

Nom: \_\_\_\_\_ ( \_\_\_\_\_ /100)

## POO—Examen final

Votre examen final pour le cours de POO présume que vous avez pris une part active à la réalisation des travaux pratiques qui vous ont été proposés durant la session. Cet examen est récapitulatif et touche l'ensemble de la session.

### Consignes

Les consignes pour cet examen sont les suivantes :

- l'examen est une activité d'évaluation *individuelle* qui doit être exécutée seul(e);
- le droit aux notes de cours et autre matériel écrit *personnel* est accordé;
- la transmission d'information pendant l'examen, que ce soit par le biais d'avions de papier, de signaux de fumée, du code morse ou par quelque autre moyen que ce soit est formellement interdite et sera sanctionnée si elle devait hélas se produire;
- la calculatrice est bannie, de même que le téléphone cellulaire, l'ordinateur portatif, le téléavertisseur, le *walkie talkie* et tout autre objet d'une *classe* apparentée;
- ayez je vous prie la courtoisie de mettre hors d'état de nuire votre téléphone cellulaire ou téléavertisseur sauf si dans une vie parallèle vous êtes un super-héros auquel on peut faire appel à tout moment pour sauver l'humanité;
- si vous bloquez sur une question, passez à la suivante; vous y reviendrez plus tard, s'il vous reste du temps;
- les questions n'ont pas toutes la même pondération; la dernière est celle qui compte le plus. Pensez à investir stratégiquement votre temps et vos efforts;
- sauf pour la dernière, la formulation des questions a tendance à être plus longue que les réponses que vous devez fournir;
- si la brume envahit votre cerveau, des exercices de respiration font toujours le plus grand bien; évitez de respirer en regardant à gauche et à droite la copie de vos voisins, le professeur pourrait y voir une intention malveillante et sentir le besoin de prendre des décisions regrettables;
- répondez directement sur le document, aux endroits prévus à cet effet.

Durée limite: ..... 150 minutes

**Bon travail !**

**Q1 Des questions simples (enfin, selon moi ...) ( \_\_\_\_/15)**

5 pts

1.1 Soit la classe suivante :

```
// Fichier Quelconque.cs
class Quelconque
{
    int valeur_;
    public int Valeur { get; private set; }

    Quelconque(int valeur)
    {
        Valeur = valeur;
    }
}
```

Ce programme client désire instancier et utiliser les services de cette classe :

```
class Program
{
    static void Main(string[] args)
    {
        Quelconque q = new Quelconque(90);

        Console.WriteLine("On a conservé dans l'instance la
                           valeur {0}", q.Valeur);
    }
}
```

Ce programme ne compile pas et donne l'erreur :

Liste d'erreurs			
1 erreur 0 avertissements 0 messages			
	Description	Fichier	Ligne
1	'_15_FinalQ1a.Quelconque' ne contient pas un constructeur qui accepte des arguments 1	Program.cs	9
Liste d'erreurs Sortie			

Quelle est la source du problème ?

5 pts

1.2 Soit la classe de base suivante :

```
namespace _15_FinalQ1b
{
    abstract class DeBase
    {
        int valeur_;

        protected int Valeur
        {
            get
            {
                return valeur_;
            }
            set
            {
                valeur_ = value;
            }
        }

        public DeBase()
        {
        }

        public abstract void Afficher();
    }
}
```

Une classe en dérive :

```
namespace _15_FinalQ1b
{
    class Derivée : DeBase
    {
        public Derivée()
        {
        }

        // Le new indique que cette propriété masque volontairement
        // la propriété de même nom dans la classe de base
        new public int Valeur
        {
            get
            {
                return base.Valeur;
            }
            set
            {
                base.Valeur = value;
            }
        }
    }
}
```

Le programme suivant cherche à instancier cette classe et utiliser l'objet :

```
using System;

namespace _15_FinalQ1b
{
    class Program
    {
        static void Main(string[] args)
        {
            Dérivée objet = new Dérivée();

            objet.Valeur = 517;

            Console.WriteLine(objet.Valeur);
        }
    }
}
```

Pourtant, le compilateur indique une erreur à l'instanciation de la classe **Dérivée**, la ligne que j'ai mise en caractères gras. Ce programme ne peut être compilé. Pourquoi ?

5 pts

- 1.3 À la question précédente, en supposant que le nécessaire a été fait pour que la classe **Dérivée** puisse être instanciée, aurait-il été possible de faire **objet.Valeur** dans le programme client sans implanter la propriété **Valeur** dans la classe **Dérivée** ? Répondez par oui ou par non et justifiez votre réponse.

Q2

Une question de base

( \_\_\_\_/20)

Nous désirons créer une classe **Récipient** qui servira de classe de base à plusieurs classes dérivées (Tasse, Verre, Pichet, Cruche, etc.).

Nous élaborons la classe suivante :

```
1  class Récipient
2  {
3      // Aucun récipient ne peut avoir une capacité de moins
4      // qu'un certain volume minimal ici fixé à 5 millilitres
5      const int VOLUME_MIN = 5;
6
7      // un récipient contient à tout moment un certain
8      // volume de liquide indiqué en millilitres.
9      int nbMl_;
10
11     // volume maximal du récipient instancié. Ce volume n'est pas constant
12     // puisqu'il peut changer selon le récipient; il sera donc fixé au
13     // moment de la construction.
14     int volumeMax_;
15
16     public int NbMl
17     {
18         get    { return nbMl_; }
19         protected set
20         {
21             nbMl_ = value;
22         }
23     }
24
25     protected int VolumeMax
26     {
27         get    { return volumeMax_; }
28         private set
29         {
30             volumeMax_ = value;
31         }
32     }
33
34     protected Récipient(int volumeMax, int nbMl = 0)
35     {
36         VolumeMax = volumeMax;
37         NbMl = nbMl;
38     }
39 }
```

5 pts

- 2.1 La propriété **VolumeMax** ne fait aucune validation de la valeur qui est mise dans l'attribut associé. Corrigez cette lacune et, s'il y a une situation qui mérite l'attention du programme client utilisant la classe, envoyez une exception de type **VolumeInvalideException** dont vous aurez préalablement indiqué la déclaration dans votre réponse.

5 pts

- 2.2 La propriété **NbMI** fait aussi un très mauvais travail de validation dans son mutateur. Corrigez la situation en utilisant au besoin le nouveau type **VolumeInvalideException** que vous avez créé.

5 pts

- 2.3 Nous désirons aussi qu'il soit possible de vérifier si une instance de récipient est égale à une autre instance de la même classe. Deux instances sont égales si elles ont le même **volumeMax** et le même nombre de **nbMI**. Écrivez tout le code qui est requis pour implanter correctement cet opérateur.

- 2.4 Est-il possible d'instancier la classe **Récipient**? Répondez par *oui* ou par *non* et justifiez brièvement votre réponse.

5 pts



**Q3****Une question qui dérive****( \_\_\_\_/20)**

Nous désirons maintenant créer une classe dérivée de la classe **Récipient** de la question précédente de cet examen. En voici une déclaration incomplète :

```
class Tasse : Récipient
{
    // volume max de ce type de récipient qui sera transmis
    // au constructeur du récipient pour y être conservé
    // à l'instanciation.
    const int VOLUME_MAX = 500;

    // constructeur par défaut à écrire...

    // constructeur paramétrique à écrire...

    public int Verser(int volume)
    {
        // méthode à écrire...
    }

    public int Ajouter(int volume)
    {
        // méthode à écrire...
    }
}
```

5 pts

- 3.1 Écrivez le constructeur par défaut de cette classe **Tasse** qui veille à initialiser son contenu à 0. N'oubliez pas que le volume maximal d'une instance de **Tasse** est conservé dans l'attribut de la classe **Récipient** prévu à cet effet.

5 pts

- 3.2 Écrivez le constructeur paramétrique de la classe **Tasse** qui reçoit pour seul paramètre du programme client le nombre de millilitres contenus dans l'instance; le nombre maximal de millilitres de la tasse est fixé par la constante et n'est pas un paramètre du constructeur de la **Tasse**.

5 pts

- 3.3 Écrivez le code de la méthode **Verser** qui retire de la tasse un certain nombre de millilitres reçu en paramètre et retourne la nouvelle quantité présente dans cette instance de la tasse une fois l'opération complétée.

5 pts

- 3.4 Nous désirons qu'il soit possible d'additionner le volume d'une instance de la classe **Tasse** au volume d'une autre instance de la classe **Tasse**; la nouvelle tasse ainsi créée aura pour **NbMI** la somme des deux quantités; si la somme dépasse le volume maximal, votre modification en 2.2 se chargera de faire le nécessaire. Écrivez la surcharge d'opérateur qui permettra d'implanter cette opération

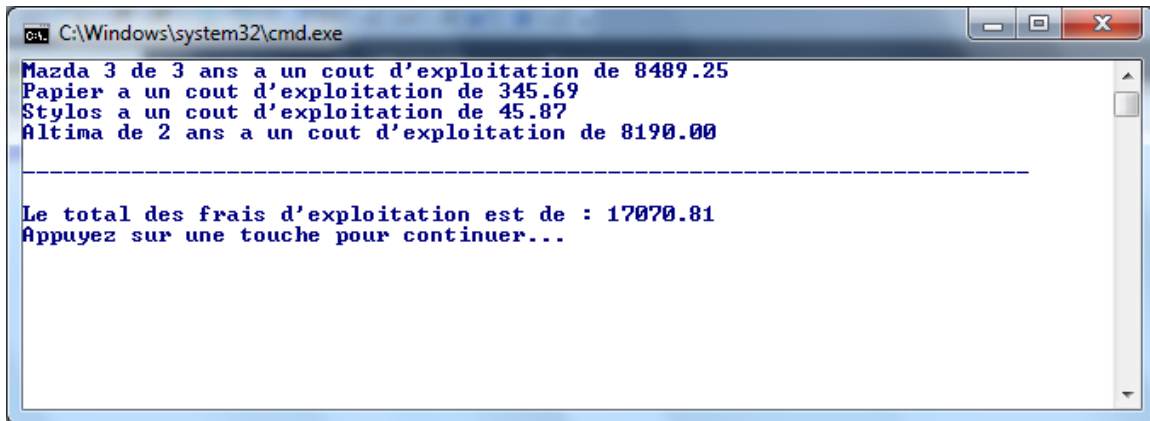
**Q4****Une question de coûts****( \_\_\_\_/20)**

Soit le code suivant qui tire profit de l'héritage d'interface :

```
1  //----- interface ICoutEvaluable -----
2  interface ICoutEvaluable
3  {
4      double CalculerCout();
5  }
6
7
8  //----- classe Automobile -----
9  class Automobile : ICoutEvaluable
10 {
11     // par souci de simplification, toutes les propriétés sont
12     // automatiques; il faudrait normalement implanter mieux les
13     // mutateurs, bien entendu.
14     //-----
15     string Nom    { get; set; }
16     int NbAnnées { get; set; }
17     double Prix   { get; set; }
18     float PourcentageDépréciation { get; set; }
19
20     public double CalculerCout()
21     {
22         double prixDéprécié = Prix;
23         int compteur = 0;
24
25         while (compteur < NbAnnées)
26         {
27             prixDéprécié = prixDéprécié * (1 - PourcentageDépréciation);
28             ++compteur;
29         }
30         return Prix - prixDéprécié;
31     }
32
33     public Automobile(string nom, int nbAnnées, double prix, float taux)
34     {
35         Nom = nom;
36         NbAnnées = nbAnnées;
37         Prix = prix;
38         PourcentageDépréciation = taux;
39     }
40
41     public override string ToString()
42     {
43         return Nom + " de " + NbAnnées.ToString() + " ans";
44     }
45 }
46
47
```

```
48 //----- classe FournitureDeBureau -----
49 class FournitureDeBureau : ICoutEvaluable
50 {
51     string Nom { get; set; }
52     double Prix { get; set; }
53
54     public FournitureDeBureau(string nom, double prix)
55     {
56         Nom = nom;
57         Prix = prix;
58     }
59
60     public double CalculerCout()
61     {
62         return Prix;
63     }
64
65     public override string ToString()
66     {
67         return Nom;
68     }
69 }
70
71
72 //----- program.cs -----
73 class Program
74 {
75     static void Main(string[] args)
76     {
77
78         // ... déclaration de la référence fraisExploitation
79         //      et instantiation requises
80
81
82         fraisExploitation.Add(new Automobile("Mazda 3", 3, 22000, 0.15f));
83         fraisExploitation.Add(new FournitureDeBureau("Papier", 345.69));
84         fraisExploitation.Add(new FournitureDeBureau("Stylos", 45.87));
85         fraisExploitation.Add(new Automobile("Altima", 2, 25000, 0.18f));
86
87         AfficherFrais(fraisExploitation);
88
89         Console.WriteLine();
90         Console.WriteLine(new string('-', 74));
91         Console.WriteLine();
92
93         Console.WriteLine("Le total des frais d'exploitation est de : {0:f2}",
94                           CalculerFrais(fraisExploitation));
95     }
96 }
```

Ce programme, une fois complété, produira la sortie suivante :



```
C:\Windows\system32\cmd.exe
Mazda 3 de 3 ans a un cout d'exploitation de 8489.25
Papier a un cout d'exploitation de 345.69
Stylos a un cout d'exploitation de 45.87
Altima de 2 ans a un cout d'exploitation de 8190.00

-----
Le total des frais d'exploitation est de : 17070.81
Appuyez sur une touche pour continuer...
```

- 4.1 Écrivez la déclaration complète de l'objet permettant de conserver tous nos frais d'exploitation. La référence de cet objet se nommera **fraisExploitation** et vous devez vous assurer de son instantiation lors de la déclaration.

4 pts

8 pts

- 4.2 Écrivez le code complet de la méthode **AfficherFrais(...)** appelée en ligne 87 du programme client qui permet d'afficher la première partie de ma sortie illustrée ci-haut, c'est-à-dire le coût de chaque élément conservé dans la liste.

- 4.3 Écrivez le code complet de la méthode **CalculerFrais(...)** appelée en ligne 94 du programme client qui permet d'évaluer le total des coûts de tout ce qui se trouve dans la liste.

8 pts



Q5

Ce qui se conçoit bien...

( \_\_\_\_\_ /25)

La célèbre compagnie électronique *Dinosaurus electronics*, celle qui file vers l'avenir comme un météore, fait appel à votre équipe de développement pour concevoir une hiérarchie de classes lui permettant de bien représenter les produits qu'elle offre sur le marché.

Parmi ses produits, on retrouve des téléphones traditionnels que l'on branche au mur, des téléphones sans fil et des téléphones cellulaires avec écran à affichage par cristaux liquides (ACL, en anglais LCD). La compagnie est aussi présente sur le marché de la photo avec des appareils classiques à film 35 mm et, dernièrement, des appareils numériques avec écran à affichage par cristaux liquides (ACL). Ils fabriquent même des télécommandes pour voiture de course et des télécommandes de téléviseurs qui se partagent la même technologie à ondes infrarouges. Vous avez également aperçu dans leur catalogue des produits des caméra vidéo HD dont le viseur comporte un écran à affichage par cristaux liquides.

Contre une promesse de confidentialité, un responsable de la compagnie vous met dans le secret des dieux : on compte commercialiser bientôt une caméra vidéo numérique avec un tout nouveau modèle de viseur beaucoup plus grand dont l'affichage comporte un écran à Diode Électro Luminescente (DEL). Cet écran sera produit par la division des téléviseurs qui fabrique déjà des téléviseurs à affichage DEL, en plus des modèles à écran Plasma et des modèles dont l'écran est à cristaux liquides (ACL).

Suite à cette première prise de contact, faites l'ébauche la plus complète possible d'une hiérarchie de classes qui permettra de représenter les différents appareils électroniques que fabrique la compagnie. Bien sûr, vous désirez établir une forme de standardisation de l'interface de ces appareils afin de simplifier le développement des programmes qui permettront de les contrôler. De plus, vous désirez pouvoir appliquer certaines opérations sur l'ensemble de tous les produits.

Cette ébauche doit prendre une forme inspirée des diagrammes UML et comporter *au minimum* le nom des classes et les relations de composition ou d'héritage entre les classes. Puisque vous cherchez à impressionner votre client, vous voudrez sans doute introduire aussi dans chaque partie du diagramme ce que votre brève analyse vous rend déjà en mesure de préciser : attributs, méthodes, méthodes polymorphiques, abstraites, etc.

Donnez votre esquisse la plus complète et la plus **lisible** possible à la page suivante<sup>1</sup>. Je vous recommande de faire un brouillon d'abord et de disposer votre réponse en mode 'paysage' sur la page réponse pour améliorer la lisibilité. Si votre réponse se poursuit sur un verso d'une autre page, soyez clair dans vos indications afin que je puisse évaluer le plus avantageusement possible votre réponse.

---

<sup>1</sup> Compte tenu du temps que vous avez, bien entendu.

