

Ada 2012 introduction



11/03/2019

Generalities

Contract programming

Expressions and predicates

SMP & iterators

Introduction to Ada 2012 - 1/9

- Briefly, Ada 2012 introduced :
 - ✓ Contract programming,
 - ✓ Aspect specifications,
 - ✓ More flexible and new kind of expressions,
 - ✓ SMP programming,
 - ✓ Iterators,

√...

 Like its predecessors, Ada 2012 integrates new features that are still focused on security and integrity of the developed applications while embracing new technologies such as SMP technology.



Introduction to Ada 2012 - 2/9

Generalities

Contract programming

Expressions and predicates

SMP & iterators

- A software application (CSCI) is generally composed of several components (CSC/CSU) that are interacting with each others.
- These modules must be as independent as possible from each other and offer a high level of abstraction
 throughout their interfaces.
- Ada 2012 provides the capability to provide a formal
 description of components interfaces by associating semantics via pre-post / conditions, invariants and

predicates.

CSCI : Computer Software Configuration Item CSC : Computer Software Component CSU : Computer Software Unit

> Systerel Solutions temps réel sécurisées

Introduction to Ada 2012 - 3/9

Generalities

Contract programming

Expressions and predicates

SMP & iterators

- For instance, this formalization of description makes it easier to match Low Level Requirements (LLR).
- Ada 2012 allows you to say what the program should do and the language (RTS) ensures that what you said will be done → respect of the contract.





Introduction to Ada 2012 - 4/9

Generalities

Contract programming

Expressions and predicates

SMP & iterators

- Through clear and explicit syntax, the following examples provide a brief overview of how Ada2012 implements the following paradigms or technologies :
 ✓ Contract programming,
 - ✓ New kind of expressions and predicates,
 - ✓ SMP programming,
 - ✓ Iterators.



Generalities

Contract programming

Expressions and predicates

SMP & iterators

Introduction to Ada 2012 - 5/9

• Contract programming (1/2) :

```
function Pop(Stack : in out Stack_T) return Integer
with Pre => not Stack.Empty,
Post => not Stack.Full;
```

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(x-t) \cdot g(t) dt$$



Generalities

Contract programming

Expressions and predicates

SMP & iterators

Introduction to Ada 2012 - 6/9

• Contract programming (2/2) :

```
package Polyhedron is
   type Polyhedron_T is tagged private;
...
private
   type Polyhedron_T is tagged record
        F, -- number of faces.
        E, -- number of edges.
        V : Natural; -- number of vertices.
   end record
```



```
Leonhard Euler
(1707-1783)
```

```
-- Euler theorem (topological invariant)

with Type_Invariant =>

Polyhedron_T.V - Polyhedron_T.E + Polyhedron_T.F = 2;

end Polyhedron;
```



Introduction to Ada 2012 - 7/9

Generalities

Contract programming

Expressions and predicates

SMP & iterators

• New kind of expressions and predicates (1/2) :





Introduction to Ada 2012 - 8/9

Generalities

Contract programming

Expressions and predicates

SMP & iterators

• New kind of expressions and predicates (2/2) :

```
procedure Sort (Vector: in out Vector T) is
with Post =>
 (for all I in Vector'Range =>
   I = Vector'Last or else Vector(I) <= Vector(I+1));</pre>
(for some Element of Matrix => Element /= 0)
(for all Element of Matrix => Element = 0)
package Polyhedron is
    type Solid T is record
    end record;
    subtype Polyhedron T is Solid T
    with Dynamic Predicate =>
         Polyhedron T.V - Polyhedron T.E + Polyhedron T.F = 2;
```

end Polyhedron;



Introduction to Ada 2012 - 9/9

Generalities

Contract

predicates

programming

• SMP programming :







Merci !