

booléen.

## Littéraux

Nous avons déjà vu quelques littéraux, même si nous ne savons pas de quoi il s'agit. Un littéral est un morceau de données dans le code source comme le nombre 125 ou la chaîne "Bonjour, de la part de Ada." dans le listing 1 de la partie 1 ; ce sont des littéraux numériques et chaînes, respectivement.

En Ada également, il existe des caractères littéraux, tels que 'C'. Une chaîne est constituée de caractères, de telle sorte que des caractères peuvent être ajoutés à des chaînes en utilisant l'opérateur &, comme dans le listing 2 de la partie 1, par exemple "Hello " & 'G' est identique à "Hello G". Il est possible de créer des variables de type Character comme cela a été fait précédemment avec Integer, Natural et Positive.

Pour le type booléen, tel qu'il a été indiqué plus haut, les valeurs True et False sont des littéraux booléens.

## Plus d'attributs

Nous avons déjà fait connaissance avec l'attribut 'Image, mais nous allons maintenant aborder trois autres attributs de type entier qui peuvent se révéler très utiles. Chaque type d'entier possède un

### Fonctionnalités sympas : les attributs

Beaucoup d'entités du langage possèdent des attributs qui peuvent fournir des informations au programmeur. Pour y accéder, il faut utiliser l'apostrophe ('). Dans nos exemples, nous avons vu les attributs des 3 types d'entiers : 'Image (), 'First, 'Last et 'Range.

intervalle de valeur acceptable, cet intervalle étant défini par les attributs 'First (premier) et 'Last (dernier), il est également possible d'y accéder en utilisant l'attribut 'Range, qui concrètement retourne 'First .. 'Last où les deux points (tréma) signifient "c'est un intervalle de valeurs."

Si vous essayez d'affecter une valeur en dehors de cette plage à une variable d'un type, cela provoquera une erreur et votre programme ne fonctionnera pas correctement.

Une autre utilisation de 'First consiste à affecter une valeur initiale à une variable. Dans le listing 2, la valeur initiale affectée à une variable vient après le symbole := qui est lui-même situé après le nom du type.

## Exercices

1. Utilisez l'attribut Integer'Image pour afficher les valeurs retournées par les attributs Integer'First et Integer'Last.
2. Refaites l'exercice 1 avec les types Positive et Natural.
3. Essayez de remplacer les valeurs initiales avec l'attribut 'First.

### UN PEU D'HISTOIRE

Dans les années 70, le Ministère de la Défense des États-Unis (DoD) décida qu'un langage de programmation devrait être utilisé pour l'ensemble de ses projets. Plusieurs équipes à travers le monde envoyèrent des propositions pour un nouveau langage, chacune nommée d'après une couleur : Rouge, Verte, Bleue et Jaune. Parmi toutes, c'est l'équipe verte dirigée par Jean Ichbiah de CII Honeywell Bull en France qui gagna.

La première version du langage vert fut adoptée par les équipes de développement du DoD et il a été dit de l'utiliser pour tous les projets ultérieurs. Le langage vert fut ensuite renommé Ada d'après le nom de la fille de lord Byron, Augusta Ada (comtesse Lovelace), qui a travaillé aux côtés de Charles Babbage et qui est considérée

comme la première programmeuse du monde en raison du travail sur la Machine Analytique de ce dernier.

La première version du langage Ada sortit vers 1983, sous le nom Ada 83, la version suivante parut au cours des années 90, initialement appelée Ada9X et renommée plus tard Ada95, la suivante fut Ada 2005 et cette année fut distribuée une autre révision appelée Ada 2012. Le compilateur GNAT est toujours en cours de développement pour pouvoir prendre en charge les fonctionnalités du nouvel Ada 2012, donc malheureusement, certaines d'entre-elles ne fonctionneront pas si vous les essayez. Pour nos programmes, nous pouvons assurément utiliser Ada95 mais aussi la plupart des fonctionnalités d'Ada 2005.