Tabla Genericos

Generic formal type	Acceptable actual types
`type T (<>) is limited private;`	Any type at all. The actual type can be limited or not, indefinite or definite, but the *generic* treats it as limited and indefinite, i.e. does not assume that assignment is available for the type.
`type T (<>) is private;`	Any nonlimited type: the generic knows that it is possible to assign to variables of this type, but it is not possible to declare objects of this type without initial value.
`type T is private;`	Any nonlimited definite type: the generic knows that it is possible to assign to variables of this type and to declare objects without initial value.
`type T (<>) is abstract tagged limited private;`	Any tagged type, abstract or concrete, limited or not.
`type T (<>) is tagged limited private;`	Any concrete tagged type, limited or not.
`type T (<>) is abstract tagged private;`	Any nonlimited tagged type, abstract or concrete.
`type T (<>) is tagged private;`	Any nonlimited, concrete tagged type.

`type T (<>) is new Parent;`	Any type derived from `Parent`. The generic knows about `Parent`'s operations, so can call them. Neither `T` nor `Parent` can be abstract.
`type T (<>) is abstract new Parent with private;`	Any type, abstract or concrete, derived from `Parent`, where `Parent` is a tagged type, so calls to `T`'s operations can dispatch dynamically.
`type T (<>) is new Parent with private;`	Any concrete type, derived from the tagged type `Parent`.
`type T is (<>);`	Any discrete type: integer, modular, or enumeration.
`type T is range <>;`	Any signed integer type
`type T is mod <>;`	Any modular type
`type T is delta <>;`	Any (non-decimal) fixed point type
`type T is delta <> digits <>;`	Any decimal fixed point type
`type T is digits <>;`	Any floating point type
`type T is array (I) of E;`	Any array type with index of type `I` and elements of type `E` (`I` and `E` could be formal parameters as well)
`type T is access O;`	Any access type pointing to objects of type `O` (`O` could be a formal parameter as well)