

2.3 The first object

Description

In order to generate objects as part of a model in the computer, we must describe the objects in a language that can instruct the computer to generate the objects. As mentioned, such a language is called a programming language.

Below we show the *description* of an object representing the account of John Smith:

```
account_1010: obj
  owner: val "John Smith"
  balance: var float
```

The keyword `obj` specifies that the program element is a description of an object. The name of the object is `account_1010`.

The object has two *attributes* characterizing the account:

- The owner of the account is represented by the data-item `owner` holding the *string value* "John Smith". A string value (also called *string literal*) is a sequence of characters enclosed in double quotes (").
- The balance of the account is represented by the data-item `balance` that may hold the value of a *real* number. A real number may be a number with a decimal point as 350.56. The variable `balance` may hold this number. For this example we assume that the balance is in Euro. If `balance` holds the value 350.56, this means that `balance` is 350.56 Euro.

The names `String` and `float` are examples of *types*. A type defines the set of values that a given data-item may hold. The data-item `owner` is said to be of type `String` since it may only hold `String` values. Similarly the data-item `balance` is said to be of type `float` since it may only hold real numbers.

Note! A computer can only represent a subset of real numbers. This subset is often called *floating point numbers* and `float` is used as the type instead of `real` in many programming languages.

The keyword `val` is a shorthand for value and specifies that the data-item holds a value that is constant as long as the object exists. The keyword `var` is a shorthand for variable and specifies that the data-item may hold different values during the life-time of the object. We return to this in chapter .

As said above, the code/program fragment above is a description of an object — it is not the object itself as represented in the computer.

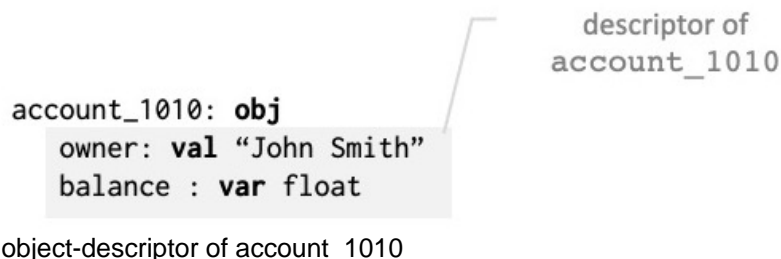
We may give this description to a compiler, which translates it into machine language. We may then execute the resulting machine language, which will generate the object in the computer.

We use the term *object-descriptor* or simply *descriptor* for the text describing the attributes of the object. For `account_1010`, the attributes are the two data-items `owner` and `balance`.

```
account_1010: obj
  owner: val "John Smith"
  balance : var float
```

object-descriptor of account_1010

descriptor of
account_1010



As we shall see later, an object-descriptor may include other elements than data-items.

The description of John Smiths account is made in a programming language called qBeta. As mentioned, a programming language is a formal notation for instructing a computer. The elements of a program are phrases that may describe

declarations, objects, data-items, statements to be executed and combinations of these. The description of `account_1010` is an example of a declaration of a data-item – in this case an object.

A description written in a programming language is called a *computer program* or just a *program*, but is also referred to as *source code* or simply *code*. A collection of programs is often referred as *software*. The description of the above object is thus an example of a program although it is not very useful.