

Algorithmique / Python Généalogie

Manuel Munier

Version 10 mars 2022 (08:38)

Expression des besoins

On désire gérer par des moyens informatiques (par exemple une programme écrit en Python. . .) les liens de parenté entre diverses personnes. Soyez rassurés, ici point de structures de données "complexes" telles que des arbres, des graphes, ou autres joyeusetés. Nous nous contenterons pour ce projet d'un premier tableau pour stocker les informations des différentes personnes qui seront donc numérotées de 0 à N selon leur indice dans ce tableau. Un second tableau servira à "relier" un enfant avec un parent via leurs numéros. Chaque ligne de ce tableau représentera un lien de parenté : la personne n°1⁵ (numéro de l'enfant dans la 1^{ère} colonne) est un fils ou une fille de la personne n°2 (numéro du parent dans la 2^{nde} colonne).

L'objectif de ce second TP sous forme de mini projet informatique est d'asseoir les compétences de base en algorithmique acquises au cours du module R107 "fondamentaux de la programmation". Vous travaillerez individuellement afin que chacun d'entre vous puisse prendre conscience de ses (éventuelles) lacunes, rattraper son (éventuel) retard et progresser en algorithmique sans devoir attendre son binôme/groupe/promo. Nous vous conseillons donc vivement de prendre ce projet au sérieux car ces notions d'algorithmique vous seront essentielles pour la suite de votre cursus (et pas qu'en BUT!). D'après le PPN du BUT R&T les notions "de base" en algorithmique susnommées sont :

- variables, structures de données (tableaux, tuples, listes)
- structures de contrôle (tests, boucles, structures imbriquées, . . .)
- sous-programmes (fonctions, avec passage de paramètres et retour de résultat, visibilité des variables)
- concevoir un algo à partir d'un énoncé, le traduire en Python
- ~~compiler~~, corriger et tester un programme

Consignes de programmation

Voici quelques consignes (de bon sens) pour prendre de bonnes habitudes en programmation :

- Pensez à commenter votre code.
- Au début de chaque fonction indiquez en commentaire le rôle de chaque variable passée en paramètre et le type du résultat qui sera retourné.
- Une fonction ne doit pas effectuer d'entrées/sorties, c'est-à-dire ne doit pas demander à l'utilisateur de saisir des valeurs au clavier ou lui afficher des informations dans le terminal. Sauf, bien évidemment, les fonctions dont le rôle est explicitement de saisir une valeur ou d'afficher une information. . .
- Prenez l'habitude que vos fonctions s'exécutent jusqu'à la fin de leur code dont la dernière instruction est le **return**. Évitez à tout prix de faire un **return** dans une boucle ou dans un test ; ça fonctionne, certes, mais c'est source de gros problèmes lors du debugage. . .

Travail à réaliser

D'un point de vue technique, ce projet sera constitué d'un seul programme (composé bien évidemment des diverses fonctions) exécuté en ligne de commande dans un terminal texte. Dans un premier temps, il n'y aura aucune interaction avec l'utilisateur : toutes les "actions" seront directement hard-codées dans le programme principal, ce qui d'ailleurs vous facilitera la phase de test. Dans un second temps uniquement, les interactions avec l'utilisateur se feront au travers d'un menu textuel (nous vous guiderons pour cela...) où le choix d'une fonctionnalité par l'utilisateur déclenchera l'exécution d'une fonction dédiée.

D'un point de vue conceptuel, ce projet repose sur l'utilisation de 2 tableaux. Le premier, celui qui stocke les personnes, devra à minima contenir les informations suivantes sur chaque personne : nom, prénom, sexe, date de naissance. Le second ne contiendra que des couples d'entiers (les indices des 2 personnes dans le 1er tableau ; une sorte de "pointeur").

Pour simplifier le travail, nous ferons dans un premier temps les suppositions suivantes :

- on suppose que les numéros utilisés pour les liens de parenté (dans le 2nd tableau) référencent effectivement des personnes existantes (dans le 1er tableau) → notion d'intégrité référentielle
- pas de vérification qu'un enfant a plus de 2 parents, 2 pères, 2 mères,...
- pas de comparaison des dates de naissance entre parents et enfants
- pas de détection de cycle dans le graphe des liens de parenté (quoique mère Nature pourrait le permettre...)

Voici dans les grandes lignes votre planning de travail. Bien évidemment, à chaque étape il vous faudra écrire un petit programme de test afin de valider les fonctions que vous aurez écrites.

1. Implémenter la fonctionnalité permettant d'ajouter une nouvelle personne. Cette fonction prend en paramètres les différentes informations, construit la nouvelle ligne et l'ajoute au tableau.
2. Implémenter la fonctionnalité permettant d'afficher, dans l'ordre d'enregistrement, toutes les personnes d'un tableau passé en paramètre. Pour simplifier, l'affichage se fera de manière textuelle dans la console.
3. Implémenter la fonctionnalité permettant de sélectionner le numéro d'une personne.
4. Implémenter la fonctionnalité permettant, à partir de 2 numéros de personnes, de créer un lien de parenté dans le 2nd tableau.
5. Implémenter la fonctionnalité permettant de connaître, à partir d'un numéro de personne fourni en paramètre, la liste de tous ses ascendants connus (c'est-à-dire sur la base des liens de parenté enregistrés). Le résultat sera retourné sous la forme d'un tableau d'entiers (les numéros des personnes identifiées).
6. Idem mais pour les descendants d'une personne cette fois.
7. Idem pour les frères et sœurs d'une personne donnée.
8. Implémenter la fonctionnalité qui, à partir d'un tableau de numéros de personnes, affiche les informations de ces personnes dans l'ordre alphabétique (nom puis prénom).
9. Idem de la personne la plus âgée à la plus jeune.
10. Écrire le squelette de votre menu principal : affichage des actions disponibles, saisie du choix de l'utilisateur en vérifiant la validité de ce choix, appel à la fonction adéquate, puis retour au menu. La fonction adéquate dont nous parlons ici devra faire 3 choses :
 - a) lecture des paramètres nécessaires
 - b) appel de la fonction de traitement (une des fonctions programmées dans les points précédents)
 - c) affichage du résultat

Document rédigé en \LaTeX sous Linux.