

Objectifs:

- Utiliser l'environnement Qt Creator
- Produire un algorithme et le coder en langage C++;
- Mise en oeuvre des instructions traitant les opérateurs booléens et logiques.

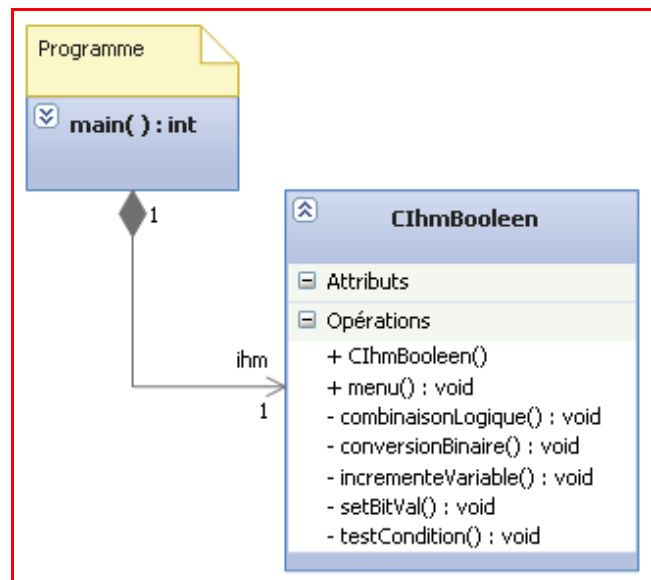
Travail à effectuer:

Coder, en langage C++, dans un seul programme, les 5 exercices proposés puis conclure.
N'oubliez pas de fournir un algorithme des procédures si nécessaire.

Remarque :

Les 5 exercices sont définis dans 5 méthodes différentes appartenant à la classe **CIhmBooleen**.
Le programme principale est composé d'un objet ihm de la classe **CIhmBooleen** et exécute la méthode `menu()` de la classe. La méthode `menu()` permet de choisir l'exercice à effectuer.

Le programme, sous Qt, doit respecter le diagramme de classe ci-dessous.



Exercice 1

- Ecrire la méthode `testCondition()` qui correspond à l'algorithme suivant:



Conclure.

Algorithme: Méthode `testCondition()`

Test de la variable val

variable: val = numérique

début

saisir val

si (val)

alors Afficher "val est vrai"

sinon Afficher "val est faux"

fin si

fin

Exercice 2

- Ecrire la méthode **combinaisonLogique()** qui saisit deux entiers *val1* et *val2*, puis afficher le résultat des combinaisons suivantes:

- a) $\sim val1$
- b) $val1 \& val2$
- c) $val1 / val2$
- d) $! val1$
- e) $val1 \&\& val2$
- f) $val1 \parallel val2$



Tester avec les valeurs: **val1 = 0,** **val2 = 1**
 val1 = 4, **val2 = 0**
 val1 = 6, **val2 = 8**

Conclure sur le comportement des opérateurs logiques et des opérateurs logiques bit à bit.

Exercice 3

- Ecrire la méthode **conversionBinaire()** qui saisit un entier et affiche l'entier en binaire sur 8 bits. On utilise les décalages bit à bit et la technique de masquage.



Afficher la variable **char code** lorsqu'elle vaut **0, 2, 127, -2, -128**.

Conclure sur la capacité de comptage et sur le rôle du bit 7.

Exercice 4

- Ecrire la méthode **incrimenteVariable()** qui incrémente une variable *code* de type **char** en boucle infini, et affiche le résultat de la variable *code*; on veut de surcroît que l'affichage attende une frappe au clavier lorsque **code vaut '1'**.

Commenter le résultat.

- Idem mais avec un **entier court (short int)**.



Exercice 5

- Ecrire la méthode **setBitVal()**, qui saisit une valeur *val* de type entier, une valeur *numero* de type **entier**, et qui met à "1" le *numero*-ième bit de *val*.

Exemple: si **val = 132** et **numero = 5** alors **val = 148**

132 = % 10000100

148 = % 10010100

