

4 TP 4

On peut trouver la documentation pour les classes de la BCL sur :
<http://docs.go-mono.com> ou <https://msdn.microsoft.com/fr-fr/library/>

Exercice 24 (Structures - Surcharge des opérateurs - Opérateurs de conversion). Pour représenter une durée (ou une heure de la journée) comme 10h 5min ou 45min on peut utiliser le nombre de minutes correspondant (605 et 45 dans les exemples précédents). Une structure Temps peut être déclarée de la manière suivante :

```
public struct Temps{
    private readonly int minutes;
    public Temps(int hh, int mm){
        this.minutes=60*hh+mm;
    }
    public override String ToString(){
        return minutes.ToString();
    }
}
```

1. Dans la structure Temps ajouter une propriété Heure qui renvoie le nombre d'heures et une propriété Minute qui renvoie le nombre de minutes (les deux propriétés doivent être en lecture seule).
Par exemple, `new Temps(23,45).Minute` doit donner la valeur 45.
2. Modifier le méthode ToString() de sorte qu'elle renvoie une représentation dans le format *hh : mm*. Par exemple 10 : 05 au lieu de 605
(utiliser `String.Format("{0:00}:{1:00} ", Heure, Minute)`).
Ecrire un exemple d'utilisation dans la méthode Main.
3. Déclarer un constructeur supplémentaire Temps (int) et redéfinir les opérateurs + et - pour la structure Temps. Par exemple vous devez pouvoir exécuter le code
`Temps t1 = new Temps(9,30);`
`Console.WriteLine(t1 + new Temps(1,15));`
`Console.WriteLine(t1 - new Temps(1,15));`
4. Dans la structure Temps définir les opérateurs de conversion suivant :
 - une conversion implicite du type int vers le type Temps ;
 - une conversion explicite du type Temps vers le type int.Dans la méthode Main, exécutez le code
`Temps t1 = new Temps(9,30);`
`Temps t2 120;`
`int m1 = (int) t1;`
`Console.WriteLine("t1={0} and t2={1} and m1={2}", t1, t2, m1);`
`Temps t3 = t1 + 45;`
Expliquer pourquoi l'addition dans l'initialisation de t3 est légale. Quelle est la valeur de t3?

5. Dans la structure Temps, essayez de déclarer un attribut non-statique de type Temps. Pourquoi est ce illégal ? Que se passe-t-il si vous déclarez le type Temps comme une classe ?
6. Dans la structure Temps, rendre l'attribut minutes public (et non readonly) à la place de private readonly. Exécutez le code

```
Temps t1 = new Time(9,30);
Temps t2 = t1;
t1.minutes= 100;
Console.WriteLine("t1={0} and t2={1}",t1,t2);
```

Obtient-on le même résultat si on change le type Temps en classe ? Expliquer.

Exercice 25 (Surcharge des opérateurs). On souhaite concevoir une classe pour représenter des fractions de la forme p/q où p et q sont des entiers de taille arbitraire. Pour disposer de la classe BigInteger qui permet de manipuler des entiers arbitrairement grands, on ajoutera la bibliothèque Systems.Numerics dans les références du projet et on ajoutera la directive using Systems.Numerics au début des fichiers sources.

1. Déclarer une classe Fraction munie de deux attributs numérateur et dénominateur de type BigInteger. Définir deux propriétés (publiques) Numérateur et Dénominateur pour accéder aux attributs. On s'assurera que dénominateur ne puisse pas être affecté à la valeur 0.
2. Définir deux constructeurs Fraction(BigInteger n, BigInteger d) et Fraction(string n, string d). Pour le second on utilisera la méthode BigInteger.Parse.
3. Redéfinir la méthode ToString de Fraction.
4. Définir une méthode donnant le développement décimal de la fraction avec d chiffres après la virgule, où d est donné en argument. Indication : considérer le quotient $(10^d \times p)/q$, la méthode Substring de la classe String pourra servir.
5. Redéfinissez pour cette classe les opérations usuelles : +, -, *, /
6. Ajouter une méthode qui permet de simplifier la fraction (c'est-à-dire obtenir un numérateur et un dénominateur premiers entre eux). Indication : utiliser la méthode GreatestCommonDivisor de BigInteger
7. On souhaite obtenir une valeur approchée de $\sqrt{p/q}$. Pour cela on considère la suite (y_n) définie par $y_{n+1} = y_n(3 - (p/q)y_n^2)/2$ qui converge vers $\sqrt{q/p}$ si y_0 est suffisamment proche de $\sqrt{q/p}$. Ecrire une méthode qui fournit une approximation de $\sqrt{p/q}$ à 10^{-d} près. (On calculera les y_n jusqu'à ce que $|y_n^{-2} - p/q|/(y_n^{-1}) \leq 10^{-d}$.)
8. Retrouver les 500 premières décimales de $\sqrt{2}$:

```
1,41421356237309504880168872420969807856967187537694807317667973799073247846210703885038753432764157273501384623091
2297024924836055850737212644121497099935831413222665927505592755799950501152782060571470109559971605970274534596862
014728517418640889198609552329230484308714321450839762603627995251407989687253396546331808829640620615258352395054
745750287759961729835575220337531857011354374603408498847160386899970699004815030544027790316454247823068492936918
62158057846311159666871301301561856898723723
```