@startuml component-diagram

!theme plain

package "Application Console GnuCOBOL" {

[Programme Principal] as Main

package "Modules Utilitaires" {

[Date Utils] as DateUtils

[String Utils] as StringUtils

}

package "Modules Fonctionnels" {

[Calculator] as Calculator

[File Handler] as FileHandler

}

Main --> DateUtils : utilise

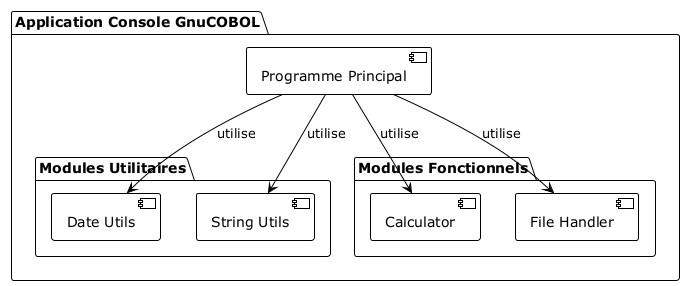
Main --> StringUtils : utilise

Main --> Calculator : utilise

Main --> FileHandler : utilise

}

@enduml



@startuml execution-flow

!theme plain

start

:Démarrage de l'application;

:Affichage du message d'accueil;

:Préparation du nom d'utilisateur;

:Appel à STRING\_UTILS pour générer une salutation;

:Affichage de la salutation;

:Récupération de la date actuelle;

:Appel à DATE\_UTILS pour formater la date;

:Affichage de la date formatée;

:Demande du premier nombre;

:Saisie du premier nombre;

:Demande du deuxième nombre;

:Saisie du deuxième nombre;

:Appel à CALCULATOR pour additionner les nombres;

:Affichage du résultat;

:Préparation du contenu du fichier;

:Appel à FILE\_HANDLER pour écrire dans un fichier;

if (Écriture réussie?) then (oui)

:Affichage du message de succès;

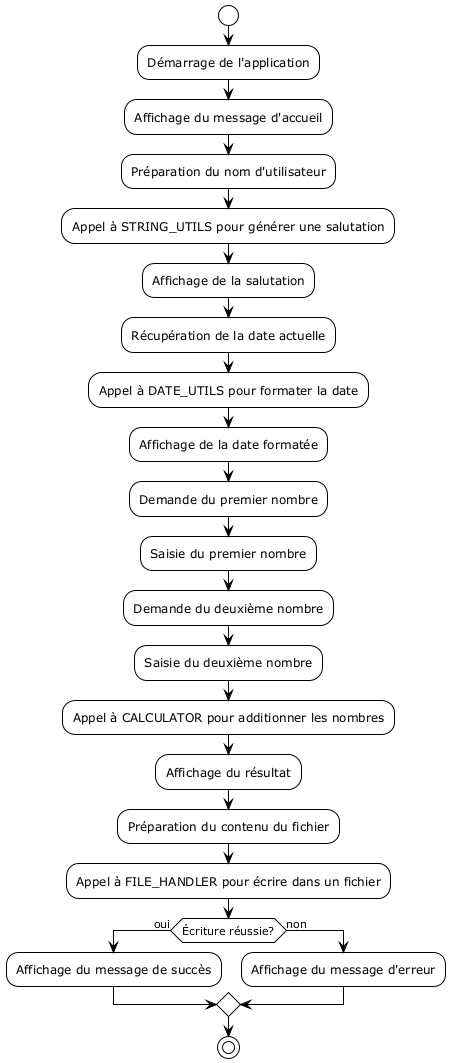
else (non)

:Affichage du message d'erreur;

endif

stop

@enduml



@startuml build-system

!theme plain

package "Système de Build CMake" {

[CMakeLists.txt principal] as MainCMake

[cmake/CobolFunctions.cmake] as CobolFunctions

[src/CMakeLists.txt] as SrcCMake

[tests/CMakeLists.txt] as TestsCMake

MainCMake --> CobolFunctions : inclut

MainCMake --> SrcCMake : ajoute sous-répertoire

MainCMake --> TestsCMake : ajoute sous-répertoire

note right of CobolFunctions

Définit les fonctions:

- add\_cobol\_executable()

- add\_cobol\_module()

end note

note right of SrcCMake

- Compile les modules

- Compile l'exécutable principal

- Crée les liens symboliques

- Génère run\_app.sh

end note

note right of TestsCMake

- Compile les tests

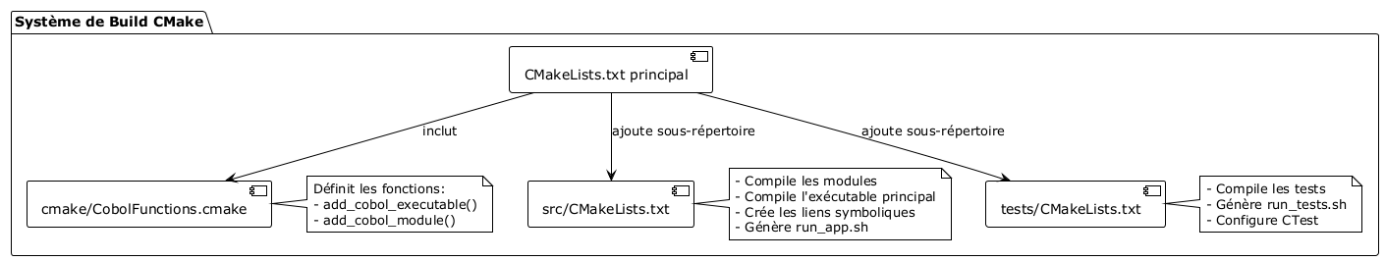
- Génère run\_tests.sh

- Configure CTest

end note

}

@enduml



@startuml structure-diagram

!theme plain

package "Programme Principal" {

class "MAIN" as Main {

+ WS-NUM1 : PIC 9(5)V99

+ WS-NUM2 : PIC 9(5)V99

+ WS-RESULT : PIC 9(6)V99

+ WS-CURRENT-DATE : structure

+ WS-FORMATTED-DATE : PIC X(10)

+ WS-USER-NAME : PIC X(30)

+ WS-GREETING : PIC X(50)

+ WS-FILENAME : PIC X(100)

+ WS-FILE-CONTENT : PIC X(1000)

+ WS-FILE-STATUS : PIC 9(2)

--

+ MAIN-PROCEDURE()

}

}

package "Modules Utilitaires" {

class "DATE\_UTILS" as DateUtils {

--

+ PROCEDURE DIVISION USING LS-CURRENT-DATE, LS-FORMATTED-DATE

}

class "STRING\_UTILS" as StringUtils {

--

+ PROCEDURE DIVISION USING LS-USER-NAME, LS-GREETING

}

}

package "Modules Fonctionnels" {

class "CALCULATOR" as Calculator {

--

+ PROCEDURE DIVISION USING LS-NUM1, LS-NUM2, LS-RESULT

}

class "FILE\_HANDLER" as FileHandler {

+ OUTPUT-FILE : file

+ WS-FILENAME : PIC X(100)

+ WS-FILE-STATUS : PIC X(2)

--

+ PROCEDURE DIVISION USING LS-FILENAME, LS-CONTENT, LS-STATUS

}

}

package "Tests" {

class "TEST\_RUNNER" as TestRunner {

+ WS-TEST-COUNT : PIC 9(3)

+ WS-TEST-PASSED : PIC 9(3)

+ WS-TEST-FAILED : PIC 9(3)

--

+ MAIN-PROCEDURE()

}

class "TEST\_DATE\_UTILS" as TestDateUtils {

+ WS-TEST-RESULT : PIC 9(1)

--

+ MAIN-PROCEDURE()

}

class "TEST\_STRING\_UTILS" as TestStringUtils {

+ WS-TEST-RESULT : PIC 9(1)

--

+ MAIN-PROCEDURE()

}

class "TEST\_CALCULATOR" as TestCalculator {

+ WS-TEST-RESULT : PIC 9(1)

--

+ MAIN-PROCEDURE()

}

class "TEST\_FILE\_HANDLER" as TestFileHandler {

+ WS-TEST-RESULT : PIC 9(1)

--

+ MAIN-PROCEDURE()

}

}

Main --> DateUtils : CALL

Main --> StringUtils : CALL

Main --> Calculator : CALL

Main --> FileHandler : CALL

TestRunner --> TestDateUtils : exécute

TestRunner --> TestStringUtils : exécute

TestRunner --> TestCalculator : exécute

TestRunner --> TestFileHandler : exécute

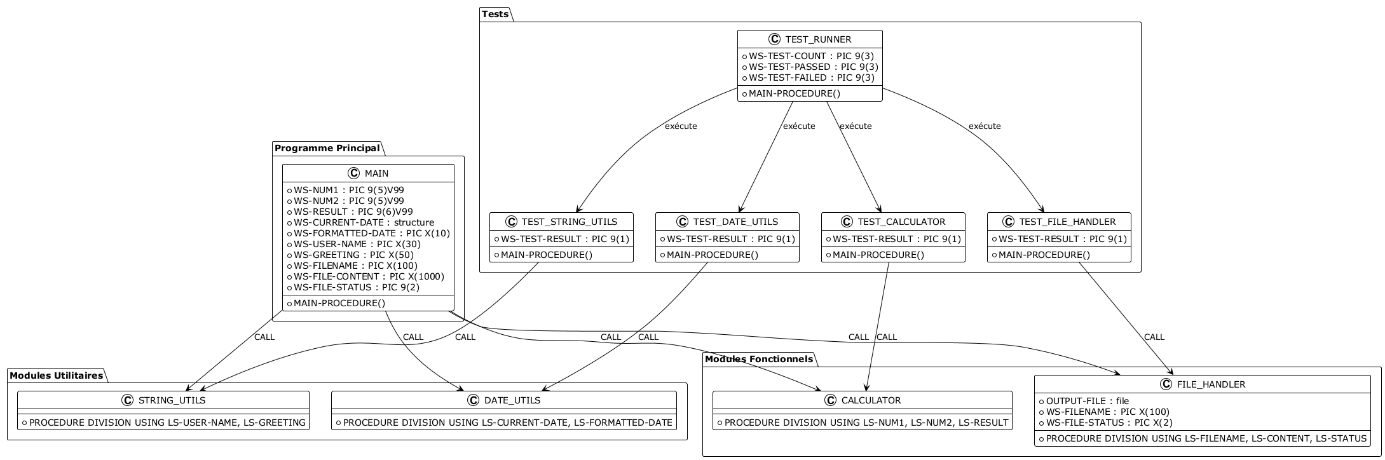
TestDateUtils --> DateUtils : CALL

TestStringUtils --> StringUtils : CALL

TestCalculator --> Calculator : CALL

TestFileHandler --> FileHandler : CALL

@enduml



@startuml test-execution

!theme plain

start

:Exécution du script run\_tests.sh;

:Initialisation des compteurs;

:Exécution de test\_date\_utils;

if (Test réussi?) then (oui)

:Incrémentation du compteur de succès;

else (non)

:Incrémentation du compteur d'échecs;

endif

:Exécution de test\_string\_utils;

if (Test réussi?) then (oui)

:Incrémentation du compteur de succès;

else (non)

:Incrémentation du compteur d'échecs;

endif

:Exécution de test\_calculator;

if (Test réussi?) then (oui)

:Incrémentation du compteur de succès;

else (non)

:Incrémentation du compteur d'échecs;

endif

:Exécution de test\_file\_handler;

if (Test réussi?) then (oui)

:Incrémentation du compteur de succès;

else (non)

:Incrémentation du compteur d'échecs;

endif

:Affichage du bilan des tests;

:Affichage du nombre total de tests;

:Affichage du nombre de tests réussis;

:Affichage du nombre de tests échoués;

if (Tous les tests ont réussi?) then (oui)

:Affichage du message de succès;

:Retourne le code 0;

else (non)

:Affichage du message d'échec;

:Retourne le code 1;

endif

stop

@enduml

