IFT1169 – TRAVAIL PRATIQUE #2 – 30 octobre 2023

« généricité surchargée »
Mohamed Lokbani

Équipes : Le travail est à faire en monôme ou en binôme, pas plus de deux. Vous ne remettez qu'un seul travail par équipe.
Remise : une seule remise est à effectuer par voie électronique le mardi 21 novembre 2023, 23h59 au plus tard, sans possibilités de prorogation.
Conseils : n'attendez pas le dernier jour avant la remise pour engager votre travail. Vous n'aurez pas le temps nécessaire pour le réaliser.

But : ce TP a pour but de vous faire pratiquer la généricité et la surcharge des opérateurs.

Énoncé : ce travail est composé de deux exercices indépendants.

Exercice 1 : Le but de cet exercice est d'écrire que les méthodes nécessaires au bon fonctionnement de la classe générique « ATrier ». Cette classe contient la méthode « tri ». Cette méthode accepte deux arguments :

- Le premier argument est un tableau d'un type quelconque
- Le second argument est un entier N, la taille de ce tableau.

Cette méthode ne retourne rien. Son rôle est uniquement de trier le tableau passé en argument et de stocker le résultat du tri dans ce même tableau. Libre à vous d'implanter l'algorithme de tri de votre choix (vous ne devez pas faire appel à un des algorithmes de tri de la « STL »).

Quelques exemples d'utilisation de cette classe:

```
int tab[]={1,10,3,5,7};
double xx[]={0.5, 3.2, -6.1, 33.9};
ATrier<int> tInt;
ATrier<double> tDoub;
tInt.tri(tab,5);
tDub.tri(xx,4);
```

Faites en sorte pour que votre classe puisse traiter aussi la classe « Personne » dont la description est comme suit :

```
class Personne{
public:
      Personne();
      Personne(string Px,string Nx);
      // n==1 pour avoir le prénom,
      // n==2 pour avoir le nom
      string getPersonne(int n);
private:
      string Prenom,Nom;
};
```

```
ATrier<Personne> sPersonne;
Personne liste[20];
...
...
sPersonne.tri(liste,20);
```

Le tri doit se faire par ordre croissant. Dans le cadre de la classe Personne, ordre croissant, nom puis prénom.

Dans le rapport associé : décrire votre démarche et l'utilisation de vos programmes.

Exercice 2 : Le fichier « tp2exo02A23.cpp » contient les différentes opérations réalisées avec l'ensemble des méthodes de la classe « TempsdAntenne ». Toutes ces méthodes doivent être définies dans le fichier « TempsdAntenne.cpp ». En plus de ces méthodes, la classe « TempsdAntenne » contient les deux membres « heures » et « minutes » initialisés au départ avec la valeur « 0 ». La description de la classe doit être faite dans le fichier d'entête « TempdAntenne.h ».

- Écrire le contenu du fichier « TempsdAntenne.h ».
- Écrire le contenu du fichier « TempsdAntenne.cpp ».
- Ajuster l'heure si le nombre de minutes est égal ou dépasse 60 minutes.
- Comparer la sortie de votre programme avec la nôtre afin de vous assurer que votre programme est bien correct. Pour comparer deux sorties, voir les détails dans l'énoncé du TP#1.

Dans le rapport associé : décrire votre approche et les différentes fonctionnalités des méthodes de la classe « TempsdAntenne ».

Fichiers à remettre

Vous allez créer le répertoire « TP2A23 » dans lequel vous allez créer deux sous-répertoires « Exo1TP2A23 » et « Exo2TP2A23 » .

Dans le répertoire Exo1TP2A23 , vous allez déposer les fichiers : « ATrier.h » qui va contenir la classe générique « ATrier ». Le fichier « Personne.h » et « Personne.cpp » pour la définition de la classe « Personne ». Le fichier « tp1exo1A23.cpp » va contenir la fonction « main » permettant de tester votre programme. Vous inclure aussi un fichier « makefile » pour compiler les composantes de cet exercice.

Dans le répertoire Exo2TP2A23 , vous allez déposer les fichiers : les fichiers « TempsdAntenne.h », « TempsdAntenne.cpp » et « tp2exo02A23.cpp ». Vous inclure aussi un fichier « makefile » pour compiler les composantes de cet exercice.

Dans le répertoire « TP2A23 », vous allez inclure un « makefile » global qui fera appel au « makefile » associé à chaque exercice pour effectuer la compilation. La cible « make tp2a23exo1 » permet d'exécuter l'exercice 1 et la cible « make tp2a23exo2 » permet d'exécuter l'exercice 2.

Regrouper le répertoire « TP2A23 » dans le fichier compressé « TP2A23.zip » et remettre par la suite ce fichier dans le dépôt Studium du devoir.

Remise
Vérifiez que le code que vous avez produit (devant normalement fonctionner correctement chez vous) fonctionne aussi bien sur les ordinateurs de la DESI.

Barème
Ce TP2 est noté sur 13 points.

Exercice 1 : 5 points

Tri [int et double] : 2 points
Tri [personne] : 2 points
Rapport : 1 point

Exercice 2 : 6 points

Temps d'Antenne : 5 points
Rapport : 1 point

Avis global : 2 points

Programmation, clarté du code, etc.

Makefile : -2 points par exercice, s'il est manquant

Ne remettez pas celui fourni par votre éditeur. Il faut que votre « makefile » puisse fonctionner en ligne de commande.

Communications

- Par courriel : une seule adresse « dift1169@iro.umontreal.ca ».

Pour faciliter le traitement de votre requête, inclure dans le sujet de votre courriel, au moins la chaîne [IFT1169] et une référence au tp02.

- En personne : à la démo ou sur rendez-vous.

Mise à jour
30-10-2023 diffusion de l'énoncé.