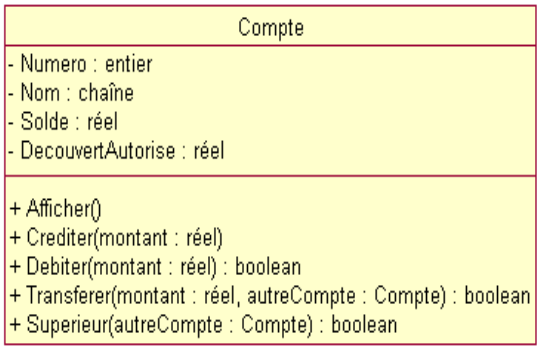
```
class Banque{
    public Banque(){
        LesComptes = new Compte[20];
        this.NbComptes = 0;
    }
    public void Init(){
        Compte c1 = new Compte(12345,"toto",1000,-500);
        Compte c2   = new Compte(45657, "titi",2000,-1000);
        Compte c3 = new Compte(32345,"dupond",1500,-1500);
        Compte c4   = new Compte(11633, "durand",1200,-500);
        Compte c5 = new Compte(2568,"dubois",-200,-500);
        Compte c6   = new Compte(8978, "duval",750,-2000);
        this.AjouteCompte(c1);
        this.AjouteCompte(c2);
        this.AjouteCompte(c3);
        this.AjouteCompte(c4);
        this.AjouteCompte(c5);
        this.AjouteCompte(c6);
        }
    private void AjouteCompte(Compte unCompte){
        LesComptes[NbComptes++] = unCompte;
        }
private int NbComptes;
private Compte[] LesComptes;
}
```



Ebauche de la classe banque.

Travail avec un tableau statique au début puis on passera au tableau dynamique les collections beaucoup plus pratique dans ce cas.

°) Les méthodes de la classe Compte.

Reconstituer le fichier en intégrant à la classe Compte la nouvelle classe Banque ainsi que la classe :

```
class AppCompte{
      static void Main(){
            Banque b = new Banque();
            b.Init();
     }
}
```

Compiler pour vérifier la bonne construction de votre programme

Travail à faire.

a) Ecrire une méthode AfficherComptes() de la classe Banque, tester en écrivant :

b.AfficherComptes();

b) Ecrire une méthode public void AjouteCompte(int n,string nom,double solde, double dec)

La tester avec :

Banque b = new Banque();

b.Init();

b.AjouteCompte(1245,"dutronc",4500,-500);

b.AfficherComptes();

c) Ecrire une méthode publique CompteSup() de la classe Banque qui retourne le compte au solde maximum. Tester avec :

b.CompteSup().Afficher();

d) Ecrire une méthode RendCompte de la classe Banque qui retourne un compte en fonction de son numéro. La fonction retourne null si le compte n'est pas trouvé. Pour cela vous rajouterez d'abord dans la classe Compte un accesseur public sur le numéro de compte. Tester avec

Compte c = b.RendCompte(1245);

if(c!=null)

c.Afficher();

e) Ecrire une méthode qui va transférer une somme d'un compte vers un autre compte.

Tester avec :

if(b.Tranferer(1245,2568,1000)) // 1000 euros du compte 1245 passe vers le compte 2568

Console.WriteLine("transfert effectué);

else

Console.WriteLine("transfert impossible);

Tester une deuxième fois avec : b.Tranferer(11633,32345,2000) // le transfert sera impossible