

Descriptif de module 63-31

Domaine : Economie & Services
Filière : Informatique de gestion

1. Intitulé de module *Programmation collaborative* 2023-2024

Code :
63-31

Type de formation :

Bachelor Master MAS DAS CAS Autres :

Niveau :

- Module de base
 Module d'approfondissement
 Module avancé
 Module spécialisé
 Autres :

Caractéristique :

Module dont l'échec peut entraîner l'exclusion définitive de la filière selon l'art.15, al.1 des directives cadres "statut des étudiants-e-s"

Type :

- Module principal
 Module lié au module principal
 Module facultatif ou complémentaire
 Autres :

Organisation temporelle :

- Module sur 1 semestre
 Module sur 2 semestres
 Semestre d'automne
 Semestre de printemps
 Autres :

2. Organisation

Crédits ECTS

5

Langue principale d'enseignement :

- Français Italien
 Allemand Anglais
 Autres :

3. Prérequis

- Avoir validé les modules 63-11 Fondement de la programmation & 63-21 Approfondissement de la programmation
 Avoir suivi le module 62-21 Gestion des données
 Pas de prérequis
 Autres :

4. Compétences visées / Objectifs généraux d'apprentissage

L'étudiante ou l'étudiant doit être capable, en fin de module, de justifier des compétences professionnelles suivantes :

- Maîtriser les algorithmes appliqués aux structures de données avancées
- Être capable de partager, de réutiliser et d'appliquer des concepts éprouvés.

5. Objectifs détaillés des enseignements

- Être capable de choisir et mettre en œuvre une structure de données adaptée à une situation
 - Vecteurs, listes, piles, files, tableaux associatifs
 - Arbres, graphes
 - Ensembles, énumérations
- Être capable de sélectionner l'implémentation d'une structure de données abstraite en fonction des situations
 - Structure linéaire ou arborescente ; structure statique ou dynamique ; structure ordonnée ou non
- Maîtriser les algorithmes appliqués aux structures de données avancées
 - Recherche dans une structure linéaire
 - Parcours de graphe
- Être capable d'appliquer des concepts éprouvés et des pratiques standards de développement logiciel
 - Classes abstraites et interfaces
 - Types génériques de données
 - Extension de fonctionnalité par héritage ou par encapsulation (*wrapper*)
 - Architecture applicative : packages, bibliothèques
 - Patrons de conception (*design patterns*) standards

6. Plan et chapitres des cours

- Structures de données et Algorithmes :
 - Types abstraits de données : concept, intérêt et exemples
 - Concept d'interface : séparation interface / implémentation
 - Extension de fonctionnalité : intérêt, méthodes usuelles, hiérarchie de classes
 - Structures de données avancées et interfaces standards :
 - Pile, File, Liste
 - Graphe, Arbre
 - Tableau associatif, table de hachage (résolution des collisions)
 - Ensembles, énumérations
 - Opérations standard, complexité et implémentation des structures de données avancées :
 - Ajout, recherche, suppression d'un élément
 - Exploration exhaustive des structures de données (parcours de graphe, ...)
 - Choix entre structures linéaires et structures arborescentes
 - Choix entre structures statiques et structures dynamiques
 - Choix entre structure ordonnée ou non
 - Récursivité
- Bonnes pratiques de développement logiciel et de programmation orientée objets :
 - Classes abstraites et interfaces
 - Types génériques de données
 - Extension de fonctionnalité par héritage ou par encapsulation (*wrapper*)
 - Architecture applicative : packages, bibliothèques
 - Patrons de conception (*design patterns*) standards :
 - Patrons de création : Singleton, Factory, Abstract Factory, Builder
 - Patrons structurels : Adapter
 - Patrons de comportement : Strategy, Observer (Listener, PropertyChangeListener).

7. Forme et méthodes pédagogiques

Le cours se donne sur quinze semaines à raison de quatre périodes d'enseignement par semaine. Chaque session de cours comportera une présentation de concept par l'enseignant, illustrée de quelques exemples commentés et suivie d'une mise en application lorsque celle-ci est pertinente.

Un assistant est à disposition pour répondre aux questions des étudiant-e-s, et apporte ses conseils et son soutien pour la réalisation des travaux pratiques obligatoires que les étudiant-e-s doivent réaliser. En dehors de ces séances, l'assistant est également à disposition des étudiant-e-s sur rendez-vous pour répondre à des questions personnelles concernant le cours et/ou les exercices pratiques proposés.

Étant fortement orienté sur l'acquisition de compétences pratiques, ce cours demande de la part des étudiants une forte implication quant à la réalisation des travaux pratiques proposés.

8. Modalités d'évaluation et de validation

Acquis : A-E
Remédiation : Fx
Répétition : F

L'évaluation du module se fera en principe de la manière suivante :

Contrôle continu : 50% 2 contrôles continus pratiques individuels de 120 minutes Chaque travail pratique rendu à temps pourra octroyer 0,1 point de bonus sur le contrôle continu suivant (avec un maximum de 0,5 par CC) en fonction de la solution proposée et de la qualité du code fourni.	Examen : 50% 1 Examen pratique individuel de 180 minutes La réalisation personnelle, attestée par l'assistant, de 75% des travaux pratiques proposés est exigée pour avoir le droit de se présenter à l'examen.
---	--