

MP1

facebook

Introduction

On désire modéliser une version très simplifiée du réseau Facebook. Dans ce projet, nous allons simuler le réseau Facebook par un tableau de taille fixe. Pour cela nous allons définir 2 classes qui seront la classe Personne et la classe Facebook.

Étant une version simplifiée, le programme doit être capable d'effectuer assez peu de fonctionnalités comparé au véritable Facebook, qui sont donc globalement :

- La création d'un compte
- L'effacement d'un compte
- Ajouter un ami
- Supprimer un ami
- Vérifier une amitié entre deux personnes
- Vérifier si deux personnes ont un ami en commun
- Pouvoir lister des amis en communs
- Pouvoir dire qui a le plus d'amis sur le réseau
- Lister les personnes présentes sur le réseau ainsi que toutes les relations entre elles

Durant ce mini-projet nous avons tenté de répondre au mieux aux fonctionnalités attendues.

Déroulement du développement

Comme conseillé par le sujet, nous avons tout d'abord commencé à écrire la définition de la classe Personne ainsi que ses méthodes, en testant celles qui étaient indépendantes par rapport à la classe Facebook (méthodes operator, saisir, afficher...).

Suite à cela nous avons défini la classe Facebook et ses méthodes que nous avons pu tester au fur et à mesure afin de s'en assurer le bon fonctionnement.

Pour la première partie nous avons implémenté des sous-programmes supplémentaires permettant de :

- Afficher un menu agréable et clair permettant d'utiliser toutes les fonctionnalités du programme, renvoyant un entier correspondant à la saisie de l'utilisateur quant à son choix

int menu()

- Rechercher de manière séquentielle une valeur dans un tableau (non trié) à une dimension comportant un certain nombre de valeurs, la fonction renvoie un booléen quant à sa présence, ainsi que l'indice où se trouve la valeur recherchée si jamais elle s'y trouve

bool rechercheTab1D (int x, int tab[MAXP], int nbVal, int& indice)

- Vérifier qu'une personne est bien existante dans un réseau Facebook donné en renvoyant un booléen

bool Personne::existe(Facebook& f)

- Trouver et renvoyer l'ID d'une personne dans Facebook grâce à son nom et son prénom

int trouverID(Personne& p);

- Rechercher de manière séquentielle une personne dans Facebook (le tableau n'étant pas trié dans la Partie I) qui renvoie un booléen et l'indice

bool Facebook::rechercheBasiqueTabFacebook(Personne& p, int& indice)

- Initialiser le tableau de relation à « faux » pour chaque case

void initialiseTabRelation();

- Afficher les tableaux de personne/id/relation pour faciliter et uniquement pour les tests

void afficheFbId()

void afficheAmities()

Pour la deuxième partie, lorsqu'il est demandé de trier le tableau `Personne` par ordre alphabétique nous avons rajouté trois autres méthodes permettant de :

- Vérifier qu'une ID est bien libre (lors de la création de compte), renvoie un booléen

`bool Facebook::idLibre(int id)`

- Rechercher de manière dichotomique une personne dans Facebook, renvoie un booléen et l'indice où se trouve la personne
`bool Facebook::rechercheDichotomiqueTabFacebook(Personne& p, int& indice)`
- Trouver où placer une personne lors de son inscription, de la même manière que l'on fait une recherche dichotomique renvoie un entier représentant l'indice où placer la personne dans le tableau

`int trouverPlaceOrdreAlphabetique(Personne& p);`

Le seul « problème » connu, est le fait que la méthode `lePlusdAmi` n'est pas capable de dire s'il existe un ex æquo et renvoie par conséquent la première personne trouvée avec le nombre maximum d'amis.

Conclusion

Les objectifs établis pour ce mini-projet ont été remplis, le programme est capable d'effectuer les fonctionnalités demandées.

De notre côté, ce mini-projet nous a permis de travailler sur un programme fonctionnant majoritairement grâce à ses classes et ses méthodes, ce qui est à la fois un bon entraînement et une manière de programmer intéressante ainsi qu'une sensation d'avoir un programme plus « propre », plus « lisible ».

Jeux d'essais

Certaines méthodes ne seront pas présentées dans les jeux d'essais, soit car elles sont totalement dépendantes de méthodes qui seront elles testées soit car il est trop compliqué d'en faire le test, ces méthodes suivantes ne seront donc pas dans le jeu d'essai :

- affiche() (de la classe Personne ainsi que celle de Facebook)
- saisir()
- operator >= (dépendante de operator== et operator>)
- operator < (dépendante de operator>=)
- operator <= (dépendante de operator== et operator<)
- existe (uniquement dépendante de rechercheDichotomiqueTabFacebook)
- quitter (uniquement dépendante de existe)
- inviterUnAmi (uniquement dépendante de existe)
- supprimerUnAmi (uniquement dépendante de existe)
- initialiseTabRelation
- afficheFbId
- afficheAmities
- supprimeRelationAmitie (similaire à ajoutRelationAmitie)

Classe Personne

Numéro Test	Méthode Testée	Valeurs des paramètres données	Justification du test	Valeur attendue pour indice
1	getNom	Personne ciblée : Jean Dupont	Fonctionnement normal	Dupont
2	getPrenom	Personne ciblée : Jean Dupont	Fonctionnement normal	Jean
3	operator==	Jean Dupont et Jean Dupont	Cas « true » noms et pré-noms similaires	True

4	operator==	Jean Dupont et John Smith	Cas « false » noms et pré- noms différents	False
6	operator==	Jean Dupont et Georges Dupont	Cas « false » prénom diffé- rents	False
7	operator>	Georges Bourbon et Jean Astier	Cas « true » Bourbon>Astier B>A	True
8	operator>	Jean Astier et Georges Bourbon	Cas « false » Astier<Bourbon A<B	False
9	operator>	Jean Astier et Jean Astier	Cas « false » noms et pré- noms similaires	False
10	operator>	Jean Astier et Alexandre As- tier	Cas « true » noms similaires mais Jean>Alexandre J>A	True
11	operator>	Alexandre Astier et Jean Astier	« Cas false » noms similaires et Alexandre<Jean A<J	False

12	inscription	MAXP=21 nbPers=10 Jean Dupont Existe ren- voie faux (la personne n'existe pas dans face- book)	Cas normal où le nombre de per- sonnes est infé- rieur au nombre maximum pos- sible et où per- sonne ayant ce nom n'est inscrit dans Facebook	True
13	inscription	MAXP=21 nbPers=20 Jean Dupont Existe ren- voie faux	Cas où le nombre maxi- mum de per- sonnes a été at- teint	False
14	inscription	MAXP=21 nbPers=10 Jean Dupont Existe ren- voie vrai	Cas où la per- sonne existe dé- jà dans Face- book	False
15	inscription	MAXP=21 nbPers=20 Jean Dupont Existe ren- voie vrai	Les deux cas précédents en même temps	False

Classe Facebook

16	creerCompte	Jean Dupont inscription renvoie vrai idLibre ren- voie vrai	Cas normal	Compte créé
17	creerCompte	Jean Dupont inscription renvoie vrai idLibre ren- voie faux	Si l'ID prédéfini pour la création du compte n'est pas libre	L'ID s'incrémente jusqu'à ce que id- Libre renvoie vrai , ensuite le compte est créé
18	creerCompte	JeanDupont inscription renvoie faux	Cas où la per- sonne existe dé- jà	La procédure lance une excep- tion de type en- tier
19	fermerCompte	Jean Dupont quitter ren- voie vrai	Cas normal	Le compte est ef- facé
20	fermerCompte	Jean Dupont quitter ren- voie faux	Cas où Jean Du- pont n'existe pas dans Face- book	La procédure lance une excep- tion de type en- tier
21	ajoutRelationA- mitie	Jean Dupont et Alex Lavoie inviterUnAmi renvoie vrai amis renvoie faux	Cas normal	Les deux per- sonnes sont dé- sormais indiquées amies, la fonction renvoie vrai

22	ajoutRelationAmitie	Jean Dupont et Alex Lavoie inviterUnAmi renvoie faux	Cas où au moins l'une des deux personnes n'existe pas dans Facebook	La fonction renvoie faux et un message d'erreur
23	ajoutRelationAmitie	Jean Dupont et Jean Dupont inviterUnAmi renvoie vrai $p=p2$	Cas où l'on essaie de mettre une personne ami avec elle-même	La fonction renvoie faux et un message d'erreur
24	ajoutRelationAmitie	Jean Dupont et Alex Lavoie inviterUnAmi renvoie vrai amis renvoie vrai	Cas où les deux personnes sont déjà amies	La fonction renvoie faux et un message d'erreur
25	amis	Jean Dupont et Alex Lavoie existe renvoie à chaque fois vrai	Cas normal	S'ils sont amis, la fonction renvoie vrai, sinon faux
26	amis	Jean Dupont et Alex Lavoie existe renvoie au moins une fois faux	Cas où au moins l'une des deux personnes n'existe pas dans Facebook	La fonction affiche un message d'erreur et lance une exception

« listerAmisCommuns » reprend les mêmes tests que « amis »

27	listAmisCommuns	Jean Dupont et Alex Lavoie nbAmis>0	Cas où les deux personnes ont au moins un ami en commun	La procédure affiche la liste des amis en commun
28	listAmisCommuns	Jean Dupont et Alex Lavoie nbAmis=0	Cas où les deux personnes n'ont aucun ami en commun	La procédure indique qu'il n'y a pas d'ami en commun

« amisIndirects » reprend les mêmes tests que « amis »

29	amisIndirects	Jean Dupont et Alex Lavoie	Cas où les deux personnes ont au moins un ami en commun	La fonction renvoie vrai
30	amisIndirects	Jean Dupont et Alex Lavoie	Cas où les deux personnes n'ont aucun ami en commun	La fonction renvoie faux

Avant d'utiliser « lePlusdAmi » on vérifie toujours que nbPers>0

31	lePlusdAmi		Cas normal	La fonction renvoie la personne avec le plus d'amis
32	lePlusdAmi		Cas où il n'y aucune amitié sur le réseau	La fonction affiche un message et lance une exception

On envoie toujours un nombre compris entre 0 et MAXP-1 en paramètre lorsqu'on fait appel à idLibre

33	idLibre	nbPers=0	Cas où le réseau est vide	La fonction renvoie vrai sans effectuer une recherche dans le tableau d'ID
34	idLibre	nbPers>0	Cas normal	La fonction fait appel à rechercheTab1D et renvoie vrai ou faux selon le retour de cette dernière

On vérifie toujours qu'une personne existe avant d'appeler trouverID

35	trouverID	Jean Dupont	Cas normal	La fonction renvoie un entier correspondant à l'ID
36	trouverPlaceOrdreAlphabetique	Jean Dupont	Dans le cas où Dupont est la première personne par ordre alphabétique par rapport au reste des personnes du tableau	La fonction renvoie un entier qui correspond à l'indice où doit être placé la personne. Ici 0.

37	trouverPlaceOrdreAlphabetique	Jean Dupont	Dans le cas où Dupont est la dernière personne par ordre alphabétique par rapport au reste des personnes du tableau	La fonction renvoie un entier qui correspond à l'indice où doit être placé la personne. Ici l'entier égal à nbPers.
38	trouverPlaceOrdreAlphabetique	Jean Dupont	Dans le cas où Dupont est à placer dans le « tas »	La fonction renvoie un entier qui correspond à l'indice où doit être placé la personne. Ici dépendant du contenu du tableau.
39	rechercheDichotomiqueTabFacebook	Jean Dupont	Dans le cas où la personne indiquée est trouvée dans le tableau	La fonction retourne vrai ainsi que la valeur de l'indice en paramètre
40	rechercheDichotomiqueTabFacebook	Jean Dupont	Dans le cas où la personne indiquée n'est pas trouvée dans le tableau	La fonction retourne faux

Sous-programmes supplémentaires

41	menu	Valeur saisie comprise entre 1 et 10	Cas normal	La fonction retourne la valeur saisie
42	menu	Valeur saisie non comprise entre 1 et 10 ou Un caractère est entré à la place d'un entier	Cas d'une mauvaise saisie de l'utilisateur	La fonction redemande une saisie tant que ce n'est pas un entier compris entre 1 et 10
43	rechercheTab1D	Valeur passée en paramètre 1	Cas où la valeur est trouvée	La fonction renvoie vrai, plus l'indice où se trouve la valeur 1
44	rechercheTab1D	Valeur passée en paramètre 1	Cas où la valeur n'est pas trouvée	La fonction renvoie faux