

TP6 – Types de Données Abstraits

- Ouvrez le dossier *Partage* puis *profs\TP info S2\TP* ; sélectionnez le dossier *TP6* ; copiez ce dossier dans votre répertoire *Mes documents* ; ouvrez le fichier *TP6-source.cpp* avec Dev C++.

I - Quelques conseils

- Ce TP se présente sous la forme d'un « programme à trous ». Notez bien que toutes les fonctions nécessaires à l'exécution du programme ont été déclarées. Votre tâche consiste seulement à compléter le corps de certaines d'entre elles que nous avons malencontreusement effacé.
- Le fichier source dont vous disposez se compose de plus de 1500 lignes de code. De nombreuses lignes correspondent à l'**affichage d'illustrations ASCII** dans la console. Ne vous découragez donc pas en voyant sa taille, ce programme n'est pas si compliqué que ça.
- **Pour en faciliter la lecture**, il vous est conseillé de modifier le paramètre relatif à la couleur d'affichage des commentaires. Pour ce faire, passez par le menu *Outils > Options de l'Editeur* ; dans l'onglet *Syntaxe*, repérez la liste déroulante *Type* et sélectionnez *Comment*. Vous pouvez alors paramétrer la couleur de *Avant-Plan* (le texte : sélectionnez un bleu foncé) et celle de *Arrière-Plan* (le fond : gardez du blanc) puis *Ok* pour valider. Ne modifiez rien d'autre dans ce menu, **SVP**.
- **Pour garder une vision globale et fonctionnelle** du fichier proposé, il vous est conseillé de vous appuyer sur l'explorateur de classes de Dev C++. Pour l'activer, passez par le menu *Outils > Options de l'Editeur* ; dans l'onglet *Explorateur de Classes*, cochez *Activer l'explorateur de classes*. Vous pouvez ainsi visualiser à tout moment les structures, fonctions et variables globales qui ont été déclarées par nos soins.
- Il est clair qu'un travail de **compréhension global d'environ 30 minutes** est indispensable pour appréhender sereinement la résolution du problème proposé. Il est fortement déconseillé de passer outre cette phase essentielle. **Aidez-vous des nombreux commentaires, de l'explorateur de classes et surtout des noms explicites choisis pour les différentes variables et fonctions déjà définies.**

II - Notions à acquérir au cours du TP

- Concepts de Type de Données Abstrait (*Pile* et *File*) ;
- Incarnation ou implémentation de ces TDA ;
- Manipulation et utilisation de ces TDA ;
- Compréhension d'un projet existant.

III - EXERCICE 1 : Compréhension du programme (30 min)

III.1 - Les fonctionnalités du programme

Ce programme constitue une ébauche de la nouvelle version de Facebook. Différents modules sont proposés dans le but d'effectuer la gestion électronique du réseau social de l'utilisateur. Voici une description rapide de ces modules.

Le premier service concerne la gestion du profil de l'utilisateur. Celui-ci peut créer son profil, l'afficher, le mettre à jour, modifier son avatar ou encore effacer son profil.

Le deuxième service concerne la gestion de son réseau social. L'utilisateur peut afficher la liste de ses amis (qui est stockée dans une *File*), modifier le niveau d'amitié d'un de ses amis ou encore rompre une amitié.

Le troisième service concerne la gestion de sa boîte aux lettres électronique. L'utilisateur peut afficher la liste de ses messages (qui est stockée dans une *Pile*), écrire un nouveau message ou encore supprimer un des messages qu'il a reçus.

Le quatrième service concerne la gestion de ses futurs amis ou ennemis. L'utilisateur peut afficher la liste de tous les inscrits (qui est stockée dans un tableau classique), effectuer une recherche parmi tous les inscrits (par nom, lieu ou genre), inviter un inscrit à devenir son ami ou encore insulter un de ses ennemis.

III.2 - Le menu principal et ses sous menus

Le programme principal se charge de l'initialisation des différentes variables globales et de la gestion du menu principal. Rappelons qu'une variable globale est accessible depuis toute fonction définie après la déclaration de la variable en elle-même. Cette méthode est utilisée pour simplifier les échanges entre le menu principal et ses sous menus.

- Le menu principal propose 5 choix à l'utilisateur, visualisez la fonction d'affichage de ce menu et le *switch case* qui y correspond.

On dispose ensuite de 4 sous menus : « You » (votre profil), « Your Pals » (vos amis ou potes), « Your Mails » (votre boîte aux lettres) et « New Pals ? » (vos nouveaux amis).

- Chacun de ces sous menus propose 4 ou 5 actions à effectuer, visualisez les fonctions d'affichage de ces sous menus et les *switch case* qui y correspondent.

III.3 - Les types structurés et leurs fonctions

Plusieurs types structurés ont été déclarés. Vous devez les utiliser sans les modifier. Pour chacun d'entre eux, de nombreuses fonctions plus ou moins élémentaires ont été déclarées.

- Visualisez la définition du type *message* puis les 8 fonctions qui suivent.
- Visualisez la déclaration du type *Pile* puis les 5 fonctions élémentaires nécessaires à l'utilisation de ce TDA. Les deux dernières fonctions sont ou seront codées, entre autre, grâce aux 5 précédentes.
- Visualisez la définition du type *profil* puis les 17 fonctions qui suivent.
- Visualisez la déclaration du type *File* puis les 5 fonctions élémentaires nécessaires à l'utilisation de ce TDA. Les quatre dernières fonctions sont ou seront codées, entre autre, grâce aux 5 précédentes.

IV - EXERCICE 2 : menu 1 - Votre profil (20 min)

IV.1 - Exécution et constatation

- Prenez 5 minutes pour compiler, exécuter le programme et essayer le menu 1 (mettre à jour le profil, afficher le profil, modifier l'avatar, afficher le profil, supprimer le profil...).

Vous avez pu constater que la fonction de mise à jour fonctionne partiellement et que le profil préenregistré n'est pas modifié complètement.

IV.2 - Finition de la fonction *modifie_profil()*

Nous avons volontairement omis quelques lignes de cette fonction qui de fait ne fonctionne pas correctement.

- Effectuez les modifications nécessaires à la saisie des champs *prenom*, *nom*, *age*, *ville* et *pays* du type *profil*.

IV.3 - Exécution et conclusion

- Prenez 5 minutes pour compiler, exécuter le programme et essayer le menu 1 (mettre à jour le profil, afficher le profil, modifier l'avatar, afficher le profil, supprimer le profil...).

Vous devez maintenant disposer d'un module de gestion de profil complètement opérationnel.

V - EXERCICE 3 : menu 4 - Vos nouveaux amis (30 min)

V.1 - Exécution et constatation

- Prenez 5 minutes pour compiler, exécuter le programme et essayer le menu 4 (afficher la liste des inscrits, rechercher un profil, inviter un futur ami, insulter un ennemi...).

Vous avez pu constater un fonctionnement particulièrement encourageant mais trompeur. L'invitation par l'utilisateur d'un membre à devenir son ami n'est pas réellement prise en compte (vous pouvez le constater en invitant un *profil* depuis le menu 4 puis en affichant votre *File* d'amis depuis le menu 2).

V.2 - Rédaction du corps de la fonction *ajoute_profil_File()*

Vous pouvez constater que le corps de cette fonction est vide. Il paraît donc cohérent qu'on ne puisse ajouter de *profil* à la *File* des amis. Cette fonction n'est pas opérationnelle et vous allez devoir vous charger de la faire fonctionner.

- Rédigez les quelques lignes nécessaires à l'ajout d'un élément *profil* dans une *File*. Quelques indications précieuses vous sont données en commentaires, considérez-les avec attention.

V.3 - Exécution et conclusion

- Prenez 5 minutes pour compiler, exécuter le programme et essayer le menu 4. Pour vérifier que vos modifications ont porté leurs fruits, essayez d'inviter deux fois de suite le même profil. Un message d'erreur doit vous prévenir que ce contact est déjà dans votre *File* d'amis prouvant bien que la première insertion s'est faite avec succès.

Une fois ces modifications apportées, vous devez avoir un module de gestion de nouveaux amis opérationnel.

VI - EXERCICE 4 : menu 2 - Vos amis (40 min)

VI.1 - Exécution et constatation

- Prenez 5 minutes pour compiler, exécuter le programme et essayer le menu 2 (afficher la *File* des amis, modifier le niveau d'amitié, afficher la *File* des amis, rompre une amitié...).

Vous avez pu constater un fonctionnement parfaitement satisfaisant des deux premiers modules. La *File* d'amis de l'utilisateur contient maintenant 4 *profils* par défaut. Ceux-ci sont ajoutés lors de l'initialisation des variables globales en début de programme principal.

- Visualisez l'initialisation de la *File* d'amis et l'ajout des 4 *profils* en début de *main()*.

En revanche le troisième module n'a aucun effet sur la *File* des amis.

VI.2 - Rédaction du corps de la fonction *supprime_profil_File()*

Vous pouvez constater que le corps de cette fonction est vide. Il paraît donc cohérent qu'on ne puisse supprimer de *profil* de la *File* des amis. Cette fonction est inefficace et vous allez devoir la rédiger pour qu'elle effectue le service attendu.

- Rédigez les quelques lignes nécessaires à la suppression d'un élément *profil* d'une *File*. Quelques indications précieuses vous sont données en commentaires, considérez-les avec attention. Cette fonction est difficile à concevoir, n'hésitez pas à faire des schémas sur papier pour vous aider.

VI.3 - Exécution et conclusion

- Prenez 5 minutes pour compiler, exécuter le programme et essayer le menu 2. Dorénavant vous pouvez supprimer un *profil* de la *File* des amis.

Une fois ces modifications apportées, vous devez avoir un module de gestion des amis opérationnel.

VII - EXERCICE 5 : Votre boîte de réception (80 min)

VII.1 - Exécution et constatation

- Prenez 5 minutes pour compiler, exécuter le programme et essayer le menu 3 (afficher la *Pile* des messages, écrire un message, afficher la *Pile* des messages, supprimer un message...).

Vous avez pu constater que rien ne fonctionne. Cela signifie donc que de nombreuses parties du code sont manquantes.

VII.2 - Rédaction du corps de la fonction *initialise_Pile()*

Vous pouvez constater que le corps de cette fonction est vide.

- Rédigez les quelques lignes nécessaires à l'initialisation d'une *Pile*. Quelques indications précieuses vous sont données en commentaires, considérez-les avec attention.

VII.3 - Rédaction du corps de la fonction *ajoute_message_Pile()*

Vous pouvez constater que le corps de cette fonction est vide.

- Rédigez les quelques lignes nécessaires à l'ajout d'un élément *message* dans une *Pile* de *messages*. Quelques indications précieuses vous sont données en commentaires, considérez-les avec attention.

VII.4 - Rédaction du corps de la fonction *retire_message_Pile()*

Vous pouvez constater que le corps de cette fonction est vide.

- Rédigez les quelques lignes nécessaires au retrait d'un élément *message* d'une *Pile* de *messages*. Quelques indications précieuses vous sont données en commentaires, considérez-les avec attention.

VII.5 - Rédaction du corps de la fonction *est_vide_Pile()*

Vous pouvez constater que le corps de cette fonction est vide.

- Rédigez les quelques lignes nécessaires à tester si une *Pile* est vide. Quelques indications précieuses vous sont données en commentaires, considérez-les avec attention.

VII.6 - Rédaction du corps de la fonction *sommet_Pile()*

Vous pouvez constater que le corps de cette fonction est vide.

- Rédigez les quelques lignes nécessaires à la lecture de l'élément *message* du sommet d'une *Pile* de *messages*. Quelques indications précieuses vous sont données en commentaires, considérez-les avec attention.

VII.7 - Exécution et constatation

- Prenez 5 minutes pour compiler, exécuter le programme et essayer le menu 3 (afficher la *Pile* des *messages*, écrire un *message*, afficher la *Pile* des *messages*, supprimer un *message*...).

Une fois ces modifications apportées, vous devez avoir un module de gestion de la boîte aux lettres électronique qui fonctionne presque. La *Pile* de *messages* de l'utilisateur contient maintenant 5 *messages* par défaut. Ceux-ci sont ajoutés lors de l'initialisation des variables globales en début de programme principal.

- Visualisez l'initialisation de la *Pile* d'amis et l'ajout des 5 *messages* en début de *main()*.

Seul le module de suppression reste inefficace.

VII.8 - Rédaction du corps de la fonction *supprime_message_Pile()*

Vous pouvez constater que le corps de cette fonction est vide. Il paraît donc cohérent qu'on ne puisse toujours pas supprimer de *message* de la *Pile* des *messages*. Cette fonction est la dernière que vous allez devoir rédiger.

- Rédigez les quelques lignes nécessaires à la suppression d'un élément *message* d'une *Pile* de *messages*. Quelques indications précieuses vous sont données en commentaires, considérez-les avec attention. Cette fonction est difficile à concevoir, n'hésitez pas à faire des schémas sur papier pour vous aider.

VII.9 - Exécution et conclusion

- Prenez 5 minutes pour compiler, exécuter le programme et essayer le menu 3. Dorénavant vous pouvez supprimer un *message* de la *Pile* des *messages*.

Si vos modifications sont bonnes, vous devez disposer d'un module de gestion de la boîte aux lettres électronique opérationnel.

Vous avez à présent complété l'intégralité du programme. Celui-ci offre maintenant à l'utilisateur tous les services que nous vous avons décrits dans le début de ce document.

Merci de votre contribution et bonnes révisions.