



## Objectifs

- Réaliser l'itération 3 de l'application Gestion\_Log
  - Améliorer l'interaction utilisateur
  - Améliorer la modularité de la programmation
  - Valider les exigences du CDC (ESS03, REQ014) → DevOps

## Contenu technique

- Créer un module C++ (.h et .cpp)
- Programmer deux sous fonctions

## Durée 5h

# I. Améliorer la programmation : Création de fonctions

Notre programme commence à être plus complexe et nécessite de regrouper les fonctionnalités dans des modules dédiés.

Nous allons avant tout découper le programme en deux sous fonctions.

Q1. Regarder le DDC pour déterminer les deux sous fonctions à créer, leur rôle et leur prototype (signature)

## I.1 Sous fonction `afficherMenu()`

Q2. Créer la sous fonction `afficherMenu()`

Q3. Re factoriser votre programme pour appeler cette sous fonction

## I.2 Sous fonction `choisirLog()`

Q4. Créer la sous fonction `choisirLog()`

Q5. Re factoriser votre programme pour appeler cette sous fonction

## I.3 Validation

**Appeler l'enseignant pour qu'il valide votre travail**



## II. Améliorer la programmation : Création d'un module

Pour encore améliorer le programme, il est nécessaire de créer un module qui contiendra les deux sous fonctions.

Q6. Rechercher la définition de modularité d'un programme informatique

### II.1 QTCreator – Création d'un module

#### II.1.1 Composition d'un module

Un module est composé de deux fichiers :

- Le fichier .h contient les déclarations des fonctions
- Le fichier .cpp contient les définitions des fonctions

Ces deux fichiers .h et .cpp seront rangés dans un répertoire du nom du module.

#### II.1.2 CMakeList.txt

Le fichier CmakeList.txt indique au compilateur quels fichiers sont à inclure dans votre projet pour construire l'exécutable.

Il faut ajouter à ce fichier chaque nouveau module pour qu'il apparaisse dans votre arborescence de projet et soit utilisé pour construire l'exécutable.

Le fichier CmakeList.txt fournit pré-inclus des variables et il faut suffire de le modifier comme ci-après pour que votre module soit utilisé.

Les modifications que vous devez apporter à ce fichier sont en **gras**.

```
#1/3-Ajouter les fichiers sources de votre projet
set(SRCS
    Menu/menu.cpp
)
#2/3-Ajouter les fichiers headers de votre projet
set(HEADERS
    Menu/menu.h
)
#3/3-Ajouter les noms des dossiers de vos modules
include_directories(Menu)
```



### II.1.3 Création du module Menu

1. Créer un nouveau dossier appelé Menu dans votre projet
2. Créer un fichier menu.h (nouveau fichier C/C++ Header File) dans le dossier Menu
3. Créer un fichier menu.cpp (nouveau fichier C/C++ Source File) dans le dossier Menu
4. Modifier le CMakeList.txt pour ajouter votre module
5. Déplacer les deux fonctions afficherMenu() et choisirLog() dans le fichier menu.cpp
6. Déclarer les prototypes des fonctions dans menu.h
7. Inclure votre module (#include menu.h) dans votre fichier main

### II.1.4 Programmation modulaire

Q7. Vérifier que votre programme fonctionne correctement avec ce nouveau module

## II.2 Validation

**Appeler l'enseignant pour qu'il valide votre travail**



### III. Documentation

Suivre le tutoriel [Documenter son code](#) pour générer la documentation html de votre projet.

Les 3 fichiers (`main.cpp`, `menu.h`, `menu.cpp`) doivent être documentés.

Les 3 fonctions (`main()`, `afficherMenu()` et `choisirLog()`) doivent être documentés.

### IV. Tests de validation

Le document « `Gestion_Log_DDV.odt` » contient les tests à réaliser pour vérifier les exigences du projet Gestion Centralisée de logs.

- Effectuer le test ESS01
- Compléter le Dossier de Validation DDV

### V. Outils DevOps

Afin de finaliser votre travail, n'oubliez pas :

- D'indenter correctement votre code
- De documenter le code et les fonctions
- De sauvegarder votre travail sur Gitlab et de créer un (tag v3.0 correspondant à la fin de l'itération3)

### VI. Livrable

Sur le moodle de la section

- Votre compte rendu de TP avec les réponses aux questions de celui-ci
- Le dossier de validation complété
- La documentation générée (Télécharger l'artifact sur le pipeline)

Sur le serveur GIT de la section

- Le projet **<VOTRE\_NOM>\_Gestion\_log** avec le tag correspondant à la fin de l'itération3.