

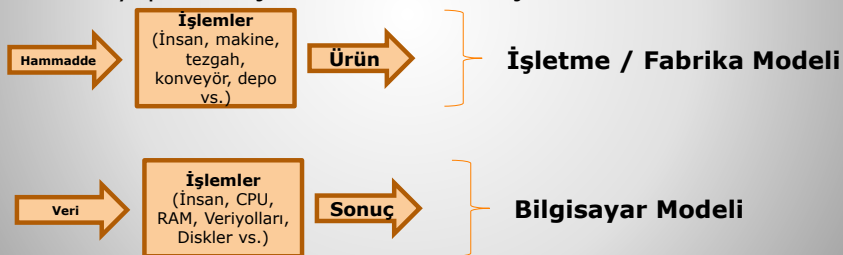
Algoritma ve Programlamaya Giriş Dersi

2. Hafta

Bilgisayar ve Program Kavramlar
Programlama Dilleri

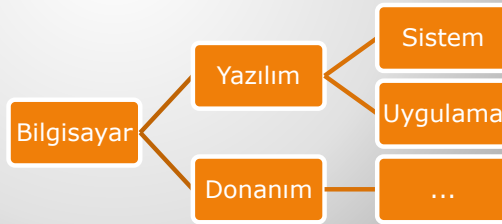
Matematiksel (Aritmetik) İşlemler
Karşılaştırma (Karar) İşlemleri
Mantıksal (Lojik) İşlemler

- **Bilgisayar;** günümüz bilgi toplumunda her alanda kullanılan vazgeçilmez bir **arac** haline gelmiştir.
- **İşlev** olarak bilgisayarın temel fonksiyonu; kendisine verilen bilgiler üzerinde tanımlanan işlemleri gerçekleştirdikten sonra istenilen formatlarda çıktı vermesidir.
- Görev yapısı bir işletmeninki ile örtüşmektedir.



Bilgisayar & Program

- Bilgisayarı **Donanım** ve **Yazılım** olarak iki bileşene ayırabiliriz.
- Donanım birimleri elektronik parçalardan oluşur. Dahili-Harici Birimler veya Giriş-Çıkış Birimleri gibi gruplarla incelenebilir.
- Yazılım bileşeni ise donanım bileşenlerini çeşitli komutlarla yönlendirerek; verilerin alınması, işlenmesi, depolanması ve çıktı olarak sunulması gibi işlevlerin gerçekleştirilebilmesini sağlar.
- Yazılım bileşeni **Sistem** ve **Uygulama** Yazılımları olarak ikiye ayrılır.
- (Programlama dili derleyicileri bazı kaynaklarda 3. bir yazılım türü olarak tanımlanır)



Bilgisayar & Program

SUBÜ - Sakarya MYO

2

Bilgisayar Programcılığı

- Bilgisayarlar programlandıkları çerçevede işlemleri gerçekleştirir.
- Hangi işlemi, nasıl gerçekleştireceği programcısı tarafından belirlenen bilgisayarın hata yapma durumu yoktur.
- Programın yapacağı işin etraflıca düşünülüp planlanması gerekir.
- Tüm ihtimal ve alternatifler dikkate alınmalıdır.
- Çok uzun ve karmaşık işlemler için hazırlanan programlar parçalara ayrılıp alt programlar halinde hazırlanabilir.

PROGRAMLAMA DİLLERİ

- Programların hazırlanması esnasında kullanılan notasyonlardır.
- Programcının yazdığı kodları makine diline (0-1) çevirirler.
- Çoğunlukla günlük kullanımdaki İngilizce kelimelerden oluşurlar.
- Böylece programcının kolayca kodlama yapabilmesini sağlarlar.







Bilgisayar & Program

SUBÜ - Sakarya MYO

3

Bilgisayar Programcılığı

- **Makine Dili:** 0 ve 1'lerden oluşan ifadeler işlemciye özgü olarak hazırlanır. Anlaşılması ve kodlaması zor, hata ihtimali yüksektir. 
- **Düşük Seviye Diller:** Makine dillerine yakın diller. Donanım bilgisi gerekir ve işlemcinin komut setlerine ulaşılabilir. Kaynakların kısıtlı, performansın önemli olduğu uygulamalarda tercih edilir. Geliştirilen uygulamaların boyutu düşüktür. (Assembly) 
- **Orta Seviye Diller:** Düşük seviye dillere göre daha anlaşılır ve esnek yapıdadırlar. Yapısal Programlama modelindedirler. (C dili) 
- **Yüksek Seviye Diller:** Olay ve nesne tabanlı programlama yapmaya imkan verirler. Kodlama işlemi konuşma diline yakın ve daha kolaydır. (VB, C#, Java, Python vb.) 
- Düşük seviyeli dillerden yüksek seviyeli dillere doğru donanım hakimiyeti azalırken, anlama ve kodlama işlemi kolaylaşır.
- Programlama dilleri farklı terim ve gramer yapısına sahip olabilirler.
- Algoritma kavramına hakim olan bir yazılım geliştirici herhangi bir programlama diline uyum sağlamakta çok zorlanmaz.

Programlama Dilleri

- Günümüzde programlarda farklı işlem türleri bulunsa da temel olarak işlemcinin görevi 3 grupta incelenebilir.
- **Matematiksel (Aritmetik) İşlemler**
- Bir program matematik işlemleri için tasarlanmış olmasa bile içinde birçok aritmetik fonksiyon barındırabilir.
- Matematiksel işlemler bilgisayar programlarında farklı sembollerle kodlanabilmektedirler. Açıklama için aşağıdaki tablo incelenebilir.

Matematikteki ifade	C dilindeki ifade	İşlem
.X veya hiçbir şey	*	çarpma
Mod	%	modüler bölme
/	/	bölme
+1	++	1 artırma (unary increment)
-1	--	1 eksiltme (unary decrement)
+	+	Toplama
-	-	çıkarma

Programlamada Kullanılan İşlemler

- Matematik işlemlerinde yer alan işlem önceliği kavramı bilgisayar dilinde yazılan formüllerde de geçerlidir. Bu sıra aşağıdaki gibidir.

1. Sayıların negatifliği
2. Parantezler
3. Özel matematiksel fonksiyonlar (Üs, karekök alma gibi)
4. Çarpma ve bölme
5. Toplama ve çıkarma

- Aynı önceliğe sahip işlemlerin sırası ise soldan sağa doğrudur.
- Birer artırma ve azaltma operatörlerinin işlem öncelikleri de aşağıdaki tablodan incelenebilir.

Unary operatörlerin ifadenin sağında yada solunda olması ile anlam değişik olur.

A++ (postincrement)	Önce A'yı kullan sonra içindeki değeri 1 artır.
++A (preincrement)	Önce A'yı 1 artır sonra bu artırılmış değeri kullan.
A-- (postdecrement)	Önce A'yı kullan sonra içindeki değeri 1 azalt.
--A (predecrement)	Önce A'yı 1 eksilt sonra bu eksiltilmiş değeri kullan.

Programlamada Kullanılan İşlemler

SUBÜ - Sakarya MYO

6

Bilgisayar Programcılığı

- Matematiksel ifadelerin bilgisayar dilinde kodlanmasına örnek olarak aşağıdaki tablo incelenebilir.

Matematiksel Yazılım	Bilgisayara Kodlanması
$a+b-c+2abc-7$	$a+b-c+2*a*b*c-7$
$a+b^2-c^3$	$a+b^2-c^3$
$a - \frac{b}{c} + 2ac - \frac{2}{a+b}$	$a-b/c+2*a*c-2/(a+b)$
$\sqrt{a+b} - \frac{2ab}{b^2-4ac}$	$(a+b)^{(1/2)}-2*a*b/(b^2-4*a*c)$
$\frac{a^2+b^2}{2ab}$	$(a^2+b^2)/(2*a*b)$

Örnek: $a=4, b=6, c=8, d=10$ değerleri için aşağıdaki ifadelerin sonuçlarını bulup inceleyiniz, daha sonra matematiksel ifade olarak yazınız.

$$c*d/(a*d)+b+c*d/a = 28$$

$$c*d/a*d+b+c*d/a = 226$$

$$c*d/a*d+(b+c)*d/a = 235$$

Programlamada Kullanılan İşlemler

SUBÜ - Sakarya MYO

7

Bilgisayar Programcılığı

• Karşılaştırma (Karar) İşlemleri

- Bilgisayarlar, işlemlerin gidişatı dallandığı zaman karar verebilmek için karşılaştırma işlemleri yapar.
- Eşit, eşit değil, büyük, küçük, büyük eşit, küçük eşit gibi kıyaslama metotlarıyla karar verilir.
- Bu işlemler özelliğine göre sayısal veya alfasayısal verilere uygulanabilir.

Matematikteki ifade	C dilindeki ifade	İşlem
>	>	Büyük
<	<	Küçük
≤	<=	Küçük veya eşit
≥	>=	Büyük veya eşit
=	==	Eşit (karşılaştırma)
≠	!=	Eşit değil

Programlamada Kullanılan İşlemler

SUBÜ - Sakarya MYO

8

Bilgisayar Programcılığı

• Mantıksal (Lojik) İşlemler

- Bilgisayarlar, temel mantık işlemlerini de gerçekleştirebilirler.
- Özellikle birden fazla karşılaştırma işlemi aynı anda kullanılacaksa araya mantıksal işlem eklemek gerekmektedir.

Matematikteki ifade	C dilindeki ifade	İşlem
^	&&	Ve (And)
∨		Veya (Or)
¬	!	Değil (not-invert)

Mantıksal Operatörlerin Doğruluk Tablosu

&& (AND)			(OR)			! (NOT)	
X	Y	Sonuç	X	Y	Sonuç	X	Sonuç
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Programlamada Kullanılan İşlemler

SUBÜ - Sakarya MYO

9

Bilgisayar Programcılığı

• Mantıksal (Lojik) İşlemlerin Elektriksel Gösterimi

VE (AND)		A	B	A VE B
		0	0	0
		0	1	0
		1	0	0
VEYA (OR)		A	B	A VEYA B
		0	0	0
		0	1	1
		1	0	1
DEĞİL (NOT)		A	DEĞİL A	
		0	1	
		1	0	
		1	0	

Programlamada Kullanılan İşlemler

SUBÜ - Sakarya MYO

10

Bilgisayar Programcılığı

$$X = A * B + C$$

- | | Opcode | Operand | |
|----|--------------------------------------|---|--|
| 1) | 000100000000000000000000000000000000 | 100000000000 (A'daki değeri reg. yükle) | |
| | Hex: 08 | 00 04 00 | |
| 2) | 001001000000000000000000000000000001 | 100000000001 (B'deki değeri reg. yükle) | |
| | Hex: 24 | 00 04 01 | |
| 3) | 001000110000000000000000000000000010 | 100000000010 (C'deki değeri reg. ile topla) | |
| | Hex: 23 | 00 04 02 | |
| 4) | 000100010000000000000000000000000011 | 100000000011 (Registerdeki değeri X'e taşı) | |
| | Hex: 11 | 00 04 03 | |

İki tabanındaki sayılar : 0, 1
 Ondalık sayılar : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
 Onaltılık (hex) sayılar : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F



Örnek Makine Dili Kodları

SUBÜ - Sakarya MYO

11

Bilgisayar Programcılığı

LOC	OBJECT	CODE	LINE	SOURCE	TEXT
00000006	00001	PORTB	EQU	06H	;PortB data register
00000010	00002	COUNT	EQU	10H	;GPR register
00000011	00003	MYREG	EQU	11H	
	00004				
	00005		ORG	0H	
0000 3000	00006		movlw	B'00000000'	
0001 0066	00007		tris	PORTB	
	00008				
0002 0190	00009		CLRF	COUNT	;COUNT = 0
0003 2???	00010	BACK	CALL	DISPLAY	
0004 2???	00011		GOTO	BACK	
	00012				
	00013	;increase value & send it to PORTB subroutine			
0005 0A90	00014	DISPLAY	INCF	COUNT,F	;count = count + 1
0006 0810	00015		MOVF	COUNT,W	
0007 0086	00016		MOVWF	PORTB	
0008 2???	00017		CALL	DELAY	
0009 0008	00018		RETURN		;return to caller
	00019				



Örnek Assembly Dili Kodları

SUBÜ - Sakarya MYO

12

Bilgisayar Programcılığı

```
// Include needed header files
#include <OC_const.h>           // OC Constants
#include <Common.h>             // Basic types
#include <stdio.h>              // Printf
#include <Data.h>               // Vectors and matrices
#include <OC_nag8.h>            // NAG function

int GetData()
{
    // Declare local variable holding error code
    int iErr;

    // *** Declare and initialize input parameters ***
    // *** passed into the NAG function nag_regsn_mult_linear. ***
    int nPts;                  // Input number of rows
    int nTdx;                  // Input number of columns
    matrix mX;                 // Input matrix containing data points
    vector vY;                 // Input vector containing data points
    vector vWT;                // Input vector containing weights

    // There are 10 rows of data in sample data
    nPts = 10;

    // There are 3 independent variables in data
    nTdx = 3;
}
```



Örnek C Dili Kodları

SUBÜ - Sakarya MYO

13

Bilgisayar Programcılığı

```

using System;  → .NET Framework Namespace

namespace HelloWorld  → Namespace Name
{
    0 references
    class Program  → Class Name
    {
        0 references
        static void Main(string[] args)  → Method
        {
            Console.WriteLine("Hello World!");  → Method to Write the Text to Console
            Console.WriteLine("Press Any Key to Exit.");
            Console.ReadLine();  → Method to Read an Input from Console
        }
    }
}

```

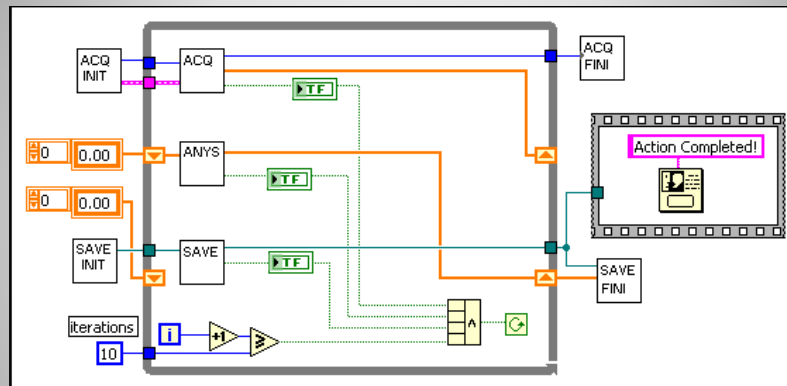
Keyword

Return Type

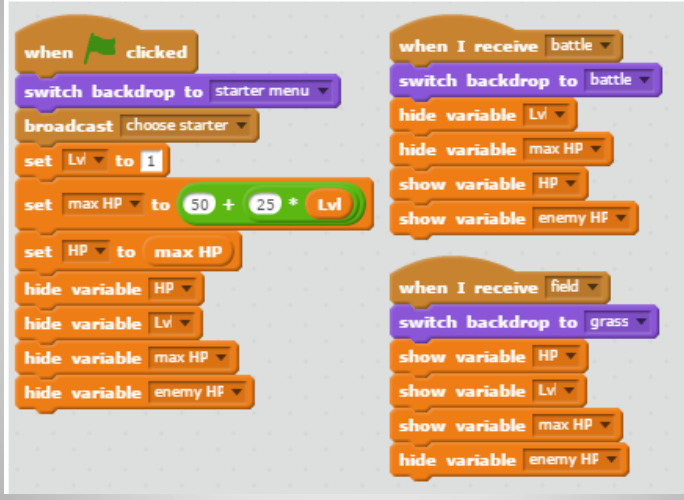
Method



Örnek C# Dili Kodları



Örnek Labview Dili Kodları (Block Diagram)



The image displays two Scratch scripts. The first script, triggered by a 'when clicked' event, switches the backdrop to 'starter menu', broadcasts 'choose starter', sets 'Lvl' to 1, calculates 'max HP' as $50 + 25 * \text{Lvl}$, sets 'HP' to 'max HP', and hides variables 'HP', 'Lvl', 'max HP', and 'enemy HP'. The second script, triggered by 'when I receive battle', switches the backdrop to 'battle', hides 'Lvl' and 'max HP', shows 'HP' and 'enemy HP'. A third script, triggered by 'when I receive field', switches the backdrop to 'grass' and shows 'HP', 'Lvl', 'max HP', and 'enemy HP'. A red arrow icon is at the bottom right of the code area.

Örnek Scratch Dili Kodları

SUBÜ - Sakarya MYO 16 Bilgisayar Programcılığı

- *Algoritma Geliştirme ve Programlamaya Giriş*, Dr. F. Vatansever, 7. Baskı, Seçkin Yay. 2009-Ankara
- *Algoritma ve C# Programlama*, Erhan ARI, 2. Baskı, Seçkin Yay. 2015-Ankara

Kaynaklar

SUBÜ - Sakarya MYO - Bilgisayar Programcılığı