INF2015 – Développement de logiciels dans un environnement Agile

Langages de programmation dynamique

Jacques Berger

Objectifs

Introduire les langages dynamiques

Présenter Groovy

Prérequis

Java

Langage dynamique

Langage de programmation dynamique

- Langage de plus haut niveau
- Utilise le typage dynamique
- Interprété à l'exécution

Langage dynamique

Exemples :

- Ruby
- Python
- Groovy

Langage dynamique

Généralement :

- Moins de ligne de code
- Productivité accrue
- Forte flexibilité
- Très peu limitatif
- Programmation fonctionnelle

Pourquoi?

Pourquoi utiliser un langage de programmation dynamique?

- Améliorer la productivité des développeurs
- Offrir plus de liberté aux développeurs

Pourquoi?

Les langages courants ont tendance à interdire les pratiques qu'ils considèrent mauvaises

Ex.

- Héritage multiple
- Surcharge d'opérateurs

Pourquoi?

Les restrictions des langages courants (ex. typage statique) limitent les développeurs

Certains savent ce qu'ils font...

Les langages courants sont comme des ciseaux à bouts ronds

Typage

Statique (ex. Java)
- Le type doit être connu à la compilation

Dynamique (ex. Groovy)
- Le type est connu à l'exécution
- Le type peut changer durant l'exécution

Groovy

Version courante : 2.1
Version avec NetBeans 7.3 : 2.0.1

Première version publiée en 2007

Groovy

Langage fortement influencé par Java et Ruby

Langage de très haut niveau
- API riche
- Code expressif
- Beaucoup de fonctionnalités en peu de lignes

Groovy

Conçu pour cohabiter avec Java

- Roule sur la JVM (bytecode)
- 100% compatible avec Java, dans les 2 sens
- Syntaxe similaire à Java

Groovy

Facilite le travail avec :

- XML
- HTML
- SQL
- JSON
- HTTP

Langage

Coding by Convention
- Même concept que Convention over configuration
- Getters / setters
- Éléments publiques

Langage

Peut faire :

- Du scriptage
- De la programmation procédurale
- De la programmation orientée-objet
- De la programmation fonctionnelle
- De la métaprogrammation

Grails

Version courante : 2.2.1

Signifiait originalement Groovy on Rails

Plateforme de développement web

Programmation fonctionnelle

Les fonctions peuvent être placées dans des variables ou passées en paramètre à une autre fonction

La closure (ou fermeture) est disponible avec Groovy

Programmation fonctionnelle

Une closure peut aider à réduire le nombre de lignes de code en favorisant la réutilisation

Une closure est l'équivalent d'un pointeur de fonction (C, C++) à l'exception que le code de la closure peut faire référence aux variables du scope qui la contient

Avantages

Se mélange très bien avec Java

Beaucoup moins de lignes de code
- Meilleure maintenabilité
- Meilleure lisibilité
- Meilleure productivité

S'apprend rapidement pour un développeur Java

Inconvénients

Temps d'exécution plus lent

Plusieurs concepts à maîtriser
- Closure
- Programmation fonctionnelle
- Typage dynamique
- Conventions

Confusion avec Java causée par le mélange des langages

Liens

Groovy

http://groovy.codehaus.org/

Grails

http://grails.org/