

**T.C.
MİLLÎ EĞİTİM BAKANLIĞI**

BİLİŞİM TEKNOLOJİLERİ

KODLAMAYA HAZIRLIK
482BK0123

Ankara, 2011

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- PARA İLE SATILMAZ.

İÇİNDEKİLER

AÇIKLAMALAR	ii
GİRİŞ	1
ÖĞRENME FAALİYETİ-1	2
1. PROGRAMLAMANIN YAPI TAŞLARI	2
1.1. Bilgisayarın Çalışma Mantığı	2
1.1.1. Makine Dili	3
1.1.2. İkili Sayı Sistemi	4
1.1.3. Derleyici	7
1.2. Yazılım	7
1.2.1. Bir Yazılımda Olması Gereken Temel Özellikler	7
1.2.2. Yazılım Çeşitleri	8
1.3. Programlama Araçları	8
1.3.1. Programlama Dili	8
1.3.2. Programlama Dilleri Çeşitleri	9
UYGULAMA FAALİYETİ	11
ÖLÇME VE DEĞERLENDİRME	12
ÖĞRENME FAALİYETİ-2	13
2. ALGORİTMA VE AKIŞ DİYAGRAMI	13
2.1. Algoritma Yazım Aşamaları	13
2.1.1. Problemi Tanımlama	13
2.1.2. Problemi Geliştirme	14
2.1.3. Sisteme Uyumluluğunu Tespit Etme (Girdi-Çıktı Belirleme)	14
2.1.4. Çözümü Kâğıt Üzerinde Gösterme (Prototip Oluşturma)	14
2.1.5. Çözümü Deneme	14
2.1.6. Çözümü Geliştirme	14
2.1.7. Oluşabilecek Hatalar	14
2.2. Akış Diyagramı	17
2.2.1. Akış Diyagramı Şekilleri	18
UYGULAMA FAALİYETİ	21
ÖLÇME VE DEĞERLENDİRME	22
ÖĞRENME FAALİYETİ-3	23
3. PROGRAMLAMA YAZILIMI	23
3.1 Programlama Yazılımının Arayüzü	23
3.1.1 Başlangıç Sayfası(Start Page)	25
3.1.2 Solution Explorer Paneli	27
UYGULAMA FAALİYETİ	30
ÖLÇME VE DEĞERLENDİRME	31
MODÜL DEĞERLENDİRME	32
CEVAP ANAHTARLARI	34
KAYNAKÇA	35

AÇIKLAMALAR

KOD	482BK0123
ALAN	Bilişim Teknolojileri
DAL/MESLEK	Alan Ortak
MODÜLÜN ADI	Kodlamaya Hazırlık
MODÜLÜN TANIMI	Bir program yazılması için gerekli olan hazırlıklara ait bilgilerin verildiği öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	
YETERLİK	Kodlama öncesi hazırlıkları yapmak
MODÜLÜN AMACI	<p>Genel Amaç Bu modül ile gerekli ortam sağlandığında programlamanın kodlama öncesi hazırlıklarını yapabileceksiniz.</p> <p>Amaçlar</p> <ol style="list-style-type: none">1. Program yazmaya hazırlık yapabileceksiniz.2. Algoritma ve akış diyagramı hazırlayabileceksiniz.3. Programlama dili yazılımı ile çalışabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	<p>Ortam: Bilgisayar laboratuvarı</p> <p>Donanım: Kâğıt, kalem, akış diyagramları ile ilgili panolar, bilgisayar, lisanslı programlama yazılımı</p>
ÖLÇME VE DEĞERLENDİRME	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.

GİRİŞ

Sevgili Öğrenci,

Okul yaşantınızda öğreneceğiniz her konu, yaptığınız her uygulama ve tamamladığınız her modül bilgi dağarcığınızı geliştirecek ve ileride atılacağınız iş yaşantınızda size başarı olarak geri dönecektir. Eğitim sürecinde daha özverili çalışır ve çalışma disiplini kazanırsanız; başarılı olmamanız için hiçbir neden yoktur.

Bu modül sonunda; istediğiniz herhangi bir problemin bilgisayardaki çözüm aşamalarını yazabileceksiniz. Çözüm aşamalarını belirledikten sonra istediğiniz bir programlama dilini kullanarak gerekli yazılımı oluşturulabileceksiniz. Akış diyagramları ile yazdığınız algoritmayı şekillerle daha kolay, anlaşılır kılacaksınız.

ÖĞRENME FAALİYETİ-1

AMAÇ

Bu öğrenme faaliyetinde, program yazmaya hazırlık yapabileceksiniz.

ARAŞTIRMA

- Günlük hayatta bir olayın insan üzerinde girdi, işleniş ve çıktı durumlarını maddeler hâlinde listeleyiniz.
- Çevrenizde sayı sistemlerinin hangi alanlarda nasıl kullanıldıklarını araştırınız.

1. PROGRAMLAMANIN YAPI TAŞLARI

1.1. Bilgisayarın Çalışma Mantığı

Bilgisayar, kullanıcıdan aldığı verilerle mantıksal ve aritmetiksel işlemleri yapan yaptığı işlemlerin sonucunu saklayabilen, sakladığı bilgilere istenildiğinde ulaşılabilen elektronik bir makinedir. Bu işlemleri yaparken veriler girilir ve işlenir. Ayrıca, istenildiğinde yapılan işlemler depolanabilir ve çıkışı alınabilir.

Giriş: Kişi tarafından veya bilgisayar tarafından sağlanan verilerdir. Bu veriler, sayılar, harfler, sözcükler, ses sinyalleri ve komutlardır. Veriler giriş birimleri tarafından toplanır.

İşlem: Veriler insanların amaçları doğrultusunda, programın yetenekleri ölçüsünde işlem basamaklarından geçer.

Bellek: Verilerin depolandığı yerdir. Giriş yapılan ve işlenen veriler bellekte depolanır.

Çıkış: Bilgisayar tarafından işlem basamaklarından geçirilerek üretilen yazı, resim, tablo, müzik, grafik, görüntü, vb.nin ekrandan ya da yazıcı, hoparlör gibi değişik çıkış birimlerinden alınmasıdır.

Bilgisayarın nasıl çalıştığını öğrenmek için onun bilgileri nasıl kullandığını anlamak gerekir. Harfler ve rakamlar bilgisayarda kodlar şeklinde ifade edildikten sonra kullanılır. Bilgisayarlarda kodlar elektrik olarak voltajın olup olmaması ile ifade edilir. Voltaj var, lamba yanıyor 1; voltaj yok, lamba yanmıyorsa 0 kodlarını alır. İki durumlu olan bu kodlamaya "ikilik sistem" denir. Bilgisayara tuşlardan verilen her bilgi 1 ve 0 kodlarına çevrilir. Her 0 ve 1, bit olarak; sekiz bitlik grup ise byte olarak tarif edilir. Bilgisayar, işlemlerini ikilik sayı sistemi ile yapar. İşlemler çok sade ve basit olmakla beraber çok hızlıdır.

Bilgisayarlar sadece sayıları saymakla kalmayıp karar da verebilir. Bu kararlar, Boolean matematiği denilen mantık kaidelerine göredir. Çeşitli şartlara göre bilgisayar EVET, HAYIR, VE, VEYA, DEĞİL gibi kararlar alabilir. Mesela; evi taşımak için bir kamyon VE bir şoföre ihtiyaç vardır. Bu kamyon bir dar köprüden geçmek zorundaysa kamyon geniş VEYA yüksekse köprüye çarpar. Taşınacak ev boş DEĞİL ise taşıma işlemi gecikecektir. Burada VE, VEYA, DEĞİL kararları verilmiştir.



Şekil 1.1: İnsan ve bilgisayar arası benzerlik

1.1.1. Makine Dili

Makine dili mikroişlemci ya da mikrodenetleyici gibi komut işleme yeteneğine sahip entegrelerin işleyebilecekleri komutlardan ve buna uygun söz diziminden oluşan dile verilen addır. Makine dili, işlemcinin verilen komutlar doğrultusunda çalıştırılmasını sağlayan ve işlemci mimarisine göre değişen en alt seviyedeki programlama dilidir. Bu dil sadece 0 ve 1 ikililerinin anlamlı kombinasyonlarından meydana gelmektedir. Bu nedenle, makine dilinin anlaşılması çok güçtür.

Sonuç olarak bilgisayar programı hangi dilde olursa olsun formatı makine diline çevrilmesi gereklidir. Her emrin yerine getirilmesine 'Instruction Cycle' denir ve 4 ana kısımdan oluşur;

Fetch: Bir sonraki emri hafızadan getir.
Decode: Emrin ne demek istediğini çöz.
Execute: Emir yerine getirilir, işlem yapılır.
Store: Sonucu sakla.

Kısaca özetleyecek olursak;

- Bilgisayarın doğal dilidir.
- Bilgisayarın donanımsal tasarımına bağlıdır.
- Makine dili “010101001”lerden oluşan komutlardır ve bunlar doğrudan makineye hitap eder.
- Makine dili işlemci türüne özel olarak yazılmaktadır. Dolayısıyla, bir işlemci türü için makine diliyle yazılmış program, diğer işlemciler için uygun olmamaktadır.

1.1.2. İkili Sayı Sistemi

Bilgisayar ortamında dört çeşit sayı sistemi kullanılmaktadır. Bunlar;

- İkili sayı sistemi(Binary)
- Sekizlik sayı sistemi(Oktal)
- Onluk sayı sistemi(Desimal)
- Onaltılık sayı sistemi(Heksadesimal)

Sekizlik sayı sistemi (oktal): Bu sayı sisteminde sadece 0,1,2,3,4,5,6,7 rakamları kullanılır. Bu sayı sisteminin tabanı 8’dir.

ÖRNEK: $(25)_8$, $(376)_8$, $(27510)_8$

Onluk sayı sistemi (desimal): Bu sayı sistemi günlük hayatta kullandığımız sayılardan oluşur. Yani 0,1,2,3,4,5,6,7,8,9 sayılarını kapsar. Bu sayı sisteminin tabanı 10’dur.

ÖRNEK: $(10)_{10}$, $(9999)_{10}$, $(820918)_{10}$

Onaltılık sayı sistemi (heksadesimal): Bu sayı sisteminde 16 tane rakam bulunmaktadır. 1,2,3,4,5,6,7,8,9,A,B,C,D,E,F rakamlarından oluşur. Sayı sisteminin tabanı 16’dır. 16’lık sayı sisteminde her bir harf bir rakamı temsil eder. Bunlar;

A = 10, B=11, C=12, D=13, E=14, F=15 ‘tir. Bilgisayar sisteminde 1 ve 0’ ın yan yana yazılmasıyla 10 sayısı aynı şekilde ifade edildiğinden karmaşıklık olmasın diye böyle bir çözüm üretilmiştir.

ÖRNEK: $(1029)_{16}$, $(AFDC)_{16}$, $(290A3DC)_{16}$

İkili sayı sistemi(Binary): İkili sayı sistemi 2 tabanını kullanır. 0 ve 1 olmak üzere iki rakamı vardır. Bu sayılar bilgisayarda akımın geçmesi veya geçmemesi esasına göre belirlenir.

ÖRNEK: $(001010)_2$, $(111011)_2$, $(10)_2$

➤ **2'li sistemden 10'lu sisteme dönüştürme:**

ÖRNEK:

$$(101101)_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$= 32 + 0 + 8 + 4 + 0 + 1 = (45)_{10}$$

➤ **10'lu sistemden 2'li sisteme dönüştürme:**

Bu işlemi yapabilmek için 10'lu sistemde verilen sayı sürekli olarak ikiye bölünmektedir. Bölmelerin kalanları bize o sayının ikili sistemdeki karşılığını vermektedir.

ÖRNEK: 10'lu tabandaki 53 sayısını ikili tabana çeviriniz?

Bölünen	Bölüm	Kalan
Bölüm		
53	2	1
26	2	0
13	2	1
6	2	0
3	2	1
1		1

$(53)_{10} = (110101)_2$

1 1 0 1 0 1

53	64	32	16	8	4	2	1
- 32	0	1	1	0	1	0	1
21							
- 16							
5							
- 4							
1							
- 1							
0							

$(110101)_2$

➤ **2’li sistemden 8’li sisteme dönüştürme:**

$(11001111011101)_2$ sayısını sekizli sayı sistemine dönüştürelim. Üçerli kümelerle ayırma ve eksik bitleri tamamlama sonucunda,

$$\begin{array}{cccccc} 011 & 001 & 111 & 011 & 101 \\ \hline 3 & 1 & 7 & 3 & 5 \end{array}$$

Her bir kümenin temsil ettiği sekizli sayı yazılırsa $(11001111011101)_2 = (31735)_8$ eşitliği elde edilir.

➤ **8’li sistemden 2’lik sisteme dönüştürme**

$(567)_8$ sayısını ikilik sisteme çevirelim.

$$\begin{array}{c|c|c} 5 & 6 & 7_8 \\ \hline & & \end{array} \quad \begin{array}{c|c|c} 5 & 6 & 7_8 \\ \hline 101 & 110 & 111_2 \end{array} \quad (567)_8 = (101110111)_2$$

➤ **2’li sistemden 16’lı sisteme dönüştürme**

$(11001111011101)_2$ sayısını onaltılı sayı sistemine dönüştürelim. Dörderli kümelerle ayırma ve eksik bitleri tamamlama sonucunda,

$$\begin{array}{cccc} 0011 & 0011 & 1101 & 1101 \\ \hline 3 & 3 & D & D \end{array}$$

Her bir kümenin temsil ettiği onaltılı sayı yazılırsa $(11001111011101)_2 = (33DD)_{16}$ eşitliği elde edilir.

➤ **16’lı sistemden 2’lik sisteme dönüştürme**

$(3A8F)_{16}$ sayısını ikilik sisteme çevirelim.

$$\begin{array}{c|c|c|c} 3 & A & 8 & F_{16} \\ \hline & & & \end{array} \quad \begin{array}{c|c|c|c} 3 & A & 8 & F_{16} \\ \hline 0011 & 1010 & 1000 & 1111_2 \end{array} \quad (3A8F)_{16} = (0011101010001111)_2$$

1.1.3. Derleyici

Programcılıkta, bir programlama dilinde yazılmış olan kaynak kodunu başka bir dile(genellikle makine koduna) çeviren yazılımlara derleyici denir.

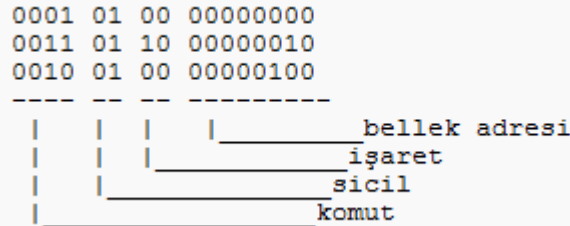
Örneğin, şu satırı bir programın kaynak kodunda (programın okunabilir hali) düşünelim:

$X = 2 + Y$

Altta assembly'de yazılmış satırlar, aynı programın derlenmiş hâlidir:

```
LOAD A [0]      ;; belleğin 0 adresindeki veriyi A siciline yükle
ADD  A BX       ;; A siciline BX sicilindeki adreste bulunan veriyi ekle
STOR A [100]    ;; sonucu 100 adresine yaz
```

Bu örnekte çevirinin hedefi, programcının anladığı kaynak kodundan işlemcinin anladığı 0 ile 1'den oluşan makine dili kodunu üretmektir (LOAD, ADD ve STOR komutları 0001, 0011 ve 0010 olarak yorumlanır.).



1.2. Yazılım

Yazılım, elektronik aygıtların belirli bir işi yapmasını sağlayan programların tümüne verilen isimdir.

Bir başka deyişle var olan bir problemi çözmek amacıyla bilgisayar dili kullanılarak oluşturulmuş anlamlı anlatımlar bütünüdür.

Kelime işlemci programları, bilgisayarınız açıldığında CD sürücünüzü, sabit sürücüler, RAM'i tanıyan BIOS; işletim sistemi, web tarayıcınız, virüslerin kendileri, antivirüs programları hep birer yazılımdır.

1.2.1. Bir Yazılımda Olması Gereken Temel Özellikler

Doğruluk: Yazılımın belirtilmiş ihtiyaçlarını karşılamasıdır.

Güvenilirlik: Gerekli işlevi ne hassaslıkla yerine getireceği beklentisidir.

Verimlilik: İşlevin gerçekleştirilmesi için kullanılması gereken bilgisayar kaynakları ve kod miktarıdır.

Güvenlik(Bütünlük): Yazılım ve bilgilerine, istenmeyen insanlarca ulaşımın ne derece engellenebildiğidir.

Kullanılabilirlik: Programın öğrenilmesi, çalıştırılması, girdi hazırlama ve çıktı yorumlama işlemlerinin kolaylık derecesidir.

Hata bulma kolaylığı: Hatanın yerini bulma ve düzeltme kolaylığıdır.

Esneklik: Yazılımda değişiklik yapma kolaylığıdır.

Taşınabilirlik: Programın farklı donanımlarda ve yazılım sistemi ortamlarında kullanılmasıdır.

Tekrar kullanılabilirlik: Yazılım tamamının ya da bir bölümünün farklı bir uygulamada kullanılabilmesidir.

Birlikte çalışabilirlik: Bir yazılım sisteminin diğerleri ile bağlantı sağlaması kolaylığıdır.

1.2.2. Yazılım Çeşitleri

Bilgisayar yazılımları genel olarak 2 ana grupta incelenebilir.

- **Sistem yazılımları (system software) :** Bilgisayarı yöneten, denetleyen, kontrol eden yazılımlardır.

Örneğin: Linux, Pardus, Windows vb.

- **Uygulama yazılımları (application software) :** Belli bir alana ve uygulamaya ilişkin olarak kullanıcılar için geliştirilmiş yazılımlardır.

Örneğin: Kelime işlemci programları, web hazırlama araçları, programlama araçları vb.

Bütün sistem programları içinde en temel yazılım işletim sistemidir ki, bilgisayarın bütün donanım ve yazılım kaynaklarını kontrol ettiği gibi kullanıcılara ait uygulama yazılımlarının da çalıştırılmalarını ve denetlenmelerini sağlar.

1.3. Programlama Araçları

1.3.1. Programlama Dili

İster genel ister özel amaçlı olsun tüm uygulama ve sistem yazılımları programlama dilleriyle yazılır. Bir programlama dili, insanların bilgisayara çeşitli işlemler yaptırmasına imkân veren her türlü sembol, karakter ve kurallar grubudur. Programlama dilleri insanlarla bilgisayarlar arasında tercümanlık görevi yapar. Programlama dilleri, bilgisayara neyi, ne zaman, nasıl yapacağını belirten deyim ve komutlar içerir.

Bir programlama dili şunlardan oluşur.

- **Genel komutlar:** Programlama dilinin anlayacağı komutlardır.
- **Gelişmiş komutlar:** Genel komutları kullanarak oluşturulmuş komutlardır.
- **API komutları:** İşletim sisteminin sunduğu özellikleri kullanan komutlardır.
- **Derleyici komutları:** Komut içinde çalışmayıp derleme esnasında alınan bilgilere göre derleme yapılmasını sağlar.
- **Aktif nesneler:** 'Buton, Menü, Gösterge çubuğu ve Tabpanel' gibi bileşenlerin genel adıdır.

1.3.2. Programlama Dilleri Çeşitleri

Bir programlama dili ya insan ya da makine anlayışına yakındır. İnsan anlayışına daha yakın programlara dillerine yüksek seviyeli programlama dilleri, makineye yakın olanlara ise düşük seviyeli programla dilleri denir.

- Yüksek seviye programlama ile yazılan projelerin kaynak kodları kısa, derlenmiş hâlleri ise uzun olur. Çalışma hızları ise yavaştır.
- Alçak seviye programlama ile yazılan projelerin kaynak kodları uzun, derlenmiş hâlleri ise kısadır olur. Çalışma hızları ise en yüksek seviyededir.

Programlama dillerini seviyelerine göre 5 ana gruba ayırabiliriz:

- **Çok yüksek seviyeli diller ya da görsel diller**

Access, Foxpro, Paradox, Xbase, Visual Basic, Oracle Forms

- **Yüksek seviyeli diller (Bunlara algoritmik diller de denir.)**

Fortran, Pascal, Basic, Cobol

- **Orta seviyeli diller**

C, C++(C Plus) , C#(C Sharp) Orta seviyeli diller daha az kayıpla makine diline çevrilebildiğinden daha hızlı çalışır.

- **Alçak seviyeli programlama dilleri**

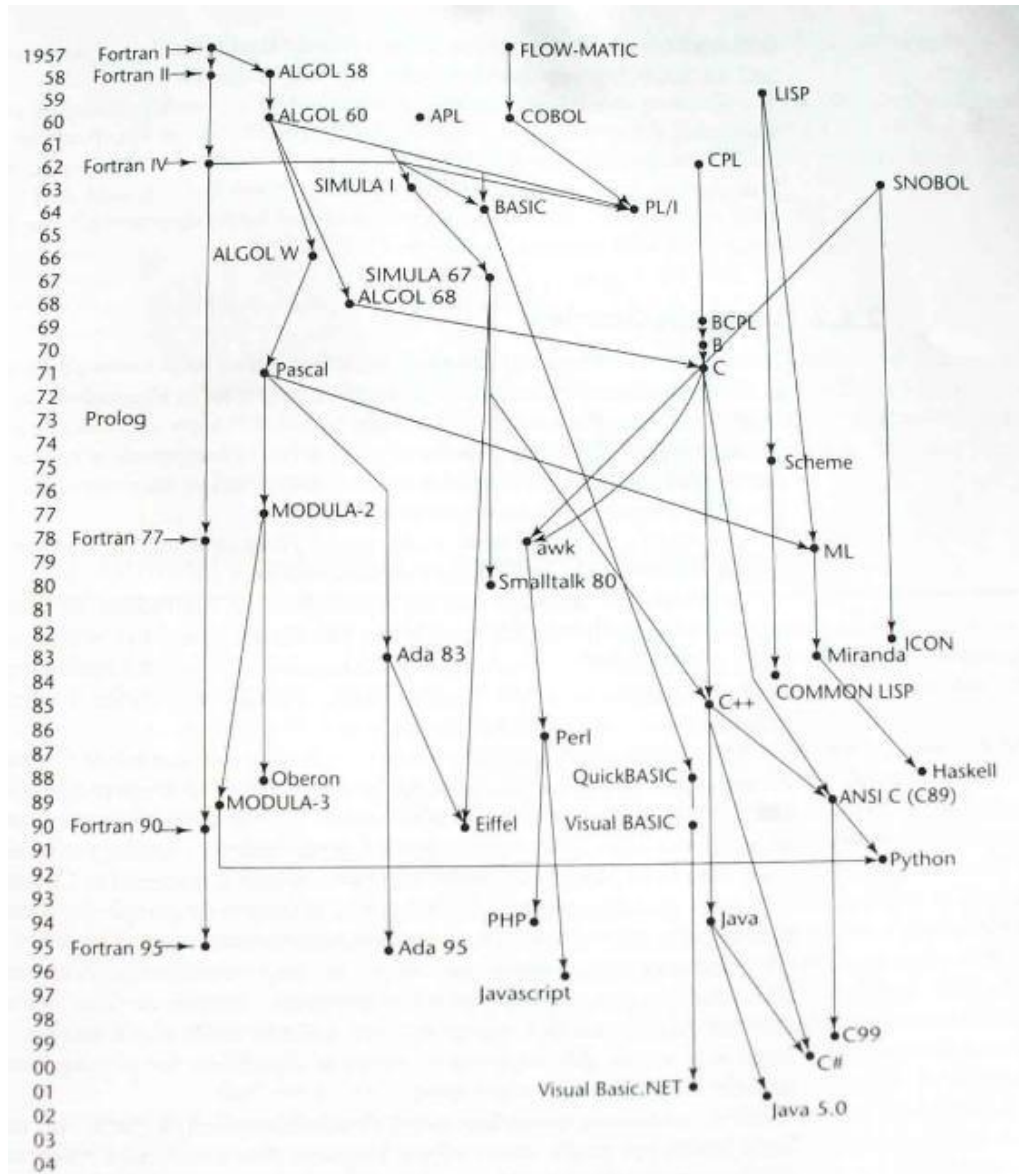
Sembolik makine dili (Assembler).

- **Makine dili**

En aşağı seviyeli programlama dilidir (Saf makine dili tamamen 1 ve 0 lardan oluşuyor.) .

Kuşak	Programlama Dili	Periyod
1	Makine dili	1940 – 1950 arası
2	Assembly dili	1950’li yıllardan itibaren
3	Yüksek seviyeli diller	1960’lı yıllardan itibaren
4	Çok yüksek seviyeli diller	1970’li yıllardan itibaren
5	Yapay zekaya yönelik diller	1980’li yıllardan itibaren

Tablo 1.1: Programlama dillerinin tarihi gelişimi



Şekil 1.2: Programlama dili soy ağacı

UYGULAMA FAALİYETİ

Sayı sistemleri arasında dönüşümler yapınız.

İşlem Basamakları	Öneriler
➤ 2'lik sayı sisteminden 8'lik sayı sistemine dönüşüm yapınız.	➤ Sayıları üçerli gruplara bölününüz. ➤ Eksik kalan bitleri tamamlayınız. ➤ Her grubun temsil ettiği 8'li sayıyı yanyana yazarak sonucu bulunuz.
➤ 8'lik sayı sisteminden 2'lik sayı sistemine dönüşüm yapınız.	➤ 8'lik her bir sayıyı kendi içinde 2'lik sisteme dönüştürünüz. ➤ Elde eden sayıları yanyana yazarak sonucu bulunuz.
➤ 8'lik sayı sisteminden 16'lık sayı sistemine dönüşüm yapınız.	➤ 8'lik sayıyı önce 2'lik sayı sistemine dönüştürünüz. ➤ Elde edilen 2'lik sayı sistemindeki sayıyı 16'lık sayı sistemine dönüştürerek sonucu bulunuz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümleleri dikkatlice okuyarak boş bırakılan yerlere doğru sözcüğü yazınız.

1., elektronik aygıtların belirli bir işi yapmasını sağlayan programların tümüne verilen isimdir.

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

2. $(01110101)_2$ 2'lik tabandaki sayının 10'luk tabanındaki karşılığı aşağıdakilerden hangisidir?
- A) 121 B) 115 C) 117 D) 119
3. 359 10'luk tabandaki sayının 2'lik tabandaki karşılığı aşağıdakilerden hangisidir?
- A) 101100101 B) 101000111
C) 100101011 D) 101100111
4. $(A55C)_{16}$ sayısının 2'lik tabandaki karşılığı aşağıdakilerden hangisidir?
- A) 1101001100111010 B) 1010010101011100
C) 1101010101011100 D) 1010001100111100
5. Aşağıdakilerden hangisi bir yazılımda olması gereken temel özellikler arasında değildir?
- A) Doğruluk B) Esneklik
C) Taşınabilirlik D) Problem çözme

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

Bu öğrenme faaliyeti sonunda algoritma ve akış diyagramı hazırlayabileceksiniz.

ARAŞTIRMA

- Günlük hayatta karşılaştığınız problemleri nasıl çözümlediğinizi dikkatlice gözlemleyiniz. Örneğin; okula geliş ve gidiş yolunda hangi taşıt araçları sıra ile kullandığınızı, hava ve yol durumuna göre çözüm olarak neler yaptığınızı anlatınız.
- Problemleri çözerken işlemleri belli bir sıra ile mi yapıyorsunuz yoksa çözüm için gerekenleri rastgele mi uyguluyorsunuz, buna dikkat ederek arkadaşlarınızla paylaşınız.
- Bir yemek tarifini araştırınız. Yemek yapımında izlenen adımları bir kâğıda yazınız. Alternatif durumları da belirtiniz.

2. ALGORİTMA VE AKIŞ DİYAGRAMI

2.1. Algoritma Yazım Aşamaları

Algoritma*, verilen herhangi bir sorunun çözümüne ulaşmak için uygulanması gerekli adımların hiçbir yoruma yer vermeksizin açık, düzenli ve sıralı bir şekilde söz ve yazı ile ifadesidir.

2.1.1. Problemi Tanımlama

Her şeyden önce çözülecek problem tam olarak anlaşılmalıdır. Bu adımda yapılacak en ufak bir hata daha sonraki adımların yeni baştan yapılmasını gerektirebilir. Problemin tanımı yapılırken var olan bilgiler, anlamları ve birbirleri ile ilişkileri tanımlanmalıdır. Daha sonra istenilenler belirlenmeli ve bunların var olan bilgiler ile ilişkileri öğrenilmelidir. Son olarak yapılacak işlemler belirlenir. Mümkün ise örnek veriler ile elde edilen sonuçlar değerlendirilmelidir.

* Algoritma kelimesi, Özbekistan'ın Harezmi, bugünkü Türkmenistan'ın Khiva kentinde doğmuş Ebu Abdullah Muhammed bin Musa el Harezmi isimli Türk matematikçinin adından gelir. Bu alim 9. yüzyılda cebir alanındaki algoritmik çalışmalarını kitaba dökerek matematiğe çok büyük bir katkı sağlamıştır.

2.1.2. Problemi Geliştirme

Problem tanımını tam olarak yaptıktan sonra çözüm için yol aramak gerekir. Genellikle bir problemin birden fazla çözüm yolu olabilir. Bunlardan en uygunu seçilmeye çalışılır. Problem ne kadar karışık olursa olsun, alt birimlere bölünür. Her birimin çözümü ayrı ayrı yapılır. Bu yapılırken birimler arası ilişki sürekli olarak korunur.

2.1.3. Sisteme Uyumluluğunu Tespit Etme (Girdi-Çıktı Belirleme)

Sonuçların dış ortama, dolayısıyla insana aktarımı düzgün bir biçimde yapılmalıdır. Programcı program çıktısı olarak almak istediği dökümün biçimini tasarlar. Bir döküm biçimi tasarlanırken anlaşılır ve kullanılabilir olmasına özen gösterilmelidir. Genellikle programa, çözdüğü soruna ilişkin bazı verilerin dışarıdan verilmesi gerekir.

2.1.4. Çözümü Kâğıt Üzerinde Gösterme (Prototip Oluşturma)

Algoritma geliştirildikten sonra daha iyi anlaşılabilir olması ve programlama dillerine aktarımı daha kolay olması nedeniyle, prototip hâline getirilir. Böylece sorunun çözüm basamakları, birbirleri ile ilişkileri ve bilgi akışı daha kolay görülebilir ve yanlışlıklar düzeltilebilir.

2.1.5. Çözümü Deneme

Algoritma yazıldıktan sonra sonuçları daha önceden bilinen veriler girilerek eldeki sonuçlarla çıkan sonuçlar karşılaştırılır. Algoritmanın doğruluğu kontrol edilir.

2.1.6. Çözümü Geliştirme

Bu aşamada doğruluğu kontrol edilmiş çözümün tekrar edilen blokların daha kısa yoldan sonuca ulaşılması durumları gözden geçirilir.

2.1.7. Oluşabilecek Hatalar

Hazırlanan algoritmalarda en sık karşılaşılan hatalar mantık hatalarından oluşmaktadır. Mantık hataları ise problemin çözüm adımları belirlenirken yapılan yanlışlıklardan kaynaklanan hatalardır.

Algoritma örnekleri

Örnek: Örneğimiz bir öğrencinin evden çıkıp okula giderken izleyeceği yolu ve okula girişinde ilk yapacaklarını tanımlamaktadır.

Çözüm:

- Evden dışarıya çık.
- Otobüs durağına yürü.

- Durakta gideceğin yöndeki otobüsü bekle .
- Otobüsün geldiğinde otobüse bin .
- Biletini bilet kumbarasına at.
- İneceğin yere yakınlaştığında arkaya yürü.
- İneceğini belirten ikaz lambasına bas .
- Otobüs durunca in.
- Okula doğru yürü.
- Okul giriş kapısından içeriye gir.
- Sınıf arkadaşlarınla selamlaş.
- Sırana otur.
- Öğretmenin gelmesini bekle.

Örnek: İki sayıyı toplamak için gerekli programa ait algoritmanın oluşturulması

Çözüm:

- A1 : Birinci sayıyı gir.
- A2 : İkinci sayıyı gir.
- A3 : İki sayının toplamını yap .
- A4 : Toplamın değerini yaz.
- A5 : Bitir.

Örnek: İki sayının ortalamasını bulmak için gerekli programa ait algoritmanın oluşturulması

Çözüm:

- A1 :Birinci sayıyı gir.
- A2 :İkinci sayıyı gir.
- A3 :İki sayının toplamını yap.
- A4 :Toplamı ikiye böl.
- A5 :Bölümün sonucunu yaz.
- A6 :Bitir.

Sözcüklerin ortaya çıkaracağı yanlış anlamaların ortadan kaldırmak amacıyla semboller ve matematik dilini gerektiren bazı kısaltmalar kullanmak daha uygun olacaktır.

Bu açıklamaya göre yukardaki örnekleri tekrar yapalım.

Örnek: İki sayıyı toplamak için gerekli programa ait algoritmanın oluşturulması

Çözüm:

- Birinci sayı için X
- İkinci sayı için Y
- Toplam adı için Z kullanılırsa

A1 :X değerini gir.
A2 :Y değerini gir.
A3 : $Z = X + Y$
A4 :Z'yi yaz.
A5 :Bitir.

Örnek: İki sayının ortalamasını bulmak için gerekli programa ait algoritmanın oluşturulması

Çözüm:

Birinci sayı için X
İkinci sayı için Y
Toplam adı için Z
Ortalama adı için Ort kullanılırsa

A1 :X değerini gir.
A2 :Y değerini gir.
A3 : $Z = X + Y$
A4 : $\text{Ort} = Z / 2$
A5 :Ort değerini yaz.
A6 :Bitir.

Görüldüğü üzere bu şekilde bir algoritma ile çözüm yolunu izlemek daha kolaydır.

Örnek: Kenar uzunlukları verilen dikdörtgenin alan hesabını yapan programa ait algoritmanın hazırlanması

Kenar uzunlukları negatif olarak girildiği durumda veri girişi tekrarlanacaktır.

Çözüm:

Dikdörtgenin kısa kenarı : a
Dikdörtgenin uzun kenarı : b
Dikdörtgenin alanı : Alan
Algoritma

A1 :a değerini gir.
A2 :Eğer $a < 0$ ise A1'e git.
A3 :b değerini gir.
A4 : Eğer $b < 0$ ise A3'e git.
A5 : $\text{Alan} = a * b$
A6 :Alan değerini yaz.
A7 :Bitir.

Örnek: Beş sayının toplamını ve ortalamasını veren programa ait algoritmanın oluşturulması

Çözüm:

Toplam adı için Top
Ortalama adı için Ort
Girilen sayılar için X
Arttırma için Sayac kullanılırsa

A1 :Top = 0, Sayac = 0
A2 :X değerini gir.
A3 :Top= Top+X
A4 : Sayac = Sayac +1
A5 :Eğer Sayac <5 ise A2'ye git.
A6 :Ort= Top/5
A7 :Top ve Ort değerlerini yaz.
A8 :Bitir.

Sorular :







1. Girilen sayının pozitif, negatif veya sıfıra eşit olduğunu gösteren algoritmayı hazırlayınız.
2. Girilen sayının faktöriyelini hesaplayan algoritmayı hazırlayınız.
3. Girilen üç sayıdan en büyüğünü bulup ekrana yazan algoritmayı hazırlayınız.
4. Bir öğrenciye ait vize ve final notlarının ortalamasını hesaplayan ve ortalamaya göre ekrana “Geçti” – “Kaldı” yazan algoritmayı hazırlayınız (Not: Ortalama hesaplanırken vizenin %40, finalin %60'ı alınacak, geçme notu 45'dir.) .
5. Bir komisyoncu sattığı mallardan fiyatı 50 TL'ye kadar olanlardan %3, daha fazla olanlardan ise %2 komisyon almaktadır. Klavyeden girilen 5 malın komisyonlarını bularak toplam komisyonu hesaplayan algoritmayı hazırlayınız.

2.2. Akış Diyagramı

Herhangi bir problem için oluşturulan algoritmanın, görsel olarak simge ya da sembollerle ifade edilmiş şekline “akış diyagramı” denir. Akış diyagramları, yazılımı oluşturacak program parçalarını ve bu parçaların birbirleri ile olan ilişkilerini belirler.

Akış diyagramlarının algoritmadan farkı, adımların simgeler şeklinde kutular içinde yazılmış olması ve adımlar arasındaki ilişkilerin oklar ile gösterilmesidir.

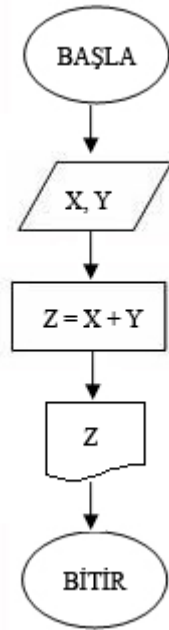
2.2.1. Akış Diyagramı Şekilleri

	Akış diyagramının başlangıç ve bitiş yerlerini gösterir. Başlangıç simgesinden çıkış oku vardır. Bitiş simgesinde giriş oku vardır.
	Klavyeden yapılacak girişler için kullanılır. Girilecek olan bilgi bir değişkene aktarılır. Sembol içine değişken yazılmalıdır.
	Her türlü hesaplama ve atama işlemleri için kullanılır.
	Bir karar verme işlemini temsil eder.
	Kullanıcıya yansıtılacak bilgileri belirtmek için kullanılır.
	Diyagramın akış yönünü, yani herhangi bir adımdaki işlem tamamlandıktan sonra hangi adıma gidileceğini gösterir.

Tablo 2.1: Akış diyagramı şekilleri ve açıklamaları

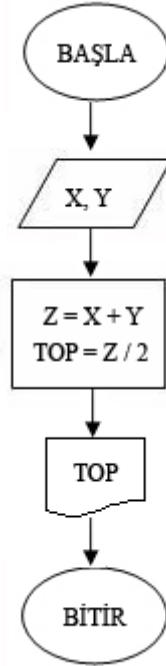
Örnek: İki sayıyı toplamak için gerekli programa ait akış şemasının oluşturulması.

Çözüm:



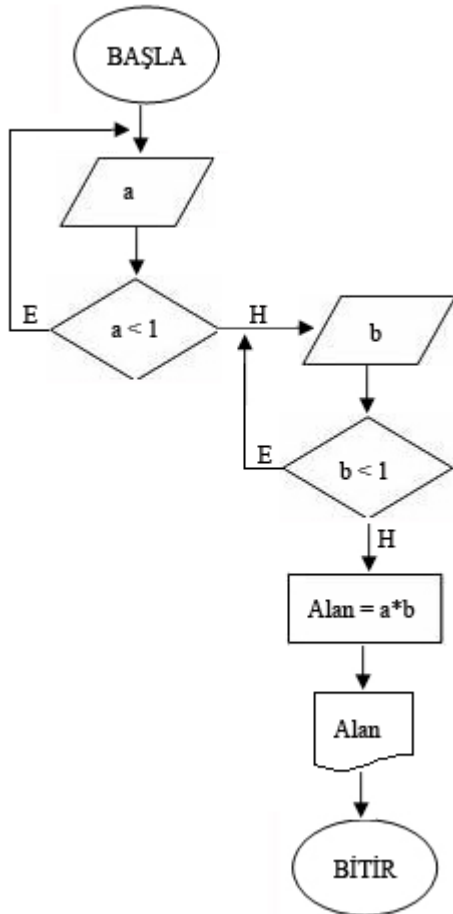
Örnek: İki sayının ortalamasını bulmak için gerekli programa ait akış şemasının oluşturulması

Çözüm:



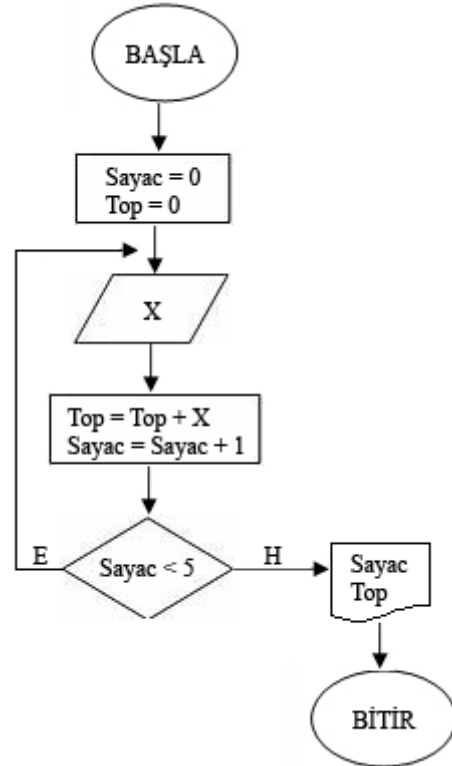
Örnek : Kenar uzunlukları verilen dikdörtgenin alan hesabını yapan programa ait algoritmanın hazırlanması
Kenar uzunlukları negatif veya sıfır olarak girildiği durumda veri girişi tekrarlanacaktır.

Çözüm:



Örnek : Beş sayının toplamını ve ortalamasını veren programa ait algoritmanın oluşturulması

Çözüm:



UYGULAMA FAALİYETİ

Günlük hayatta karşılaştığımız bir problem için algoritma ve akış şemasını hazırlayınız.

İşlem Basamakları	Öneriler
➤ Problemi tanımlayınız.	➤ Var olan bilgileri, anlamları ve birbirleriyle olan ilişkileri belirleyiniz.
➤ Problemi için çözümler geliştiriniz.	➤ Problemi alt birimlere bölünüz. ➤ Her birim için ayrı ayrı çözümler geliştiriniz. ➤ En uygun olan çözümü seçiniz.
➤ Girdi ve çıktıları belirleyiniz.	➤ Seçtiğimiz çözüm için dışarıdan veriler girerek sonuçlarını belirleyiniz.
➤ Çözümü kâğıt üzerinde gösteriniz.	➤ Problemi, ayırdığımız her birim ve çözümleri ile kâğıt üstüne aktarınız.
➤ Çözümü deneyiniz.	➤ Çözümüne dışarıdan veriler girerek çıkan sonuçları doğruluğu kanıtlanmış doğrularla karşılaştırıp doğruluğunu test ediniz.
➤ Çözümü geliştiriniz.	➤ Doğruluğu tespit edilen çözümümüzde tekrar eden birimleri tek bir birime düşürmek için yeni yollar geliştiriniz.
➤ Oluşabilecek hataları gözden geçirin.	➤ Oluşabilecek mantık hatalarını tespit ediniz.
➤ Akış şemasını çizin.	➤ Oluşturulan çözümün en son hâlini akış şeması olarak çizin.





ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümleleri dikkatlice okuyarak boş bırakılan yerlere doğru sözcüğü yazınız.

1. Verilen herhangi bir sorunun çözümüne ulaşmak için uygulanması gerekli adımların hiç bir yorumu yer vermeksizin açık, düzenli ve sıralı bir şekilde söz ve yazı ile ifadesine denir.
2. Algoritma yazıldıktan sonra, sonuçları daha önceden bilinen veriler girilerek, eldeki sonuçlarla çıkan sonuçlar karşılaştırıldığı aşama'dır.

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

3. Aşağıdakilerden hangisi algortima yazım aşamaları arasında yer almaz?
A) Problemi tanımlama
B) Çözümü geliştirme
C) Oluşabilecek hatalar
D) Güvenirlilik
4. Her türlü hesaplama ve atama işlemlerinin yapıldığı akış diyagramı şekli aşağıdakilerden hangisidir?

- A)  B) 
C)  D) 

A1 :sayac = 0
A2 :a değerini gir.
A3 :Eğer $a < 0$ A2'ye git.
A4 :toplam = toplam + a
A5 :.....
A6 :Toplam değerini yaz.
A7 :Bitir.

5. Yukardaki algoritmanın klavyeden girilen 5 adet pozitif sayının toplamını ekrana yazdırması için boş bırakılan satıra aşağıdaki seçeneklerden hangisi yazılmalıdır.

- A) Eğer sayac < 6 A2'ye git
B) Eğer sayac < 5 A2'ye git
C) Eğer sayac < 5 A4'e git
D) Eğer sayac > 5 A4'e git

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ-3

AMAÇ

Bu öğrenme faaliyeti sonunda programlama dili yazılımı ile çalışabileceksiniz.

ARAŞTIRMA

- En çok hangi programlama yazılımının kullanıldığını araştırınız.
- Bir program yazmak için kullanılabilecek araçları listeleyip bunlar içinden en rahat kullanabileceklerinizi işaretleyiniz.

3. PROGRAMLAMA YAZILIMI

Bilgisayar programlama konusunda şu ya da bu programlama dili daha iyidir demek yanlıştır. Önemli olan yazılımın içeriğidir. Seçilecek programlama dili ile yazılımın en efektif şekilde üretilmesi amaçlanır. Programlama yazılımlarının görevi yazmış olduğumuz kodları belirle kurallar çerçevesinde bilgisayarın anlayacağı dile çevirmektir. Her programlama yazılımı bütün dilleri derleyemeyeceğinden yazacağımız kodlara uygun bir yazılım seçmemiz gerekmektedir. Günümüzde kullanılan çok çeşitli programlama yazılımları mevcuttur.

IDE: Tümleşik geliştirme ortamı (Integrated development environment), bilgisayar programcılarının hızlı ve rahat bir şekilde yazılım geliştirebilmesini amaçlayan, geliştirme sürecini organize edebilen birçok araç ile birlikte geliştirme sürecinin verimli kullanılmasına katkıda bulunan araçların tamamını içinde barındıran bir yazılım türüdür.

Tümleşik geliştirme ortamlarında olması gerekli en temel özellikler:

- Programlama diline göre sözdizimi renklendirmesi yapabilen kod yazım editörü
- Kod dosyalarının hiyerarşik olarak görülebilmesi amacıyla hazırlanmış gerçek zamanlı bir dizelge
- Tümleşik bir derleyici, yorumlayıcı ve hata ayıklayıcı
- Yazılımın derlenmesi, bağlanması, çalışmaya tümüyle hazır hâle gelmesi ve daha birçok ek işi otomatik olarak yapabilmek amacıyla küçük inşa araçları

3.1. Programlama Yazılımının Arayüzü

Visual Studio, çok gelişmiş özelliklere ve yardımcı araçlara sahip bir dosya editörüdür. NET platformu üzerinde geliştirilen proje dosyaları dışında metin dosyaları, sql, rtf uzantılı dosyalar da düzenlenebilir.

➤ Çalışma sayfaları (tab pages)

Visual Studio ortamında dosyalar, birer çalışma sayfası olarak açılır. Bu dosyalar sekmeler hâlinde sıralanır. Sayfalar arasında CTRL-TAB kısayolu ile geçiş yapılır.

Bu çalışma modelinde, sadece bir sayfa görünür ve üzerinde çalışma yapılır.

➤ Araç çubukları (toolbars)



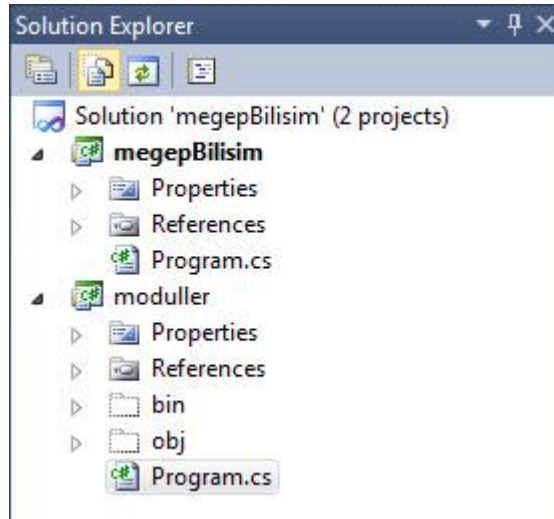
Resim 3.1: Programlama yazılımı araç çubukları

Visual Studio, menü komutları için görsel kısayolları araç çubukları ile sunar. Benzer işlemler için kullanılan komutlar bir araç çubuğunda gruplanır. Örneğin standart araç çubuğu, yeni dosya oluşturmak, bir dosyayı açmak - kaydetmek gibi genel dosya işlemleri için kullanılır.

Araç çubukları, varsayılan olarak menülerin altında bulunur. Ancak çubukları taşınarak yerlerini değiştirebilir veya kayan duruma getirilebilir. Ayrıca istenen çubuklar saklanılabilir veya gösterilebilir. Araç çubuklarını listesini görmek için 'View' menüsünden 'Toolbars' alt menüsüne işaret ediniz.

➤ Paneller

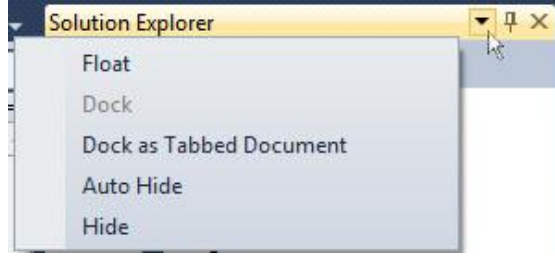
Paneller, Visual Studio içindeki pencerelerdir. Çalışma ortamında birçok panel bulunmasıyla beraber, 'Solution Explorer, Toolbox, Database Browser, Properties, Find Result, Error List' gibi sıkça kullandığımız paneller vardır.



Resim 3.2: Solution explorer paneli

İPUCU: Görmek istenilen paneller 'View' menüsünden seçilebilir.

Paneller, Visual Studio ortamı içinde istenilen yere taşınabilir veya sabitlenebilir. Panellerin birkaç genel özelliği vardır:



Resim 3.3: Panel özellikleri

Auto hide(otomatik gizle):

Panelin, fare üzerindeyken görünmesi ve fare çekildikten sonra gizlenmesidir.

Dockable(sabitlenebilir):

Panelin, Visual Studio ortamı içinde bir yerde sabitlenebilme özelliğidir.

Floating(kayan):

Kayan paneller herhangi bir yere sabitlenemez. Ancak her sayfanın üstünde durur ve böylece sürekli görünür.

Panellerin bu özelliklerine 'Window' menüsünden erişilebilir.

3.1.1 Başlangıç Sayfası(Start Page)

Başlangıç sayfası projelere erişmeyi ve yeni projeler oluşturmayı kısayoldan sağladığı gibi ürün bültenlerini, yaklaşan konferansları ve en son gelişme makalelerini okuma olanağı da tanır. Başlangıç sayfasına ulaşmak için 'View(görünüm) → Start Page(Başlangıç Sayfası)' menüsünü kullanabilirsiniz.

Başlangıç sayfası command 'section(komut bölümü), recent projects(son projeler) ve content area(içerik alanı)' olmak üzere üç bölümden oluşur. Sayfanın sol alt kısmında ise başlangıç sayfasının görünüm ayarları için seçenekler mevcuttur.



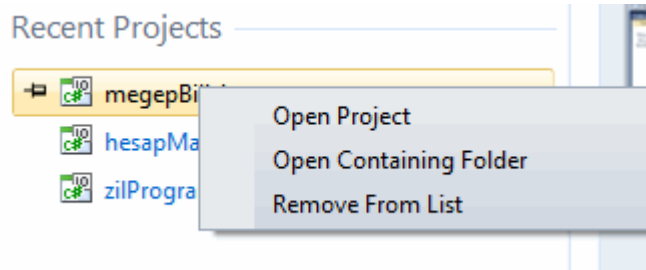
Resim 3.4: Programlama yazılımı başlangıç sayfası(start page)

➤ **Command section(komut bölümü)**

Yeni proje ve bilgisayarda kayıtlı olan projeleri açmak için kullanılan bölümdür.

➤ **Recent projects(son projeler)**

Son projeler listesi için bağlantılar görüntülenir. Bir link tıklandığında ‘Visual Studio’ ile ilgili proje açılır. Link üzerinde sağ tıklandığında aşağıdaki tablodaki seçeneklerden birinin seçilmesini sağlayan bir menü açılır.



Resim3.5: Son açılan projeler listesi üzerinde yapılacak işlemler menüsü

Seçenek	Açıklama
Open Project	Projeyi açar.
Open Containing Folder	Projeyi içeren klasörü Windows explorer’da açar.
Remove From List	Projeyi son projeler listesinden kaldırır.

Tablo 3.1: Son açılan projeler üstünde yapılacak işlemlerin açıklaması

Son projeler listesinde bir öge üzerine gelindiğinde, ögeyi vurgular ve bir raptiye simgesi görünür. Raptiye simgesi bir kez tıklandığında o proje listede kalıcı olarak saklanır.

➤ **Content area(içerik alanı)**

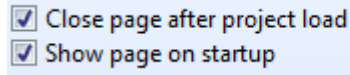
İçerik alanı ‘Get Started(Başlarken)’ ve Latest News(Son Haberler)’ sekmelerini içermektedir.

‘Get Started’ sekmesi altında bulunan kategorilerden birini seçerek aşağıdaki listeyi değiştirebiliriz. Bu sekmede teknik makaleler, yardım konuları, verimlilik artışı ve ürün özellikleri hakkında yardımcı bilgiler yer almaktadır.

‘Latest News’ sekmesinde kutuya girmiş olduğumuz adreste bulunan makalelerin listesi yayınlanır. RSS besleme adresini değiştirerek farklı bir haber kanalına geçebiliriz. ‘MSND Web’ sitesinde kullanabileceğimiz RSS besleme adreslerini bulabilirsiniz.

➤ **Display options(gösterme seçenekleri)**

Başlangıç sayfasının görünüm ayarlarının yapıldığı bölümdür.



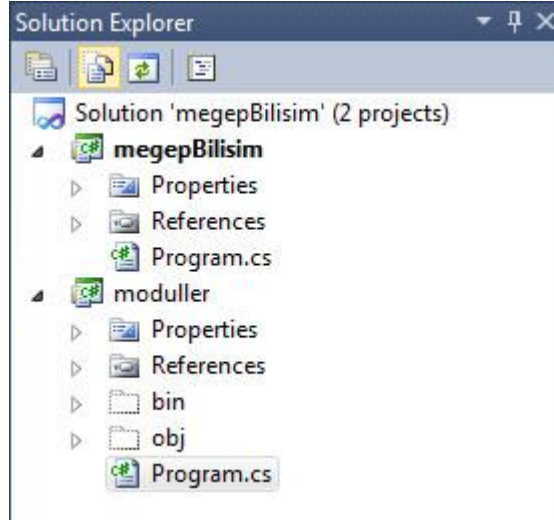
Resim 3.6: Programlama yazılımı başlangıç sayfası görünüm seçenekleri

Seçenek	Açıklama
Close page after project load	Bir proje açıldığında başlangıç sayfasını kapatır.
Show page on startup	Visual Studio her başlatıldığında başlangıç sayfasının görüntülenmesi sağlanır.

Tablo 3.2: Başlangıç sayfası görünüm seçenekleri açıklaması

3.1.2 Solution Explorer Paneli

‘Visual Studio’ çalışma ortamında projeler bir solution (çözüm) altında açılır. Bir solution içine farklı dilde ve tipte projeler dâhil edilebilir. ‘Visual Studio’ ile bir ‘solution2’ açıldığında, ‘Solution Explorer’ paneli ile ‘solution’ içinde bulunan tüm projeleri, ilgili dosya ve klasörleri görüntüler. Panelde koyu yazı tipinde gözüken proje, solution içindeki başlangıç projesidir.




Resim 3.7: Solution explorer paneli

Panelin üst kısmında, seçilen öge üzerinde basit işlemler gerçekleştirmek için bir araç çubuğu bulunur.

-  Refresh (yenile)


Proje dosyaları üzerindeki değişikliklerin görünmesini sağlar.

-  Show all files (bütün dosyaları göster)

Seçilen projenin bulunduğu klasördeki tüm dosyaları ve alt klasörleri gösterir. Panelde görünen beyaz ögeler proje içine dâhil edilmemiş öğelerdir. Projede kapsamında kullanılmak istenen ögeler (örneğin arka plan resmi), üzerine sağ tıklanıp 'Include In Project' komutu ile projeye dâhil edilmelidir.

-  Properties (özellikler)

Paneldeki tüm öğelerin özellikleri, 'Properties' komutu ile görülebilir. Bu komut seçildiğinde öğenin özellikleri 'Properties paneli' ile görüntülenir.

-  View code (kodu göster)

Panel üzerinde seçmiş olduğumuz dosyanın kodlama sayfasının görüntülenmesini sağlar.

'Solution Explorer paneli, View → Other Windows → Solution Explorer' menüsünden görülebildiği gibi varsayılan klavye seçeneklerinde CTRL-ALT-L kısayolu ile de görülebilir.

- **Projeye yeni bir öge eklemek için**
 - Solution Explorer'da, bir hedef proje seçiniz.
 - Proje üzerinde sağ tıklayarak Add→New Item'ı seçiniz.
 - Kategori bölümünde bir kategori seçiniz.

- **Projeye varolan bir öge eklemek için**
 - Solution Explorer'da, bir hedef proje seçiniz.
 - Proje üzerinde sağ tıklayarak Add→Existing Item'ı seçiniz.
 - Öge iletişim kutusunda eklemek istediğiniz proje veya ögeyi bulup seçiniz.





UYGULAMA FAALİYETİ

Aşağıdaki işlemleri uygulayınız.

İşlem Basamakları	Öneriler
➤ Yeni bir proje oluşturunuz.	➤ Programı çalıştırınız. ➤ Komut bölümü veya File menüsünden New Project seçeneğini seçiniz. ➤ Geliştireceğimiz uygulama türünü seçip isimlendiriniz.
➤ Kayıtlı olan bir projeyi açınız.	➤ Komut bölümü veya 'File' menüsünden 'Open Project' seçeneğini seçiniz. ➤ Projemizin kayıtlı olduğu yeri bulup açınız.
➤ Son projeler bölümündeki proje listesini temizleyiniz.	➤ Son projeler bölümde proje isimleri üzerine sağ tıklayınız. ➤ 'Remove From List' seçeneğini seçip projeyi listeden kaldırınız.
➤ Projeyi açtıktan sonra başlangıç sayfasının otomatik olarak kapanacak şekilde ayarlayınız.	➤ Başlangıç sayfasında 'Display Options(Gösterme Seçenekleri)' bölümündeki 'Close page after project load' kutucuğunu onaylayınız.
➤ Debug araç çubuğunu görünür hâle getiriniz.	
➤ Database penceresini görünür hâle getiriniz.	

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

1. Aşağıdaki eşleştirmelerden hangisi yanlıştır?
 - A)  -----> Refresh (Yenile)
 - B)  -----> Show All Files (Bütün Dosyaları Göster)
 - C)  -----> Categorized (Kategorile)
 - D)  -----> View Code (Kodu Göster)
2. Başlangıç sayfası aşağıdaki bölümlerden hangisini içermez?
 - A) Connect to Database(Veritabanı Bağlantısı)
 - B) Command Section(Komut Bölümü)
 - C) Recent Projects(Son Projeler)
 - D) Content Area(İçerik Alanı)
3. Panelin, Visual Studio ortamı içinde bir yerde sabitlenebilmesini hangi seçenek sağlar.?
 - A) Auto Hide
 - B) Dockable
 - C) Floating
 - D) Property

Aşağıdaki cümleleri dikkatlice okuyarak boş bırakılan yerlere doğru sözcüğü yazınız.

4. Çalışma sayfaları arasında geçişi klavye kısayolu ile sağlarız.
5. Görmek istediğimiz panele menüsünden erişilir.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise “Modül Değerlendirme”ye geçiniz.


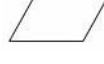



MODÜL DEĞERLENDİRME

Aşağıdaki cümleleri dikkatlice okuyarak boş bırakılan yerlere doğru sözcüğü yazınız.

1. Kişi tarafından veya bilgisayar tarafından sağlanan verilere denir.
2. Kaynak kodu kısa, derlenmiş hâlleri uzun ve çalışma hızları yavaş olan projeler dil ile yazılmıştır.
3. Herhangi bir problem için oluşturulan algoritmanın, görsel olarak simge ya da sembollerle ifade edilmiş şekline denir.
4. Menü komutları için oluşturulmuş görsel kısayollara denir.

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

5. Aşağıdakilerden hangisi makine dili için yanlıştır?
A) Bilgisayarın doğal dilidir.
B) Bilgisayarın donanımsal tasarımına bağlıdır.
C) Doğrudan makinaya hitap eder.
D) Her işlemci için ortakdır.
6. Hangi sayı sistemi 0-7 arası rakamları kapsar?
A) Yedilik
B) İkilik
C) Sekizlik
D) Onluk
7. $(0011010)_2$ sayısının 10'luk sistemde karşılığı aşağıdakilerden hangisidir?
A) 21 B) 26 C) 17 D) 19
8. $(39)_{10}$ sayısının 2'lik sistemde karşılığı aşağıdakilerden hangisidir?
A) 100101 B) 110111 C) 100111 D) 101011
9. $(1101110)_2$ sayısının 8'lik sistemde karşılığı aşağıdakilerden hangisidir?
A) 155 B) 146 C) 158 D) 156
10. $(271)_8$ sayısının 2'lik sistemde karşılığı aşağıdakilerden hangisidir?
A) 10111001 B) 10101001 C) 01111001 D) 10101101
11. $(101100111110101)_2$ sayısının 16'lık sistemde karşılığı aşağıdakilerden hangisidir?
A) B3E5 B) B3F5 C) B5F3 D) B5E3

12. $(8FD2)_{16}$ sayısının 2'lik sistemde karşılı aşağıdakilerden hangisidir?
A) 1000110011010010 B) 1000111111001010
C) 1010111111010110 D) 1000111111010010
13. Aşağıdakilerden hangisi kullanıcıya yansıtılacak bilgileri belirtmek için kullanılır?
A)  B) 
C)  D) 
14.  butonunun görevi aşağıdakilerden hangisidir?
A) Properties (özellikler)
B) View code (kodu göster)
C) Refresh (yenile)
D) Show all files (bütün dosyaları göster)

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ 1 CEVAP ANAHTARI

1	Yazılım
2	C
3	D
4	B
5	D

ÖĞRENME FAALİYETİ 2 CEVAP ANAHTARI

1	Algoritma
2	Çözümü Deneme
3	D
4	A
5	B

ÖĞRENME FAALİYETİ 3 CEVAP ANAHTARI

1	C
2	A
3	D
4	Ctrl + Tab
5	View

MODÜL DEĞERLENDİRME CEVAP ANAHTARI

1	Giriş
2	Yüksek Seviyeli
3	Akış Diyagramı
4	Araç Çubukları
5	D
6	C
7	B
8	C
9	D
10	A
11	B
12	D
13	A
14	C

KAYNAKÇA

- <http://www.ecma-international.org/>
- <http://www.gorselprogramlama.com/> (15.03.2011- 12:00)
- <http://www.yazilimutfagi.com/> (15.03.2011- 12:00)
- <http://www.csharpnedir.com/> (15.03.2011- 12:00)
- <http://www.programmersheaven.com/> (15.03.2011- 12:00)
- http://tr.wikipedia.org/wiki/Ana_Sayfa