

Contenu du cours

Alexis Nasr (d'après les slides de Arnaud Labourel)



S'initier à toutes les étapes du développement de logiciels

- 1 **Analyser** les besoins
- 2 **Spécifier** les comportements du programme
- 3 **Choisir** et éventuellement **concevoir** les solutions techniques
- 4 **Implémenter** le programme (coder)
- 5 **Vérifier** que le programme a le comportement spécifié (tester)
- 6 **Déployer** le programme dans son environnement, **fournir** une documentation dans le cas de bibliothèque
- 7 **Maintenir** le programme (corriger les bugs, ajouter des fonctionnalités)

Cours 1

- Le penser objet
- Vocabulaire programmation objet
- La gestion de version avec le logiciel git

Cours 2

- Classes utiles et types primitifs
- Structures de contrôles
- Types paramétrés
- Tests et développement

Cours 3

- Interfaces

Cours 4

- Composition et délégation
- Classes abstraites et extension
- Extension et accessibilité

Cours 5

- Surcharge de méthode/constructeurs
- Exceptions
- Les énumérations
- Mot-clé `final`

Cours 6

- Commentaires et documentation
- Structure d'un projet et paquets
- Classe interne

Cours 7

- Types paramétrés (notions avancées)
- Interfaces (notions avancées)

Cours 8

- Une méthodologie pour bien nommer
- Des principes pour bien programmer
- La gestion de projet
- Tests unitaires et développement par les tests (TDD)
- Les principes solides

- 6 ou 7 TP à réaliser en une semaine
 - ▶ début la semaine prochaine
 - ▶ le TP est à rendre avant **dimanche minuit** sur etulab
- 1 mini projet à réaliser en 3 ou 4 semaines
- Les TP sont faits en binôme
 - ▶ la composition des binômes doit être faite avant la première séance de TP
 - ▶ une fois un binôme constitué, vous envoyez un mail à votre encadrant de TP

Outils pour les TP

- un ordinateur
- une machine virtuelle JAVA (JVM)
- un compilateur JAVA
- un éditeur
- git
- gradle
- junit
- vous pouvez utiliser IntelliJ mais ce n'est pas obligatoire et déconseillé pour les premiers TP
 - ▶ pour compiler et exécuter un programme JAVA depuis le terminal
 - `javac MaClasse.java`
produit `MaClasse.class`
 - `java MaClasse`
exécute la méthode `main` de `MaClasse`