

Plan du cours

Slide

1) introduction (environ 1/2 de journée)

- Qu'est-ce que Java, une JVM
- Le framework Java EE, les différents serveurs Applicatif J2EE ?
- Qu'est-ce qu'Eclipse ? Comment le configurer ?
- Le debuggage

Travaux pratiques

Installation du JDK, gestion de la variable d'environnement classpath. Installation puis utilisation de l'IDE Eclipse en perspective Java, utilisation de Eclipse en mode debug.

2) Les constructions de base du langage (environ 3/4 de journée)

- Les variables : déclaration et typage.
- Les méthodes : définition.
- Les expressions.
- Les instructions de contrôle : les instructions conditionnelles, de boucle, de branchement.
- Les tableaux.
- Les unités de compilation et packages : le contrôle de la visibilité des classes, le mécanisme d'import.
- Les imports statiques.

Travaux pratiques

Suite d'exercices simples permettant la prise en main de l'environnement de développement (Eclipse en perspective Java), notamment la documentation et la réalisation d'un programme

simple. Utilisation des packages.

3) Tests logiciels (environ 3/4 de journée)

- Pourquoi faire des tests ?
- Présentation des différents types de tests : tests unitaires, fonctionnels, de robustesse et de performance.
- Quels tests lancer et quand ?
- Utilité des objets "Mock" et "Fake" durant les tests unitaires. Couverture des tests unitaires.

Travaux pratiques

Installer JUnit sous Eclipse. Ajouter des tests unitaires avec JUnit sur les projets Eclipse déjà écrits.

4) Bonnes pratiques de conception d'une application (environ 1/2 de journée)

- Découpage en couches (données, métier, présentation).
- Présentation des enjeux d'un développement d'entreprise.
- Introduction à l'écosystème JAVA (JEE, Spring, Hibernate, JSF ou Struts...).

Travaux pratiques

Echanges quant au choix technique du framework Java EE Web Profile. Réflexion sur la conception en couche.

5) Les techniques Objet (environ 1/2 journée)

- Les principes généraux de la modélisation et de la programmation Objet.
- L'abstraction et l'encapsulation : les interfaces.
- Les différentes formes d'héritage, le polymorphisme.
- Introduction à la modélisation UML

Travaux pratiques

La notion d'identité simple sera étendue dans l'application de registre d'identité afin de faire apparaître des particularités.

6) La gestion des versions : introduction à SVN (environ 3/4 de journée)

- Les concepts généraux liés à la gestion de versions.
- Les concepts SVN : dépôt, projets, révisions, tronc, branches et tags.
- Les principales opérations offertes au développeur. La gestion des conflits.
- La gestion des branches. Les perspectives SVN proposées par les plug-ins Eclipse.

Travaux pratiques

Installation d'une solution de versioning, installation du bon plug-in dans Eclipse. Création de projets associés à un repository. Gestion des versions de l'application développée - récupération d'une copie locale, modifications, fusion, commit.

7) Définition de la structure d'un projet avec Maven (environ 3/4 de journée)

- Définition de la structure d'un projet.
- Les conventions. Les dépendances entre projets. Les tâches prédéfinies : compilation, génération d'archives...
- Les perspectives Maven proposées par les plug-ins Eclipse.

Travaux pratiques

Utilisation de Maven sous Eclipse pour automatiser une succession de traitements notamment de test.

8) Quelques Design Pattern (environ 1/4 de journée)

- Les objectifs et les avantages.
- Les Design Patterns les plus populaires dans les architectures logicielles modernes.

9) mesures de la qualité (environ 1/4 de journée)

- Synthèse des mesures qualité.
 - La convention de codage et la documentation.
 - La couverture de tests et l'automatisation des procédures.
 - Mise en place d'un tableau de bord de la qualité.
-

Revision #2

Created 27 October 2019 14:16:57 by Admin

Updated 8 September 2020 09:48:34 by ggpilou2