



C# .NET

LABORATOR 1

De ce aveti nevoie



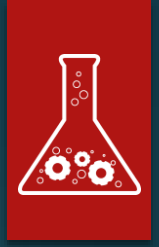
- ▶ Cunostinte teoretice de POO (Programare Orientata Obiect)
- ▶ Cunostinte de C++ (de baza) si implementarea POO in C++
- ▶ Visual Studio 2010, 2012 sau 2013 (recomandat), Visual Studio C# Express Edition sau orice alt IDE opensource (SharpDevelop, MonoDevelop)
- ▶ Pentru studentii eligibili Microsoft DreamSpark, Visual Studio poate fi downloadat gratuit de pe platforma
- ▶ Pentru ceilalti exista varianta Visual C# 2010 Express, ce poate fi folosita in scopuri comerciale si este disponibila la adresa <http://www.microsoft.com/visualstudio/eng/downloads#d-2010-express>

Ce este C# (C Sharp)



- ▶ Un limbaj de programare puternic obiectual (asemanator Java, de acolo fiind si inspirat)
- ▶ Numele lui provine din muzica
- ▶ A aparut in anul 2000 impreuna cu proiectul .NET
- ▶ Face parte din .NET Framework
- ▶ Este case sensitive

.NET Framework



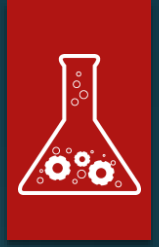
- ▶ Framework software dezvoltat de Microsoft ce ruleaza in primul rand pe sistemul de operare Windows
- ▶ Ajuns in prezent la versiunea 4.5 (odata cu lansarea Windows 8)
- ▶ In cadrul acestor laboratoare vom lucra cu versiunea 4.0
- ▶ Este independent de platforma
- ▶ Este format din CLR si BCL

CLR – Common Language Runtime



- ▶ Masina virtuala .NET, asemanatoare JVM (Java Virtual Machine)
- ▶ Transforma codul din limbajul de programare in care a fost scris in cod masina
- ▶ Foloseste JIT (Just In Time compilation)
- ▶ Unul din avantajele JIT este faptul ca erorile ne sunt afisate imediat dupa scrierea codului

BCL – Base Class Library



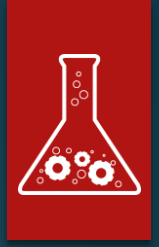
- ▶ Include clase scrise de dezvoltatorii limbajului pentru diferite functionalitati
- ▶ Aceste clase sunt organizate ierarhic in namespace-uri (notiune asemanatoare cu cea din C++)
- ▶ Prima clasa din ierarhie este **Object**
- ▶ Intalnim BCL nu numai in .NET Framework, ci si in .NET CF (Compacted Framework – pentru dispozitive mobile), Silverlight (alternativa Microsoft la Adobe Flash) si Mono (proiect open source ce permite dezvoltarea de aplicatii C# pe Linux)

In C# putem dezvolta aplicatii



- ▶ Desktop (numite si WinForms – scopul acestor laboratoare)
- ▶ RIA, folosind Silverlight
- ▶ Web in ASP.NET sau ASP MVC
- ▶ Jocuri – XNA pentru Xbox
- ▶ Mobile pentru Windows Mobile sau Windows Phone

Principalele diferente fata de C++



- ▶ Nu se pot instantia variabile globale, acestea trebuind sa fie in clase sau structuri
- ▶ Constructorii se apeleaza doar la instantierea cu **new**
- ▶ Nu mai avem nevoie de destructori (asta nu inseamna ca nu ii mai putem folosi), obiectele sunt dezalocate automat de o clasa speciala numita Garbage Collector
- ▶ Dezalocarea poate fi apelata si explicit prin **GC.Collect()**;
- ▶ Nu exista notiunea de pointer decat in modul de lucru unsafe (nerecomandat)
- ▶ Totusi tipurile referentiale (vezi continuarea) au comportament asemanator pointerilor din C++



Tipuri de date

Tipurile valorice (manipulate direct prin numele lor)

- ▶ Tipuri de baza (int, float, double, decimal, bool, etc)
- ▶ Tipul **struct**
- ▶ Tipul **enum**



Tipuri de date

Tipurile referentiale (alocate dinamic si manipulate prin referinte)

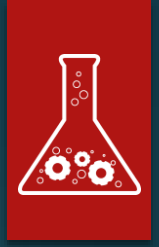
- ▶ Tipul **class**
- ▶ Tipul **interface**
- ▶ Tipul **delegate**

Principalele diferente fata de C++



- ▶ Tipul referinta se apropie mai mult de pointerul din C++, doar ca se derefera automat
- ▶ Tipurile valorice sunt structuri si detin metode (de ex: `14.ToString();`)
- ▶ Derivare multipla este permisa doar in anumite circumstante (dintr-o clasa si mai multe interfete)
- ▶ Trebuie sa existe o singura functie *Main*, indiferent de numarul claselor sau al fisierelor

Clasa String



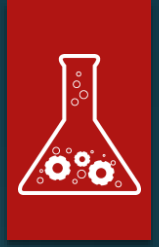
- ▶ Se poate folosi cu litera mare sau cu litera mica (string este alias la String)
- ▶ Se gaseste in namespace-ul System
- ▶ Este derivate direct din Object
- ▶ Este sealed (nu se mai pot deriva alte clase din ea)
- ▶ Este immutable (un obiect odata creat nu mai poate fi modificat)

Interfata (interface)



- ▶ Asemănătoare noțiunii de clasă abstractă din C++ (clasă ce conține doar funcții virtuale pure)
- ▶ Nu se poate instanția
- ▶ Defineste de obicei un comportament
- ▶ Interfetele din BCL sunt denumite CamelCase cu I în față
(Exemplu: **IComparable** ce conține metoda **CompareTo()**)

Delegatul (delegate)



- ▶ Asemănător noțiunii de pointer la funcție din C++
- ▶ Se folosește în special pentru apelul de funcții între threaduri (fire de execuție) diferite, apelul direct fiind imposibil
- ▶ Un alt caz unde trebuie utilizat este evitarea referirii circulare (nu putem adăuga într-o clasă A o referință la clasă B, pentru că mai apoi în clasă B să adăugăm referință la clasă A – astfel apelul funcțiilor din A poate fi făcut doar prin intermediul delegatilor)

Prima noastra aplicatie .NET



- ▶ Exemplu de structura (tip valoric)
- ▶ Exemplu de clasa (tip referential)
- ▶ Diferenta intre tipurile valorice si referentiale
- ▶ Metoda Clone()
- ▶ Vectori, matrice, foreach
- ▶ Tipul String, metode specifice lui
- ▶ Proprietati
- ▶ Containere din BCL