

Programmation Orientée Objet

Programmation JAVA

FISE 3A ICy / FISA 3A Informatique

CM0 : Introduction et présentation

Mohamed Amine BOUDIA

UPHF, CNRS, UMR 8201 - LAMIH, F-59313 Valenciennes, France

Email : mohamedamine.boudia@uphf.fr



Objectif Zéro Papier



Intervenants

- ▶ Responsable du module : Mohamed Amine BOUDIA
- ▶ A groupes TD : Mohamed Amine BOUDIA
- ▶ 2 groupes TP : Mohamed Amine BOUDIA

(Pas de changement de groupe possible)



app.wooclap.com/IYEPLQ



1

Allez sur wooclap.com

2

Entrez le code d'événement dans le bandeau supérieur

Code d'événement
IYEPLQ



1

Envoyez [@IYEPLQ](https://twitter.com/IYEPLQ) au **06 44 60 96 62**

2

Vous pouvez participer

 Désactiver les réponses par SMS

The background features abstract, overlapping geometric shapes in various shades of blue, ranging from light sky blue to deep navy blue. The shapes are primarily triangles and polygons, creating a dynamic, layered effect. The central text is positioned in the white space between these blue elements.

Organisation et Instructions de travail pour les étudiants

Bienvenue 😊 !

▶ **Joie** et **bien être**

▶ Mes recommandations :

- Présence et **Concentration**
- Le **plaisir** d'apprendre et de découvrir.
- **éloquent**
- « L'avenir nous tourmente, le passé nous retient, c'est pour ça que le présent nous échappe. »
Gustave Flaubert

Bienvenue 😊 !

- ▶ Mes demandes
 - ▶ Respect : **Silence** en CM, TD et TP
 - ▶ Participation en CM et TD : principe de **SORA** : se **S**ituer, **O**bserver, **R**éfléchir et **A**gir, prenez le temps de réfléchir, écrivez des notes puis levez la main avant de parler.
 - ▶ « L'erreur est une formidable opportunité d'apprentissage. » Jane Nelsen
 - ▶ « **Travailler**, c'est prier. » Saint Bernard

Pendant le Cours magistral (CM)

- ▶ « si la parole est d'argent le silence est d'or » en CM
- ▶ Pour parler vous devez **demander la parole** en levant la main et attendre.
- ▶ Chaque début de cours, 10 minutes sont réservé pour répondre aux questions concernant le cours précédent.
- ▶ Après **la fin des cours**, vous pouvez trouver les supports de cours sur **Moodle**.
- ▶ En cours, vous êtes inviter à **prendre des notes** en complément du support de cours

Après le CM

- ▶ **Réviser** votre cours et apprenez ce que doit être appris.
- ▶ Faites une **fiche de synthèse** pour chaque cours.
- ▶ Créez et partagez entre vous des **QCM** de révision, n'hésitez pas à mettre votre enseignant en boucle
 - Notez les points que vous ne comprenez pas .

Pendant le TD :

Une liste d'exercices est distribuée (disponible sur moodle, **objectif 0 papier**)

La séance de TD est divisé en trois parties

1. *Travail "individuel"*

- Vous devez résoudre les exercices individuellement.
- Vous pouvez échanger vos solutions avec vos voisins et en discuter doucement
- Passez à l'exercice suivant.

2. *Proposition et discussion de correction*

- Un étudiant ou étudiante **désigné** par l'enseignant **propose** une solution au tableau
- Les étudiants **discutent** la solution, l'enseignant **distribue** la parole.

3. *Lorsque la solution est **validé**, vous pouvez la recopier.*

Après le TD

- ▶ Essayez de **refaire** seul les exercices **sans** lire les solutions.
- ▶ Notez **vos lacunes** et **vos questions** .
- ▶ Travaillez en **groupe** .

- ▶ En cas où vous arrivez pas à résoudre l'exercice :
 - Demander **à vos camarades** .
 - **Revenez** vers l'enseignant au début du TD suivant .

En TP

- ▶ Vous pouvez travailler pendant le TP en **binôme ou seul**.
- ▶ Pour chaque question de TP:
 - Prenez le temps pour **comprendre** la question
 - **Ecrivez** le programme en **LDA** (Langage Descriptive Algorithmique)
 - **Testez** votre algorithme avant de passer à la machine
 - **Programmez**
 - **Testez plusieurs fois** en exécutant, et vérifiez la correction (ce que vous espérez voir) et la robustesse (les cas différents à tester)

Après les TP

- ▶ Si vous disposez d'un ordinateur, **installez** Java avec des environnement de travail (NetBeans ou Eclipse..)
- ▶ Sinon, accès libre dans *certaines* salles TP lorsqu'elles sont libres
 - **Renseignez-vous !**
- ▶ Le travail personnel est **le secret** de la réussite :
 - terminer (si possible) le sujet TP,
 - faire les Devoirs maisons (DM),
 - programmer les exercices de TD

Ce qui peut vous faire perdre des points en TP

- ▶ Ne pas respecter **les consignes écrites** dans le TP
- ▶ **Pas de nom d'étudiant** dans le fichier :
- ▶ Commentaire **absent** ou **inadapté**
- ▶ **Pas de type** des paramètres et des **variables locales**
- ▶ **Problème d'exécution**
- ▶ **Affichage pas clair** : on ne sait pas à quoi correspond l'affichage,
- ▶ On n'a pas de **message pour saisir** une valeur

Compétences

- ▶ BC2. **Concevoir, développer et tester des solutions informatiques**
 - BC2.1. Modéliser un problème ou un besoin fonctionnel exprimé par un client et spécifier une solution informatique
 - BC2.2. Etudier, comparer et sélectionner les outils et méthodes nécessaires à la conception, au développement et au test d'une solution informatique
 - BC2.3. Concevoir et développer les applications informatiques : web, mobiles, logicielles
 - BC2.4. Documenter une étude et une solution informatique
 - BC2.5. Analyser et identifier les aspects réglementaires et techniques
- ▶ BC3. **Etudier et optimiser une solution informatique existante**
 - BC3.2. Assurer et optimiser les performances des systèmes d'information
 - BC3.3. Proposer, planifier et développer des évolutions
 - BC3.5. Fiabiliser et sécuriser
- ▶ BC4. **Déployer et superviser une solution informatique**
 - BC4.3. Déployer une solution informatique
 - BC4.5. Former l'utilisateur à l'usage de la solution informatique

A l'issue de cette UE, l'apprenant aura progressé sur les compétences suivantes du référentiel de la formation

- ▶ **Modéliser** les **besoins fonctionnels** en objet pour réaliser des applications **fiables**, **maintenables** et **réutilisables**.
- ▶ Utiliser des **paradigmes de la programmation objet** pour la **conception d'application**.
- ▶ **Comprendre** les **cycles de vie** du développement, du **test** et de la **maintenance** des applications.
- ▶ **Commenter** le **code** réalisé.
- ▶ **Rédiger** une **documentation** orientée utilisateur.

Syllabus

- ▶ Rappel notions de base du développement Objet,
 - ▶ Développement Objet avancé, héritage, généricité
 - ▶ Manipulation d'objets (listes, tables, etc.),
 - ▶ Programmation fonctionnelle, lambda expression,
 - ▶ Gestion de processus, communication réseau
 - ▶ Conception et programmation d'applications graphiques (JavaFX)
 - ▶ Optimisation de code java, recherche de données, Optimisation de code par utilisation de classes spécialisées
-
- ✓ **Activités pédagogiques** : Les apprenants participeront à des séances de travaux pratiques (TP) sous forme de projets tutorés, avec un focus particulier sur l'application concrète des concepts appris en cours.
 - ✓ **Environnements de développement intégrés (IDE)** : Eclipse, IntelliJ IDEA