Declaring the class

MatDeck supports a object-oriented programming environment. In object-oriented programming, a class is a program-code-template used for creating objects, providing initial values for the state (member variables) and for implementations of behavior (member functions or methods). In MatDeck, the class name is used as the name for the class (the template itself), the name for the default constructor of the class (a subroutine that creates objects), and as the type of objects generated by the class. In the following segment we give two examples of class declarations, object creation and manipulation in MatDeck.

Declaring class named person

```
1  class person
2  {
3    fnm := 0
4    lnm := 0
5     // Class constructor
6    person(firstName, lastName)
7    {
8      fnm = firstName
9      lnm = lastName
10   }
11    // Member function
12   getName()
13   {
14     return(fnm + " " + lnm)
15   }
16 }
```

After class declaration, we can create objects of the given type in the following manner

```
17  p := person("John", "Smith")
```

We can use the member functions (methods) to manipulate with objects. For example, we can see values related to the object, p, of the type person.

```
18  pp := p.getName()
```

We can show results within a Canvas as follows.

pp = "John Smith" Result obtained by method getName()

p = object of type person Properties of the object p

type(p) = "person" Type of the object p

Declaring class employee

In the following example we create a class employee and two objects of the type employee. Previous coding is done in the Text Mode editor, however we illustrate that the code can be edited in a Canvas as well.

```
class employee
{
  1  per := person("" , "")
  2  pos := ""
     employee(firstName , lastName , position)
     {
  3    1  per = person(firstName , lastName)
       2  pos = position
     }
     getName( )
     {
  4    1  v := per.getName() + " " + pos
       2  return(v)
     }
  5
}
e1 := employee("John" , "Doe" , "unknown")      Object e1 created
e2 := employee("Jane" , "Doe" , "unknown")      Object e2 created

        e1 = object of type employee
        e2 = object of type employee
        type(e1) = "employee"
        type(e2) = "employee"
ep1 := e1.getName( )
ep2 := e2.getName( )
        ep1 = "John Doe unknown"
        ep2 = "Jane Doe unknown"
```