**Alegerea limbajului de programare C#**

Conform seriei de artiocole publicate in Gazeta de Informatica la sectiunea programare de Mihai Scortaru, motivul principal pentru care am ales si eu limbajul C# este acela că acest limbaj este cel mai folosit pentru crearea de aplicații pentru platforma Microsoft. Chiar daca există si alte alternative, C# s-a impus desi s-ar zice ca C# este varianta Java a celor de la Microsoft. Elementele de bază sunt foarte asemănătoare și există o mulțime de concepte introduse de Java care au fost preluate de C#.

C# este un limbaj de programare modern, simplu, cu circa 80 de cuvinte cheie şi 12 tipuri de date predefinite. Este foarte asemănător cu Java şi C++. Limbajul de programare C# este intregrat în platforma Microsoft .NET Framework . Platforma .Net este un cadru(Framework) de dezvoltare software unitară care permite realizarea, distribuirea şi rularea aplicaţiilor de tip forme Windows şi aplicaţiilor WEB. Pentru a dezvolta aplicaţii în limbaju C# avem nevoie de mediul de dezvoltare Visula C# 2010 Express, care se poate descărca de pe siteul oficial a lui Microsoft.  instalarea programului fiind foarte simplă.

**Utilizare**

Limbajul este utilizat de mulți programatori ceea ce permite foarte ușor să folosim codul scris de alții. Există  multe tehnologii care utilizează limbajul C#. Folosind acest limbaj vom putea dezvolta aplicații software care vor rula pe calculatoarele obișnuite, pe tablete sau pe telefoane. Poate fi utilizat și în alte scopuri (de exemplu aplicații care rulează pe servere) dar cam tot ce are legătură cu aplicațiile mobile pentru Windows va fi scris, în principiu, în C#.

În C# se scriu de asemenea aplicaţii pentru sisteme complexe care funcţionează sub o mare varietate de sisteme de operare, cât şi pentru sisteme dedicate (*embeded systems*). Acestea din urmă se întind pe o arie largă, de la dispozitive portabile cum ar fi ceasuri digitale, telefoane mobile, MP3 playere, până la dispozitive staţionare ca semafoare de trafic, sau controlere pentru automatizarea producţiei.

**Puncte tari**

* Este un limbaj de programare simplu, modern, de utilitate generală, cu productivitate mare în programare.
* Este un limbaj orientat pe obiecte.
* Permite dezvoltarea de aplicaţii industriale robuste, durabile.
* Oferă suport complet pentru dezvoltarea de componente software, foarte necesare de pildă în medii distribuite. De altfel, se poate caracteriza C# ca fiind nu numai orientat obiect, ci şi orientat spre componente.

În strânsă legatură cu ***Arhitectura .NET*** (***.NET Framework***) pe care funcţionează, C# gestionează în mod automat memoria utilizată. Eliberarea memoriei ocupate (*garbage collection*) de către obiectele care nu mai sunt necesare aplicaţiei, este o facilitate importantă a limbajului. Programatorii nu mai trebuie să decidă singuri, aşa cum o fac de pildă în C++, care este locul şi momentul în care obiectele trebuie distruse.

C# este un limbaj compilat. In urma compilării unui program C#, nu se crează un cod executabil. Se creează un fişier numit ***assembly*** care de regulă se identifică cu extensia **.exe** sau **.dll**. Un asemenea fişier nu poate fi executat pe un sistem pe care nu există infrastructura **.NET**. Fişierul conţine un tip special de cod, numit ***Limbaj Intermediar***, pe scurt **CIL** (*Common Intermediate Language*). Limbajul **CIL** defineşte un set de instrucţiuni portabile, independente de orice tip de procesor şi platformă.

C# funcţionează având .NET ca infrastructură, dar .NET suportă şi alte limbaje, cum este C++, Visual Basic sau Java. În oricare dintre aceste limbaje se face programarea, sunt la dispoziţie aceleaşi biblioteci de clase. .NET realizează în acest fel **interoperabilitatea limbajelor**.

La aceste caracteristici generale se adaugă şi alte trăsături, cum este de pildă suportul pentru internaţionalizare, adică posibilitatea de a scrie aplicaţii care pot fi adaptate cu uşurinţă pentru a fi utilizate în diferite regiuni ale lumii unde se vorbesc limbi diferite, fără să fie necesare pentru aceasta schimbări în arhitectura software.

**Date de intrare**

Programele ne furnizează date de ieșire. De cele mai multe ori ele au nevoie de informații din exterior. Acestea le sunt comunicate prin intermediul așa numitelor date de intrare. Dacă prin intermediul datele de ieșire programele ne furnizează informații, prin intermediul datelor de intrare noi suntem cei care oferim informații. Și datele de intrare pot lua diverse forme:

* informații introduse de la tastatură,
* mișcarea mouse-ului,
* apăsarea butoanelor mouse-ului,
* apăsarea unui touchscreen,
* rostirea în microfon, etc.

Acele informații codificate pe care le vom obtine la sfârșitul unei secțiuni anterioare nu pot fi înțelese decât de alte programe, ele reprezentand datele de intrare pentru aceste alte programe. O observație importanta este aceea că, în C#, fiecare instrucțiune se încheie cu semnul punct și virgulă.