

Eiffel et OCaml—deux langages, deux approches pour obtenir la qualité des logiciels

De nos jours, des logiciels sont partout : des portables, des avions, des lave-vaisselle, des lampes.

Pour citer Marc Andreessen, « les logiciels mangent le monde ». Malheureusement, ces logiciels sont souvent bogués, même dangereuses.

Depuis de nombreuses années, l'industrie du logiciel essaye de trouver des méthodes pour améliorer la qualité de leur logiciel. Plusieurs éminents informaticiens français ont développé des nouveaux langages à cette fin. Dans ce rapport, je décrirai deux langages de programmation : Eiffel et OCaml. Les deux tentent d'assurer la sécurité mais ils ont des approches très différentes. Je vais examiner et comparer leurs fonctions et leurs caractéristiques.

Bien qu'OCaml—datant de 1996—soit plus jeune qu'Eiffel—de 1986—ses origines sont plutôt plus âgées. OCaml dérive de ML, développé à Edimbourg en 1973. OCaml lui-même a débuté à INRIA, l'institut de recherche français. Eiffel a été créé par Bertrand Meyer, un informaticien français qui est actuellement professeur de génie logiciel à ETH Zurich.

Eiffel est un langage « orienté objet », ce qui signifie qu'il modèle le monde comme une collection d'objets. Chaque objet peut effectuer des actions (« commandes »), et chaque objet a des propriétés (« requêtes »).

Par contre, OCaml est un langage multi-paradigme—mais, premièrement, il est un langage fonctionnel. Dans un tel langage, des informations et des actions sont strictement séparées. Nous examinerons les conséquences de ces conceptions dans le reste du projet.

Pensez
ROSS, c'est assez bien mais à bien préciser ce qui constituera chacune des parties - Pour la langue, soyez attentif aux accords grammaticaux et aux articles choisis. Travaillez bien !