

TD n°2 C++

1 Exercice.1

Soit la classe suivante définissant les nombres complexes :

```
class Complexe
{
    float reel;
    float imag;
public :
    Complexe(float, float);
    ~Complexe(){}
};
```

Ajouter les données membres et les méthodes membres pour faire fonctionner le programme suivant :

```
int main()
{
    Complexe z1(1);
    Complexe z2(5,4);

    if (z1.reelPur()) cout<<"Complexe reel pur"<<endl;

    return 0;
}
```

2 Exercice.2

1. Ecrire la déclaration de la classe **Chat** incluant les données membres suivantes : ***mAge, *mPoids**. Ces variables membres seront de type **int *** (pointeur sur un entier). Pour l'instant procédez seulement à l'écriture des prototypes des méthodes membres.
2. Ecrire le corps du constructeur par défaut, en donnant pour valeur par défaut 1 pour mAge et 2 pour mPoids. Il convient bien évidemment d'utiliser le mot clé **new**.
3. Ecrire le corps du destructeur de la classe **Chat**.
4. Rajouter les méthodes d'accès **LireAge** et **DefAge**.
5. Finalement écrire la fonction **main()** qui crée une instance sur la classe **Chat** qui aura la forme suivante : **Chat *Poussy**; et utilisez l'ensemble des méthodes écrites ci-dessus.

3 Exercice.3

Le mot clé **new** permet d'allouer de la mémoire dont l'adresse sera stockée dans un pointeur. Ce mot clé permet en outre d'allouer plusieurs espaces mémoires, exemple : **float *pPointeur = new float[10]**; dans ce cas pPointeur contient l'adresse d'un tableau sur des flottants qui aura 10 cases numérotées à partir de 0. On accède à chaque élément du tableau soit en utilisant la notion de déréférencement du pointeur ***(pPointeur+i)** ou bien tout simplement **pPointeur[i]** avec i=0..9.

Question : écrire une classe permettant de gérer un ensemble d'entiers représentés par un tableau alloué dynamiquement. Cette classe doit pouvoir s'utiliser avec le programme suivant :

```
int main()
{
    TabFloat ens(10); // creation d'un tableau de 10 flottants
    int i;
    for (i=0;i<ens.Taille();i++) ens.ajoute(i); // affecte des valeurs
                                                //à toutes les cases du tableau.

    if (ens.appartient(3)) // si 3 est présent dans le tableau
        cout<<"3 appartient au tableau"<<endl;
    else
        cout<<"3 n'appartient pas au tableau"<<endl; //sinon
    // Calcul de la somme des éléments présents dans le tableau.
    cout<<"la somme des éléments saisis du tableau : "<<ens.somme();
    return 0;
}
```