## Типы данных

В языке C# существуют 2 типа данных. Ссылочный и значимый. Вот несколько примеров:

Типы значений:

1. перечисления enum, например, LightType и т. д.
2. структуры (struct), например, byte, int, float, bool, Vector3, Quaternion и т. д.

Ссылочные типы:

1. Тип object
2. Тип string
3. Классы (class)
4. Интерфейсы (interface)
5. Делегаты (delegate)

Важно понимать между ними различие! В C# устройство памяти следующие, есть стек и управляемая куча. Значимые типы данных создаются в стеке, а ссылочные в куче.

**Модификаторы доступа**

В C# .Net существует 6 модификаторов доступа:

* **public** – публичный, общедоступный класс или член класса. Он доступен из любого места в коде, а также из других программ и сборок;
* **private** – закрытый класс или член класса, полная противоположность модификатору **public**. Доступен только из кода в том же классе или контексте;
* **protected** – доступен из любого места в текущем классе или в производных классах;
* **internal** – класс и члены класса с подобным модификатором доступны из любого места кода в той же сборке. Но он недоступен для других программ и сборок (как модификатор **public**);
* **protected internal** – совмещает функционал двух модификаторов. Классы и члены класса с таким модификатором доступны из текущей сборки и из производных классов.
* **private protected** – комбинация ключевых слов private protected является модификатором доступа к члену. К члену private protected имеют доступ типы, производные от содержащего класса, но только в пределах содержащей сборки. Сравнение модификатора private protected с другими модификаторами доступа (<https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/private-protected>).

Если мы явно не указываем модификатор доступа, он выставляется автоматически. Для класса и структуры это **internal**, поля класса и функции – **private**.

## Основные конструкции

Любой класс либо структура могут содержать в себе следующий набор элементов

1. Поля
2. Свойства
3. Конструктор
4. Метод
5. Перегруженный оператор