

Objectifs:

- Utiliser Qt Creator en mode console.
- Produire un programme et le coder en langage C++;
- Mise en oeuvre des instructions traitant les tableaux.
- Manipulation des données d'un tableau
- Déclaration et développement de la classe CIhmTableauEntier

Résumé de cours:Déclaration et utilisation d'un tableau:

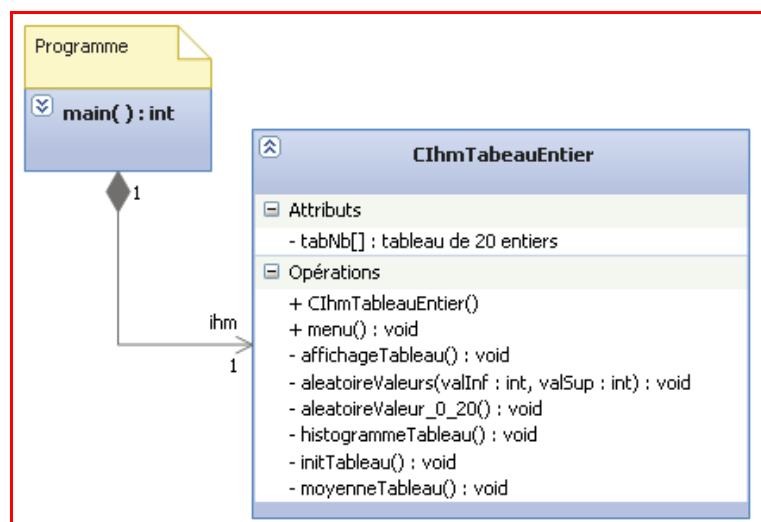
```
int tab[10] ; // déclare un tableau de 10 entiers de l'index 0 à l'index 9
tab[0]=33; // on met 33 dans l'élément n°0
```

```
// balayage du tableau (tableau de 10 éléments)
for ( int i = 0 ; i<10 ; i++ )
{
    tab[i] = ... // traitement
}
```

But : Définir la classe **CIhmTableauEntier** constituée d'un attribut **tabNb** (tableau de 20 entiers) et des procédures qui permettent l'initialisation du tableau, son affichage, son affectation par des valeurs aléatoires, et l'affichage de statistiques sur les valeurs du tableau (moyenne, valeurs minimale et maximale, histogramme des valeurs).

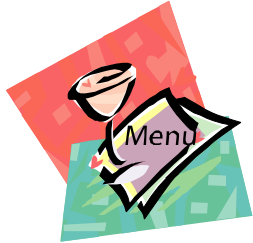
Le programme principale est composé d'un objet ihm de la classe **CIhmTableauEntier** et exécute la méthode `menu()` de la classe. La méthode `menu()` permet de choisir la manipulation que l'on souhaite effectuer au tableau d'entiers.

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



Travail à effectuer:

- a) Déclarer la classe **CihmTableauEntier** définie ci-dessus et développer le programme principal **main()** qui exécute la procédure **menu()** de la classe.
- b) Développer la procédure **menu()** de la classe **CihmTableauEntier**, qui effectue la gestion de l'affichage, et accède au choix de l'utilisateur en faisant appel à une procédure de la classe.



Utiliser les fonctions suivantes pour améliorer l'affichage:

```
system("cls"); // qui permet d'effacer l'écran  
getch(); // renvoie le caractère saisi au clavier  
gotoxy(int x, int y); // qui permet de positionner le  
 curseur d'affichage avant d'utiliser un affichage par la méthode cout.
```

Remarque : Pour l'utilisation de la fonction **gotoxy()** dans votre programme, copier le fichier **consoleJR.h** dans votre répertoire source puis inclure ce fichier dans votre fichier **cihhtableaumentier.h**

- c) Ecrire la procédure **void initTableau()** qui permet d'initialiser le tableau avec des valeurs à 0.
- d) Ecrire la procédure **void AffichageTableau()** qui permet de visualiser entièrement le tableau **tabNb**. On souhaite visualiser les indices associés aux valeurs du tableau.
- e) Ecrire la procédure **void aleatoireValeurs_0_20()** qui remplit le tableau de 20 entiers avec des valeurs aléatoires comprises entre 0 et 20.
Utiliser l'aide de la fonction **rand()** dans le fichier **QT - nbres aléatoires.pdf** pour résoudre l'exercice
- f) Ecrire la procédure **void aleatoireValeurs(int valInf, int valSup)** qui remplit le tableau de 20 entiers avec des valeurs aléatoires comprises entre **valInf** et **valSup**.
Tester la procédure par un choix dans le menu tel que : **aleatoireValeurs(5, 18);**
- g) Ecrire la procédure **void moyenneTableau()** qui calcule et affiche la moyenne, la valeur max et la valeur min des valeurs du tableau.
- h) Ecrire la procédure **void histogrammeTableau()** qui affiche successivement les valeurs 0 à 20 suivies du résultat du nombre de notes correspondant dans le tableau et compléter cette fonction pour visualiser les résultats sous forme d'un histogramme.

Exemple:

```
0 : 3 notes      : ***  
1 : 2 notes      : **  
2 : 0 note       :  
3 : 6 notes      : *****  
...  
19 : 1 note      : *  
20 : 0 note      :
```

