TP CINE-SALLE

Travail à faire	
1	Démarrez Visual Studio Ouvrir la solution « **TpCinema** » Depuis l'explorateur de solution, ouvrir les documentations des API utilisées par l'application et présentent dans les répertoires : *documentationApiCinema, documentationApiPersistance* Il convient d'ouvrir les fichiers *index.html* dans votre navigateur web. Répondez aux questions ci-dessous

1 Documentation *ApiCinema*

a. Classe *Cinema*

Quelle(s) information(s) est(sont) nécessaire(s) pour initialiser un objet de type *Cinema* ?

Le nom du cinéma est nécessaire et doit être transmis en paramètre

Le nom du cinéma est-il modifiable une fois l'objet créé ? justifiez.

Oui, une propriété permet de lire et de modifier le nom du cinéma

Quel type de variable retourne la méthode *GetLaPlusGrandeSalle* ?

La méthode retourne une variable objet de type Salle

Combien de paramètres sont nécessaires pour ajouter une salle de cinéma via la méthode *AjouterSalle* ?

La méthode prend trois paramètres

Quelle est la particularité (technique) de la collection de salle d'un objet de type *Cinema* ?

La collection est en lecture seule. Il n'est pas possible de modifier la liste au travers de la

Que fait la méthode *GetCapaciteTotalDuCinema* ?

Elle retourne le nombre de place dont dispose le cinéma

b. Classe *Salle*

Combien de paramètres sont nécessaires pour créer un objet de type *Salle* ?

Trois paramètres sont nécessaires pour créer un objet de type Salle

Quels types de paramètres sont nécessaires pour créer un objet de type Salle ?

Une chaine de caractères et deux entiers

2 Documentation *ApiPersistance*

a. Classe *Persistance*

Quel type de valeur retourne la méthode *ChargerDonnees* ?

Elle retourne une valeur de type ILIST

Combien de paramètres sont attendus par la méthode *SauvegarderDonnees* ?

Deux paramètres sont attendus

Travail à faire	
2	En vous appuyant sur la documentation *ApiPersistance*, complétez le « *case 0* » de la méthode *Main* de la classe *Programs*.

```
Persistance.SauvegarderObjet("Cinema.save", unCinema);
```

Travail à faire	
3	En vous appuyant sur la documentation *ApiCinema*, complétez les autres « *case* » de la méthode *Main* de la classe *Programs*.

```
Case 1

foreach (Salle s in unCinema.LesSallesDuCinema)
{
  Console.WriteLine(s.NomDeLaSalle);
}

Case 2

Console.WriteLine("Saisir le nom de la salle ");
nomSalle = Console.ReadLine();
Console.WriteLine("Saisir le numéro de la salle ");
numeroDeSalle = SaisirEntier();
Console.WriteLine("Saisir la capacité de la salle");
capactiteSalle = SaisirEntier();
unCinema.AjouterSalle(nomSalle, numeroDeSalle, capactiteSalle);

Case 3

Console.WriteLine("Saisir le numero de la salle ");
numeroDeSalle = SaisirEntier();
bool suppression = unCinema.SupprimerSalle(numeroDeSalle);
if (!suppression)
{
  Console.WriteLine("Ce numéro de salle ne semble pas référencer une salle du
cinéma");
}
```

Case 4

```
Salle uneSalle = unCinema.GetSalleLaPlusGrande();
Console.WriteLine("Nom : {0} - Numero : {1} - Capacité : {2}", uneSalle.NomDeLaSalle,
uneSalle.NumSalle, uneSalle.Capacite);
```

Case 5

```
Console.WriteLine("{0} places",unCinema.GetCapaciteTotalDuCinema());
```

Travail à faire	
4	En vous appuyant sur la documentation *ApiCinema*, écrire le code source des méthodes de classes ci-dessous : - MemeSalle de la classe Salle - GetCapaciteTotalDuCinema de la classe Cinema - GetSalle de la classe Cinema - GetSalleLaPlusGrande de la classe Cinema - SupprimerSalle de la classe Cinema

```
public bool MemeSalle(Salle uneSalle)
{
 return this.NumSalle == uneSalle.NumSalle && this._nomDeLaSalle ==
uneSalle.NomDeLaSalle;
}

public int GetCapaciteTotalDuCinema()
{
  int capaciteTotalDuCinema = 0;
  foreach (Salle salle in _lesSallesDuCinema)
  {
   capaciteTotalDuCinema += salle.Capacite;
  }
  return capaciteTotalDuCinema;
}

public Salle GetSalle(int numSalle)
{
   return _lesSallesDuCinema.Find(s => s.NumSalle == numSalle);
}

public bool SupprimerSalle(int numSalle)
{
       bool supprimer = false;
       Salle uneSalle = GetSalle(numSalle);
       if(uneSalle!=null)
       {
              supprimer = _lesSallesDuCinema.Remove(uneSalle);
       }
       return supprimer;
}
```

```
public Salle GetSalleLaPlusGrande()
{
	Salle salleLaPlusGrande = null;
	if (_lesSallesDuCinema.Count > 0)
	{
		int maxCapacite = _lesSallesDuCinema[0].Capacite;
		salleLaPlusGrande = _lesSallesDuCinema[0];
		foreach (Salle salle in _lesSallesDuCinema)
		{
			if (salle.Capacite > maxCapacite)
			{
				maxCapacite = salle.Capacite;
				salleLaPlusGrande = salle;
			}
		}
	}
	return salleLaPlusGrande;
}
```