

Programmation Orientée Objet - BUT 2^e année

Séance 1 - présentation du cours et rappels en Python

Lentschat Martin - LIG - SIGMA GETALP

6 Septembre 2023

Objectifs

- Introduction aux paradigmes de la programmation orientée objet
- La conception orientée objet
- La notion d'héritage
- Algorithmes basés sur les objets

Détails importants

- Enseignant : martin.lentschat@univ-grenoble-alpes.fr
- Evaluation :
 - QCM - examen final (40%)
 - TP individuel noté (30%)
 - Projet de groupe (30%)
- présence obligatoire

Planning

- 3 séances en salle B106 → Projet
- 5 séances en salle B107 → Rendu Projet
- 2 séances en salle B108 → TP noté

QCM après la Toussaint

Infos utiles

- Les slides seront déposées sur Chamilo après chaque cours
- attention au plagiat dans les rendus !
- pensez à faire les exercices et/ou à consulter des ressources externes

Un langage :

- Débutés en 1989, version 3.11.5 actuellement
- Incompatibilité entre python 2 et 3
- 2^e langage de programmation le plus utilisé au monde

Un langage :

- interprété
- impératif, fonctionnel et orienté objet
- versatilité, nombreuses bibliothèques, plus proche du langage naturel que d'autres
- lent pour certaines tâches, gourmand en mémoire, peu utilisé sur mobile

Faites attention à l'indentation et à la priorité des opérateurs !

Exercice I - Chaînes de caractères, Affichage, Types

- Écrire un programme dialoguant avec un utilisateur. Celui-ci lui demande successivement son **nom**, **prénom**, **date de naissance** (*ddmmyyyy*), **adresse mail**
- Ces informations sont ensuite affichées dans une phrase

Faites attention à l'indentation et à la priorité des opérateurs !

Exercice I - Chaînes de caractères, Affichage, Types

- Écrire un programme dialoguant avec un utilisateur. Celui-ci lui demande successivement son **nom**, **prénom**, **date de naissance** (*dd-mm-yyyy*), **adresse mail**
- Ces informations sont ensuite affichées dans une phrase

Exercice I bis - Conditions, Boucles

- Pour chaque information, mettez en place une vérification sur la donnée entrée et l'opportunité à l'utilisateur de les corriger
- Le programme s'arrête si un nombre maximum d'erreurs autorisées est atteint

Exercice II - Nombres, Listes, Itération, Imports, Fonctions

- Écrire un programme générant une liste de **dates** de naissances (*dd-mm-yyyy*)
`from random import randint`
- Créer des fonctions pour :
 - ajouter une nouvelle date à la liste
 - retourner une liste des dates converties en âge
 - retourner l'âge minimal, maximal, médian et leur somme
 - retirer de la liste toute date antérieure à l'âge médian
 - trier les dates en ordre croissant selon l'âge

Exercice III - Dictionnaires Les résultats aux examens d'étudiants sont stockés dans des dictionnaires. Ceux-ci incluent les différentes matières ainsi que leurs coefficients :

```
{'bob': {'notes': {'math': 16, 'histoire': 14, ...},  
        'coeffs': {'math': 5.0, 'histoire': 3.0, ...}  
... }
```

Écrire des fonctions permettant :

- de calculer la moyenne d'un étudiant
- d'afficher les moyennes de tout les étudiants
- d'afficher les résultats du meilleur étudiant
- de trouver la matière la plus difficile
- de détecter les étudiants n'ayant pas passé tout les examens
- de retourner un dictionnaire de la forme :

```
{'math': {'coeff': 5.0, 'notes': { 'bob': 16, ... } }  
... }
```