

Inférence de types

L'**inférence de types** est un mécanisme qui permet à un compilateur ou un interpréteur de rechercher automatiquement les types associés à des expressions, sans qu'ils soient indiqués explicitement dans le code source.

Il s'agit pour le compilateur ou l'interpréteur de trouver le type le plus général que puisse prendre l'expression. Les avantages à disposer de ce mécanisme sont multiples : le code source est plus aéré, le développeur n'a pas à se soucier de retenir les noms de types, l'interpréteur fournit un moyen au développeur de vérifier (en partie) le code qu'il a écrit et le programme est peu modifié en cas de changement de structure de données. Les expressions restent les plus générales possibles — l'inférence de types va de pair avec le polymorphisme, puisque le compilateur génère des types abstraits au besoin. Enfin, le compilateur, lui, dispose de toutes les informations de type, il y a un typage fort et statique, ce qui lui permet de produire du code plus sûr et plus efficace.

Le mécanisme d'inférence de type est présent dans certains langages de programmation fonctionnelle.

Exemple

OCaml est un exemple de langage qui utilise l'inférence de type :

```
let rec longueur = function
| [] -> 0
| h::t -> 1 + (longueur t);;
```

Le code présenté montre qu'il s'agit d'une fonction qui prend en entrée une liste quelconque (vide ou non) et qui renvoie un entier. En fait, cette fonction permet de calculer la longueur d'une liste. Lorsque ce morceau de code est envoyé à l'interpréteur, celui-ci répond :

```
longueur : 'a list -> int = <fun>
```

Caml indique ainsi qu'il reconnaît que la fonction `longueur` prend en entrée un quelque chose de type `'a list`, c'est-à-dire une liste quelconque (d'entiers, de réels, de chaînes de caractères, etc.). Caml reconnaît aussi que la fonction renvoie une valeur de type entier (`int`).

C# est un autre exemple de langage qui utilise l'inférence de type.

```
var name = "James Bond";
var age = 25;
var isSecret = true;
var r = 2.52;

Type nameType = name.GetType();
Type ageType = age.GetType();
Type isSecretType = isSecret.GetType();
Type rType = r.GetType();

Console.WriteLine("name is type : " + nameType.ToString());
Console.WriteLine("age is type : " + ageType.ToString());
Console.WriteLine("isSecret is type : " + isSecretType.ToString());
Console.WriteLine("r is type : " + rType.ToString());
```

Exemples de langages utilisant l'inférence de type

- [Boo](#)
- [C#](#)
- [C++11](#)
- [Caml](#)
- [Crystal](#)
- [D](#)
- [F#](#)
- [Go](#)
- [Haskell](#)
- [Haxe](#)
- [Java 8](#)¹
- [Kotlin](#)
- [ML](#)
- [OCaml](#)
- [Rust](#)
- [Scala](#)
- [Scol](#)
- [Swift](#)
- [Vala](#)

Références

- (en)** « Type Inference » (<https://docs.oracle.com/javase/tutorial/java/generics/genTypeInference.html>), The Java™ Tutorials, Oracle Java Documentation.

Ce document provient de « https://fr.wikipedia.org/w/index.php?title=Inf%C3%A9rence_de_types&oldid=154171629 ».

La dernière modification de cette page a été faite le 22 novembre 2018 à 18:44.

Droit d'auteur : les textes sont disponibles sous [licence Creative Commons attribution](#), [partage dans les mêmes conditions](#) ; d'autres conditions peuvent s'appliquer. Voyez les [conditions d'utilisation](#) pour plus de détails, ainsi que les [crédits graphiques](#). En cas de réutilisation des textes de cette page, voyez [comment citer les auteurs et mentionner la licence](#).

Wikipedia® est une marque déposée de la [Wikimedia Foundation, Inc.](#), organisation de bienfaisance régie par le paragraphe [501\(c\)\(3\)](#) du code fiscal des États-Unis.