

MGL7460 – Réalisation et maintenance des logiciels

Langages de programmation dynamique

Jacques Berger

Objectifs

Introduire les langages de programmation dynamique

Prérequis

Programmation

Langages courants

Les langages usuels tentent d'empêcher les développeurs de faire les "mauvaises" choses

Analogie du ciseau à bouts ronds

Langages courants

Exemples de "mauvaises" choses :

Héritage multiple

Surcharge d'opérateurs

Métaprogrammation

Langages courants

La lourdeur du typage statique freine l'expressivité des développeurs

La programmation fonctionnelle n'est pas vraiment utilisable

La métaprogrammation est souvent absente

Typage

Statique : le type de l'objet est connu à la compilation, il ne peut pas être modifié durant l'exécution

Dynamique : le type de l'objet est défini à l'exécution; l'objet est défini par ce qu'il fait et non ce qu'il est

Typage

Fort : lors de l'exécution, le type est vérifié et respecté; impose une conversion explicite lors d'un changement de type

Faible : ne respecte pas le type lors de l'exécution (ex: int + char)

Typage

Statique fort : Java, C#

Statique faible : C, C++

Dynamique fort : Groovy, Python, Ruby

Dynamique faible : Perl, Javascript, PHP

Métaprogrammation

Introspection

Introduction (intercession)

Protocole Méta Objet (MOP)

Groovy

Compatible avec Java, compile sur la JVM

Plus connu dans la communauté Java

Ruby

Programmation fonctionnelle, orienté-objet,
impérative, réflexive

Communauté active

Python

Très répandu dans l'industrie et dans le milieu académique

Très forte communauté