Eiffel

* Design-by-contract, invariants, code-as-documentation
* Command-query separation
* Code re-use, extensibility
* Null safety, genericity, type safety

Eiffel a été conçu avec un seul objectif : améliorer la qualité de logiciel. La plupart de sa fonctionnalité est destinée à servir cet objectif. Peut-être la caractéristique la plus inhabituelle est « la conception par contrat ».

# La conception par contrat

Très souvent, lorsque les informaticiens veulent prouver qu’un algorithme est correct, ils utilisent les « préconditions », les « postconditions » et les « invariants ». Si on examine un fragment de code, une précondition est un fait qui doit être vrai juste avant l’exécution du code. De même, une postcondition doit être vrai juste après l’exécution. Un invariant doit être vrai en tout temps.

La plupart des langages n’encodent pas ces concepts mathématiques, mais Eiffel en encourage.

# Glossary

Reusability réutilisabilité

Extensibility extensibilité

Efficiency efficacité

Libraries bibliothèques

Reliability fiabilité

Calling function la fonction appelante

Called function la fonction appelée

Keywords les mots-clés

Design by contract la conception par contrat

Generic classes les classes génériques

Inheritance l’héritage

Precondition la précondition

Postcondition la postcondition