

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4  class Personne
5  {
6  private:
7      string nom;
8      int age;
9  public:
10     void incrementerAge(int);
11     void setAge(int);
12     int getAge();
13     void setnom(string);
14     string getnom();
15     void afficher();
16
17 };
18 int Personne::getAge()
19 {
20     return age;
21 }
22 void Personne::incrementerAge(int)
23 {
24     age++;
25 }
26 void Personne::setAge(int x)
27 {
28     if(x>age)
29     {
30         age=x;
31     }
32 }
33 void Personne::setnom(string ch)
34 {
35     nom=ch;
36 }
37 string Personne::getnom()
38 {
39     return nom;
40 }
41 void Personne::afficher()
42 {
43     cout << nom << age << endl;
44 }
45 class Compte
46 {
47     int numCompte;
48     Personne proprietaire;
49     double solde;
50     public:
51         void setSolde(double);
52         double getSolde();
53         void setnumCompte(int);
54         void setProprietaire(Personne);
55         Personne getProprietaire();
56         double retirerArgent(Compte,double montant);
57         double deposterArgent(Compte,double montant);
58         void afficherCompte();
59
60 };
61 void Compte::setSolde(double s)
62 {
63     solde=s;
64 }
65 double Compte::getSolde()
66 {

```

```

67     return solde;
68 }
69 void Compte::setnumCompte(int num)
70 {
71     numCompte=num;
72 }
73 void Compte::setProprietaire(Personne p)
74 {
75     proprietaire=p;
76 }
77 Personne Compte::getProprietaire()
78 {
79     return proprietaire;
80 }
81 double Compte::retirerArgent(Compte,double montant)
82 {
83     if(solde>montant)
84     {
85         solde-=montant;
86     }
87     return solde;
88 }
89 double Compte::deposerArgent(Compte,double montant)
90 {
91     solde+=montant;
92     return solde;
93 }
94 void Compte::afficherCompte()
95 {
96     //cin >> numCompte;
97     //cin >> proprietaire;
98     //cin >> solde;
99     // cout << numCompte << endl;
100    // cout << proprietaire.getnom << endl;
101    cout << solde << endl;
102 }
103 class Banque
104 {
105     Compte account;
106     Personne proprietaire;
107     int nombreComptes;
108     Compte accounts[nombreComptes];
109     public:
110         string nom;
111         Compte ouvrirCompte(int , Personne);
112         void transfertArgent(Compte,Compte,double montant);
113         void store(int nombreComptes);
114         //void deleteCompte();
115 };
116 void Banque::transfertArgent(Compte comptel,Compte compte2,double montant)
117 {
118     double d1,d2;
119     d1=Compte::retirerArgent(comptel,montant);
120     d2=Compte::deposerArgent(compte2,montant);
121 }
122 Compte Banque::ouvrirCompte(int num_compte, Personne proprietaire)
123 {
124     cin << num_compte << endl;
125     cin << proprietaire << endl;
126     //setnumCompte(int);
127     //setProprietaire(Personne);
128     cout << "compte crée" << endl;
129     return account;
130 }
131 void store(int nombreComptes)
132 {

```

```
133     if account=ouvrirCompte(int num_compte,Personne)
134     {
135         nombreComptes+=1;
136     }
137     for(int i=0; i<nombreComptes; i++)
138     {
139         accounts[i]=account;
140     }
141 }
142 /*
143 Compte Banque::deleteCompte()
144 {
145
146 }*/
```