

ACTIVITE POO

Programmation Orientée Objet (Activité :Programmer une application informatique)			
ECA	Durée :	Date dernière MAJ	19/07/2021
Nom			
Contexte technique :			
Visual Studio.net Langage C#			
Documents autorisés :			
Tous <input checked="" type="checkbox"/> Aucun <input type="checkbox"/> Liste jointe <input type="checkbox"/>			
Autres moyens nécessaires:			
Matériel <input type="checkbox"/> PC Logiciel <input type="checkbox"/> Autre <input type="checkbox"/>			
Scénario :			
..			
Tâches à confirmer :			
Dérivation d'une classe existante.....			
Redéfinition de méthodes.....			
Constructeurs.....			
Encapsulation.....			
Collection.....			
Classe de test.....			

Description du contexte (organisationnel, technique, ...)

La fédération européenne de sports veut enregistrer les informations concernant les sports dont elle souhaite suivre les records. Certaines informations sont très générales :

- ✓ le nom du sport
- ✓ le type de sport : athlétisme, sport cycliste, sport automobile, sport de combat, ...

Par exemple : **100 m brasse dames** est le nom du sport
Natation est le type de sport

A chaque fois qu'un record tombe pour un sport individuel, on conserve :

- ✓ le nom du champion ou de la championne,,
- ✓ le record proprement dit,
- ✓ le lieu.
- ✓ la date

Par exemple **Emma Igelstrom** (nom de la championne) a battu le record du monde du 100m brasse dames en petit bassin en **1 min 05 sec 11/100** (le record proprement dit) en finale des Championnats de Suède de natation à **Stockholm** (le lieu) le **10 avril 2003** (la date).

Concernant les sports d'équipe, on conserve ;

- ✓ le nom de l'équipe championne,
- ✓ le nombre de personnes dans l'équipe.

Par exemple : en basket féminin , c'est l'équipe de **Valenciennes** (nom de l'équipe championne) : l'équipe sur le terrain comportant **5** joueuses (le nombre de personnes dans l'équipe).

Description du problème à traiter

Il vous est demandé de créer une « **assembly dll** » nommée : « **sports.dll** » réalisant les classes :

- ✓ **Sport** (nom du fichier : **Sport.cs**) : qui comporte les informations générales,
 - ✓ **SportIndiv** (nom du fichier : **SportIndiv.cs**) : qui traite les informations d'un sport individuel,
 - ✓ **SportEquip** (nom du fichier : **SportEquip.cs**) : qui traite les informations d'un sport collectif.
- dans un espace de noms nommé "**Sports**" selon l'analyse fonctionnelle ci-dessous :

1) A partir du squelette de classe "**Sport**" fournie en annexe :

1. On doit pouvoir construire un objet
 - ✓ en ne connaissant que le nom du sport
 - ✓ le type de sport peut-être indiqué ultérieurement.
2. L'encapsulation doit être mise en œuvre.
3. la méthode ToString() doit renvoyer :
 - ✓ *le nom du sport est*
 - ✓ *suivi du nom du sport*
 - ✓ *suivi de ce sport a pour type*
 - ✓ *suivi du contenu du type de sport.*

2) A partir du squelette de classe "**SportEquip**" fournie en annexe :

1. Cette classe doit héritée de la classe Sport.
2. On doit pouvoir construire un objet SportEquip :
 - ✓ en lui communiquant
 - le nom du sport,
 - le nom de l'équipe championne
 - et le nombre de personnes par équipe.

CRM	Auteur	Activité	Compétence	Version	Date MAJ	Page 2/10
	AR	Programmer	POO	1	19/07/2021	

- ✓ ou en ne lui communiquant rien : dans ce cas les variables prennent des valeurs par défaut
 - Nom du sport : "Beach Volley"
 - Nom du champion : "Brésil"
 - Nombre de personnes : 4
- 3. L'encapsulation doit être mise en œuvre
- 4. la méthode ToString() doit renvoyer
 - ✓ le nom du sport est
 - ✓ suivi du nom du sport
 - ✓ suivi de ce sport a pour type
 - ✓ suivi du contenu du type de sport.
- ainsi que
 - ✓ le nom des champions est suivi du nom de l'équipe "championne"
 - ✓ le nombre de joueurs est de suivi du nombre de personnes dans l'équipe.

3) A partir du squelette de classe **"SportIndiv"** fournie en annexe :

- 5. Cette classe doit héritée de la classe Sport.
- 6. On doit pouvoir construire un objet SportIndiv :
 - ✓ en lui communiquant
 - le nom du sport,
 - le nom du champion
 - le record
 - le lieu et la date où est le record a été battu.
- 7. la méthode ToString() doit renvoyer
 - ✓ le nom du sport est
 - ✓ suivi du nom du sport
 - ✓ suivi de ce sport a pour type
 - ✓ suivi du contenu du type de sport.
- ainsi que
 - ✓ le nom du champion est suivi du nom du champion
 - ✓ son record est de suivi du record .
 - ✓ le record date de suivi de la date du record
 - ✓ à suivi du lieu du record.

4) Créer une méthode **gagne()** qui renseigne :

Pour un sport individuel :

Le nom du champion

Le record

Pour un sport d'équipe

Le nom de l'équipe championne.

CRM	Auteur	Activité	Compétence	Version	Date MAJ	Page 3/10
	AR	Programmer	POO	1	19/07/2021	

Description du problème à traiter (suite)

II) Il vous est demandé de créer une « **assembly exe en mode Console** » nommée :

« **TestAppliSport.exe** » réalisant la classe de Test de l'applatif:

1) Complétez la classe "**TestAppliSport**" (nom du fichier : **TestAppliSports.cs**) fournie en annexe pour tester les classes "**Sport**", "**SportIndiv**" et "**SportEquip**". Les opérations à mener sont signalées par les commentaires, les objets seront construits à partir de données saisies au clavier .

2) Une des opérations demandées pose problème expliquez pourquoi par des commentaires dans le source de la classe

III) Il vous est demandé de créer une « **assembly exe en mode Console** » nommée :

« **TestCollectionSport.exe** » réalisant une collection de sports:

Complétez ou modifiez la classe "**TestCollectionSport**" (nom du fichier : **TestCollectionSport.cs**) fournie en annexe:

- ✓ pour y alimenter un objet collection de "SportIndiv" au fur et à mesure de la saisie.
- ✓ Pour afficher la liste des "SportIndiv" lorsque l'utilisateur ne souhaite plus enregistrer un nouveau record. Pour ce faire, vous devrez utiliser la méthode ToString() déjà réalisée

Production attendue

I) Réalisez ce travail en créant « **une Solution sous VS.net** » formée de 3 projets : « **Sports** », « **TestAppliSports** », « **TestCollectionSports** »

Conseil : vérifiez qu'une structure arborescente de la sorte est créée : AppSports contenant 3 dossiers : Sports, TestAppliSports, TestCollectionSports

II) Fournissez sur papier les **sources des 3 « assembly »** réalisées

CRM	Auteur	Activité	Compétence	Version	Date MAJ	Page 4/10
	AR	Programmer	POO	1	19/07/2021	

Annexe N°1 : La super classe Sport

```
/**
 * Titre : Sports
 * Description : Ebauche de Sport.cs
 */
using System ;

namespace Sports
{

public class Sport
{
    private string strNomSport;
    protected string strTypeSport;

    public Sports()
    {
        // TODO : implement code
    }

    public override string ToString()
    {
        // TODO : implement code
    }

}
}
```

CRM	Auteur	Activité	Compétence	Version	Date MAJ	Page 5/10
	AR	Programmer	POO	1	19/07/2021	

Annexe N°2 : la sous classe SportIndiv

```

/*****
* Ebauche de: SportIndiv.cs
*****/

using System ;

namespace Sports
{

public class SportIndiv
{
    private string strChampion; // nom du champion
    private string strRecord; // record
    private string strLieu; // lieu du record
    private DateTime datRecord; // date du record (structure de type DateTime)

    public SportIndiv(....., string dateR)
    { .....
        datRecord=DateTime.Parse(dateR) ; // date « chaine (jj/mm/aaaa) ou (jj moisenlettre aaaa)» au format
                                           // DateTime
    }

    public string StrChampion
    {
        set { strChampion=value;}
        get { return strChampion;}
    }

    public string StrRecord
    {
        set { strRecord =value;}
        get { return strRecord;}
    }

    public string StrLieu
    {
        set { strLieu =value;}
        get { return strLieu;}
    }

    public DateTime DatRecord
    {
        set { datRecord =value;}
        get { return datRecord;}
    }

    // la méthode toString
    public override string ToString()
    {
        // TODO implement code
    }
}
}

```

CRM	Auteur	Activité	Compétence	Version	Date MAJ	Page 6/10
	AR	Programmer	POO	1	19/07/2021	

Annexe N°3 : la sous classe SportEquip

```

/*****
* Ebauche de: SportEquip.cs
*****/
using System ;

namespace Sports
{

public class SportEquipe
{
    private string strEquipChamp; // nom de l'équipe championne
    private int  nbPers;          // nombre de personne dans l'équipe

    public SportEquipe()
    {
    }

    // la méthode ToString
    public override string ToString()
    {
    }
}
}

```

CRM	Auteur	Activité	Compétence	Version	Date MAJ	Page 7/10
	AR	Programmer	POO	1	19/07/2021	

Annexe N°3 : classe TestAppliSport

* Fichier de test de l'applicatif : TestAppliSport.cs

*****/

```
using System ;
using Sports ;

namespace TestAppliSports
{
    public class TestAppliSport
    {
        public static void main(String[] args)
        {
            /* Créer une instance de Sport ayant pour nom : sport1
               avec comme nom du sport : Saut en longueur */

            /* Indiquer que ce sport fait partie de l'athlétisme */

            /* Créer seq1 qui est un sport d'équipe le polo à bicyclette
               qui a pour équipe championne Chambéry
               et un nombre de personnes de 5 par équipe */

            /* Créer seq2 qui est le sport d'équipe par défaut */

            /* Afficher les caractéristiques de sport1 */

            /* Afficher les caractéristiques de seq1 */

            /* Afficher les caractéristiques de seq2 */

            /* Créer un sport individuel sind1 "saut à la perche"
               dont le champion est "Galfione" avec un record de 6 mètres
               le 25 août 1996 à Séville
               Traiter l'exception (avec try et catch): tant que le format de date est incorrect, obliger la reconstruction
               de l'objet sind1
            */

            /* Afficher les caractéristiques de l'instance ainsi créée */

            /* Puis enregistrer sur sind1 le fait que "BUBKA" gagne
               le record avec 6,15mètres */

            /* Changer par la méthode gagne l'équipe championne de seq1
               en "Quimper" */

            /* Changer par la méthode gagne l'équipe championne de sport1
               en "Paris" */
        }
    }
}
```

CRM	Auteur	Activité	Compétence	Version	Date MAJ	Page 8/10
	AR	Programmer	POO	1	19/07/2021	

Annexe N°4 : classe TestCollectionSport

* Fichier de collection de sports : TestCollectionSport.cs

*****/

```
using System ;
using System.Collections ;
using Sports ;
```

```
namespace TestCollectionSports
{
```

```
/**
 * Titre : enregistrement des records et affichage de la liste
 * des records enregistrés en fin de saisie
 */
```

```
public class TestCollectionSport
{
```

```
    public static string lire()
    {
        string strSaisie = null;
        strSaisie=Console.ReadLine();
        return strSaisie;
    }
```

```
    public static void main(String[] args)
    {
```

```
        string strSport;
        string strNom;
        string strRecord;
        string strDate;
        string strLieu;
```

```
        Console.Write("Souhaitez-vous enregistrer un nouveau champion ? (oui/non)");
        string strAutreChampion = lire();
        while (strAutreChampion.equals("oui"))
        {
```

```
            Console.Write("nom du sport :");
            strSport = lire();
            Console.Write("nom du champion :");
            strNom = lire();
            Console.Write("nouveau record :");
            strRecord = lire();
            Console.Write("date du record :");
            strDate = lire();
            Console.Write("le lieu où est le record a été battu :");
            strLieu = lire();
```

```
            Console.Write("Souhaitez-vous enregistrer un nouveau record ? (oui/non)");
            strAutreChampion = lire();
```

```
        }
        /*      affichage des résultats      */
        for () ou while()
        {
```

```
            Console.Write("????????????????????????????????????");
```

```
        }
    }
```

CRM	Auteur	Activité	Compétence	Version	Date MAJ	Page 9/10
	AR	Programmer	POO	1	19/07/2021	

}

CRM	Auteur	Activité	Compétence	Version	Date MAJ	Page 10/10
	AR	Programmer	POO	1	19/07/2021	