

Reguli redactare proiect .NET

1. Fiecare clasa declarata in cadrul proiectului va fi continuta intr-un singur fisier. Exceptie pentru clasele si metodele declarate "partial".
2. Fiecare clasa va avea un comentariu de tip "summary" in care se va descrie rolul acesteia.
3. Proprietatile si metodele publice vor avea un comentariu de tip "summary" in care se vor descrie, pe scurt, rolul acestora, semnificatia parametrilor. Aceste informatii apar in Intellisense.
4. Tot ce este public in clasa se declara folosind notatia *Pascal*, in caz contrar se va folosi notatia *Camel*, exceptie clasele imbricate – vezi 5. Parametrii metodelor, datele membru private, variabile locale, pot incepe si cu _. Variabilele locale si parametrii metodelor incep cu litera mica.

Exemplu corect :

```
int _contor=20;
int contor = 30;
```

Exemplu incorect – nume variabila locala:

```
int Contor = 20;
```

Exemplu corect:

```
public void M(int numarElemente, string sir)
{
    //cod
}
```

Exemplu incorect – nume parametri:

```
public void M(int NumarElemente, string Sir)
{
    //cod
}
```

5. Se va folosi notatia *Pascal* la denumirea claselor indiferent de modifierul de acces.
6. Codul trebuie sa fie indentat si cate o singura instructiune pe un rand.
7. Perechile {, } trebuie sa fie indentate pe aceeasi pozitie.

Corect	Incorect
<pre>if (conditie) { // cod } else { // cod }</pre>	<pre>if (conditie) { // cod } else { // cod }</pre>

8. In general codul dintr-o metoda ar trebui sa poata fi afisat intr-un singur ecran. Daca acest lucru nu e posibil veti folosi directivele **#region** si **#endregion** pentru anumite portiuni de cod.
9. In fisierul ce contine metoda **Main** veti specifica intr-un comentariu, plasat la inceputul fisierului, urmatoarele informatii (veti respecta aceasta ordine):

```
/// <summary>
/// Nume si prenume: Popescu Ioan
/// Laborator: Vineri 18-20
/// Tema laborator: Rollback pe obiecte
/// Data predare proiect: 17.11.2016
/// Declaratie: Declar pe propria raspundere ca acest proiect nu a fost
/// copiat din alte surse
/// Bibliografie, surse de inspiratie: MSDN, http://..., carti
/// </summary>
```

10. Metoda **Main** va contine apel la alte metode plasate in acelasi fisier cu Main. Exemplu:

```
public static void Main()
{
    TestPolimorfism();
}

// Metoda ce testeaza polimorfismul pentru ierarhia de clase creata
private void TestPolimorfism()
{
    //cod
}
```

11. Un proiect (tema) pentru laborator poate sa contina mai multe "Projetcs" in Visual Studio. Toate aceste proiecte vor fi continute intr-o singura solutie din Visual Studio daca nu se specifica altfel.