

Exercice 1:QCM

(4 points) Sur votre feuille, écrivez la (ou les) bonne réponse pour les questions suivantes. Répondez sur le modèle suivant :

- A1 et B1
- B2
- ne sais pas
- B4 et D4

Toute mauvaise réponse **enlève 1 point**. Toute bonne réponse ajoute 1 point. Vous pouvez ne pas répondre à une des questions (vous écrivez je ne sais pas).

Question 1

```
public class Entier {
    private int i;
    public Entier(int i) {
        this.i = i;
    }
    public void setVal(int i) {
        this.i = i;
    }
    public int theentier{
        get { return i; }
        set { i = value; }
    }
    public override string ToString() {
        return i + "";
    }
}
class Program
{
    static void Main(string[] args)
    {
        Entier i = new Entier(1);
        Entier j = i;
        j.setVal(2);
        Console.WriteLine(i);
        Console.ReadKey();
    }
}
```

Réponses possibles :

- A1. L'exécution de Main affiche 1
- B1. L'exécution de Main affiche 2
- C1. L'exécution de Main affiche autre chose que 1 ou 2.

Question 2

Dans la question précédente on modifie la ligne

```
public override string ToString() {
```

en

```
public string toString() {
```

Réponses possibles :

- A2. L'exécution de Main affiche 1
- B2. L'exécution de Main affiche 2
- C2. L'exécution de Main affiche autre chose que 1 ou 2.

Question 3

```
public class A
{
    public void m()
    {
        Console.WriteLine("A");
    }
}

public class B:A
{
    public void m()
    {
        Console.WriteLine("B");
    }
}
```

```

}

class Program
{
    static void Main(string[] args)
    {
        A b = new B();
        ((A)b).m();
    }
}

```

Réponses possibles :

- A3. L'exécution de Main affiche A.
- B3. L'exécution de Main affiche B.
- C3. Ce code ne compile pas (indiquez la ligne en cause).
- D3. Ce code compile mais le compilateur affiche un warning. L'exécution affiche A.
- E3. Ce code compile mais le compilateur affiche un warning. L'exécution affiche B.

Question 4

```

public abstract class A
{
    public abstract int m();
    public int n()
    {
        return 2 * m();
    }
}

```

Réponses possibles :

- A4. La méthode n doit être déclarée abstraite.
- B4. La méthode n ne doit pas être déclarée abstraite.
- C4. On peut choisir de déclarer la méthode n abstraite ou pas abstraite. Les 2 solutions sont bonnes

Exercice II :

Dans cet exercice, il n'y a pas d'erreur d'énoncé mais des erreurs de code. Soit les classes suivantes :

```

class Personne {
    public virtual void parler() {
        Console.WriteLine("hum!");
    }
}
class Adulte : Personne{
    public override void parler()
    {
        Console.WriteLine("Bonjour tout le monde!");
    }
}
class AdulteDistinguee : Adulte {
    public override void parler()
    {
        Console.WriteLine("Mes chers amis, bonjour!");
    }
}
class Jeune : Personne { }
class Ado : Jeune {
    public override void parler()
    {
        Console.WriteLine("salut les mecs!");
    }
}
class Enfant : Jeune { }
class Bebe : Enfant {
    public override void parler()
    {
        Console.WriteLine("J'chui pas un bebe!");
    }
}
class BebeCadum : Bebe
{
    public override void parler()

```

```

    {
        Console.WriteLine("Agheu, agheu!");
    }
}

```

1) Donner ce qui est affiché par le programme suivant :

```

static void Main(string[] args)
{
    Personne P1 = new Personne();
    Personne P2 = new AdulteDistinguee();
    Adulte P3 = new AdulteDistinguee();
    Personne P4 = new Bebe();
    Ado P5 = new Jeune();
    Enfant P6 = new BebeCadum();
    BebeCadum P7 = new BebeCadum();
    Enfant P8 = new Bebe();
    P1.parler();
    P2.parler();
    P3.parler();
    P4.parler();
    P5.parler();
    P6.parler();
    P7.parler();
    P8.parler();
}

```

Faire les corrections qui s'imposent (si vous avez un choix entre deux classes à faire, prenez la classe la plus générale).

2) On effectue les affectations suivantes (ces affectations sont indépendantes les unes des autres, elles ne sont pas exécutées en séquence!). Pour chaque affectation, dites si elle fonctionne sans erreur, si elle provoque une erreur à la compilation ou si elle provoque une erreur à l'exécution. Si il y a une erreur expliquez (rapidement) pourquoi.

```

P1 = P2;
P4 = P1;
P3 = P4;
P3 = P1;
P4 = P5;
P7 = P6;
P7 = (BebeCadum)P4;
P6 = (Bebe)P4;
P3 = (AdulteDistinguee)P2;
P8 = (Bebe)P5;

```

3) On crée un tableau de jeunes de la manière suivante:

```

Jeune[] e = new Jeune[4];
e[0] = (Jeune)P4;
e[1] = P5;
e[2] = P6;
e[3] = P7;
e[4] = P8;

```

a) Que faut-il faire pour que les affectations suivantes soient toutes valides (à la compilation et à l'exécution ; on reprendra les variables dans l'état de la question 1):

```

e[0] = P4;
e[1] = P5;
e[2] = P6;
e[3] = P7;
e[4] = P8;

```

b) Que produit l'exécution suivante :

```

for (int i=0; i<e.Length;i++)
    e[i].parler();

```

c) Que faut il faire pour que les affectations suivantes soient toutes valides (à la compilation et à l'exécution):

```

P5 = e[0];
p4 = e[1];
p7 = e[2];
P8 = e[3];
P6 = e[4];

```