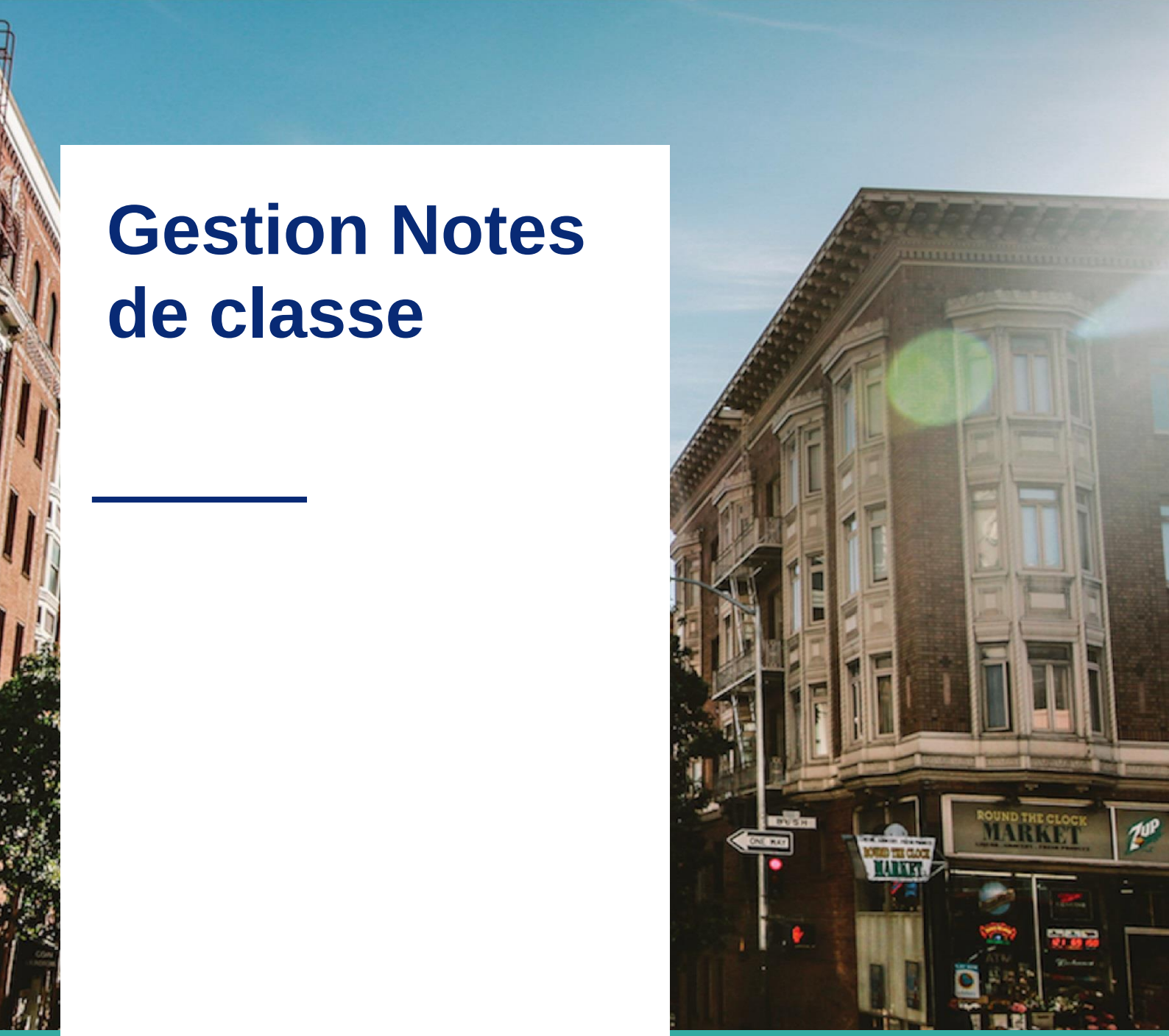
Gestion Notes de classe



2 MARS

ESP

Créé par : Serigne Modou Mbacke Fedior

1. Remplacer la classe Employé par une classe Étudiant

Commençons par examiner la classe employée qui se trouve dans le fichier employé.cs.

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

namespace UsageCollections
{
    public class Employé
    {
        public string Nom { get; set; }
        public string PréNom { get; set; }

        public string Matricule { get; set; }
    }
}
```

Ce code en C# définit une classe Employé dans le namespace UsageCollections, avec trois propri
automatiques : Nom, PréNom et Matricule, permettant de stocker les informations d'un employ
également plusieurs bibliothèques (System, System.Collections.Generic, System.Linq, System.Te
ne soient pas utilisées dans ce code. La classe est accessible publiquement et peut être utilisée p
manipuler des objets représentant des employés avec leurs informations personnelles.

Remplaçons la classe employée par la classe Etudiant :

Etudiant.cs

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

namespace UsageCollections
{
    public class Etudiant
    {
        public string Nom { get; set; }
        public string PréNom { get; set; }

        public string Matricule { get; set; }
    }
}
```

2. Utiliser les propriétés Numéro d'ordre (NO), Préno NoteCC et NoteDevoir.

Etudiant.cs

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

namespace UsageCollections
{
    public class Étudiant
    {
        public string NO { get; set; }
        public string Prénom { get; set; }
        public string Nom { get; set; }
        public int NoteCC { get; set; }
        public int NoteDevoir { get; set; }
    }
}
```

3. Lire les noms de la liste de classe au clavier. Pour étudiant saisir ses 2 notes. Ranger ces données dan SortedList, la clé est le NO.

Pour lire les noms des étudiants au clavier et enregistrer leurs notes dans une SortedLis par demander à l'utilisateur d'entrer le nombre d'étudiants. Ensuite, pour chaque étudian matricule, son nom et son prénom, puis on demande ses deux notes. On stocke ces info un objet Étudiant, en ajoutant les notes sous forme de tableau ou de liste. L'objet Étudian inséré dans une SortedList, où la clé est le matricule (NO) de l'étudiant. Enfin, on peut af informations de tous les étudiants enregistrés avec leurs notes en parcourant la SortedLi

Program.cs

```
using System;
using System.Collections;

namespace UsageCollections
{
    class Program
    {
        static void Main(string[] args)
        {
            SortedList lstÉtudiant = new SortedList();

            Console.Write("Entrez le nombre d'étudiants : ");
            int nombreÉtudiants = int.Parse(Console.ReadLine());

            for (int i = 0; i < nombreÉtudiants; i++)
            {
                Console.WriteLine($"\nSaisie de l'étudiant {i + 1}:");
                Console.Write("NO : ");
                string no = Console.ReadLine();

                Console.Write("Prénom : ");
                string prénom = Console.ReadLine();

                Console.Write("Nom : ");
                string nom = Console.ReadLine();

                Console.Write("NoteCC : ");
                int noteCC = int.Parse(Console.ReadLine());

                Console.Write("NoteDevoir : ");
                int noteDevoir = int.Parse(Console.ReadLine());
```

```
                Étudiant étudiant = new Étudiant
                {
                    NO = no,
                    Prénom = prénom,
                    Nom = nom,
                    NoteCC = noteCC,
                    NoteDevoir = noteDevoir
                };

                lstÉtudiant.Add(no, étudiant);
            }

            Console.WriteLine("\nAppuyez sur Entrée pour afficher la liste des étudiants...");
            Console.ReadLine();

            foreach (DictionaryEntry entry in lstÉtudiant)
            {
                Étudiant étudiant = (Étudiant)entry.Value;
                Console.WriteLine($"NO: {étudiant.NO}, Prénom: {étudiant.Prénom}, Nom: {étudian
{étudiant.NoteCC}, NoteDevoir: {étudiant.NoteDevoir}");
            }

            Console.ReadLine();
        }
    }
}
```

4. Afficher le NO, le nom, le prénom, les 2 notes et la ainsi que la moyenne de la classe, à la fin. La note d continu compte pour 33%.

Pour résoudre ce problème, nous devons afficher les informations des étudiants, leurs m
individuelles et la moyenne de la classe. La moyenne individuelle est calculée en pondér
contrôle continu (33%) et la note de devoir (67%). La moyenne de la classe est la moyer
moyennes individuelles.

Program.cs

```
using System;
using System.Collections;

namespace UsageCollections
{
    class Program
    {
        static void Main(string[] args)
        {
            SortedList lstÉtudiant = new SortedList();

            Console.Write("Entrez le nombre d'étudiants : ");
            int nombreÉtudiants = int.Parse(Console.ReadLine());

            for (int i = 0; i < nombreÉtudiants; i++)
            {
                Console.WriteLine($"\nSaisie de l'étudiant {i + 1}:");
                Console.Write("NO : ");
                string no = Console.ReadLine();

                Console.Write("Prénom : ");
                string prénom = Console.ReadLine();

                Console.Write("Nom : ");
                string nom = Console.ReadLine();

                Console.Write("NoteCC : ");
                int noteCC = int.Parse(Console.ReadLine());

                Console.Write("NoteDevoir : ");
                int noteDevoir = int.Parse(Console.ReadLine());

                Étudiant étudiant = new Étudiant
                {
                    NO = no,
                    Prénom = prénom,
                    Nom = nom,
                    NoteCC = noteCC,
                    NoteDevoir = noteDevoir
                };

                lstÉtudiant.Add(no, étudiant);
```

```
            }

            Console.WriteLine("\nAppuyez sur Entrée pour afficher la liste des étudiants...");
            Console.ReadLine();

            double totalClassAverage = 0.0;

            foreach (DictionaryEntry entry in lstÉtudiant)
            {
                Étudiant étudiant = (Étudiant)entry.Value;
                double moyenne = (étudiant.NoteCC * 0.33) + (étudiant.NoteDevoir * 0.67);
                totalClassAverage += moyenne;

                Console.WriteLine($"NO: {étudiant.NO}, Nom: {étudiant.Nom}, Prénom: {étudiant.P
{étudiant.NoteCC}, NoteDevoir: {étudiant.NoteDevoir}, Moyenne: {moyenne:F2}");
            }

            if (lstÉtudiant.Count > 0)
            {
                double moyenneClasse = totalClassAverage / lstÉtudiant.Count;
                Console.WriteLine($"\nMoyenne de la classe: {moyenneClasse:F2}");
            }
            else
            {
                Console.WriteLine("\nAucun étudiant dans la liste.");
            }

            Console.ReadLine();
        }
    }
}
```

5. Gérer la pagination. Le nombre de lignes par page par défaut, le maximum 15. Demander à l'utilisateur un nombre de lignes par page compris entre 1 et 15.

Pour implémenter la pagination avec un nombre de lignes personnalisé, il faudra modifie program.cs :

Program.cs

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

```
namespace UsageCollections
{
    class Program
    {
        static void Main(string[] args)
        {
            SortedList lstÉtudiant = new SortedList();

            Console.Write("Entrez le nombre d'étudiants : ");
            int nombreÉtudiants = int.Parse(Console.ReadLine());

            for (int i = 0; i < nombreÉtudiants; i++)
            {
                Console.WriteLine($"\nSaisie de l'étudiant {i + 1}:");
                Console.Write("NO : ");
                string no = Console.ReadLine();

                Console.Write("Prénom : ");
                string prénom = Console.ReadLine();

                Console.Write("Nom : ");
                string nom = Console.ReadLine();

                Console.Write("NoteCC : ");
                int noteCC = int.Parse(Console.ReadLine());

                Console.Write("NoteDevoir : ");
                int noteDevoir = int.Parse(Console.ReadLine());

                Étudiant étudiant = new Étudiant
                {
                    NO = no,
                    Prénom = prénom,
                    Nom = nom,
                    NoteCC = noteCC,
                    NoteDevoir = noteDevoir
                };

                lstÉtudiant.Add(no, étudiant);
            }

            // Conversion en liste triée et calcul des moyennes
            List<Étudiant> étudiants = new List<Étudiant>();
            List<double> moyennes = new List<double>();
            double totalMoyenne = 0;

            foreach (DictionaryEntry entry in lstÉtudiant)
            {
                Étudiant e = (Étudiant)entry.Value;
```

```
            double m = (e.NoteCC * 0.33) + (e.NoteDevoir * 0.67);
            étudiants.Add(e);
            moyennes.Add(m);
            totalMoyenne += m;
        }

        double moyenneClasse = étudiants.Count > 0 ? totalMoyenne / étudiants.Count : 0;

        // Configuration de la pagination
        int linesPerPage = 5;
        do
        {
            Console.Write("\nNombre de lignes par page [1-15] (défaut 5) : ");
            string input = Console.ReadLine();
            if (string.IsNullOrEmpty(input)) break;
            if (!int.TryParse(input, out linesPerPage) || linesPerPage < 1 || linesPerPage
            {
                Console.WriteLine("Valeur invalide !");
                linesPerPage = 5;
            }
            else break;
        } while (true);

        // Affichage paginé
        int page = 0;
        while (page * linesPerPage < étudiants.Count)
        {
            Console.Clear();
            Console.WriteLine($"=== Page {page + 1} ===\n");
            Console.WriteLine("NO".PadRight(10) + "Nom".PadRight(15) + "Prénom".PadRight(15
                              "CC".PadRight(5) + "Devoir".PadRight(7) + "Moyenne");
            Console.WriteLine(new string('-', 60));

            for (int i = page * linesPerPage; i < (page + 1) * linesPerPage && i < étudiant
            {
                Étudiant e = étudiants[i];
                Console.WriteLine(
                    $"{e.NO.PadRight(10)}{e.Nom.PadRight(15)}{e.Prénom.PadRight(15)}" +
                    $"{e.NoteCC.ToString().PadRight(5)}{e.NoteDevoir.ToString().PadRight(7)
                );
            }

            page++;
            if (page * linesPerPage < étudiants.Count)
            {
                Console.WriteLine("\nAppuyez sur Entrée pour la suite...");
                Console.ReadLine();
            }
        }
```

```
            Console.WriteLine($"\nMoyenne générale de la classe : {moyenneClasse:F2}");
            Console.ReadLine();
        }
    }
}
```

6. Proposer une option pour sortir du programme.

Voici la solution finale avec gestion de la pagination et option de sortie :

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

namespace UsageCollections
{
    class Program
    {
        static void Main(string[] args)
        {
            SortedList lstÉtudiant = new SortedList();

            Console.Write("Entrez le nombre d'étudiants : ");
            int nombreÉtudiants = int.Parse(Console.ReadLine());

            for (int i = 0; i < nombreÉtudiants; i++)
            {
                Console.WriteLine($"\nSaisie de l'étudiant {i + 1}:");
                Console.Write("NO : ");
                string no = Console.ReadLine();

                Console.Write("Prénom : ");
                string prénom = Console.ReadLine();

                Console.Write("Nom : ");
                string nom = Console.ReadLine();

                Console.Write("NoteCC : ");
                int noteCC = int.Parse(Console.ReadLine());

                Console.Write("NoteDevoir : ");
                int noteDevoir = int.Parse(Console.ReadLine());

                Étudiant étudiant = new Étudiant
                {
```

```
            NO = no,
            Prénom = prénom,
            Nom = nom,
            NoteCC = noteCC,
            NoteDevoir = noteDevoir
        };

        lstÉtudiant.Add(no, étudiant);
    }

    List<Étudiant> étudiants = new List<Étudiant>();
    List<double> moyennes = new List<double>();
    double totalMoyenne = 0;

    foreach (DictionaryEntry entry in lstÉtudiant)
    {
        Étudiant e = (Étudiant)entry.Value;
        double m = (e.NoteCC * 0.33) + (e.NoteDevoir * 0.67);
        étudiants.Add(e);
        moyennes.Add(m);
        totalMoyenne += m;
    }

    double moyenneClasse = étudiants.Count > 0 ? totalMoyenne / étudiants.Count : 0;

    int linesPerPage = 5;
    do
    {
        Console.Write("\nNombre de lignes par page [1-15] (défaut 5) : ");
        string input = Console.ReadLine();
        if (string.IsNullOrEmpty(input)) break;
        if (!int.TryParse(input, out linesPerPage) || linesPerPage < 1 || linesPerPage
        {
            Console.WriteLine("Valeur invalide !");
            linesPerPage = 5;
        }
        else break;
    } while (true);

    int page = 0;
    bool exit = false;

    while (page * linesPerPage < étudiants.Count && !exit)
    {
        Console.Clear();
        Console.WriteLine($"=== Page {page + 1} ===\n");
        Console.WriteLine("NO".PadRight(10) + "Nom".PadRight(15) + "Prénom".PadRight(15
                          "CC".PadRight(5) + "Devoir".PadRight(7) + "Moyenne");
        Console.WriteLine(new string('-', 60));
```

```
                for (int i = page * linesPerPage; i < (page + 1) * linesPerPage && i < étudiant
                {
                    Étudiant e = étudiants[i];
                    Console.WriteLine(
                        $"{e.NO.PadRight(10)}{e.Nom.PadRight(15)}{e.Prénom.PadRight(15)}" +
                        $"{e.NoteCC.ToString().PadRight(5)}{e.NoteDevoir.ToString().PadRight(7)
                    );
                }

                page++;
                if (page * linesPerPage < étudiants.Count)
                {
                    Console.WriteLine("\nAppuyez sur :");
                    Console.WriteLine("[Entrée] - Page suivante");
                    Console.WriteLine("Q - Quitter");
                    var key = Console.ReadKey();

                    if (key.KeyChar.ToString().ToUpper() == "Q")
                    {
                        exit = true;
                    }
                }
            }

            Console.WriteLine($"\nMoyenne générale de la classe : {moyenneClasse:F2}");
            Console.WriteLine("\nAppuyez sur n'importe quelle touche pour quitter...");
            Console.ReadKey();
        }
    }
}
```