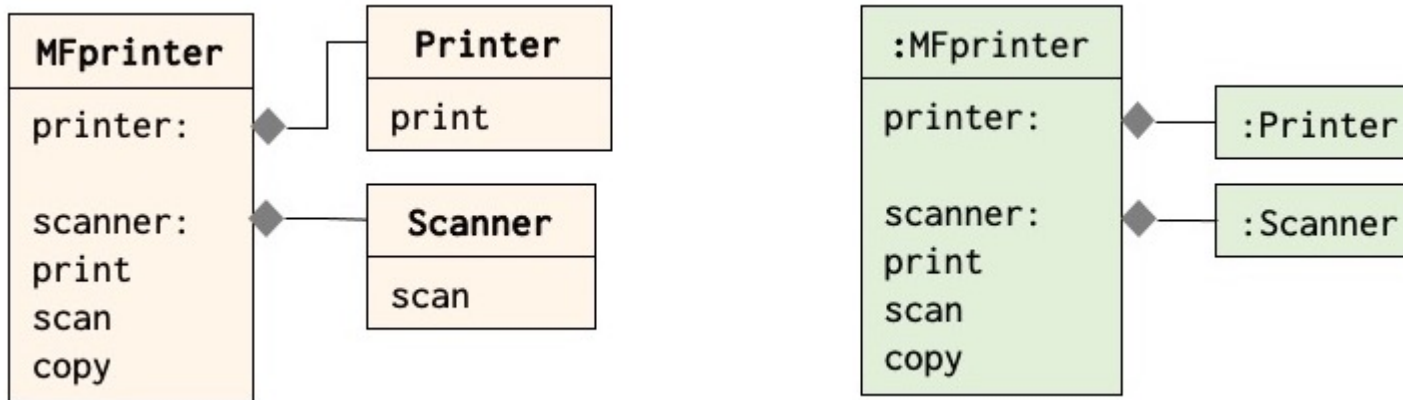


```
class Printer:
  print(doc: ref Document):
  ...
class Scanner:
  scan -> doc: ref Document:
  ...
class MultiFunctionPrinter:
  thePrinter: obj Printer
  theScanner: obj Scanner

  print(doc: var Document):
    thePrinter.print(doc)
  scan -> doc: ref Document:
    doc := theScanner.scan
  copy:
    thePrinter.print(theScanner.scan)
```

The `copy` method of the composite `MutiFunctionPrinter` uses the properties of the two parts, and it is usually a function that is used directly by a human user. The method `print` is invoked by computers linked to a multi-function printer. The `scan`-method is typically used by a human who places a document in the scanner and activate the scanner, which sends the scanned document to an email address or to a computer.

In addition to the well-known notation for composition above, it may in some cases be an advantage to see the obj data-items that represent the parts of the composite object. The following figure shows this for both a class and for an object of that class.



## Representing a position on earth

A position on earth is defined using latitude and longitude, which both are coordinates from -90 to 90 or shown as 0-90 followed by North/South and East/West. These coordinates may be further spilt into minutes and seconds – often abbreviated *DMS* standing for degrees, minus and seconds.

We may represent these values using composition of value objects as follows:

```

GlobalPosition: Value
  la: var Latitude
  lo: var Longitude
  
```

A latitude and longitude may be represented by :

```

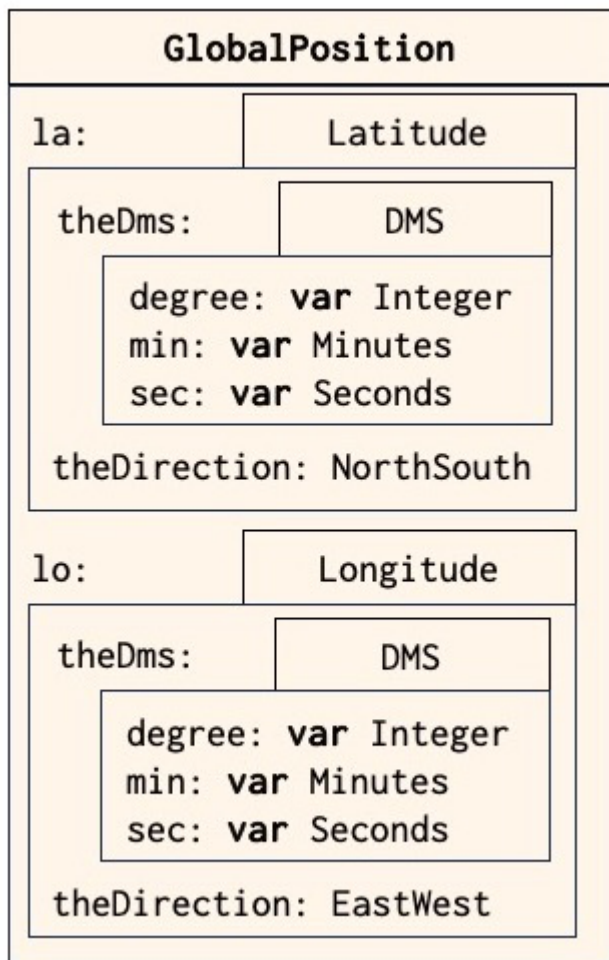
Latitude: Value
  theDms: var DMS
  theDirection: var NorthSouth
Longitude: Value
  theDms: var DMS
  direction: var EastWest
  
```

And finally a DMS may be represented by:

```

DMS: Value
  degree: var float
  min: var Time.Minutes
  sec: var Time.Seconds
  
```

The figure illustrates the composition hierarchy of a `GlobalPosition`-class:



Note that `min` and `sec` are defined using units of dimension `Time` as described in section . The variable `degree` has no unit. We leave the definition of `NorthSouth` and `EastWest` to the reader.

## A model of a university

The next example shows part of a representation of a university with the purpose of representing the educations given by the university. A university consists of a number of faculties. Each faculty consist of a number of departments. In this model, a university offers the degrees bachelor, masters and PhD. This structure may be represented in the following way:

```

class University:
    class Faculty:
        ...
    class Department:
        ...
    class Degree:
        ...
    class Bachelor: Degree
        ...
    class Masters: Degree
        ...
    class PhD: Degree
        ...
    arts: obj Faculty
        ...
    naturalScience: obj Faculty
        computerScience: obj Department
    
```

```

bachelorInCS: obj Bachelor
  ProgrammingLab: obj ...
  ...
mastersInCS: obj Masters
  ...
...
physics: obj Department
  bachelorInPhysics: obj Bachelor
  ...
  ...
...

```

Note that the faculties Arts, and Natural Science, etc. are represented by singular objects being subclassed from `Faculty`. Each faculty has departments belonging to the faculty. Within each department there are objects representing the degrees (bachelor, masters, PhD) offered by the department.

The above example is quite sketchy and we leave to the reader to fill it out to a complete example.

## Representing composition

In section , we discussed classification as a fundamental means for organizing and understanding phenomena and concepts. *Composition* is another fundamental means for organizing and understanding phenomena. Composition is a means to organize phenomena in terms of other phenomena.

In many situations it is useful to consider a *whole* as constructed from *parts*. The parts may again consist of smaller, simpler parts i.e. there is a distinction between whole phenomena and their *parts*/phenomena. The notion of a *whole/part composition* is an important means of understanding and organizing complex phenomena.

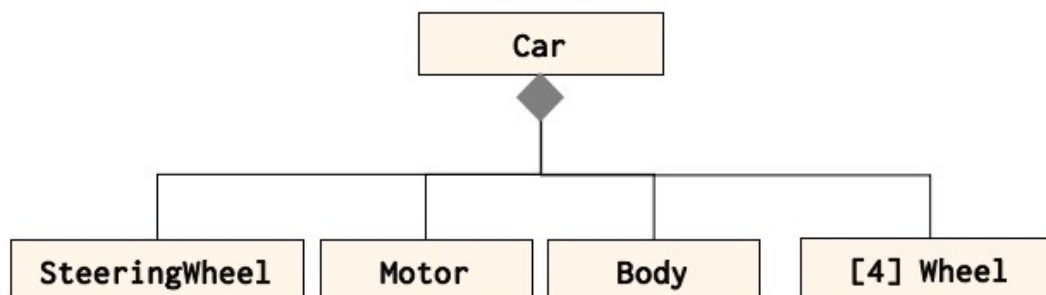
The *part-of relation* gives rise to a *whole/part hierarchy*. The above figure shows an example of such a hierarchy for the MFprinter consisting of parts like `Printer`, and `Scanner`.

A car may be considered as consisting of parts like steering wheel, motor, body and wheels, i.e. wheel is a *part of* a car, a motor is a part of a car, etc. These components are physical parts of the car. We may represent a car as shown below.

```

class Car:
  aSteeringWheel: obj SteeringWheel
  aMotor: obj Motor
  aBody: obj Body
  wheels: obj Array(4, #Wheel)

```



There are numerous other examples of whole/part hierarchies:

- A tree may be viewed as consisting of branches, a trunk, roots and leaves.
- A Hotel Reservation may be viewed as a composition of a Person, a Hotel, a Room and a Date. It is, however, not meaningful to view, for instance, the person as a physical part of the hotel reservation. The corresponding attribute is better viewed as a reference to a person.

- A Person may naturally be viewed as consisting of parts like Head, Body, Arms and Legs. In turn, the Legs consist of Lower leg, Foot, etc.

We define composition in the following way:

To *compose* is to form a compound phenomenon by means of a number of component phenomena. Properties in the intension of the compound phenomenon are described using the component phenomena. The extension of the compound phenomenon consists of phenomena which have components belonging to the extension of the component phenomena.

To *decompose* is to identify a component phenomenon of a phenomenon. Decomposition is the inverse of composition.

Composition gives rise to the following relations:

The *component-of* relation is a relationship between a phenomenon and one of its component phenomena.

The *part-of relation* is a relation between a phenomenon and one of its part phenomena.

## Composition of activities

It is not just physical phenomena as shown in the above examples that may be organised in terms of wholes and parts. This is also the case for activities. An activity described by a method may consist of other activities.

A activity of making a pizza may be understood as a composition of several sub-activities, including: making the dough, making the tomato sauce, preparing the topping, etc. Like an activity may consist of sub-activities, a computation may consist of sub-computations.

As an example from the bank domain, we have that a transfer activity consists of a withdraw activity followed by a deposit activity. This is represented by the transfer method, which is describes a composition of a withdraw computation followed by a deposit computation.