

# A3P (AL)

**A**pprentissage **P**ar **P**roblème  
de la **P**rogrammation avec java puis C

IGI-3007

*Responsable : Denis BUREAU*

(+ Algèbre Linéaire / SFM-3007 )

# Précédente présentation

- Inversion TP -> TD -> cours
- Objectif 100% de validation
- Modalités d'évaluation
- Filières, cursus informatique E3ST
- Implication personnelle
- **Poser des questions**
- Trouver soi-même grâce à une « question »
- Ne pas attendre ni regarder la solution

# Les TP

- TP « tout court » (8 x 2h + 2 x 4hC) :  
découverte de nouvelles notions java par la  
pratique ; chercher à comprendre,  
pas à aller le plus vite possible  
(pas de rapport à rendre, **un par poste**)  
Il peut être demandé une lecture préalable.
- TP proj (6 x 2h) : point d'avancement du projet  
avec un enseignant technique,  
et aide technique pour avancer.

# Les TD

- TD « tout court » (5 x 2h + 2 x 1hC) :
  - écrire un programme différent du TP, mais utilisant les mêmes notions => compréhension
  - écrire du java sur papier => mémorisation, intégration, et **préparation aux contrôles**
- TD coach (1 x 1h) : **pause réflexive** au début de l'unité, sur votre apprentissage dans l'unité, votre future gestion du projet, les écueils à éviter, vos capacités à les surmonter, ...

# Les cours

- Cours « spéciaux » (2 x 1h) :  
introduction à la pédagogie et à l'unité
- Cours « normaux » (6 x 1h + 2 x 1hC) :  
cours courts !
  - restructuration de ce qui a été vu en TP/TD,  
voire en projet ==>  
Vous comprenez ce qui se dit ! Écouter 100% ?
  - basé sur un résumé de cours et des Q/R  
(**indispensable** à la véritable compréhension)

# Les séances Résa (ex-Pers)

- Résa « dédiés » (1h ou 2h ou 4h) : salles réservées pour accomplir une tâche, comme rendre une version du projet ou remplir un formulaire
- Résa « simples » (15 x 1 ou 2h) : salles réservées pour avancer le projet ou pour terminer un TP
- Tout à fait déraisonnable et non rentable de ne pas utiliser ces créneaux !

# Travail personnel

- Travail non planifié dans ADE :
- Terminer les TP
- Terminer les TD (et les passer sur machine)
- Avancer le projet
- Relire les « apports de connaissance »
- Formuler et **poser des questions**
- ==> environnement de développement sur son ordinateur personnel (ou rester tard à l'ESIEE)

# Le projet = un jeu d'aventure

- Le but n'est pas de réaliser un jeu vidéo !
- Le but est d'apprendre plus de java, et d'apprendre à utiliser une bonne conception objet, en réalisant un jeu d'aventure dont vous choisissiez le scénario (*impossible à apprendre autrement, nécessité de refactoring*).
- Un but secondaire est d'être fier d'avoir réalisé un jeu tout seul, si possible d'être sélectionné pour le Jour des Projets, et pourquoi pas, de gagner un prix ?



# En savoir plus sur le projet 1/2

- Jeu d'aventure, tour par tour
- Projet décrit dans un livre (**anglais/français**)
- Liste officielle des exercices (**oblig./optionnels**)
- **Créativité, contraintes, créativité**
- Versions successives :
  - programmation améliorée
  - fonctionnalités ajoutées
- **Explications** données dans le livre, et **exemples de programmation** donnés au fur et à mesure, mais **résolution de problèmes** !
- **Commencez à réfléchir** à votre histoire ...

# En savoir plus sur le projet 2/2

- Pris par la main au début  
(ne pas traîner)
- De plus en plus en autonomie  
(ne pas se décourager)
- Découverte de notions nouvelles
- Comprendre le pourquoi du re-factoring
- **Ne pas attendre le prochain TP**  
pour poser des questions sur un exercice  
(utiliser le forum !)
- **Attention au plagiat !**  
(pour le moindre morceau de code)

# Le langage Java (1/2)

- Plus facile à apprendre que le C.  
Erreurs moins graves pour un débutant.
- Après le C en E2, vous pourrez apprendre par vous-mêmes le C++ ou le C#.
- Famille de langages la plus utilisée au monde  
(date de 1995, 11 versions majeures)
- JRE = Java Runtime Environment (exécution)
- JDK = Java Development Kit (compilation)
- ==> installé avec BlueJ sur son ordi. Personnel  
(gratuit, open source)

# Le langage Java (2/2)

- On ne peut pas apprendre « tout » Java.
- Ni le langage, encore moins l'A.P.I.  
(l'essentiel, javadoc, « pied à l'étrier »)
- Besoin de tout : TP + TD + cours + projet
- Conception Orientée Objet (philosophie)  
petit à petit, surtout par le projet
- Parti pris, vocabulaire, exigences pédagogiques

# La Java Virtual Machine

- Processeur virtuel = programme interpréteur d'instructions « assembleur »
- Fourni sur toutes les plateformes (PC, smartphone, box, carte à puce, ...)
- Programme compilé tourne tel quel entre Mac / PC / Windows / Linux / ...
- Erreurs à la compilation ou à l'exécution
- Dire un maximum de choses au compilateur ! (ne pas économiser des caractères)

# Environnement de développement

- BlueJ, prononcez « bloudjè » (IDE)
- Installer sur son ordinateur personnel (gratuit, open source, comprend le JDK 11)
- Plus adapté qu'Eclipse ou NetBeans
- Peu de menus (tous les regarder ?)
- Schéma des classes / interaction
- Object bench / interaction / inspect
- Debugger
- Tests automatiques
- Code pad

# iCampus

- Tout est sur iCampus, semaine après semaine, séance après séance.
- Vous devez cliquer sur chaque séance, y compris Résa/Pers, pour lire les consignes.
- Toute unité est sur Blackboard (→ Moodle)
- Donc s'inscrire à l'unité sur Moodle
- (puis aussi à l'unité du projet Zuul)

# Les bons conseils

- Il n'est pas attendu de vous que vous sachiez résoudre tous les problèmes par vous-mêmes.
- Ne pas accepter de ne pas comprendre qqch (retrouvez cette révolte en vous !)
- **Prévenir** quand vous ne pouvez pas assister à une séance (en cas d'imprévu, s'excuser après)
- Et surtout : **rattraper au plus vite !** (1h encadrée ==> 2h en autonomie)



# L'approche pédagogique

- Elle est évidemment non négociable.
- Toute l'équipe pédagogique est persuadée qu'elle est plus efficace que la « classique ».
- Nous comprenons qu'elle puisse ne pas convenir à une minorité d'étudiants.
- Nous essaierons de les convaincre dans un premier temps, mais ensuite, nous les aiderons à compenser les inconvénients pour eux de la méthode.
- **Votre persévérance est indispensable !**

# Les contraintes absolues

1. Lire ses **mails TOUS LES JOURS ! [A3P:]**
2. **Toutes les séances sont utiles** : difficile de réussir en manquant un TP, TD, Cours, Résa.
3. Lire tous les mots de toutes les phrases des énoncés, des documents, **ET DES MAILS.**

Unité iCampus : IGI-3007/18

4. Bien gérer son temps :
  - travailler un peu (**quasiment**) **tous les jours**
  - ne pas rester bloqué plus d'une demi-heure sans poser de question (10mn en TP)

*(il est anormal d'y passer 20h/semaine !)*

# Bien commencer

1. L'unité démarre beaucoup plus intensément qu'il n'y paraît : beaucoup de notions nouvelles (même simples) et de **vocabulaire** (indispensable pour comprendre la suite) ; à apprendre/travailler **dès la 1<sup>ère</sup> semaine**.
2. *Chaque séance utilise ce qui a été vu précédemment. N'accumulez pas de retard !*
3. *Vous pouvez toujours essayer de rattraper un retard de cours peu avant un partiel, mais vous n'aurez pas acquis de savoir-faire en programmation ...*

# Votre réussite est entre vos mains

Votre « pouvoir personnel » est immense,  
2 attitudes possibles :

1. Utiliser toute son énergie et son temps à être capable de démontrer pourquoi on a échoué (garder les problèmes en réserve → + de repos)

2. Utiliser toute son énergie et son temps pour essayer de réussir (résoudre les problèmes dès qu'ils se présentent → + de travail)

3. Valider en ligne la charte de l'unité ==> **Résa0.1**

# Dès que quelque chose ne va pas :

## Dites-le moi au plus vite !

- Vous ne comprenez pas ce qui est demandé
- Vous ne comprenez pas comment faire
- Vous n'arrivez pas à résoudre un problème
- Un lien ne fonctionne pas
- Un document/renseignement manque
- Un intervenant pose problème (son nom !)
- ...

# What's next ?

- **Résa 0** (charte de l'unité et découverte des ressources)
- **TP 1** (découverte de BlueJ et des concepts classe, objet, attribut, méthode, ...)
- **Résa 1** (suite TP1 et lecture ressources)
  
- Même lien :  
**<https://perso.esiee.fr/~bureaud/E3ST/>**
- Suite de l'unité sur moodle
- Cliquer sur chaque séance !

Merci de votre attention

*Des questions ?*

# Rattraper un TP

- Sujet sur iCampus
- Le faire en autonomie.
- Utiliser le forum de ce TP  
(ne pas rester bloqué **longtemps** ==>  
un peu tous les jours plutôt que **bcp une fois**)
- Si possible, avant la séance suivante



# Rattraper un TD

- Sujet sur iCampus
- Le faire sur papier en autonomie.
- Le passer sur machine pour vérifier.
- Utiliser le forum de ce TD  
(ne pas rester bloqué **longtemps** ==>  
un peu tous les jours plutôt que **bcp une fois**)
- Si possible, avant la séance suivante

# Rattraper un cours

- Résumé et PDF sur iCampus
- Le lire en essayant de TOUT comprendre.
- Utiliser le forum de ce cours  
(ne pas rester bloqué **longtemps** ==>  
un peu tous les jours plutôt que **bcp une fois**)
- Si possible, avant la séance suivante

# Rattraper une séance Pers

- Réserver dans son emploi du temps une séance de 2h pour avancer le projet.
- Ne pas oublier de **lire les consignes** de la séance Pers correspondante (dont le remplissage du Compte-Rendu d'Avancement)

# Rattraper un TP proj

- Impossible.
- Évaluation d'avancement = 00/20.
- Toujours avoir la dernière version de son projet
- Réserver dans son emploi du temps une séance de 2h pour avancer le projet.
- Ne pas oublier de **lire les consignes** du TP+t correspondant (dont le remplissage du Compte-Rendu d'Avancement)