Add to section 12.23 the text shown in blue

The variable count in the example above can only be accessed by the corresponding disp_count method. Each specialization of the class *vector* has its own unique copy of count.

A specialization is the combination of a specific generic class with a unique set of parameters. Two sets of parameters shall be unique unless all parameters are the same as defined by the following rules:

1- A parameter is a type parameter and the two types are matching types.

2- A parameter is a value parameter and both their type and their value are the same.

All matching specializations of a particular generic class shall represent the same type. The set of matching specializations of a generic class is defined by the context of the class declaration. Since generic classes in a package are visible throughout the system, all matching specializations of a package generic class are the same type. In other contexts, such as modules or programs, each instance of the scope containing the generic class declaration creates a unique generic class, thus, defining a new set of matching specializations.

To avoid having to repeat the specialization either in the declaration or to create parameters of that type, a **typedef** should be used:

In Section A.4.1.1 and Syntax 19-5: parameter_value_assignment (BNF)

REPLACE

parameter value assignment ::= # (list of parameter assignments)

WITH

parameter_value_assignment ::= # ([list_of_parameter_assignments])