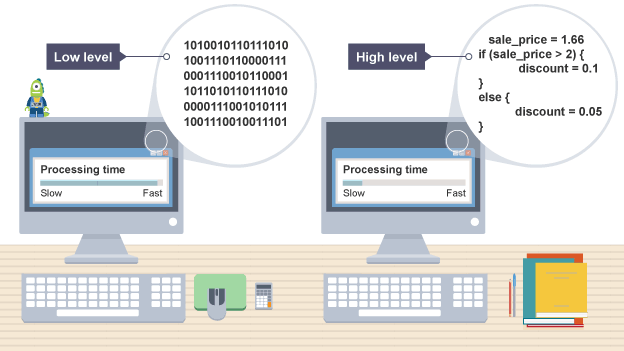
Въведение в C# и .NET

1. Какво е **C#**?



C# (Си Шарп) е език за програмиране, създаден от Майкрософт през 2001 г. Той е обектно-ориентиран език, т.е. програмите, които биват изграждани чрез него са изградени на базата на взаимодействието на обекти. C# е език от високо ниво, което значи, че е изключително разбираем и достъпен за начинаещи програмисти, тъй като е много по-близък до хората като синтаксис, отколкото до машините, които боравят единствено с 0-ли и 1-ци. Това удобство прави процеса на писане на програмите по-лесен и дава възможност на програмиста да се фокусира върху разработката, а не върху специфични детайли характерни за машината върху/за която пишем.



На картинката по-горе може да видите как машината „вижда“ нашия код (чрез 0-ли и 1-ци) и как ние го виждаме, когато пишем на език като C# (от високо ниво).

C# може да се използва за създаването на почти всякакъв вид приложения:

* Десктоп приложения

Технологии:

WPF, WinForms, MAUI, Console

* Уеб приложения

Технологии:

ASP.NET MVC, ASP .NET Core, Razor Pages, Web API и др.

* Мобилни приложения

Технологии:

Xamarin Forms, MAUI

* Игри

Технологии:

Unity

* Библиотеки(Class libraries)
* Тестови проекти

Технологии:

XUnit, NUnit и др.

Разбрахме какви видове проекти можем да създаваме със C#, но нека разберем какво ни е нужно, за да можем да стартираме C# програма (проект) върху нашият компютър. Тук ще се запознаем с **„Какво е това .NET?“.**

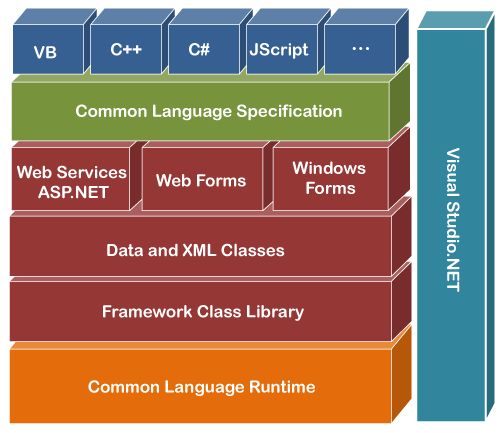
****

1. Какво е .**NET**?

**.NET** се нарича платформата, върху която изграждаме нашите приложения. Накратко, за да можем да изпълним **C#** програма на нашият компютър, ние трябва да сме инсталирали преди това **.NET** на същия компютър. **.NET** е **безплатна**, **cross-platform** **платформа** за разработка. Cross-Platform значи, че можем да създаваме приложения (проекти), които работят както на **Windows**, така и на **Linux** и **Mac OS**.

.**NET** се състои от две основни части в своята структура:

* **CLR (Common Language Runtime)** – Това е средата, в която се изпълнява нашят код. Представлява виртуална машина, която контролирано изпълнява **.NET** кода и осигурява различни услуги като управление на паметта, управление на сигурността и др. Можем да мислим за нея, като за специална среда, в която кодът ни работи. Кодът, който работи в **CLR** се нарича **managed** код и е абсолютно защитен от заобикалящата го компютърна среда. Кодът, работещ в тази среда се ползва с много от „благините“ на **CLR**, но за това ще говорим в следващите ни видеа/материали.
* **FCL (Framework Class Library)** – Библиотека от класове, които са ни дадени на готово, и чрез които можем да разработваме нашите програми (разбирайте инструментариума ни). Съдържа функционалност, даваща ни възможност да работим с бази данни, XML, JSON, файлова система, мрежата, вход/изход от конзолата и т.н.
  1. Архитектура на **.NET**



**Бележка:** Диаграмата е до голяма степен точна, но има и разлики, които се появиха в последните 5-6 години.

Както виждате на картинката по-горе, най-отдолу е **CLR**, който реално е най-близо до нашата операционна система (**Windows**, **Linux**, **Mac** **OS**). Това е както казахме средата, в която се изпълнява нашият код. Върху **CLR** стои **Framework Class Library** инструментариума, който казахме, че ни е даден. Отгоре над него стоят още класове и библиотеки за достъп до бази данни и др. Над тях са различните технологии, които можем да ползваме като **WinForms**, **WPF**, **ASP .NET Core**, **Web API** и др. И най-отгоре стоят съответно езиците, които можем да използваме върху **.NET** платформата. Много са езиците, които могат да се ползват върху **.NET**, но ние ще се съсредоточим само върху **C#.**