

## Programme Compte bancaire (code)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace compte_Bancaire
{
    /// <summary>
    /// Classe Program
    /// </summary>
    public class Program
    {
        /// <summary>
        /// Structure Compte
        /// </summary>
        public struct compte
        {
            /// <summary>
            /// numero de Compte en entier
            /// </summary>
            public int num;
            /// <summary>
            /// Nom du titulaire du Compte en caractere
            /// </summary>
            public string nom;
            /// <summary>
            /// Solde du Compte en decimal
            /// </summary>
            public double solde;
            /// <summary>
            /// Decouvert autoriser du Compte en entier
            /// </summary>
            public int decouvert;
        }
        /// <summary>
        /// Procedure init qui permet d'initialiser un compte
        /// </summary>

        public static void init(compte[] tab, compte c, ref int i)
        {
            tab[i].num = c.num;
            tab[i].nom = c.nom;
            tab[i].solde = c.solde;
            tab[i].decouvert = -(c.decouvert);
            i++;
        }
    }
}
```

```
/// <summary>
/// Fonction qui permet d'afficher un compte
/// </summary>
```

```
public static string afficher1compte(compte c)
{
    string Affichage;
```

```
    Affichage = ("Numero de compte " + c.num + "\n Nom du proprietaire: " + c.nom + "\n
Solde du compte: " + c.solde + "\n Decouvert autorise " + c.decouvert);
```

```
    return (Affichage);
}
```

```
/// <summary>
/// Procedure qui permet de debiter un compte
/// </summary>
```

```
public static void debiter(compte[] tab, int i, double debit)
{
    if (tab[i].solde - debit > tab[i].decouvert)
    {
        tab[i].solde = tab[i].solde - debit;
    }
}
```

```
/// <summary>
/// Procedure qui permet de crediter un compte
/// </summary>
```

```
public static void crediter(compte[] tab, int i, double credit)
{
    tab[i].solde = tab[i].solde + credit;
}
```

```
/// <summary>
/// Fonction qui permet de definir si un compte existe
/// </summary>
```

```
public static bool existCompte(compte[] tab, int nbc, int numCompte)
{
    bool trouve = false;
    int i = 0;

    while ((i < nbc) && (trouve == false))
    {
        if (numCompte == tab[i].num)
        {
            trouve = true;
        }
    }
}
```

```

    }
    else
    {
        i++;
    }
}

return (trouve);

}
/// <summary>
/// Fonction qui permet de retourner l'indice d'un compte
/// </summary>
public static int indiceCompte(compte[] tab, int nbc, int numCompte)
{
    bool trouve = false;
    int i = 0;
    int k = -1;

    while ((i < nbc) && (trouve == false))
    {
        if (numCompte == tab[i].num)
        {
            trouve = true;
        }
        else
        {
            i++;
        }

        if (trouve == true)
        {
            k = i;
        }
    }

    return (k);
}

/// <summary>
/// Fonction qui permet de rechercher un compte
/// </summary>
public static compte getCompte(compte[] tab, int nbc, int recherche)
{
    compte compteRecherche = new compte();
    bool trouve = false;
    int i = 0;

    while ((i < nbc) && (trouve == false))
    {

```

```

        if (recherche == tab[i].num)
        {
            trouve = true;
        }
        else
        {
            i++;
        }

        if (trouve == true)
        {
            compteRecherche = tab[i];
        }
    }

    return (compteRecherche);
}
/// <summary>
/// Programme principal
/// </summary>
static void Main(string[] args)
{

    compte[] tabcpt = new compte[100];
    compte c = new compte();
    string nom;
    int nbc = 0;
    int reponse, numero;
    int indice = -1;
    string affichage;
    double credit = 0;
    double debit = 0;
    int i = 0;
    double solde;
    int decouvert;

    do
    {

        Console.WriteLine("1 - Saisir un compte");
        Console.WriteLine("2 - Afficher un compte");
        Console.WriteLine("3 - Crediter un compte");
        Console.WriteLine("4 - Debiter un compte");
        Console.WriteLine("5 - Afficher tous les comptes\n");
        Console.WriteLine("Quel est votre choix ??");

        reponse = Convert.ToInt32(Console.ReadLine());

        switch (reponse)
        {

```

```

case 1:
{
    Console.WriteLine("Saisir un numero de compte:");
    numero = Convert.ToInt32(Console.ReadLine());

    while (existCompte(tabcpt, nbc, numero) == true)
    {
        Console.WriteLine("Ce compte existe deja !! Saisissez un nouveau numero de
compte:");
        numero = Convert.ToInt32(Console.ReadLine());
    }

    Console.WriteLine("Saisir un nom:");
    nom = Console.ReadLine();
    Console.WriteLine("Saisir un solde:");
    solde = Convert.ToDouble(Console.ReadLine());
    Console.WriteLine("Saisir un decouvert:");
    decouvert = Convert.ToInt32(Console.ReadLine());
    c.num = numero;
    c.nom = nom;
    c.decouvert = decouvert;
    c.solde = solde;

    init(tabcpt, c, ref nbc);
    break;
}

case 2:
{
    Console.WriteLine("Saisir le numero de compte à afficher :");
    numero = Convert.ToInt32(Console.ReadLine());

    if (indiceCompte(tabcpt, nbc, numero) == -1)
    {
        Console.WriteLine("Numero de compte incorrect");
    }
    else
    {
        compte cpt = getCompte(tabcpt, nbc, numero);
        affichage = afficher1compte(cpt);
        Console.WriteLine(affichage);
    }
    break;
}

case 3:
{
    Console.WriteLine("Saisir le numero de compte à crediter :");
    numero = Convert.ToInt32(Console.ReadLine());
    indice = indiceCompte(tabcpt, nbc, numero);

```

```

        if (indice == -1)
        {
            Console.WriteLine("Ce compte n'existe pas !!");
        }
        else
        {
            Console.WriteLine("Saisir le montant à créditer :");
            credit = Convert.ToSingle(Console.ReadLine());
            crediter(tabcpt, indice, credit);
            Console.WriteLine(" Crédit de " + credit + " sur le compte numero: " +
tabcpt[indice].num + "\n Le nouveau solde du compte est de: " + tabcpt[indice].solde);
        }
        break;

    }
    case 4:
    {
        Console.WriteLine("Saisir le numero de compte a debiter :");
        numero = Convert.ToInt32(Console.ReadLine());
        indice = indiceCompte(tabcpt, nbc, numero);

        if (indice == -1)
        {
            Console.WriteLine("Ce compte n'existe pas !!");
        }
        else
        {
            Console.WriteLine("Saisir le montant a debiter :");
            debit = Convert.ToSingle(Console.ReadLine());
            debiter(tabcpt, indice, debit);
            Console.WriteLine(" Debit de: " + debit + " sur le compte numero: " +
tabcpt[indice].num + "\n Le nouveau solde du compte est de: " + tabcpt[indice].solde);
        }

        break;
    }
    case 5:
    {
        for (i = 0; i < nbc; i++)
        {
            affichage = (" Numero de compte: " + tabcpt[i].num + "\n Nom du propriétaire
du compte: " + tabcpt[i].nom + " \n Solde du compte: " + tabcpt[i].solde + "\n Decouvert autorise
pour le compte: " + tabcpt[i].decouvert);
            Console.WriteLine(affichage + "\n");
        }
        break;
    }
}

```

```
    } while (reponse < 6);
```

```
    }
```

```
  }
```

```
}
```