

TP2 : TicTacToe

13 novembre 2007

1 Reprise

1.1 Fonctions Virtuelles

(Environ 0 :40 h) Réprise : Testez le programme ci-dessous. Qu'est-ce qui se passe si on remplace `abstract` par `virtual` (override by new)? Peut `C` dériver de `B` ET `B2`? Pourquoi pensez vous qu'il y a cette restriction?

Pour dériver attacher plusieurs méthodes, on utilise des interfaces. Mais on en parlera une autre fois...

```
using System;
using System.Text;

public abstract class A
{
    public abstract void test();
}

public class B2 : A
{
    public override void test()
    {
        Console.WriteLine("B2B2B2B2");
    }
}

public class B : A
{
    public override void test()
    {
        Console.WriteLine("BBBB");
    }
}

public class C : B
{

```

```

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

    public class test
    {
        public static void Main()
        {
            C test = new C();
            test.test();

            B test2 = (B)test;
            test2.test();
        }
    }
}

```

1.2 Références

(Environ 20 min) Pouvez-vous écrire une fonction qui échange deux valeurs ?
ARRETEZ de lire ici et essayez le !!! Pourquoi ca ne marche pas avec ce que vous connaissez jusqu'à maintenant ?

Le mot clé "ref" à été ajouté pour ceci.

Essayez la fonction

```

swap(ref int t1, ref int t2) {
    int temp;
    temp=t1;
    t1=t2;
    t2=temp;
}

```

Pour appeler la fonction il faut aussi mettre "ref" devant les paramètre.

Essayez le programme suivant :

```

using System;
using System.Text;

public class A {
    public int a;
}

public class Prog {
    public static void setA(A v, int t)
    {
        v.a = t;
    }
}

```

```

    }

    public static void Main()
    {
        A temp=new A();
        temp.a=3;

        setA(temp, 0);

        Console.WriteLine("{0}", temp.a);
    }
}

```

Qu'est-ce qui se passe? Les objets sont toujours passés en référence... MAIS est-ce que swap marche sans mot "ref" sur les objets???

2 TicTacToe

Le but aujourd'hui est de créer un TicTacToe. Le jeu de 3×3 cases où le but est de remplir une colonne, ligne ou diagonale.

2.1 Le jeu

(Environ 2 :00 h) Créez un TicTacToe en utilisant la ligne de commande. Le jeu sera pour deux joueurs.

Déroulement : Utilisez les deux types suivantes :

```

enum Winner {X, O, NONE};
enum GameValue {X, O, EMPTY};

```

N'oubliez pas d'utiliser des exceptions :

```

GameValue [] table= new table[9];

try {
    table[10]=GameValue.X;
} catch (IndexOutOfRangeException e){
    Console.WriteLine("INVALID NUMBER");
}

```

Pour lire l'entrée d'un joueur, vous pouvez utiliser *Console.ReadLine()*. Ensuite *Convert* va vous permettre de transformer le string résultant en `int32` (`int`). Évidemment *Convert* aussi peut lancer des exceptions.

Créez une fonction pour vérifier si un joueur a gagné. Pour ceci utilisez une boucle `for` :

```
for (int i=0; i<3;++i)
    ...
```

2.2 Interface

(Environ 1 :00 h) **ATTENTION** cet exercice à été rapporté à la semaine prochaine!!!!