

TP Exceptions

I. Compte.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace TPException
{
    public class Compte
    {
        private int c_numero;
        private String c_nom;

        private double c_solde;
        private static int nbcomptes = 0;

        public String C_nom
        {
            get { return c_nom; }
            set { c_nom = value; }
        }
        // le constructeur de la classe
        public Compte(int numero, String nom, double montant)
        {
            c_numero = numero;
            c_nom = nom;
            c_solde = 0;
            if (nom == "")
            {
                throw new TitulaireInexistantException("Il n'y a pas de titulaire");
            }
            else if (montant == 0)
            {
                throw new SoldeAZeroException("Le solde est nul");
            }
            else
            {
                crediter(montant);
            }
            nbcomptes++;
        }
        public bool compte_est_superieur(Compte CO)
        {
            bool c;
            if (c_solde > CO.c_solde)
                c = true;
            else
                c = false;
            return c;
        }
    }
}
```

```

    public static int nbelements()
    {
        return nbcomptes;
    }
    public void crediter(double m)
    {
        c_solde = c_solde + m;

    }
    public void debiter(double m)
    {
        if ((c_solde - m) > 0)
        {
            c_solde = c_solde - m;
        }
        else
        {
            throw new SoldeNegatifException("Le solde est négatif vous ne pouvez
pas débiter");
        }

    }

    public double getsolde()
    {
        return c_solde;
    }

    private void fraisgestion()
    {
        debiter(1);
    }
    public void afficher()
    {
        Console.WriteLine(" le numero du compte est " + c_numero);
        Console.WriteLine("le titulaire      s'appelle " + c_nom);
        Console.WriteLine("le solde est de " + c_solde);
    }
    public void editionReleve()
    {
        Console.WriteLine(" ");
        Console.WriteLine("-----");
        Console.WriteLine("Titulaire: " + c_nom);

        Console.WriteLine("-----");
        Console.WriteLine("Solde du compte : " + getsolde());
        Console.WriteLine("-----");

    }

}
} // fin classe Compte
}

```

II. LivretA.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace TPException
{
    class LivretA : Compte
    {
        private static double _taux;
        private static int _plafond;

        public LivretA(int numero, string nom, double montant) : base(numero, nom,
montant)
        {
            _taux = 0.025;
            _plafond = 20000;
        }
        public new void crediter(double m)
        {
            double epargne = base.getsolde() + m;
            if (epargne > 20000)
            {
                throw new DepassementPlafondException("Crédit impossible : Plafond
du livret atteint avec ce montant");
            }
            else
            {
                base.crediter(m);
            }
        }
    }
}
```

III. SoldeAZeroException.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace TPException
{
    class SoldeAZeroException : Exception
    {
        public SoldeAZeroException(string message) : base(message)
        {
        }
    }
}
```

IV. TitulaireInexistantException.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace TPException
{
    class TitulaireInexistantException : Exception
    {
        public TitulaireInexistantException(string message) : base(message)
        {
        }
    }
}
```

V. SoldeNegatifException.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace TPException
{
    class SoldeNegatifException : Exception
    {
        public SoldeNegatifException(string message) : base(message)
        {
        }
    }
}
```

VI. DepassementPlafondException.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace TPException
{
    class DepassementPlafondException : Exception
    {
        public DepassementPlafondException(string message) : base(message)
        {
        }
    }
}
```

VII. Jeux d'essai

Création d'un compte avec solde nul :

```
Compte c1 = new Compte(5486, "Richard", 0);
```

```
else if (montant == 0)
{
    throw new SoldeAZeroException("Le solde est nul");
}
else
{
    crediter(montant);
}
nbcomptes++;
```

Exception non gérée
TPException.SoldeAZeroException : 'Le solde est nul'

Afficher les détails | Copier les détails

Création d'un compte sans titulaire entré :

```
Compte c2 = new Compte(5752, "", 10);
```

```
{
    throw new TitulaireInexistantException("Il n'y a pas de titulaire");
}
else if (montant == 0)
{
    throw new SoldeAZeroException("Le solde est nul");
}
else
```

Exception non gérée
TPException.TitulaireInexistantException : 'Il n'y a pas de titulaire'

Afficher les détails | Copier les détails

Test de débit supérieur au solde du compte :

```
LivretA l1 = new LivretA(4578, "Dupont", 578);
l1.debiter(600);
```

```
solde = c_solde - m;
throw new SoldeNegatifException("Le solde est négatif vous ne pouvez pas débiter");
```

Exception non gérée
TPException.SoldeNegatifException : 'Le solde est négatif vous ne pouvez pas débiter'

Afficher les détails | Copier les détails
Paramètres d'exception

Test du dépassement du plafond de livret (Le premier **ReadLine()** nous prouve que créditer fonctionne et le deuxième génère l'exception pour dépassement) :

```
LivretA l1 = new LivretA(4578, "Dupont", 578);
l1.crediter(15000);
l1.afficher();
Console.ReadLine();
l1.crediter(15000);
l1.afficher();
Console.ReadLine();
```

```
le numero du compte est 4578
le titulaire s'appelle Dupont
le solde est de 15578
```

```
Exception non gérée
impossible : Plafond du livret atteint
```

TPException.DepassementPlafondException : 'Crédit impossible : Plafond du livret atteint avec ce montant'

Afficher les détails | Copier les détails
Paramètres d'exception

VIII. Program.cs

```
using System;

namespace TPexception
{
    class Program
    {
        static void Main(string[] args)
        {
            //Compte c1 = new Compte(5486, "Richard", 0);
            //Compte c2 = new Compte(5752, "", 10);
            LivretA l1 = new LivretA(4578, "Dupont", 578);
            l1.crediter(15000);
            l1.afficher();
            Console.ReadLine();
            l1.crediter(15000);
            l1.afficher();
            Console.ReadLine();
        }
    }
}
```