

Classes abstraites

Eleve.cs

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

namespace classAbstraites
{
    class Eleve : IComparable
    {
        private String _nom;
        private List<int> _listeNotes = new List<int>();
        private double _moyenne;

        public Eleve(String nom)
        {
            _nom = nom;
        }
        public double getMoyenne()
        {
            return _moyenne;
        }
        public String getNom()
        {
            return _nom;
        }
        public List<int> getListeNotes()
        {
            return _listeNotes;
        }
        /* Une note inferieure a 0 sera considere comme egale a 0 ;
        une note superieure a 20 sera considere comme egale a 20 */
        public void ajouterNote(int note)
        {
            if (note > 20)
            {
                note = 20;
            }
            if (note < 0)
            {
                note = 0;
            }
            this._listeNotes.Add(note);
            if (_listeNotes.Count>0)
            {
                foreach (int value in _listeNotes)
                {
                    this._moyenne += value;
                }
                this._moyenne = this._moyenne/_listeNotes.Count;
            }
        }
        public String toString()
        {
            return _nom + " (" + (int)(100 * _moyenne) / 100.0 + ")";
        }
    }
}
```

```

    }

    public int CompareTo(object obj)
    {
        Eleve e2 = (Eleve)obj;
        if (this._moyenne>e2._moyenne)
        {
            return 1;
        }
        else
        {
            return -1;
        }
    }
}
}

```

EleveECTS :

```

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

namespace classAbstraites
{
    abstract class EleveECTS : Eleve
    {
        public EleveECTS(string nom) : base(nom)
        {
        }
        public abstract int resultat();
    }
}

```

EleveMaster :

```

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

namespace classAbstraites
{
    class EleveMaster : EleveECTS
    {
        public EleveMaster(string nom) : base(nom)
        {
        }

        public override int resultat()
        {
            int total = 0;
            foreach (int note in getListeNotes())
            {
                if (note>=10)
                {
                    total += 6;
                }
                if (note<10&&note>=8)
            }
        }
    }
}

```

```

        {
            total += 3;
        }
    }
    return total;
}
}
}

```

Eleve1.cs :

```

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

namespace classAbstraites
{
    class Eleve1 : EleveECTS
    {
        public Eleve1(string nom) : base(nom)
        {
        }

        public override int resultat()
        {
            if (getMoyenne() <= 12)
            {
                return 60;
            }
            else
            {
                return 0;
            }
        }
    }
}

```

Program.cs :

```

using System;
using System.Collections.Generic;
using System.Linq;

namespace classAbstraites
{
    class Program
    {
        static void Main(string[] args)
        {
            Eleve e1 = new Eleve("Michel");
            Eleve e2 = new Eleve("Francis");
            Eleve e3 = new Eleve("Jean-eude");
            e1.ajouterNote(20);
            e3.ajouterNote(15);
            e2.ajouterNote(5);
            Console.WriteLine(e1.CompareTo(e3));
            Console.WriteLine(e3.CompareTo(e2));
            Console.WriteLine(e2.CompareTo(e1));
        }
    }
}

```

```

List<EleveECTS> listeEleve = new List<EleveECTS>();
EleveMaster em1 = new EleveMaster("Arthur");
Eleve1 eleve1 = new Eleve1("Julie");
EleveMaster em2 = new EleveMaster("Charlotte");
listeEleve.Add(em1);
listeEleve.Add(em2);
listeEleve.Add(eleve1);

em1.ajouterNote(15);em1.ajouterNote(7); em1.ajouterNote(17);
em1.ajouterNote(13);
em2.ajouterNote(3); em2.ajouterNote(5); em2.ajouterNote(20);
em2.ajouterNote(18);
eleve1.ajouterNote(5); eleve1.ajouterNote(13); eleve1.ajouterNote(8);

foreach (EleveECTS eleve in listeEleve)
{
    Console.WriteLine("\nNombre de credits de " + eleve.getNom() + " : " +
eleve.resultat()+" pour une moyenne de "+eleve.getMoyenne());
}
Console.WriteLine("Le meilleur élève est "+listeEleve.Max().getNom());

Console.ReadKey();

    }
}

```

Jeux d'essais :

```

1
1
-1

Nombre de credits de Arthur : 18 pour une moyenne de 17,7916666666667
Nombre de credits de Charlotte : 12 pour une moyenne de 14,2916666666667
Nombre de credits de Julie : 0 pour une moyenne de 12,5
Le meilleur élève est Arthur

```