FICHE UE

IDENTIFICATION
Section : Bachelier en informatique de gestion
Intitulé de l'UE : Principes algorithmiques et programmation
Code de référence : 752105U32D3
Nombre de périodes : 120
Nombre de crédits ECTS : 8

DESCRIPTION
Prérequis ou documents de référence pour une préparation préalable au cours : Néant
Finalités particulières : *L'unité d'enseignement vise à permettre à l'étudiant* - de développer des comportements professionnels - développer des compétences collectives par le travail en équipe - prendre conscience des compétences à développer en ce domaine pour répondre d'une manière appropriée à l'évolution des techniques et des besoins de la clientèle en ce domaine - de mettre en oeuvre, d'une manière appropriée, des techniques, des méthodes spécifiques pour appréhender, globalement, la diversité méthodologique de la fonction de programmation dans le secteur des métiers de l'informatique et dans les besoins de la clientèle (entreprises publique et privée) - développer des compétences de base en utilisation d'un langage largement utilisé dans le monde des entreprises - mettre en oeuvre une démarche algorithmique cohérente.

FICHE UE

-

Contenu du cours :

Contents

1. Description du dossier pédagogique ..... **Erreur ! Signet non défini.**
   1.1. Finalité de la section ..... **Erreur ! Signet non défini.**
   1.2. Finalité de l'unité de formation algorithme et programmation ..... **Erreur ! Signet non défini.**
   1.3. Programme du cours ..... **Erreur ! Signet non défini.**
   1.4. Capacités terminales - Evaluation ..... **Erreur ! Signet non défini.**
2. Démarche pédagogique ..... **Erreur ! Signet non défini.**
   2.1. Ressources ..... **Erreur ! Signet non défini.**
3. Installation de Java et Eclipse ..... **Erreur ! Signet non défini.**
   3.1. Installation de Java. ..... **Erreur ! Signet non défini.**
   3.2. Installation d'Eclipse ..... **Erreur ! Signet non défini.**
4. Principes algorithmiques et programmation ..... **Erreur ! Signet non défini.**
   4.1. Première liste d'exercices liés au projet ..... **Erreur ! Signet non défini.**
   4.2. Création d'un projet et d'une classe Java ..... **Erreur ! Signet non défini.**
   4.3. Exécution d'un programme java. ..... **Erreur ! Signet non défini.**
      4.3.1. Byte Code – Machine Virtuelle (JVM) ..... **Erreur ! Signet non défini.**
   4.4. Premier programme "élaboré" ..... **Erreur ! Signet non défini.**
      4.4.1. Variables et affectations ..... **Erreur ! Signet non défini.**
      4.4.2. Séquencement des instructions - Debug ..... **Erreur ! Signet non défini.**
      4.4.3. Importance des types de donnée ..... **Erreur ! Signet non défini.**
   4.5. Le codage de l'information ..... **Erreur ! Signet non défini.**
      4.5.1. Les nombres binaires ..... **Erreur ! Signet non défini.**
      4.5.2. Synthèse sur les nombres et types entiers en java ..... **Erreur ! Signet non défini.**
      4.5.3. Représentation des nombres à virgule flottante (norme IEEE75) ..... **Erreur ! Signet non défini.**
      4.5.4. Les conversions numériques ..... **Erreur ! Signet non défini.**
      4.5.5. Les objets nombres en Java ..... **Erreur ! Signet non défini.**
      4.5.6. Le codage hexadécimal ..... **Erreur ! Signet non défini.**
      4.5.7. Le codage ASCII ..... **Erreur ! Signet non défini.**
      4.5.8. L'Unicode ..... **Erreur ! Signet non défini.**
      4.5.9. Les caractères ..... **Erreur ! Signet non défini.**

4.6. Algorithmes - Pseudo Code ..... **Erreur ! Signet non défini.**
4.6.1. Exercice ..... **Erreur ! Signet non défini.**
4.7. Programmation structurée – les fonctions et procédures ..... **Erreur ! Signet non défini.**
4.7.1. Procédures et fonctions ..... **Erreur ! Signet non défini.**
4.7.2. Pseudo code ..... **Erreur ! Signet non défini.**
4.7.3. Exercices ..... **Erreur ! Signet non défini.**
4.8. Saisies utilisateur ..... **Erreur ! Signet non défini.**
4.8.1. Exemple ..... **Erreur ! Signet non défini.**
4.8.2. Pseudocode ..... **Erreur ! Signet non défini.**
4.8.3. Exercices ..... **Erreur ! Signet non défini.**
4.9. Chaînes de caractères. ..... **Erreur ! Signet non défini.**
4.9.1. La classe String ..... **Erreur ! Signet non défini.**
4.9.2. Exercice ..... **Erreur ! Signet non défini.**
4.10. Structures alternatives ..... **Erreur ! Signet non défini.**
4.10.1. Structure alternative if ..... **Erreur ! Signet non défini.**
4.10.2. Structure alternative - Switch Case ..... **Erreur ! Signet non défini.**
4.10.3. L'opérateur ternaire ? : ..... **Erreur ! Signet non défini.**
4.11. Tableaux (Arrays) ..... **Erreur ! Signet non défini.**
4.11.1. Tableau unidimensionnel ..... **Erreur ! Signet non défini.**
4.11.2. Tableaux multidimensionnels ..... **Erreur ! Signet non défini.**
4.12. Les structures répétitives ..... **Erreur ! Signet non défini.**
4.12.1. La boucle REPETER (faire) … TANT QUE (jusqu'à ce que) .. **Erreur ! Signet non défini.**
4.12.2. La boucle TANT QUE ..... **Erreur ! Signet non défini.**
4.12.3. La boucle FOR ..... **Erreur ! Signet non défini.**
4.12.4. Les boucles imbriquées ..... **Erreur ! Signet non défini.**
4.13. Gestion des exceptions ..... **Erreur ! Signet non défini.**
4.13.1. Pseudo code ..... **Erreur ! Signet non défini.**
4.13.2. Exercices ..... **Erreur ! Signet non défini.**
4.14. Les Arrays List ..... **Erreur ! Signet non défini.**
4.14.1. Autres possibilités de parcours et modifications des ArrayList ..... **Erreur ! Signet non défini.**
4.14.2. Exercice ..... **Erreur ! Signet non défini.**
4.15. Création de fichier texte et écriture ..... **Erreur ! Signet non défini.**
4.15.1. Pseudo code ..... **Erreur ! Signet non défini.**
4.15.2. Exercice ..... **Erreur ! Signet non défini.**

4.16. Date et Heure ..... Erreur ! Signet non défini.
4.16.1. Pseudo Code ..... Erreur ! Signet non défini.
4.16.2. Exercice ..... Erreur ! Signet non défini.
4.17. Variables globales et variables locales ..... Erreur ! Signet non défini.
4.18. Passage de paramètres par adresse ou valeur ..... Erreur ! Signet non défini.
4.19. Exercice récapitulatif – création du ticket de caisse. ..... Erreur ! Signet non défini.

Bibliographie :

Syllabus de cours : algorithmes et programmation (V. Leblond)

Internet regorge d'informations et de code Java. Parmi ceux-ci citons :
- https://www.tutorialspoint.com/java/index.htm
- https://openclassrooms.com/courses/apprenez-a-programmer-en-java
- https://java.developpez.com/

bibliographie :
- Programmer en Java (Claude Delannoy) – Eyrolles
- Exercices en Java (Claude Delannoy) - Eyrolles

PERSONNEL(S) ENSEIGNANT(S)

LEBLOND Vincent

METHODOLOGIE

IRAM
Enseignement pour Adultes

FICHE UE

L'approche pédagogique de ce cours se veut résolument pratique. Les notions théoriques sont construites au fur et à mesure qu'elles sont nécessaires à la compréhension et à la résolution d'exercices de programmation. Ceux-ci sont eux-mêmes destinés à accomplir un projet de programmation plus vaste, présenté sous forme d'un cahier de charges issu d'une demande client.

Etant donné la forte hétérogénéité du public de promotion sociale, il est nécessaire à certains moments de pouvoir différencier les tâches. L'important est d'offrir à chacun la possibilité de progresser quelles que soient ses connaissances antérieures, son expérience éventuelle en algorithme, en programmation et en java.

Pour ce faire, le projet "brasserie" à réaliser, les exercices qui y sont associés sont disponibles dès le début du cours au §4.1. Des variantes du cahier des charges sont aussi proposées dans le projet. Ces variantes sont en quelque sorte des améliorations de la demande initiale. Il existe aussi une training list comportant une série d'exercices dont les difficultés sont progressives. Contrairement au projet, ces exercices n'ont aucun rapport, ni avec le projet, ni entre eux ; ils servent à décontextualiser les notions vues dans le projet brasserie et les appliquer dans d'autres domaines. Pour les plus aguerris, les passionnés, il est possible d'aller bien plus loin que ce que demande le dossier pédagogique pour ce cours.

Historiquement, le langage de programmation enseigné à l'IRAM est le Java. Ce langage courant issu du C\ C++ permet à l'utilisateur de se former aussi indirectement à ces langages. Java ou ses dérivés sont utilisés pour le développement d'applets pour le Web et pour des applications (non liées au web).

Il s'agit d'un langage orienté objet (POO) ce qui présente le bénéfice d'introduire le cours de POO qui suivra. Paradoxalement, l'inconvénient du Java est qu'il s'agit d'une POO et qu'il inclut donc directement des notions et des spécificités liées à la POO. Beaucoup de termes visibles à l'écran et fonctionnalités dans l'environnement de développement seront donc éludés pour l'instant en attendant d'approfondir la POO. Il faut garder à l'esprit que le plus important dans ce cours n'est pas de former des experts en Java mais de former les esprits à l'analyse et à l'algorithme. Java n'est alors qu'un champ d'application de ces algorithmes, un exemple d'environnement de programmation et de mise à l'épreuve de ceux-ci.

L'apprentissage de la programmation se fait de diverses manières mais principalement, elle se construit dans le temps à force de pratiquer. Il est donc impossible de réussir en étudiant « à fond » seulement trois jours avant l'examen. La lecture d'ouvrages et la consultation de codes sources réalisés par des programmeurs confirmés sont vivement recommandées.

SUPPORTS

FICHE UE

Slides présentés au cours distribués au format pdf.

Syllabus de cours : algorithmes et programmation (V. Leblond)

MODES D'ÉVALUATION ET ACQUIS D'APPRENTISSAGE

Examen écrit en fin d'unité qui comprend une partie d'algorithmique sur papier, une partie codage et tests 'unitaires et d'intégration) sur PC. Le but est de réaliser une application console à partir d'un cahier des charges donné.

ACQUIS D'APPRENTISSAGE

Pour atteindre le seuil de réussite, l'étudiant sera capable
en disposant d'un environnement matériel ou virtuel informatique et logiciels opérationnels et d'une documentation appropriée, face à un problème mettant en jeu des algorithmes de base, dans le respect du temps imparti,

- de mettre en oeuvre une représentation algorithmique du problème posé ;
- de développer au moins un programme en respectant les spécificités du langage choisi
- de mettre en oeuvre des procédures de test
- de justifier la démarche mise en oeuvre dans l'élaboration du (ou des) programme(s).

UTILISATION DE L'IA

L'utilisation de l'IA peut être envisagée dans l'optique d'illustrer des concepts dans d'autres contextes que ceux présentés au cours. A ce stade premier de l'apprentissage du codage, il est vivement **dé**conseillé de demander les solutions et de corriger les codes via l'IA. Elle doit être utilisée après coup, pour comparer la solution proposée par l'IA avec sa propre solution. Elle peut être utilisée par l'étudiant en dernier recours lors d'un blocage de longue durée lorsqu'il travaille en autonomie. Elle ne pourra en aucun cas être utilisée lors de l'épreuve finale. Pour

résumer, l'IA peut être un faux ami pour un programmeur débutant dans la mesure où elle est utilisée pour réfléchir à la place de l'apprenant.

Elle peut s'avérer très utile si elle est utilisée, en autonomie, en complément du cours et des exercices.