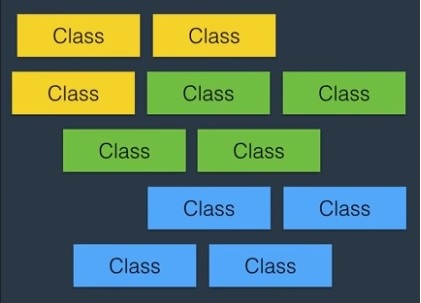
Архитектура на C# програмите

1. Основни съставни части на C# приложенията

Програма:



1. Какво е клас?

Класът е контейнер, който съдържа данни (**data**) и методи (**methods**), с които се характеризира. Той представлява обект от реалния свят.

***Пример:***

Имаме клас, казващ се “**Car**”.

**Car** може да има данни от сорта на:

* Brand
* Model
* HorsePower

**Car** може да има и методи (т.е. някакво поведение)

* Drive()
* Stop()
* ChangeGear()

Така имаме клас **Car**, който има съответните данни (**Brand**, **Model**, **HorsePower**) и съответното поведение (**Drive()**, **Stop()**, **ChangeGear()**).

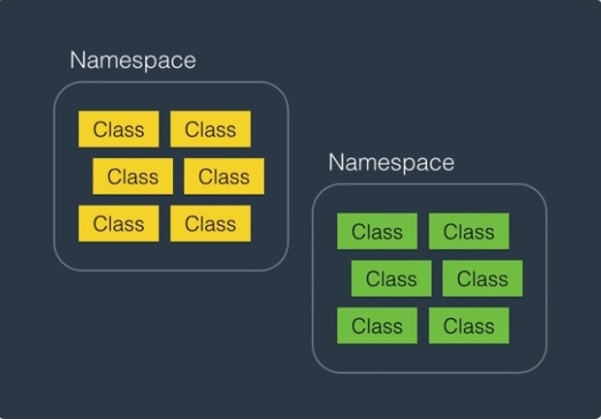
Ние избираме как да моделираме класовете в нашата програма, т.е. какви данни да има в тях и какво поведение да имат.

1. Групиране на класовете в namespace-и

Тъй като класовете, които ползваме са изключително много, то те трябва да бъдат групирани някак си. Точно тук на помощ идват namespace-ите.

**Namespace** е **контейнер**, който съдържа множество **взаимосвързани** **класове**.

***Assembly (.DLL или .EXE)***



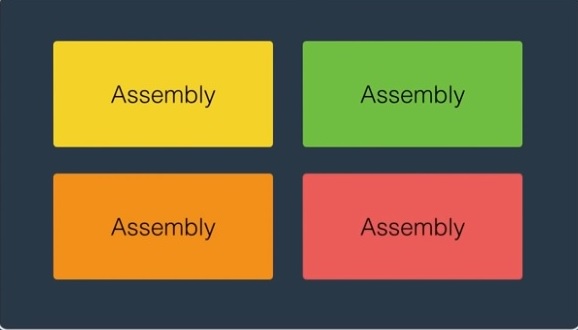
1. Групиране на namespaces

Namespace-ите се групират в нещо наречено **Assembly**. Assembly-то може да бъде **.EXE** или **.DLL**. То е контейнер, който съхранява взаимосвързани namespace-и.

1. Компилиране на C# проект

Когато имаме **C#** цялостен проект, то ние може да имаме множество assembly-та, които трябва да се компилират, за да се създаде крайният продукт на нашата програма, който е или **.DLL** или .**EXE** файл. С едно изречение, може да имаме проект, който съдържа едно **assembly**, но може да имаме и проект, който съдържа много assembly-та.

***C# application:***



1. Какво значат **.EXE** и .**DLL** абревиатурите:

**.EXE** – **executable file** (изпълним файл) т.е. файл, който като го кликнем два пъти изпълнява някаква програма.

**.DLL** – **dynamic link library** -> Файл, който съдържа преизползваем код. Няма възможност да бъде изпълнен отделно (самостоятелно), а е тип проект, който се извиква от други проекти, които искат да използват наличната логика в него наготово, а не да я пишат изначално.