

BTS SN 2^{ème} ANNÉE – Option IR

IR2NUM5

LANGAGE C++ LES CLASSES

Durée : 4h

2020-2021

Introduction

1. CREATION D'UNE FONCTION TEMPLATE

2. CREATION D'UNE CLASSE TEMPLATE

La programmation orientée objet est basée sur l'utilisation des classes. Dans cette séance de travaux pratiques nous allons nous intéresser à :

- La création et l'utilisation d'une fonction template au sein d'un programme.
- La création et l'utilisation d'une classe template au sein d'un programme.

Dans chaque partie, il vous sera demandé de réaliser une certaine tâche professionnelle. A chaque fois, en conclusion de votre travail, vous devrez :



- Montrer au professeur que le **cahier des charges** a bien été rempli et répondre à ses questions ;
- Répondre aux questions du **document-réponse**.

Seront pris en compte dans l'évaluation de votre travail :

- La bonne réalisation des **installations** ;
- Le **soin** accordé au matériel ;
- Le bon **rangement** du matériel en fin de TP ;
- La **clarté** des explications données au professeur ;
- La **qualité rédactionnelle** du document-réponse.

En fin du TP, vous veillerez à :

- **Ranger** soigneusement le matériel ;
- Modifier le nom du document-réponse, en remplaçant les « **YYY** » par vos noms ;
- **L'envoyer** par mail ou le glisser dans la BAL de votre professeur.

Matériel disponible :



- Ordinateur équipé de *Windows 10*, et des logiciels *CodeBlock* ;
- Baies informatiques câblées et équipées de switches ;
- Câbles RJ45.

Ressources disponibles :



- Cours sur langage C++;
- « *Guide du Technicien* ».

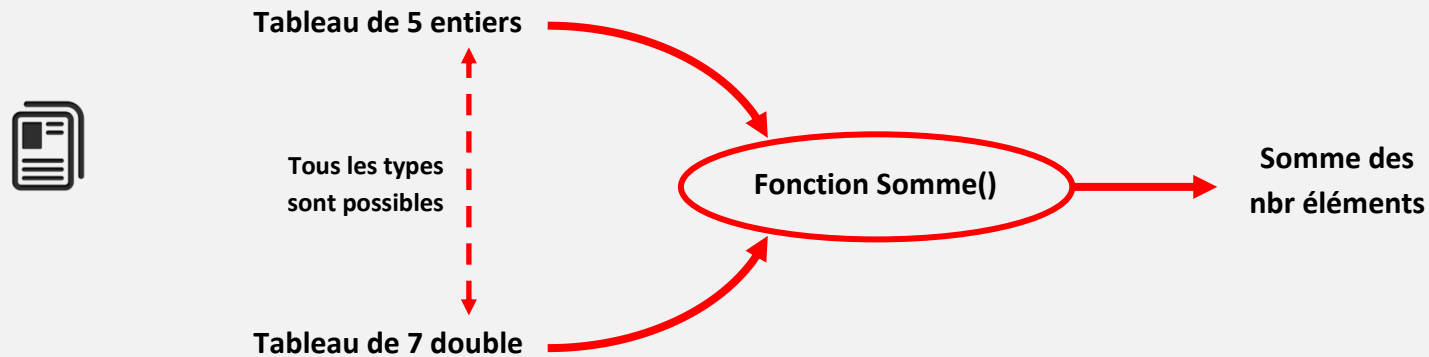
Introduction

1. CREATION D'UNE
FONCTION
TEMPLATE2. CREATION D'UNE
CLASSE TEMPLATE

1. CREATION D'UNE FONCTION TEMPLATE

Créez un projet de type console puis créez une fonction nommée somme dont les caractéristiques sont les suivantes :

- La fonction reçoit deux arguments l'un nommé tab de type tableau de générique et l'autre nommé nbr de type entier.
- La fonction renvoie une valeur de type générique correspondant à la somme des nbr éléments du tableau tab.



Dans le programme principal créez 2 tableaux de tailles différentes et de types différents que vous initialiserez. Appelez la fonction Somme() en lui passant le premier tableau avec le nombre nbr d'éléments. Renouvelez l'opération avec le second tableau puis affichez les deux sommes d'éléments calculés par la fonction somme().

Pour chacun des tableaux utilisez les deux types de déclaration de tableau. Une déclaration classique et une déclaration du type allocation dynamique de mémoire avec le mot clé "new".



- Complétez le compte-rendu de mesures.
- Commentez précisément l'affichage sur la console lors de l'exécution de votre programme

Introduction

1. CREATION D'UNE
FONCTION
TEMPLATE2. CREATION D'UNE
CLASSE TEMPLATE

2. CREATION D'UNE CLASSE TEMPLATE

Créez un projet de type console et ajoutez-y une classes template nommées Calculator. Cette classe devra comporter :

- Un constructeur sans argument dont le rôle est d'initialiser à 0 les attributs de la classe.
- Un constructeur avec deux arguments **X** et **Y** génériques permettant d'initialiser respectivement les attributs **num1** et **num2**.
- Un destructeur sans argument mais qui affiche lors de son exécution "Fin de Calculator"
- Deux attributs de type générique nommés **num1** et **num2**.
- 4 méthodes nommées **Add()**, **Sous()**, **Mult()** et **Div()** utilisant les deux attributs **num1** et **num2** de type générique pour réaliser les opérations et renvoyant un argument de type générique.
- Une méthode nommée **DisplayResult()** permettant d'afficher la somme, la soustraction, la multiplication et la division des attributs. Attention, cette méthode utilise les méthodes **Add()**, **Sous()**, **Mult()** et **Div()** précédentes.
- 2 méthodes nommées **ModifNum1** et **ModifNum2** recevant chacune un argument de type générique et permettant de modifier les attributs **num1** pour **ModifNum1** et **num2** pour **ModifNum2**.



Dans le programme principal créez 2 pointeurs de type **Calculator** que vous initialiserez avec la fonction "new". Chaque pointeur pointe sur un objet de type **Calculator** et est initialisé à des valeurs que vous choisirez mais qui doivent correspondre au choix du type utilisé lors de la création de l'objet.

Comme pour l'exercice précédent, utilisez les deux déclarations pour les instanciations des objets **Calculator**. L'une classique et l'autre avec une allocation dynamique de mémoire en utilisant les mots clés "new" puis "delete".



- Complétez le compte-rendu de mesures.
- Commentez précisément l'affichage sur la console lors de l'exécution de votre programme