

## Travaux Pratiques

### Travaux Dirigés de synthèse

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100



🚦 Ce TP réalisé par :

ZRAIDI NAJWA  
(GLSID1)

- ❖ L'objectif de ce programme permettant à résoudre le problème de gestion des employes dans une entreprise .

```
#include <iostream>
#include <string>
#include <conio.h>
using namespace std;

class Employe
{
    int Matricule;
    string Nom;
    float Salaire;
    int Age;
public :
    Employe(int,string,int,float); //Constructeur
    string ToString()const; //permettant d'afficher toutes les caracteristique
d'un employe
    float GetSalaire(); //permettant de retourner la salaire d'un employe
    void SetSalaire(float); //permettant de changer le salaire d'un employe
    void AfficherS(); //permettant d'afficher le salaire d'un employe
    int GetAge(); //permettant de retourner l'age d'un employe
    void SetAge(int); //permettant de changer l'age d'un employe
    void AfficherA(); //permettant d'afficher l'age d'un employe
    ~Employe(); //Destructeur
};

class Adherent : public Employe {
    string date_adherence;
    float montant;
public :
    Adherent(int,string,int,float,string,float);
    float GetMontant();
    void SetMontant(float);
    float GetSalaire();
    void AfficherS();
};

Employe :: Employe(int mat,string nom,int age,float sal){
    this->Matricule=mat;
    this->Nom=nom;
    this->Age=age;
    this->Salaire=sal;
}

string Employe :: ToString() const {
    return "Matricule : " + to_string(this->Matricule) + "\n Nom : " + this-
>Nom + "\n Age : " +to_string(this->Age)+"\n Salaire : " +to_string(this-
>Salaire) +"\n" ;
}

float Employe :: GetSalaire() { return this->Salaire; }
void Employe :: SetSalaire(float salaire){ this->Salaire=salaire; }
```

```

void Employee :: AfficherS(){ cout<<"Le salaire est : "<<this->Salaire;}
int Employee :: GetAge() { return this->Age; }
void Employee :: SetAge(int Age){ this->Age=Age; }
void Employee :: AfficherA(){ cout<<"L'age est : "<<this->Age;}
Adherent :: Adherent(int mat,string nom,int age,float sal,string date,float
montant) : Employee(mat,nom,age,sal)
{
    this->date_adherence=date;
    this->montant=montant;
}
float Adherent :: GetMontant() { return this->montant; }
void Adherent :: SetMontant(float Montant){ this->montant=Montant; }
float Adherent :: GetSalaire()
{
    return Employee :: GetSalaire()-this->montant; }
void Adherent :: AfficherS()
{ cout<<"Le salaire est : "<<this->GetSalaire()<<endl;}
Employee :: ~Employee()
{
    cout<<"\nTaper une touche pour continuer ...";
    getch();
    cout<<"\nDestruction du l'employe :\n Matricule : "<<Matricule<<"\n Nom :
"<<Nom<<"\n Age : "<<Age<<"\n Salaire : "<<Salaire;
}
int main()
{
    Employee E(1,"Kamis",26,3000);
    Adherent EA(1,"Kamis",26,3000,"27/01/2022",500);
    cout<<"l'employe : \n "<<E.ToString()<<endl;
    E.AfficherS();
    cout<<endl;
    E.SetAge(27);
    E.AfficherA();
    cout<<endl;
    cout<<"\n-----"<<endl;
    EA.GetSalaire();
    EA.AfficherS();
    cout<<"le montant est : "<<EA.GetMontant()<<endl;
    EA.SetMontant(250);
    EA.AfficherS();
}

```

❖ L'exécution de programme donne :

```
C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\Travaux Dirigés de synthèse.exe
l'employe :
Matricule : 1
Nom : Kamis
Age : 26
Salaire : 3000.000000

Le salaire est : 3000
L'age est : 27

-----
Le salaire est : 2500
le montant est : 500
Le salaire est : 2750

Taper une touche pour continuer ...
Destruction du l'employe :
Matricule : 1
Nom : Kamis
Age : 26
Salaire : 3000
Taper une touche pour continuer ...
Destruction du l'employe :
Matricule : 1
Nom : Kamis
Age : 27
Salaire : 3000
-----
Process exited after 2.858 seconds with return value 0
Appuyez sur une touche pour continuer...
```