



Université Internationale  
de Casablanca

LAUREATE INTERNATIONAL UNIVERSITIES

Nous innovons pour votre réussite !

Ecole d'Ingénierie  
Filières : Tronc Commun  
Classe : S5

Cours : Interface Homme-Machine  
Professeur : MOUJAHID Abdallah  
Date : 13/01/2017

# EXAMEN FINAL

Durée : 2 heures

Etudiant :	
Groupe :	
Note :	

## Notes Importantes :

- *Aucun document autorisé. Sont interdits tous les calculatrices, les téléphones, ainsi que tout autre outil de calcul et/ou de communication.*
- *Vous devez aussi remettre à votre professeur cet imprimé, portant votre nom, (Un étudiant qui n'a pas remis l'imprimé n'aura pas de note)*
- *TOUTE sortie est définitive !*
- *La propreté, la clarté et la qualité de rédaction seront pris en considération dans la notation.*
- *TOUTE tentative de fraude sera sanctionnée selon la procédure en vigueur.*



## **Partie I : QCM (3,75 points)**

Pour chaque question, il y a exactement une bonne réponse qu'il faut entourer

**+0,75 pour une bonne réponse, 0 pour absence de réponse, -0,5 pour une mauvaise réponse.**

1. **Quel est l'espace de nom (namespace) à inclure pour utiliser les listes en C# ?**
  - a. System.Lists
  - b. System.Generics
  - c. System.Collections
  - d. System.Arrays
  
2. **Comment le versionnement des assemblies permet d'éviter l'enfer des DLL ?**
  - a. L'environnement d'exécution vérifie qu'une seule version de l'assembly est présente à la fois sur une machine.
  - b. Le .NET permet aux assemblies de spécifier le nom et la version des assemblies qu'ils ont besoins d'utiliser.
  - c. Le compilateur vérifie la compatibilité descendante lors la compilation.
  - d. Il ne le permet pas.
  
3. **Si une méthode est marquée comme protected, qui peut y accéder ?**
  - a. Les classes qui sont à la fois dans le même assembly et qui dérivent de la classe dans laquelle cette méthode est déclarée.
  - b. Seules les méthodes qui sont dans les classes dérivées
  - c. Seules les méthodes qui sont dans la même classe que la méthode en question.
  - d. Les classes du même assembly et celles qui dérivent de la classe dans laquelle cette méthode est déclarée.
  - e. Aucune des propositions ci-dessous.
  
4. **Comment pouvez-vous empêcher l'héritage d'une classe en C#. NET ?**
  - a. Déclarer la classe comme shadows
  - b. Déclarer la classe comme overloads
  - c. Déclarer la classe comme sealed
  - d. Déclarer la classe comme override
  
5. **Quels sont les deux blocs utilisés pour la gestion des exceptions ?**
  - a. try, catch
  - b. try, exception
  - c. manage, exception
  - d. exception, catch



## Partie II : Questions Directes (4,5 points)

1. (1,5 pt) Quel sera le résultat de l'exécution du programme suivant :

<pre>public class ClassBase {     public int a;     protected int b;     private int c;     public ClassBase() {         Console.WriteLine("A");     }     public virtual void doIt() {         Console.WriteLine("B"); // (1)     } }</pre>	<pre>public class ClassDerived: ClassBase {     public int d;     private int e;     public ClassDerived() : base(){         Console.WriteLine("C");     }     public override void doIt(){         Console.WriteLine("D"); // (3)         base.doIt();     }     public void doIt(int i) {         Console.WriteLine("E");         d = 3;         doIt();         Console.ReadKey();     } }</pre>
<pre>public class Tester {     public int t;     public static void Main()     {         ClassBase x = new ClassBase();         ClassDerived y = new             ClassDerived(); // (2)         x.doIt(); y.doIt(); y.doIt(3);     } }</pre>	

.....

.....

.....

.....

2. (1,5 pts) Le programme suivant contient une erreur. Laquelle ? (il faudra donner des explications)

```
abstract class A_Fiche
{
    protected int valeur=10;

    public void test()
    { Console.WriteLine("test"); }

    abstract public int AccessValeur
    { get; set; }
}
```



```
class Fiche : A_Fiche
{
    public override int AccessValeur
    {
        get { return (valeur); }
        set { valeur = value; }
    }
}

class convert
{
    static void Main(string[] args)
    {
        A_Fiche af = new A_Fiche();
        af.test();
    }
}
```

.....

.....

.....

.....

.....

3. (1,5 pts) Lisez le code qui suit et répondez aux questions qui suivent :

```
bool a = true;
bool b = false;
int c = 3;
bool d = c > 3;

if (d)
{
    Console.WriteLine("alex");
}

if (!a || d || c < 2)
{
    Console.WriteLine("bart");
}

do
```



```
{  
    a = !a;  
    Console.WriteLine("car1");  
}  
while (a);  
Console.ReadLine();
```

a) Qu'est-ce que le programme affiche en sortie ?

.....  
.....  
.....

b) Qu'est-ce que le programme affiche en sortie si on remplace la première instruction par bool a = false; ?

.....  
.....  
.....



### Partie III : Etudes de cas - Programmation (11,75 points)

#### **Exercice 1 – C# & POO (4 pts)**

- Un parc auto se compose des voitures et des camions qui ont des caractéristiques communes regroupées dans la classe **Vehicule**. Chaque véhicule est caractérisé par son **matricule**, **l'année de son modèle**, son **prix**.
- Lors de la création d'un véhicule, son matricule est incrémenté selon le nombre de véhicules créés.
- Tous les attributs de la classe véhicule sont supposés privés ce qui oblige la création des propriétés.
- La classe Véhicule possède également deux méthodes **demarrer()** et **accelerer()** qui seront définies dans les classes dérivées et qui afficheront des messages personnalisés.
- La méthode ToString() de la classe Véhicule retourne une chaîne de caractères qui contient les valeurs du matricule, de l'année du modèle et du prix.
- Les classes **Voiture** et **Camion** étendent la classe Véhicule en définissant concrètement les méthodes accelerer() et demarrer() en affichant des messages personnalisés.

#### **Travail à faire :**

1. **(2 pts)** Créer la classe Véhicule.
2. **(2 pts)** Créer les classes Camion et Voiture.

#### **Exercice 2 – C# & IHM (7,75 pts)**

Nous voulons mettre en œuvre une application en C# servant à gérer les écritures dans un journal.

L'interface « **frm\_Journal** » en cours d'exécution de cette application est donnée dans la figure suivante :

The screenshot shows a Windows application interface for managing a journal. It is divided into three main sections:

- Saisie des données (Data Entry):** Contains a text box for 'Montant' (Amount) and radio buttons for 'Type d'opération' (Operation Type), with 'Débit' (Debit) selected.
- Opérations (Operations):** Contains three buttons: 'Ecrire dans le journal' (Write to journal), 'Annuler la dernière écriture' (Cancel last entry), and 'Quitter l'application' (Quit application).
- Journal des écritures (Journal of entries):** A table-like view showing a list of entries with columns for 'Crédit' and 'Débit'. At the bottom, it shows a 'Total' row with values 200, 8300, and -8100.

	Crédit	Débit	
	200	0	
	0	300	
	0	8000	
Total	200	8300	-8100



- Un utilisateur saisi le montant de l'écriture et choisi le type de l'opération qui veut effectuer (crédit ou débit).
- Il clique ensuite sur le bouton « Ecrire dans le journal » pour ajouter le montant saisi dans la liste des crédits si l'option crédit est choisie ou dans la liste des débits si l'autre option débit est choisie.
- Si l'écriture se fait dans la liste des crédits (respectivement, dans la liste des débits) l'item correspondant dans la liste des débits (respectivement, des crédits) est mis à 0. Après chaque écriture, les calculs du total crédit, débit et balance seront effectués.
- A tout moment, cet utilisateur peut annuler la dernière écriture effectuée en cliquant sur le bouton « Annuler la dernière écriture ».
- L'annulation d'une écriture enlève des deux listes crédit et débit la ligne correspondante à la dernière écriture et met à jour le total crédit, débit et balance.
- L'utilisateur peut également quitter l'application par un simple clique sur le bouton « Quitter l'application ».

Les contrôles du formulaire sont décrits dans le tableau suivant :

Contrôle	Type de contrôle	Propriété name	Remarques
Montant	TextBox	txtMontant	
Type d'opération – Crédit	RadioButton	rdCredit	Non Sélectionner
Type d'opération – Débit	RadioButton	rdDebit	Sélectionner
Journal des écritures – Crédit	ListBox	lbJournalCredit	
Journal des écritures – Débit	ListBox	lbJournalDebit	
Opérations – Ecrire dans le journal	Button	btnEcrireJournal	
Opérations – Annuler la dernière écriture	Button	btnAnnuler	
Opérations – Quitter l'application	Button	btnQuitter	
Journal des écritures - Total Débit	Label	lblTotalDebit	Initialisé à 0
Journal des écritures - Total Crédit	Label	lblTotalCredit	Initialisé à 0
Journal des écritures - Balance	Label	lblBalance	Initialisé à 0

### Travail demandé :

1. **(2 pts)** On souhaite initialiser les valeurs du total crédit, débit et balance à 0 lors du chargement du formulaire « frm\_journal ». Le RadioButton rdDebit doit aussi être sélectionné.
  - a. Quel est l'événement qu'il faut programmer ?
  - b. Donnez le code de cet événement (pour faire les opérations décrites ci-dessus).
  - c. Est-il nécessaire de déclarer des variables globales ? si oui, donner le code de déclaration et expliquer l'emplacement où il faut mettre ce code.
2. **(2 pts)** Donnez le programme du bouton « Ecrire dans le journal » **comme défini dans l'énoncé.**



Le code doit vérifier que le montant saisi par l'utilisateur est bien un réel et ne peut être vide, sinon on affiche des messages d'erreurs à l'aide d'un contrôle MsgBox.

3. **(1,5 pts)** Donnez le programme du bouton «Annuler la dernière écriture» **comme défini dans l'énoncé.**
4. **(0,75 pt)** Donnez le programme du bouton «Quitter l'application».
5. **(1,5 pts)** Proposer un code permettant de vérifier qu'un clique sur le bouton « Annuler la dernière écriture » ne peut pas être possible si aucune écriture n'est présente dans la liste crédit et débit (les deux listes débit et crédit sont vides).