

Master Traitement du signal et des images

Algorithmique et programmation

Responsable	Descriptions	Informations
Jean marc BOI jean-marc.boi@univ-amu.fr	Code : STSAU16 Nature : Unité d'enseignement Domaines : Sciences et Technologies	Composante : Faculté des Sciences

Langue(s) d'enseignement

Français

Contenu

La première partie de cet enseignement débutera donc par quelques révisions, puis introduira les notions de type abstrait, de définition fonctionnelle d'un type, dans un premier temps de manière indépendante d'un quelconque langage de programmation, puis l'implantation des opérations définies pour ce type en utilisant le Langage C. Seront abordées suivant ce modèle différentes façons de définir des collections d'objets très usitées en informatique (pile, file, arbres, ...). Associés à ces types de données, différents algorithmes de consultation et mise à jour seront décrits.

La deuxième partie de cette UE est conçue pour doter les étudiants d'une compréhension approfondie de la Programmation Orientée Objet (POO) en Python. Cette compétence est essentielle pour l'utilisation efficace des bibliothèques de Deep Learning et de Vision par ordinateur, qui seront abordées dans les cours de la deuxième année de ce Master.

Les étudiants commenceront par explorer les principes fondamentaux de la POO, en les contrastant avec le paradigme de programmation procédurale, afin de mieux comprendre les intérêts de cette approche.

Le cours se poursuivra avec des modules détaillés sur la définition des classes, la création d'objets, ainsi que l'importance des variables d'instance et des méthodes, qui sont cruciales pour encapsuler et structurer les données et les fonctions de manière cohérente. Les étudiants apprendront également à implémenter des constructeurs pour initialiser les objets et à utiliser l'encapsulation pour protéger les données à travers des modificateurs d'accès.

Le concept d'héritage, y compris l'héritage multiple, sera exploré pour montrer comment les classes peuvent étendre ou modifier le comportement d'autres classes. En outre, le cours couvrira la surcharge de méthode et le polymorphisme, permettant ainsi aux étudiants de redéfinir des méthodes et d'utiliser des interfaces unifiées pour manipuler divers objets.

Compétences à acquérir

Première partie:

Reprendre et approfondir des notions précédemment abordées, dont les phases d'élaboration d'un Programme, le cas spécifique du Langage C.

- Formuler la définition de type de données abstrait, Introduction de la description fonctionnelle, de la formulation des propriétés associées
- Définir le type de données abstrait Pile, Description fonctionnelle, Propriétés, et réaliser son implantation
- Définir le type de données abstrait File, Description fonctionnelle, Propriétés, et réaliser son implantation
- Connaître les structures de données arborescentes (Arbre n-aires, Arbre binaires)
- Connaître le type de données Arbre binaire, Description fonctionnelle, Propriétés, réaliser son implantation ; Cas particulier des opérations d'ajout en feuille et d'ajout à la racine
- Définir les Mesures sur les Arbres binaires (Taille, Hauteur, Nombre de feuilles, ...) et leurs programmations, connaître les Arbres binaires particuliers (parfaits, complets)
- Maîtriser les différents parcours d'un Arbre binaire
- Connaître les méthodes performantes de tri de tableau de complexité $O(n \log n)$

- Maîtriser le type de données Arbre binaire de recherche (Arbre binaire ordonné), Description fonctionnelle, Propriétés, et implantation
- Comprendre les coûts des opérations de recherche et de mise à jour dans un Arbre binaire de recherche, et définir les opérations de rééquilibrage d'un arbre de recherche.

Deuxième partie:

- Comprendre les principes de base de la programmation orientée objet (POO) : être capable de différencier la POO du paradigme de programmation procédurale et comprendre les intérêts de l'utilisation de la POO.
- Définition de classes et création d'objets : savoir comment créer des classes en Python et utiliser ces classes pour instancier des objets.
- Maîtrise de l'encapsulation : savoir utiliser les modificateurs d'accès (public, privé et protégé) pour contrôler l'accès aux données et protéger l'intégrité de l'objet.
- Employer l'héritage et l'héritage multiple : comprendre comment une classe peut hériter des propriétés d'une ou plusieurs autres classes, et comment gérer les complexités qui en résultent.
- Maîtriser la surcharge de méthode : savoir comment redéfinir ou ajuster le comportement des méthodes héritées dans les classes dérivées.
- Exploiter le polymorphisme : utiliser le polymorphisme pour concevoir des interfaces unifiées qui permettent l'interaction avec des objets de différentes classes de manière homogène.

Modalités d'organisation

- Première partie : "**Programmation procédurale, types abstraits**"
 - Intervenant : Jean-Marc BOI
 - Horaire : 12 h de Cours 12 h de TD 8 h de TP
 - Organisation souhaitée de l'emploi du temps :
 - 6 journées organisées à un rythme hebdomadaire comme décrites ci-après
 - 4 premières journées :
 - Matin, de 9h à 12h
 - 3 h de Cours
 - Après-midi, de 13h à 16h
 - 1 h de TD en salle informatique
 - 2 h de TP en salle informatique à la suite du TD
 - 2 dernières journées :
 - Matin, de 9h à 13h
 - 2 h de TD en salle banalisée
 - 2 h de TD en salle informatique
- Deuxième partie : "**Fondements de la Programmation Orientée Objet**"
 - Intervenant : Motassem NAWAF
 - Horaire : 8 h de Cours 8 h de TD 2 h de TP
 - Organisation souhaitée de l'emploi du temps :
 - 3 journées organisées à un rythme hebdomadaire comme décrites ci-après
 - 2 premières journées :
 - Matin, de 9h à 12h
 - 3h de Cours

- Après-midi, de 13h à 16h
 - 3 h de TD en salle informatique
- dernière journée :
 - Matin, de 9h à 12h
 - 2h de Cours
 - 1 h de TD en salle banalisée à la suite du Cours
 - Après-midi, de 13h à 16h
 - 1h de TD en salle informatique
 - 2 h de TP en salle informatique à la suite du TD

Pré-requis obligatoires

L' UE pSTSAU16J "Algorithmique et Programmation " se déroule au semestre 1 du M1 TSI.

Bien que l'on puisse considérer cette UE comme un prolongement des UEs pSPI5U70J "Algorithmique et Programmation 1" et pSPI6U85J "Algorithmique et Programmation 2" de L3SITeR, il convient de signaler qu'elle s'adressera aussi à des étudiants admis en M1 TSI sans avoir suivi le L3SITeR, mais, cependant, pour lesquels certains prérequis en Algorithmique et Programmation en Langage C seront attendus.

Ceci étant, et dans un premier temps, il conviendra donc de reprendre rapidement des notions abordées dans les UEs pSPI5U70J et pSPI6U85J.

VOLUME HORAIRE

- Volume total: 50 heures
- Cours magistraux: 20 heures
- Travaux dirigés: 20 heures
- Travaux pratiques: 10 heures

Codes Apogée

- STSAU16J [ELP]

M3C

Aucune donnée M3C trouvée

Pour plus d'informations

[Aller sur le site de l'offre de formation...](#)



Dernière modification le 04/09/2024