C# proqramlaşdırma dilində Tiplər və Dəyişənlər

Proqramlaşdırma dillərində xüsusən də C#-da tip vacib bir anlayışdır. Tip dəyişənin hansı məlumatları özündə saxlaya biləcəyini, yaddaşdakı rolunu və həmin dəyişən üzərində hansı əməliyyatları yerinə yetirə biləcəyimizi təyin edən bir anlayışdır.

Tutaq ki, tam (integer) tipində bir dəyişən təyin etdik. Integerin tipi tam olduğu üçün bu dəyişən özündə ancaq tam ədədləri saxlaya biləcək. Riyaziyyatdan bizə məlum olan tam ədədlər üzərində toplama, çıxma, bölmə, vurma, kök alma, qüvvətə yüksəltmə və s. əməliyyatları proqramlaşdırmada da tam ədədlər üçün edə bilərik. Digər tərəfdən də tam tipə aid olan bir dəyişən ram yaddaşda 4 bayt = 32 bit yer ayırır. Dəyişəni necə istifadə edəcəyimizi və yaddaşın strukturunu həmin dəyişənin tipi müəyyən edir.

Dəyişən nədir?

Dəyişən özündə hansısa bir məlumatı saxlayır və həmin məlumat ram yaddaşda saxlanılır. Bu məlumat dəyişilə bilər və bir neçə dəfə istifadə oluna bilər. Məsələn, “int a = 2;” sətri yaddaşda 4 bayt = 32 bit həcmində, adı “a” olan bir sahə ayırır və həmin yaddaş sahəsinə 2 informasiyasını yazır.

## C# Dəyişən təyin etmə

Dəyişən nə deməkdir? Dəyişənlər verilənlərin özüdür formasında açıqlıq gətirsək heç də səhv etmiş olmarıq.Verilənlərimizə daha asan çatmaq və istifadə etməkçün çeşitli dəyişənlər müəyyən edirik. Bir dəyişən müəyyən etmək üçün lazımlı formul belədir:

Xüsusi açar söz + Data tipi (int, string, boolean, və s.) + dəyişən adı;

C# üzərində dəyişən təyin etmə əslində bu qədər bəsitdir. Bu örnəkdəki xüsusi açar söz-public, private, protected və s. –yəni hər hansı bir əlçatan təyinedicisidir, data tipi isə dəyişən üzərində hansı data tipinə aid data saxlayacaqsaq onu təyin edir, sondakı isə dəyişənimizə verdiyimiz addır. Göründüyü kimi bir dəyişən təyin etmək çox sadə bir prosesdir. Nümunə olaraq; object x;-Buradakı dəyişənimizin bir dəyəri olmadığı üçün yaddaşda hər hansı bir yer tutmaz.

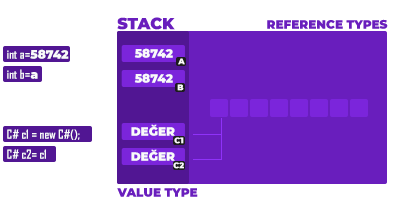
Dəyişən təyin etmə qaydaları:

* Dəyişən adında boşluq olmamalıdır. Bu tərz uzun dəyişən adlarını daha anlaşılan etmək istəsək aralarına **\_**, **$** kimi özəl işarələr əlavə edə bilərik.
* Dəyişən adı rəqəm və ya xüsusi bir xarakter ilə başlaya bilməz.
* Təsvir edəcəyimiz hər dəyişən üçün bir data tipi təyin edilməlidir
* C# dəyişənləri, bir dəyər mənimsətmədən “write” oluna bilməz: **object** dəyişən adı; Console.Write(dəyişən adı);
* Dəyişən olaraq static, void, class, və s. xüsusi əmrlərdən istifadə edilməz.

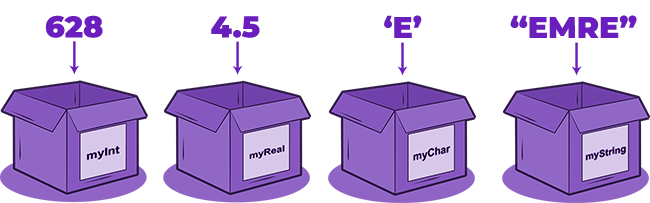
Qeyd: C#, Java-da olduğu kimi Case Sensitive, yəni böyük kiçik hərflərə həssas bir dildir. Yəni böyük kiçik hərlər bir-birindən fərqlidir. Bir dəyişən təyin edərkən istifadə etdiyimiz X ilə x fərqli nəticələr verir.

C# Data Tipləri

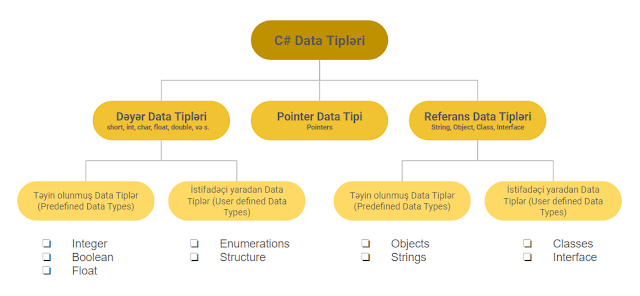
Obyekt yönümlü proqramlaşdırma dillərindəki stack və heap qavramı C# dili üçün də keçərli bir ünsürdür. Stack və heap qavramını başa düşə bilməyinizçün aşağıdakı təsvirə göz ata bilərsiniz.

C# Stack ve Heap sahəsi

İlkin olaraq dəyişənlərin stack yaddaş bölməsində və referans tiplər ilə yaradılan obyektlərin isə heap yaddaş bölməsində saxlanıldığını bilmək kifayətdir. C# dəyişən təyin edilməsi üçün lazımi formulu və təyin edilərkən diqqət edəcəyimiz mövzuları öyrəndiyimizə görə bu dəyişənlərə verəcəyimiz data tiplərinin öyrənməyə başlaya bilərik. Ümumi 3 ədəd data tipi vardır. Bu data tipləri value, referans və pointer data tipləri olaraq ayrılır. Dəyər (value) tipləri RAM-ın stack bölməsində saxlanarkən, Referans tiplər isə heap-də saxlanılır.

C# Data tipləri

**1.** Dəyər Data Tipləri (Value Data Types)  
**2.** Pointer Data Tipi (Pointer Data Type)  
**3.** Referans Data Tipləri (References Data Types)



Dəyər Data Tipləri

İlkin data tipləri olaraq da bilinən dəyər data tipləri, ümumi olaraq ədədi verilənləri saxlamaq üçün istifadə olunur. Dəyər tiplərində dəyişən özünə mənimsədilən dəyəri birbaşa özündə saxlayır. C# dili işarəli(signed) və işarəsiz(unsigned) literalların hər ikisini dəstəkləyir. Data Tiplərin yaddaş ölçüləri Əməliyyat Sistemindən asılı olaraq 32 bit yaxud 64 bit olaraq dəyişə bilər.

Bu ədədi verilənlər onluq və tam ədədlər olmaq üzrə 2 yerə ayrılır. Ədədi verilənlərin yanında tək xarakter saxlamaq üçün ya da true-false data tipləri üçün value data tipləri istifadə edirik. Əməliyyat Sistemi 32 bit olan data tiplərə baxaq.

Tam ədəd növünə aid data tipləri aşağıdakılardır:

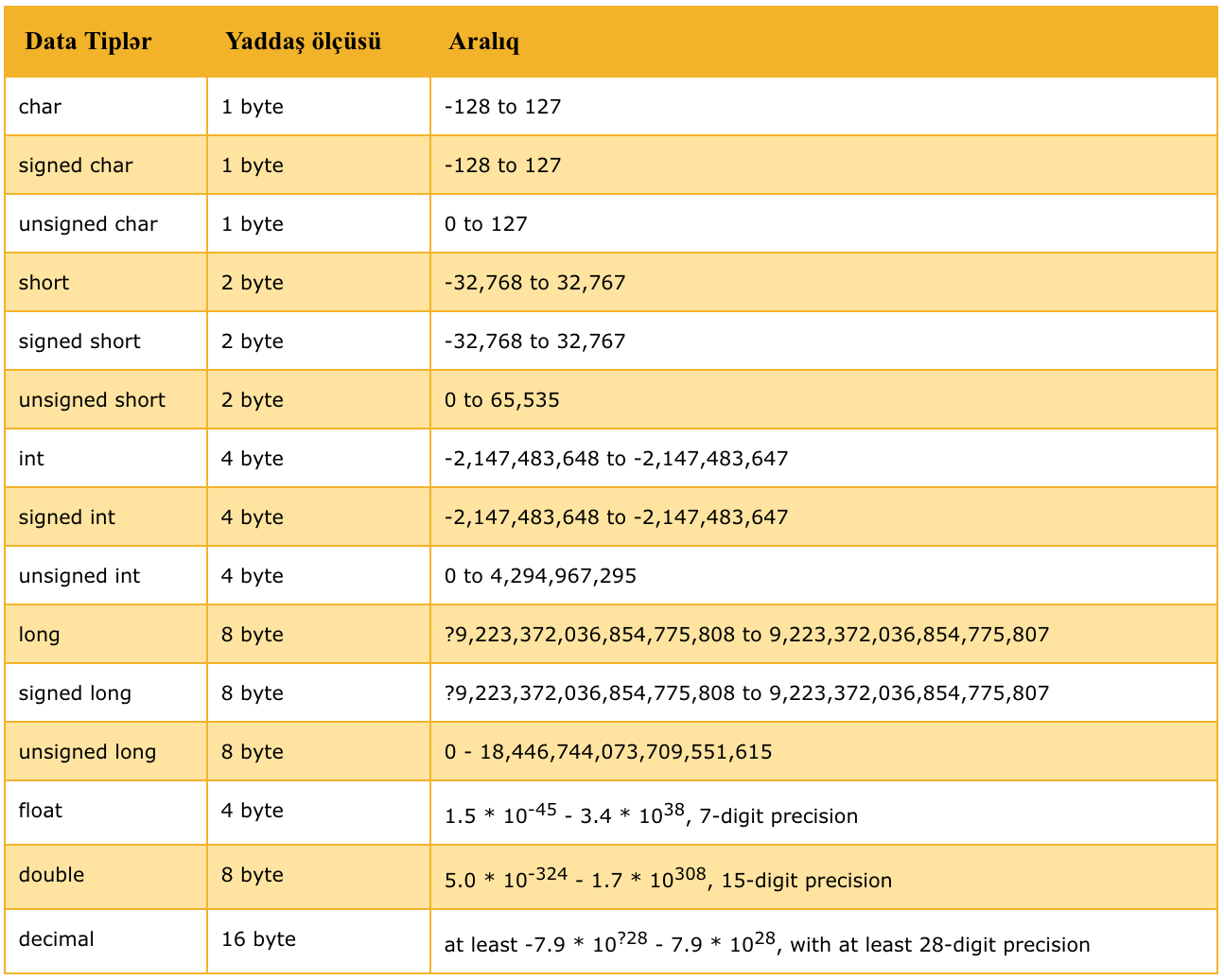
* **Byte:** Yaddaşda 1 byte yer tutar. 0’dan başlayaraq 255 dəyərinə qədər olan tam ədəd aralıöında dəyərlər ala bilər.
* **Sbyte:** Byte kimi yaddaşda tutduğu yer eynidir,-128 ilə 127 aralığında tam ədəd dəyərərini ala bilir değerlerini alabilir.
* **Short:** 2 Byte həcmindədir,-32768 ilə 32767 aralığında dəyər alır.
* **Ushort:** Sbyte kimi pozitiv (müsbət) dəyərlər alır. 0 ilə 65535 aralığında dəyər alır.
* **Integer (int):** Hər kəsin tez-tez istifadə elədiyi data tipidir. Yaddaşda 4 Byte yer tutur. -2³¹ ile 2³¹ -1 aralığında dəyər alır.
* **Uint:** Integer data tipinin müsbət dəyərlər alan halıdır. 0 ile 2×2³¹-1 aralığındakı dəyərləri alır.
* **Long:** Yaddaşda 8 Byte yer tutur. Integer data tipindən daha uzun tam ədəd dəyərlərini özündə saxlaya bilir. -2⁶³ ile 2⁶³-1 arasındaki değerler tanımlanabilir.
* **Ulong:** 0 ile 2×2⁶³ arasındakı tam ədədləri saxlayan data tipidir.

**Onluq kəsrlər növünə aid data tipləri aşağıdakılardır:**

* **Float:** Yaddaşda 4 Byte yer tutar. -3.4\*10³⁸ ile 3.4\*10³⁸ aralığında dəyərlər alır.
* **Double:** Yaddaşda 8 Byte yer tutar. -1.7\*10³⁰⁸ ile 1.7\*10³⁰⁸ aralığında dəyər alır.
* **Decmial:** Ondalıklı elemanları tutan veri tipidir. Vergüldən sonar 28-ə qədər çıxa bilən 16 byte-lıq kəsirli bir data tipidir.

Digər value tipləri:

* **Char:** Tək xarakter tipində dəyərlər alır. Təyin edilərkən yalnız bir rəqəm, işarə və ya hərf istifadə edə bilərik. Və dəyişəni mənimsədən zaman tək dırnaq içərisində yazılmalıdır.
* **Boolean:** Digər dəyər tiplərinin əksinə sadəcə 2 ədət dəyər alır. Bu dəyərlər true və ya false olaraq bilinir. Yaddaşda bir bit yer tutur. Logikal data tipi olaraq da bilinir.



**Yuxarıdan da göründüyü kimi bu data tiplərinin istifadə qaydası və istifadə yerləri də bir-birindən fərqlidir.**

Referans Data Tipləri

Referans Data Tiplərində dəyişən faktiki dəyəri özündə saxlamır. Bunun yerinə onlar ancaq dəyişənin dəyərinin saxlanıldığı yaddaşın ünvanını özündə saxlayırlar. Əgər hansısa dəyişəndə dəyər dəyişilsə avtomatik olaraq bu dəyişənə referans olan digər dəyişənlərdə də dəyişəcək. **Referans data tipləri RAM-ın heap sahəsində saxlanılır. Class, Array və İnterface olaraq təyin edə biləcəyimiz 3 referans tipi mövcuddur. Referans olaraq təyin edilən data tiplərində, ilkin data tiplərdən fərqli olaraq reallaşdırılacaq proseslər dəyişənin özü ilə həyat keçirilir. Təyin edici kimi də new açar sözündən istifadə edirik. String** referans tipindədir və yeganə bir xarakter, söz və ya cümlə kimi dəyərlərin saxlanmasında istifadə olunur. **Və təyin edilərkən string dəyişəninə dəyər mənimsədərkən cüt dırnaqdan istifadə olunur.**

Təyin olunmuş Referans Tipləri:

* Object – C# dilində bütün dəyişən tipləri bu tipdən törəyib.
* String – Unicode simvollarından ibarət mətn(string) məlumatları bu tipdə saxlanılır.

İstifadəçi tərəfindən yaradılan Referans Tiplər:

1. Classes
2. İnterfaces

## Default Dəyərlər

Hər data tipin default dəyəri vardır. Ədədi tipinki 0, bool tipinki false və char-ın default dəyəri '\0'  kimi bir dəyərdir.

int i = default(int); // 0

float f = default(float);// 0

decimal d = default(decimal);// 0

bool b = default(bool);// false

char c = default(char);// '\0'

// C# 7.1 onwards

int i = default; // 0

float f = default;// 0

decimal d = default;// 0

bool b = default;// false

char c = default;// '\0'