ADTs:

An abstract data type is a data type having the following two properties:

* the objects from the domain of the ADT are specified independently of their representation
* the operations of the ADT are specified independently of their implementation

The domain of an ADT describes what elements belong to this ADT

* if the domain is finite, we can simply enumerate them
* if the domain is not finite, we will use a rule that describes the elements belonging to the ADT

Operations of an ADT – *Interface*

* the set of all operations for an ADT is called its interface
* the interface of an ADT contains the signature of the operations, together with their input data, results, preconditions and postconditions (but no detail regarding the implementation)

A *container* is a collection of data, in which we can add new elements and from which we can remove elements. A container should provide at least the following operations:

* creating an empty container
* adding a new element to the container
* removing an element from the container
* returning number of elements
* providing access to the elements (usually iterator)