7.1 Class details

Description

A class is a template for objects with identical structure, and has the form:

```
ClassName(Parameters): SuperClass
   Declarations
   Statements
```

A class has a name as specified by *`ClassName`*.

A class may have parameters as specified by *`Parameters`* – if no parameters, the brackets are not needed. The *`Parameters`* may be one or more declarations of data-items or virtual methods and classes. Virtual methods and classes as parameters are explained in chapter .

A class may have a superclass as specified by *`SuperClass`*. We explain superclass in chapter .

The body of a class consists of a sequence of possible declarations and statements as specified by *`Declarations`* and *`Statements`*.

Class instantiation

We have seen two ways of creating objects from a class as shown here:

```
JohnSmithsAccount: obj Account(JohnSmithsProfile)
anAccount: ref Account
anAccount := Account(LindaBerrysProfile)
```

We have explained the difference between using **`obj`** and **`ref`** in sections and .

The declaration:

`JohnSmithsAccount: obj Account(JohnSmithsProfile)`,

and the statement:

`anAccount := Account(LindaBerrysProfile)`

both include an expression `Account(...)`. This expression creates an `Account`-object with actual parameters `(...)` – either `JohnSmithsProfile` or `LindaBerrysProfile`.

The evaluation of the expression returns a reference to the newly created `Account`-object. In `JohnSmithsAccount: obj Account(JohnSmithsProfile)`, `JohnSmithsAccount` is holding this reference.

In the statement `anAccount := Account(LindaBerrysProfile)`, this reference is assigned to `anAccount`.

Execution of the expression `Account(...)` also implies that first the declarations in `Account` are *generated* and second the statements in `Account` are *executed*. Note that even if declarations and statements may be mixed, the declarations are allocated before the statements are executed.

The object `X` executing `Account(...)` is called the invoker and the `Account`-object being generated is called the *callee*.

Instantiation of a singular object takes place in the same way as instantiation of a class, except that there are no actual parameters.