### 16.1.3 External functions

A function may be defined within a module or be declared as being defined externally (i.e. **external**).

***Syntactical Structure***

**external** **function** [ **@deterministic** ] *ExtFunctionIdentifier*

"(" [ { ( *FormalValuePar* | *FormalTimerPar* | *FormalTemplatePar* | *FormalPortPar* ) [","] } ] ")"

[ **return [ template** [ *Restriction* ] ] *Type* ]

***Semantic Description***

For an external function only the function interface has to be provided in the TTCN‑3 module. The realization of the external function is outside the scope of the present document.

Using the @deterministic modifier, an external function can be declared to be deterministic. Deterministic functions are safe to be used when called from specific places where non-determinism could lead to unexpected side effects (see clause 16.1.4).

***Restrictions***

In addition to the general static rules of TTCN‑3 given in clause 5, the following restrictions apply:

1. Void.
2. Void.
3. Restrictions on invoking functions from specific places are described in clause 16.1.4.

NOTE: External functions should only exchange information with the test system via return values and parameter passing. Side-effects that change the status of the test system and may influence the test outcome should be avoided. Such side-effects can occur if an external function contains default handling, configuration, communication or timer operations.

***Examples***

 **external** **function** fx\_myFunction4() **return** **integer**; // External function without parameters

 // which returns an integer value

 **external** **function** fx\_initTestDevices(); // An external function which only has an

 // effect outside the TTCN‑3 module