

Les templates

TP 6

Objectifs

- ★ Manipulez les templates
- ★ Manipulez la surcharge d'opérateur

Contraintes

- Indentez vos fichier.
- La correction tiendra compte de la brièveté des méthodes que vous écrivez (évitiez les fonctions de plus de 25 lignes) ; n'hésitez pas à découper une méthodes en plusieurs sous-méthodes (privées) plus courtes.
- Votre code ne doit pas donner d'erreurs avec Valgrind (ni fuite mémoire, ni autre erreurs).
- Vous ne devez pas utiliser de fonction C quand un équivalent C++ existe.
- Pour l'UML, vous pouvez utiliser UMLet ou Umbrello.
- Les noms de classe commencent par une majuscule.
- Les noms de méthodes et d'attributs commencent par une minuscule.
- La convention de nommage des accesseur est `get_nom_attribut()` et `set_nom_attribut(...)`.
- Vous devez fournir un Makefile qui compile vos fichiers source et contient une règle clean ainsi qu'un programme de test.
- Le **code source et les diagrammes** doivent être *pusher* sur le git du TP

Préparation du TP

- Suivre le lien reçu par mail
- Cloner votre répertoire sur votre compte

`git clone https://epudx.polytech.upmc.fr/git/epu-n7-iob-tp6-login`

 l'adresse qui vous a été attribué.
- Pendant le TP n'oublier pas de comité régulièrement
- Le répertoire contient un fichier `main.cc` qui vous servira de programme de test.
- N'oublier pas de pusher l'ensemble

Le dépôt git ne doit pas contenir d'exécutable, ni de fichier objet, ni de fichier temporaire (*.~)

Exercice 1 Tableau2D

Compléter la classe template Tableau2D avec les méthodes suivantes :

- Un constructeur qui prend en argument le nombre de colonnes et le nombre de lignes ;
- Un opérateur d'accès en lecture (`i, j`) ;
- Un opérateur d'accès en écriture (`i, j`) ;
- Un constructeur de copie
- Un opérateur `=`
- Un opérateur `==`, qui compare le contenu des tableaux.
- Un opérateur `<<`

Exemple :

```
int main()
{
    Tableau2D<float> tab(10, 10);
    tab(2, 2) = 42;
    std::cout << tab(2, 2) << std::endl;
    Tableau2D<float> tab2 = tab;
    tab2(1, 1) = 1;
    if (tab2 == tab)
        std::cout << "equal" << std::endl;
    std::cout << tab2 << std::endl;
    return 0;
}
```

Exercice 2 Matrix

Écrivez une classe Matrix qui dérive de Tableau2D et qui ajoute les opérations propres aux matrices :

- opérateurs +, -, * entre matrices ;
- opérateurs +, -, *, / matrix/réels ;
- transposition ;

```
int main()
{
    Matrix<float> a1(5, 5, 1);
    Matrix<float> a2(5, 5, 0);
    Matrix<float> a3 = a2;
    Matrix<float> a4 = 2 * (a1 + a2 + a3);
    a3 = a4 / 2.0;
    std::cout << a3 << std::endl;
    Matrix<float> a5(5, 5, Matrix<float>::identity);
    std::cout << a5 << std::endl;
    return 0;
}
```