

# Introduction au langage Ada

# Question !

Jusqu'où peut-on faire confiance au programmeur ?

Sachant que :

- Des langages comme le C ont été conçus dans l'hypothèse que le programmeur est raisonnable et qu'il sait ce qu'il fait...
- Le programmeur fait son travail mais, sachant que certains sont meilleurs (ou moins bons) que d'autres...

Ada a été conçu et imaginé pour entre autres tenir compte des ces facteurs humains...

L'histoire

Généralités

Qu'y a-t-il dans Ada?

# L'histoire - 1/2

Créé(initié) suite à un contrat du DoD.

Le nom : Ada Byron, comtesse de Lovelace (1815-1852).

Ada83 : Les bases → premier langage industriel avec exceptions, génériques, tasking.

Ada95 : POO, objets protégés, bibliothèques hiérarchiques → premier langage orienté objet normalisé.

Ada2005 : Interfaces, containers, améliorations de l'existant → on rassemble tout et on consolide!

Ada2012 : Contrats, expressions de plus haut niveau → convergence vers les méthodes formelles.

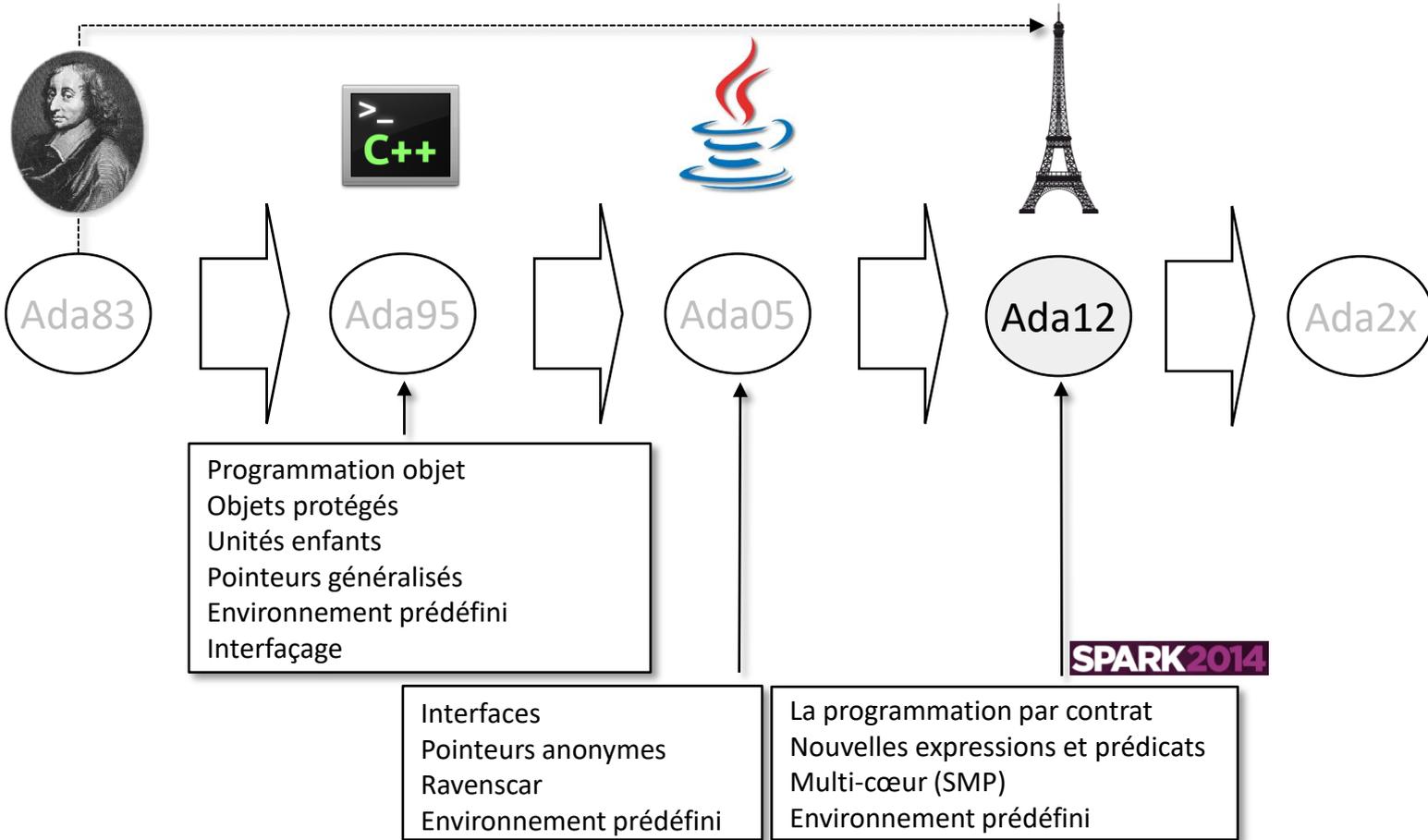
Ada202x : ...

L'histoire

Généralités

Qu'y a-t-il dans Ada?

# L'histoire - 2/2



L'histoire

Généralités

Qu'y a-t-il dans Ada?

## Le langage met l'accent sur :

- La fiabilité
- La sûreté
- La sécurité
- La détection au plus tôt des erreurs
- La maintenabilité
- La portabilité et l'interfaçage (langage et matériel)
- La Programmation Orienté Objet (POO)
- La Programmation Par Contrat (PPC) + approche formelle(SPARK)
- La programmation temps réel
- La programmation concurrente

"La vraie faute est celle qu'on ne corrige pas."

孔夫子

Du plus haut niveau au plus bas.

L'histoire

Généralités

Qu'y a-t-il dans Ada?

Ada vérifie beaucoup de choses à la compilation →  
une mauvaise conception ne compile pas !

L'histoire

Généralités

Qu'y a-t-il dans Ada?

## Pourquoi utiliser Ada?

1. Quand l'application est amenée à être utilisée sur le long terme (10-30 ans),
  - Maintenabilité
  - Portabilité
2. Quand l'application doit satisfaire à de fortes exigences de sûreté ou de sécurité ou de disponibilité (et cela peut être les trois à la fois!).

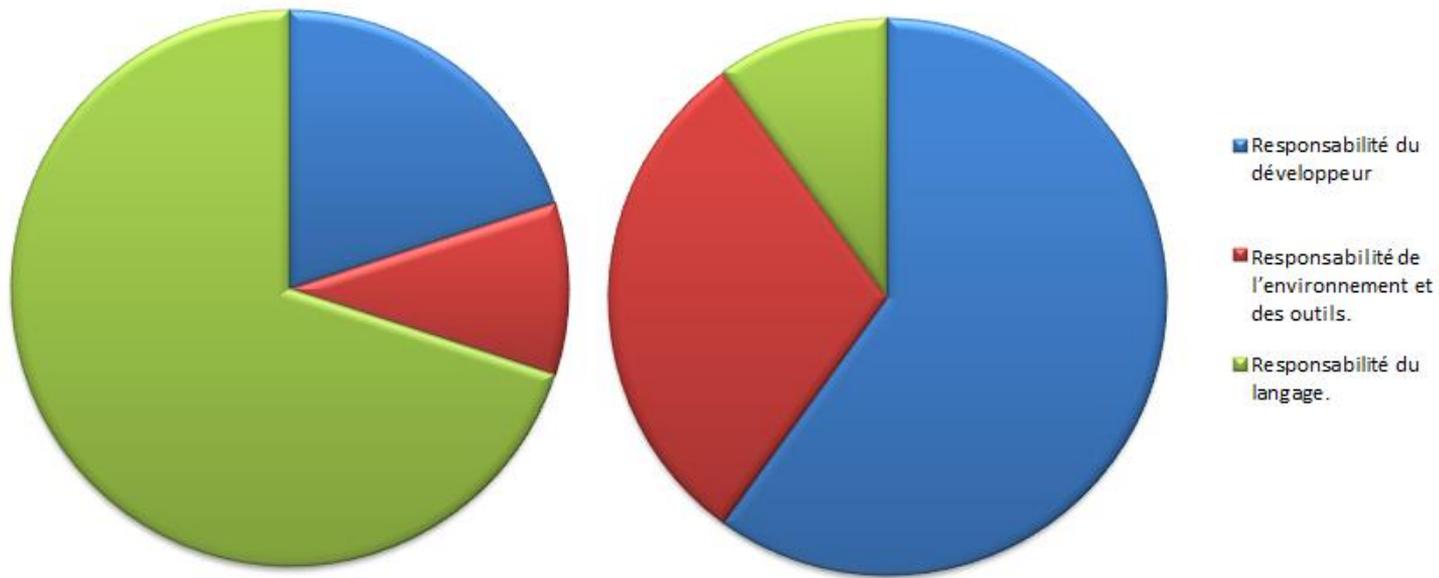


Débordements de buffer, débordements arithmétiques, pointeurs incorrects, fuites mémoire...

# Généralités - 3/6

A titre de comparaison et pour un même niveau de fiabilité on peut estimer que :

- L'histoire
- Généralités
- Qu'y a-t-il dans Ada?



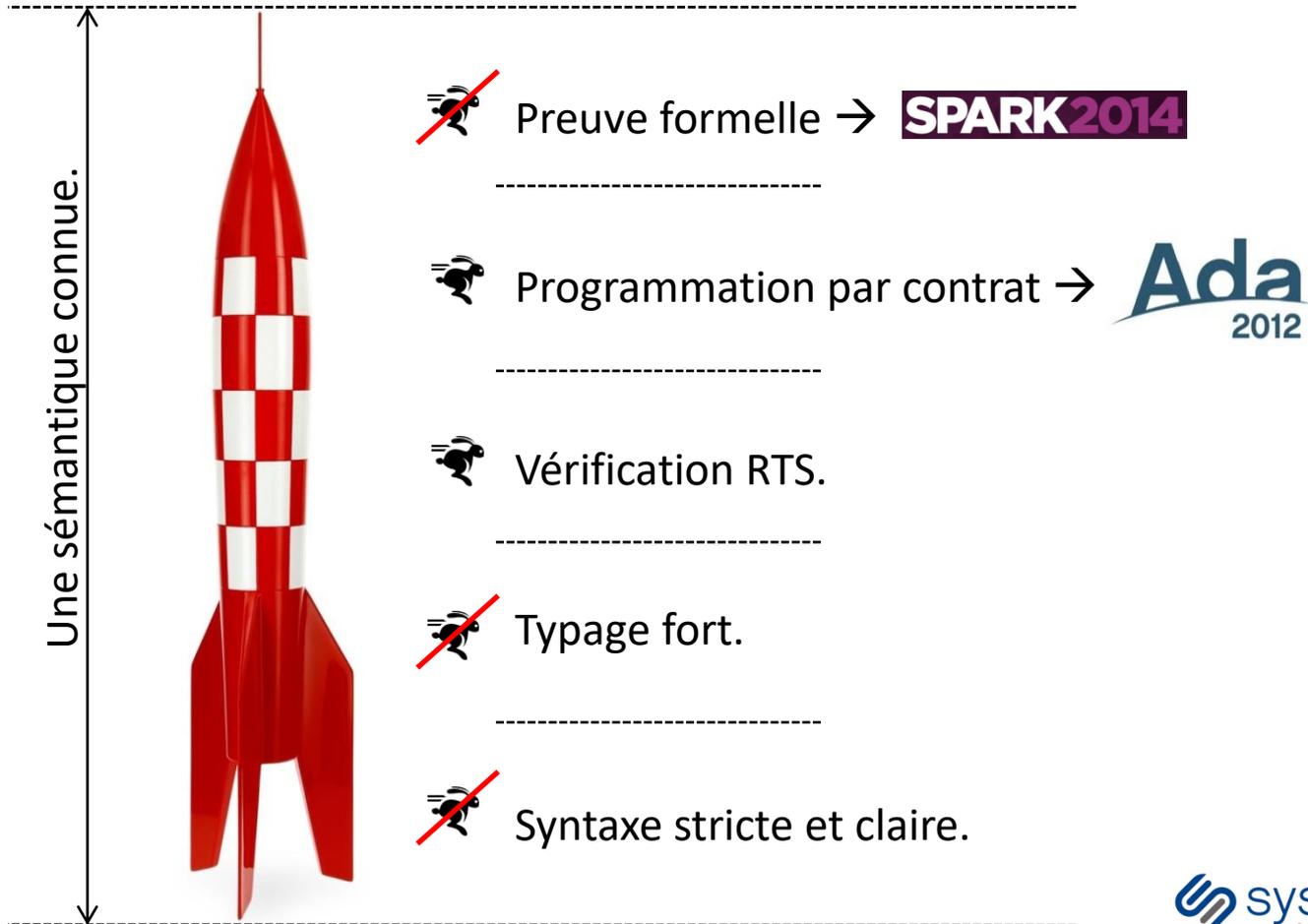
→ 73 mots réservés (Ada<sub>2012</sub>)

→ 44 mots réservés (C<sub>11</sub>)



# Généralités - 5/6

Ada est une fusée à plusieurs étages :



L'histoire

Généralités

Qu'y a-t-il dans Ada?

L'histoire

Généralités

Qu'y a-t-il dans Ada?

## Où trouve-t-on de l'Ada?

- Dans le ferroviaire : TGV, métro, ATS, ...
- Dans l'avionique : A380, A400M, B777, ...
- Dans la défense : Rafale, NH90, M51, Tigre, ...
- Dans le spatial : Ariane 5, 6, ATV, ...
- Dans le contrôle aérien.
- Dans le nucléaire.
- Dans la banque.
- Ada Web Server ...
- Autre : Canal+, ...
- ...



# Qu'y a-t-il dans Ada? - 1/9

L'histoire

Généralités

Qu'y a-t-il dans Ada?

Syntaxe claire et lisible

Typage fort

Orienté objet (classes, héritage, interfaces)

Exceptions

Génériques

Pointeurs

Parallélisme

Accès au bas niveau et interfaçages (SW et HW)

Programmation par contrat

Itérateurs

Annexes pour besoins spécialisés

# Qu'y a-t-il dans Ada? - 2/9

## Syntaxe claire et lisible :

```
Tellurique : constant Boolean := (case Planete is
when Mercure |
      Venus   |
      Terre   |
      Mars    => True,
when Jupiter |
      Saturne |
      Uranus  |
      Neptune |
      Pluton  => False);
```



## Typage fort :

```
type Personne_T is (Pere, Mere, Frere, Cousin);
type Diptere_T  is (Mouche, Moustique, Cousin);
```

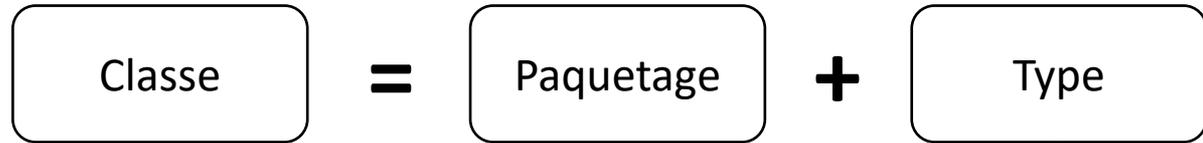
```
Personne : Personne_T := Cousin;
Diptere   : Diptere_T  := Cousin;
```

```
Personne := Diptere;
```

# Qu'y a-t-il dans Ada? - 3/9

## Orienté objet (classes, héritage, interfaces) :

- Un modèle atypique :



**package** Image **is**



```
type Image_T (Largeur  : Natural;  
              Hauteur  : Natural) is tagged private;
```

```
function Filtrer (Image : in Image_T) return Image_T;
```

```
function "+" (Gauche, Droite : in Image_T) return Image_T;
```

```
end Image;
```

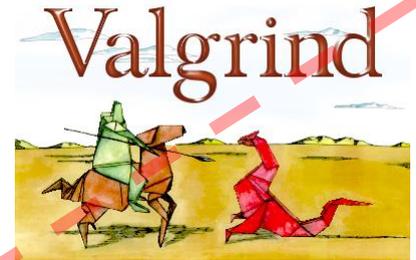
L'histoire

Généralités

Qu'y a-t-il dans Ada?

## Exceptions :

- Ce mécanisme permet de détecter au plus tôt :
  - Des erreurs de design ou système!
  - Des erreurs de codage,
  - Des erreurs sur les interfaces (Sw et Hw).
- Et d'adopter la stratégie adéquate :
  - Mise en repli sécuritaire,
  - Tentative de recouvrement,
  - Trace de debug,
  - L'ignorer!
  - ...



## Génériques :

- Mécanisme simple, efficace et sûr permettant de créer des composants logiciel applicables à un grand nombre de situations.

**generic**

```
type Index_T is (<>);
```

```
type Element_T is private;
```

```
type Vector_T is array (Index_T range <>) of Element_T;
```

```
with function Cmp (G,D : in Element_T) return Boolean;
```

```
procedure Sort (Vector : in out Vector_T);
```

```
procedure Trier_Images is new Sort (Index_T    => Natural,  
                                     Element_T  => Image_T,  
                                     Vector_T   => Tab_Image_T,  
                                     Cmp        => "<");
```



L'histoire

Généralités

Qu'y a-t-il dans Ada?

## Pointeurs :

- Un modèle pointeur souple et sûr
- Pas de références fantômes
- Une gestion de la mémoire à discrétion

## Parallélisme :

- Le parallélisme fait partie intégrante du langage
- Les tâches sont des objets de haut niveau
- Communication et synchronisation de haut niveau
- Programmation SMP (multi-cœur)
- Gestion simple des interruptions

## Accès au bas niveau et interfaçages (SW et HW) :

- Alignement, tailles, champs de bits, Endianness
- Prise en compte des conventions de représentation des autres langages
- Un mécanisme de dérivation permet d'adapter les clauses de représentations d'une donnée à son contexte d'utilisation (abstraction élevée ↔ représentation machine)

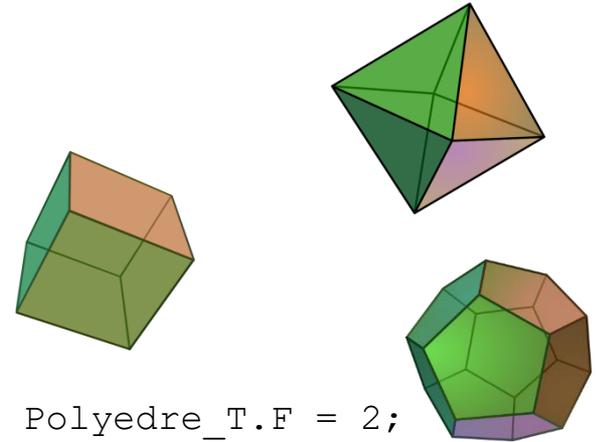


Garantit une excellente portabilité.

## La programmation par contrat :

```
procedure Trier (Vecteur: in out Vecteur_T) is  
with Post =>  
  (for all I in Vecteur'Range =>  
    I = Vecteur'Last or else Vecteur(I) <= Vecteur(I+1));
```

```
package Polyedre is  
  type Solide_T is record  
    ...  
  end record;  
  
  subtype Polyedre_T is Solide_T  
  with Dynamic_Predicate =>  
    Polyedre_T.S - Polyedre_T.A + Polyedre_T.F = 2;  
  
end Polyedre;
```



L'histoire

Généralités

Qu'y a-t-il dans Ada?

L'histoire

Généralités

Qu'y a-t-il dans Ada?

## Annexes pour besoins spécialisés :

- Programmation système
- Temps-réel
- Systèmes distribués
- Sûreté et sécurité
- ...

## Et plus...

- Restrictions et profils (exemple Ravenscar)
- L'environnement prédéfini et bindings
- ASIS (Ada Semantic Interface Specification)
- ...

*Merci*