

### Exercice 1:

Définissez une classe **Mesure** qui modélise des mesures entières (températures par exemple) :

- ✚ une mesure a une valeur de type entier.
- ✚ le constructeur de la classe **Mesure** permet d'instancier un objet de cette classe avec la valeur entière fournie en paramètre. Si la valeur fournie est en-dehors de l'intervalle  $[-100,100]$  (Min et Max), le message ``**mesure hors intervalle**'' est affiché et la valeur n'est pas prise en compte.
- ✚ **NbMesures**, le nombre d'instances de la classe **Mesure** est géré par la classe ainsi que **SomMesures** la somme des valeurs des mesures instanciées.
- ✚ la méthode **afficherMoyenne** calcule et affiche la moyenne de toutes les mesures instanciées, le nombre de mesures est également affiché. S'il n'y a pas de mesures, le message ``zéro mesures'' est affiché.

La fonction **main** permet à l'utilisateur d'entrer un nombre arbitraire de valeurs au clavier. Ces valeurs sont utilisées pour instancier des Mesures.

Quand la valeur 0 est entrée, l'objet ne sera pas compté. Voici un exemple d'affichage lors de l'exécution du programme :

```
Entrez une mesure :120
Mesure hors intervalle
Entrez une mesure :10
Entrez une mesure : 20
Entrez une mesure :30
Entrez une mesure :0
La moyenne des 3 mesures est : 20
```

### Exercice 2:

Définir une classe **Matrice** permettant de représenter des matrices dynamiques de réels à deux dimensions dont la déclaration se présente ainsi:

```
class Matrice {
    char *nom; // nom = nom de la matrice
    float** mat;
    int nl, nc ; // nl = nombre de lignes et nc = nombre de colonnes
public:
    .....
    .....
};
```

La classe devra comporter:

- un constructeur acceptant le nom de la matrice, les tailles de chaque dimension (valeurs par défaut), une valeur d'initialisation des éléments de la matrice (valeur par défaut);
- un constructeur par copie;
- un destructeur;
- une méthode lecture permettant d'obtenir une matrice à partir d'un tableau à deux dimensions passé comme argument
- une méthode afficher() affichant la matrice à l'écran
- une méthode transposee() retournant la matrice transposée d'une matrice donnée.

### Exercice 3:

Ecrivez une classe Rationnel qui définit les nombres rationnels. Les fractions doivent être stockées de manière irréductible.

La classe a les attributs suivants :

- **numérateur** : Le numérateur;
- **dénominateur** : Le dénominateur.

La classe Rationnel doit disposer des constructeurs suivants :

- **Rationnel()**;
- **Rationnel(numérateur, dénominateur)**;
- **Rationnel(Rationnel)**.

Elle doit aussi contenir les méthodes :

- **ajouter(Rationnel)** : addition de deux nombres rationnels;
- **soustraire(Rationnel)** : soustraction de deux nombres rationnels;
- **multiplier(Rationnel)** : multiplication de deux nombres rationnels;
- **diviser(Rationnel)** : division de deux nombres rationnels;
- **evaluer()** : renvoie le résultat sous la forme d'un nombre réel;
- **toString()** : donne une représentation du rationnel (a/b).

Tester la classe Rationnel.