--- Sayfa 1 ---

İZMiR KATİP ÇELEBİ ÜNİVERSİTESİ 2010  
UZAKTAN ÖĞRETİM DERS MATERYALLERİ  
İLERİ PROGRAMLAMA TEKNİKLERİ  
1 Hafta  
Algoritma Temel Kavramlar Prof. Dr. Ayşegül Alaybeyoğlu

--- Sayfa 2 ---

İLERİ PROGRAMLAMA TEKNİKLERİ  
Ders 1  
ProfDr. Ayşegül Alaybeyoğlu

--- Sayfa 3 ---

Problem Çözme  
İnsanlar sürekli düşünür ve problem çözerler Birçok problem; az Ya da hiç dügünülmeden çözülebilir  
Problem: Bugün evden çıkarken ne giymeliyim?  
Çözüm: Bunun için muhtemelen pencereden dışarıya bakılır. Hava yağmurlu ise mevsime göre giyinmenin yanı sıra dışarıya çıkarken bir de şemsiye alınması gerekir:. Hava güneşli ve sıcak ise daha ince giyinilerek dışarıya çıkılır: Böylece problemin çözümü kendiliğinden oluşturulan bir kararla sağlanır:  
3

--- Sayfa 4 ---

Problemi Kim Çözecek?  
Bilgisayar, sadece yazılımcının kendisine söylediği şeyi nasıl yapacağını bilir.  
Sonuç olarak yazılımcı bilgisayara problemi nasıl çözeceğini bildirmelidir  
4

--- Sayfa 5 ---

Problem Çözme (devam..  
Bilgisayara nasıl iş yaptıracak, nasıl iletisim kuracaksınız?  
Bilgisayarın dili makine dilidir Onunla makine mantığı ile iletişim kurabiliriz.  
Bir <Program" ile. Bilgisayarlar program olmadan çalısmazlar.  
Bu da Algoritma (talimat, rutin; reçete) ile olur .  
5

--- Sayfa 6 ---

Problem Çözme Sırası  
1. Problemi anlama (Understanding, Analyzing) 2. Bir çözüm yolu gelistirme (Designing) 3 . Algoritma ve program yazma (Vriting) 4. Tekrar tekrar test etme (Revieving)  
Polya; George (1957) 'Hov To Solve It \_ Princeton University Press, 2nd Edition  
6

--- Sayfa 7 ---

Problem Çözme Farklı Bakış  
Problemin tanmmlanması  
Problem Çözme Aşaması  
Çözümün ana hatlarının ortaya konulması  
Ana hatlara bağlı bir algoritma gelistirilmesi  
Algoritmanın doğruluğunun sıralanması  
Algoritma kodları belirli bir programlama diline dönüştürülür  
Gerçeklestirim Program bilgisayarda çalıştırılır: Aşaması  
Program belgelemesi ve bakımı yapılır.

--- Sayfa 8 ---

Problem Çözme Descartes  
Problem çözmede, soruna hemen girişmek yerine, dikkatli ve sistematik yaklaşım ilke olmalıdır. Problem iyice anlaşılmalı ve mümkün olduğu kadar küçük parçalara ayrılmalıdır: Descartes'in "Discourse on Method' isimli kitabında problem çözme teknikleri şu dört madde ile özetlenir: 1 Doğruluğu kesin olarak kanıtlanmadıkça, hiçbir şeyi doğru olarak kabul etmeyin; tahmin ve önyargılardan kaçının: 2. Karşılaştığınız her güçlüğü mümkün olduğu kadar çok parçaya bölün. 3 . Düzenli bir biçimde düşünün; anlaşılması en kolay olan şeylerle başlayıp yavaş yavaş daha zor ve karmaşık olanlara doğru ilerleyiniz: 4. Olaya bakışınız çok genel, hazırladığınız ayrıntılı liste ise hiçbir şeyi dışarıda bırakmayacak kadar kusursuz ve eksiksiz olsun. 8

--- Sayfa 9 ---

Algoritma Nedir?  
Basit tanım: Belirli bir görevi yerine getiren sonlu sayıdaki işlemler dizisidir Geniş\_tanım: Verilen herhangi bir sorunun çözümüne ulaşmak için uygulanması gerekli adımların hiç bir yoruma yer vermeksizin açık, düzenli ve sıralı bir şekilde SÖZ ve yazı ile ifadesidir. Algoritmayı oluşturan adımlar özellikle basit ve açık olarak sıralandırılmalıdır.  
9

--- Sayfa 10 ---

Algoritmaya Dair -  
Algoritmanın etkin bir şekilde olusturulması Program yazma adımından çok daha önemlidir. Hazırlanan algoritmanın programlama diliyle yazılması işin basit kısmıdır: Tasarladığınız algoritma iyi değilse; kullandığınız dilin hiçbir önemi yoktur (C, C+4, C#, Java; Visual Basic vb.) Bir sorunun çözümü için birbirinden farklı birden fazla sayıda algoritma hazırlanabilir. Bu da gösteriyor ki herhangi bir problemin çözümü için birbirinden farklı yüzlerce bilgisayar programı yazılabilir:  
10

--- Sayfa 11 ---

Algoritma Türlerine Örnekler  
Arama algoritmaları Bellek yönetimi algoritmaları Bilgisayar grafiği algoritmaları Evrimsel algoritmalar Genetik algoritmalar Kriptografik algoritmalar Optimizasyon algoritmaları Sıralama algoritmaları Veri sıkıştırma algoritmalar Veri Madenciliği algoritmaları İş Zekası algoritmaları  
Astronomi algoritmaları Dinamik Programlama algoritmaları Sağlık bilimleri algoritmaları Fizik algoritmaları Veritabanı algoritmaları İşletim sistemi algoritmaları  
11

--- Sayfa 12 ---

Algoritmaların Sahip Olması Gereken Genel Özellikler  
Giriş/çıkış bilgisi, Sonluluk, Kesinlik, Etkinlik, Başarım ve performans.  
12

--- Sayfa 13 ---

Algoritmaların Sahip Olması Gereken Genel Özellikler (Devam.  
Giriş/Çıkış Bilgisi Algoritmalarda giriş ve çıkış bilgileri olmalıdır. Dışarıdan gelen verilere giriş bilgisi denir Bu veriler algoritmada işlenir ve çıkış bilgisini oluşturur. Çıkış bilgisi her algoritmada mutlaka vardır Algoritmaların temel amacı giriş bilgisini işleyerek çıkış bilgisi olusturmaktır. Ancak her durumda bir algoritmanın çıkış bilgisi istenenleri tam olarak karşılayamaz. Böyle durumlarda ilk algoritmanın ürettiği çıkış bilgisi başka bir algoritmaya giriş bilgisi olarak gönderilir ve böylece kullanıcı istediği bilgiye sahip olmuş olur:  
13  
MNPUT  
OutPuT

--- Sayfa 14 ---

Algoritmaların Sahip Olması Gereken Genel Özellikler (Devam.  
Sonluluk Her türlü olasılık için algoritma sonlu adımda bitmelidir Algoritma sonsuz döngüye girmemelidir:  
Merhaba Merhaba Mertaba Merhaba Mcrhaba Merhaba Merhaba Merhaba Merhaba Merhaba Merhaba  
Z  
14

--- Sayfa 15 ---

Algoritmaların Sahip Olması Gereken Genel Özellikler (Devam.  
Kesinlik Her komut, kisinin kalem ve kağıt ile yürütebileceği kadar basit olmalıdır Algoritmanın her adımı anlaşılır; basit ve kesin bir biçimde ifade edilmiş olmalıdır. Kesinlikle yorum gerektirmemeli ve belirsiz ifadelere sahip olmamalıdır.  
~  
15  
s1

--- Sayfa 16 ---

Algoritmaların Sahip Olması Gereken Genel Özellikler (Devam.  
Etkinlik Yazılan algoritmalar etkin ve dolayısıyla gereksiz tekrarlardan uzak oluşturulmalıdır: Bu algoritmanın temel özelliklerinden birisidir. Ayrıca algoritmalar genel amaçlı yazılıp yapısal bir ana algoritma ve alt algoritmalardan oluşturulmalıdır. Böylece daha önce yazılmış bir algoritma daha sonra başka işlemler için de kullanılabilir Buna örnek vermek gerekirse eğer elimizde; verilen n adet sayının ortalamasını bulmakta kullandığımız algoritma varsa bu algoritma; bir sınıfta öğrencilerin yaş ortalamasını bulan bir algoritma için de kullanılabilmelidir.  
16

--- Sayfa 17 ---

Algoritmaların Sahip Olması Gereken Genel Özellikler (Devam.  
Başarım ve Performans Amaç donanım gereksinimi (bellek kullanımı gibi), çalışma süresi gibi performans kriterlerini dikkate alarak yüksek başarımlı programlar yazmak olmalıdır. Gereksiz tekrarlar ortadan kaldırılmalıdır. Bir algoritmanın performans değerlendirmesinde aşağıdaki temel kriterler göz önünde bulundurulur: Birim İşlem Zamanı Veri Arama ve Getirme Zamanı SUCCESS Kıyaslama Zamanı Aktarma Zamanı  
17

--- Sayfa 18 ---

Örnek: Çay Demleme Algoritması  
Mutfakta değilsen mutfağa git Çayı kontrol et, çay yoksa? Markete git, çay al. Çaydanlığa bak, dolu değilse su doldur Ocağı yak ve çaydanlığı ateşin üstüne koy Suyun kaynamasını bekle. Su kaynadıktan sonra çayı bırak ve üstüne suyu dök. Yine demliğe biraz daha su ilave ederek bekle. Su kaynadığında biraz dinlendirerek ateşi kapat. Çay bardağını al çayını doldur. Çayına istediğin kadar şeker at (ya da atma) ve karıştır. Geldiğin odaya geri dön. Ve çayı iç.  
18

--- Sayfa 19 ---

Yazılım Gelistirme Yaşam Döngüsünde Algoritma Nerede?  
Analiz  
Tasarım  
Algoritma ve Akış Şemaları  
Derleme (Compile  
Bağlama (Link)  
Çalıştırma Evecute  
Kodlama  
Test (Sınama)  
Bakım  
19

--- Sayfa 20 ---

Algoritma Gösterim Şekilleri  
1. Düz yazı ile gösterim, 9 2. Sözde kod (pseudocode) ile gösterim, 9 3 . Akış şeması ile gösterim.  
20

--- Sayfa 21 ---

Düz Yazı ile Gösterim  
Çözülecek problem; adım adım metin olarak yazılır: Her satıra numara verilir. BASLA ile başlanıp 'BİTİR ile sonlandırılır.  
21

--- Sayfa 22 ---

Örnek: Tahtaya Adını Yazma Algoritması  
1  
BAŞLA 2. Yerinden kalk 3 . Yönün tahtaya doğru mu? Hayırsa tahtaya dön Evetse 4. adıma git 4 Tahtaya doğru yürü 5 . Tahtaya geldin mi? Hayırsa 4.adıma git Evetse 6. adıma git 6 Kalemi al 7. Adını yaz 8. BİTİR  
22

--- Sayfa 23 ---

Örnek: Tahtaya Adını Yazma Algoritması  
Örneğin amacı, adımların tutarlılığını ve mantıksal sırasını göstermektir: Burada emirler; belli sorgulamalar yapılarak ve mantıksal bir sıra içinde verilmiştir. Yerinden kalk emri verilmeden kişiden yürümesi istenemez. Kalemi almadan adını yaz emrinin verilmesi doğru olmaz Sorgulamalarla da islemi yapıp yapmadığı kontrol edilmistir.  
23

--- Sayfa 24 ---

Örnek: Tahtaya Adını Yazma Algoritması  
Aslında bilgisayar bu tür işleri yerine getiremez. Kullanıcılar bilgisayarlara belli girdiler verir Onlar da programcının verdiği adımlara göre bu girdiler üzerinde matematiksel ve mantıksal islemler yaparak bir çıktı üretirler.  
24

--- Sayfa 25 ---

Sözde Kod (Pseudocode) ile Gösterim  
Herkesin anlayabileceği ve rahatlıkla bir programlama diline çevrilebilecek basit komutlardan oluşan bir dildir: Sözde kodun temel işlevi program gelistirmeye geçmeden algoritmayı oluşturmak ve üzerinde tartışabilmektir. Sözde kodlar, doğrudan konuşma dilinde ve programlama mantığı altında; eğer, iken gibi koşul kelimeleri ve = < gibi ifadeler ile beraber yazılır. Programda kullanılacak elemanları temsil etmek üzere uygun isimler veya değiskenler seçilir: Cebirsel notasyon ve kararlar kullanarak aritmetik islemler gerçeklestir. 25

--- Sayfa 26 ---

Örnek: İki Sayının Toplamı Algoritması  
1  
BAŞLA 2. Birinci sayıyı gir 3 . İkinci sayıyı gir 4 . İki sayıyı topla 5 . Sayıların toplam değerini yaz 6. BİTİR  
26  
koda Sözde çeviririz? nasıl

--- Sayfa 27 ---

Örnek: İki Sayının Toplamı Algoritması  
Düz Yazı  
Sözde Kod  
1  
BAŞLA 2 Birinci sayıyı gir 3 . İkinci sayıyı gir 4. İki sayıyı topla 5 Sayıların toplam değerini yaz 6 BİTİR  
Toplam için T, birinci sayı için <, ikinci sayı için Y seç 1 . BAŞLA 2. Y değerini OKU 3 . Y degerini OKU 4 T=K+ Y  
5 . Tdeğerini YAZ 6. BİTİR

--- Sayfa 28 ---

Örnek: Üçgenin Alanını Hesaplayan Algoritma  
1 .  
BAŞLA 2. Taban değerini gir 3 . Yükseklik değerini gir 4 . Taban ile yüksekliği çarp ve sonucu ikiye böl 5 . Çıkan sonucu yaz 6. BİTİR koda Sözde çeviririz? nasıl  
28

--- Sayfa 29 ---

Örnek: Üçgenin Alanını Hesaplayan Algoritma  
Düz Yazı  
Sözde Kod  
1  
BAŞLA 2 Taban değerini gir 3 . Yükseklik değerini gir 4 . Taban ile yüksekliği çarp ve sonucu ikiye böl 5 Çıkan sonucu yaz 6 BİTİR  
Taban için t; yükseklik için Ys alan için A seç 1 . BAŞLA 2. t değerini OKU 3 . y değerini OKU 4 . A = (t y)/2 5 . A değerini YAZ 6. BİTİR

--- Sayfa 30 ---

Akış Şemaları ile Gösterim  
Bir algoritmanın görsel şekiller ve sembollerle ifade edilmiş haline Akış Şemalarv adı verilir: Akış şeması sembolleri ANSI (American National Standards Institute) standardı olarak belirlenmiş ve tüm dünyada kullanılmaktadır: Algoritma doğal dille yazıldığı için herkes tarafindan anlasılamayabilir ya da başka anlamlar çıkarılabilir Ancak akış şemalarında her bir\_gekil standart bir anlam taşıdığı için\_farklı yorumlanması mümkün değildir:  
30

--- Sayfa 31 ---

Akış Şeması Şekilleri  
BAŞLA  
Akış şemasının başlangıç ve bitiş yerlerini gösterir Başlangıç simgesinden çıkış oku vardır Bitiş simgesinde giriş oku vardır.  
BİTİR  
Aritmetik işlemler ve değişik atama işlemlerinin temsil edilmesi için kullanılır:  
Dışarıdan bilgi giriş ve çıkışı için kullanılır.  
Belgeye; yazıcıya; ekrana çıktı için kullanılır:  
31

--- Sayfa 32 ---

Akış Şeması Şekilleri  
Kontrol ve karar verme işlemlerini temsil eder  
Döngü olduğunu gösterir:  
Oklar şemanın akış yönünü belirler  
Bağlantı işlemlerini temsil eder.  
32

--- Sayfa 33 ---

Örnek: İki Sayının Toplamı Akış Şeması  
Sözde Kod  
Başla  
Toplam için T, birinci sayı için <, ikinci sayı için Y seç 1 BAŞLA 2. < değerini OKU 3 . Y degerini OKU 4 T=Y+ Y  
Kve Y değerlerini OKU  
T=K+Y  
5   
Tdeğerini YAZ 6 BİTİR  
T değerini YAZ  
33  
Bitir

--- Sayfa 34 ---

Mantıksal Yapılar  
Bir bilgisayar programının geliştirilmesinde kullanılan programlama dili ne olursa olsun bu programların akış şemalarında genel olarak üç basit mantıksal yapı kullanılır: 1 . Sıralı Yapı 2. Karar Verme Yapısı 3 . Tekrarlı Yapı  
34

--- Sayfa 35 ---

Mantıksal Yapılar: Sıralı Yapı  
Sıralı yapı, hazırlanacak programdaki her işlemin mantık sIrasına göre nerede yer alması gerektiğini vurgular. Bu yapı sona erinceye kadar ikinci bir işlem başlayamaz.  
Başla  
İşlem 1  
İşlem 2  
İşlem 3  
Bitir  
35

--- Sayfa 36 ---

Mantıksal Yapılar: Karar Verme Yapısı  
Birden fazla sıralı yapı seçeneğini kapsayan modüllerde; hangi şartlarda hangi sıralı yapının seçileceğini belirler:  
Girdi  
Doğru  
Yanlış  
Koşul  
İşlem 1  
İşlem 2  
Bitir  
36

--- Sayfa 37 ---

Mantıksal Yapılar: Tekrarlı Yapı  
Algoritma içinde, bazı satırlar tekrarlı şekilde işlem görüyorsa, bir döngü söz konusudur. Döngülere belirli bir koşul geçerli olduğu sürece devam eden eylemleri tanımlamak için başvurulur.  
Başla  
Doğru  
Yanlış  
İşlemler  
Koşul  
37  
Bitir

--- Sayfa 38 ---

İşlemler ve Operatörler  
İşlemler 3?e ayrılır: 1. Matematiksel İşlemler Temel\_Aritmetik İslemler: Toplama; çıkarma; çarpma; bölme. Matematiksel Fonksiyonlar: Üstel, logaritmik, trigono-metrik, hiperbolik vb 2. Karşılaştırma İşlemleri 3. Mantıksal (Logic) İşlemler  
38

--- Sayfa 39 ---

Matematiksel İşlemler  
İşlem Toplama Çıkarma Çarpma Bölme Üs alma  
Gösterim a + b a - b a \* b a / b a ^ b  
Matematiksel Yazım a+b-c+2abc-7 a+b2-c3  
Bilgisayar Gösterim a+b-c+2\*a\*b\*c-7 a+b^2-c^3  
2 a = b+ 2ac a+b 2ab Va + b ~ b? 4ac  
a-blc+2\*a\*c-2I(a+b)  
(a+b) 1/2)-2\*a\*bl (b^2- 4\*a\*c) (a^2+b^2)/(2\*a\*b)  
a1+b2 2ab  
39

--- Sayfa 40 ---

Karşılaştırma İşlemleri  
Değiskenlerin büyük olma; küçük olma ve eşit olma durumlarını kontrol eden işlemlerdir  
İşlem Sembolü  
Anlamı Eşittir Eşit değildir Büyüktür Küçüktür Büyük eşittir Küçük esittir  
>=  
40

--- Sayfa 41 ---

Mantıksal İşlemler  
(Ve, Veya; Değily operatörleri hem matematiksel işlemlerde hem de karar ifadelerinde kullanılır.  
Mantıksal İşlem Ve Veya Değis  
Komut And Or Not  
VE bağlacı ile söylenmek istenen her iki koşulun da sağlanmasıdır: VE bağlacı ile bağlanmış önermelerden en az birinin yanlış olması sonucu yanlış yapar. VEYA bağlacı ile bağlanan koşullardan bir tanesinin doğru olması sonucu doğru yapar. 41

--- Sayfa 42 ---

Mantıksal İşlemler (devam...)  
DEĞİL bağlacı; doğruyu yanlış, yanlışı doğru yapar:  
Örnek: Yazılım departmanında çalışan erkek personellerden yaşı 30/un üzerinde olanları ekrana yazdır:  
Eğer; (perCinsiyet Erkek) VE (perYas 30) ise ekrana yazdır:  
1. KOŞUL  
2. KOŞUL  
42

--- Sayfa 43 ---

Algoritmada Kullanılan Terimler  
1 . Tanımlayıcı 2. Değisken 3 . Atama 4. Sayaç 5. Döngü  
43

--- Sayfa 44 ---

Algoritmada Kullanılan Terimler: Tanımlayıcı  
Programcı tarafindan oluşturulur. Programdaki değişkenleri, sabitleri, kayıt alanlarını, özel bilgi tiplerini vb. adlandırmak için kullanılan\_kelimelerdir Tanımlayıcılar; yerini tuttukları ifadelere çağrışım yapacak şekilde seçilmelidir: İngiliz alfabesindeki A-Z veya a-z arasındaki 26 harf ile 0-9 arası rakamlar kullanılabilir. Sembollerden sadece alt çizgi ( ) kullanılabilir Tanımlayıcı isimleri harfle veya alt çizgiyle başlayabilir: Tanımlayıcı ismi, rakamla başlayamaz veya sadece rakamlardan oluşamaz. 44

--- Sayfa 45 ---

Algoritmada Kullanılan Terimler  
1 . Tanımlayıcı 2. Değisken 3 . Atama 4. Sayaç 5. Döngü  
45

--- Sayfa 46 ---

Algoritmada Kullanılan Terimler: Tanımlayıcı  
Programcı tarafindan oluşturulur. Programdaki değişkenleri, sabitleri, kayıt alanlarını, özel bilgi tiplerini vb. adlandırmak için kullanılan\_kelimelerdir Tanımlayıcılar; yerini tuttukları ifadelere çağrışım yapacak şekilde seçilmelidir: İngiliz alfabesindeki A-Z veya a-z arasındaki 26 harf ile 0-9 arası rakamlar kullanılabilir. Sembollerden sadece alt çizgi ( ) kullanılabilir Tanımlayıcı isimleri harfle veya alt çizgiyle başlayabilir: Tanımlayıcı ismi, rakamla başlayamaz veya sadece rakamlardan oluşamaz. 46

--- Sayfa 47 ---

Algoritmada Kullanılan Terimler: Değişken  
Programın her çalıstırılmasında; farklı değerler alan bilgilbellek alanlarıdır: Değişken isimlendirilmeleri, tanımlayıcı kurallarına uygun biçimde yapılmalıdır: Örnekler: Dikdörtgenin uzun kenarının aktarıldığı değişken: uzun\_kenar, UzunKenar; uzunKenar Bir öğrenciye ait ismin aktarıldığı değişken: isim; ogrenci\_isim, 47 ogrencilsım

--- Sayfa 48 ---

Algoritmada Kullanılan Terimler: Atama  
Değiskenlere değer aktarma işlemidir. Değiskenlere atanan bu değerler daha sonra tekrar kullanılabilirler.  
Değisken Değer  
Sağdaki Değer sonucu Değişken'e aktarılır. Bu durumda Değişken'in bir önceki değeri varsa silinir  
i =0  
48

--- Sayfa 49 ---

Algoritmada Kullanılan Terimler: Sayaç  
Bazı işlemlerin belirli sayıda yaptırılması ve üretilen değerlerin sayılması gerekebilir Bu tür sayma işlemlerine algoritmada Sayaç adı veriler Sayaçlar da birer değişkendir.  
Bu işlemde Sayac değiskenine 1 eklenmekte ve oluşan sonuç yine kendisine yani Sayac değişkenine aktarılmaktadır.  
Sayac Sayac + 1  
49

--- Sayfa 50 ---

Algoritmada Kullanılan Terimler: Döngü  
Bir çok programda bazı işlemler; belirli ardışık değerlerle gerçeklestirilmekte veya belirli sayıda yaptınılmaktadır: Programlardaki belirli işlem bloklarını, verilen sayıda gerçekleştiren işlem akış çevrimlerine döngü" denir. Örneğin; 1 ile 1000 arasındaki tek sayıların toplamını hesaplayan programda T=1+3+5 yerine 1 ile 1000 arasında ikişer artan bir döngü kurulu ve döngü değişkeni ardışık toplanır:  
50

--- Sayfa 51 ---

Örnek: 1-5 arasındaki sayıların ekrana yazdırılması  
1 . BAŞLA 2. Sayac = 1 3 . Sayac değerini YAZ 4. Sayac Sayac + 1 5. Eğer Sayac < 6, GİT 3 6. BİTİR  
Başla  
Sayac = 1  
Sayac YAZ  
Sayac = Sayac + 1  
Evet  
Sayac <6  
Hayır  
Bitir

--- Sayfa 52 ---

Örnek: 1-5 arasındaki sayıların ekrana yazdırılması  
1. BAŞLA 2. Sayac 1 3 . Sayac değerini YAZ 4 Sayac Sayac + 1 5 . Eğer Sayac < 6, GİT 3 6. BİTİR  
Değişken İzleme Tablosu Eski Sayac Yeni Sayac Ekran 1 2 1  
2  
3  
2  
3  
4  
3  
4  
5  
4  
5  
6  
5  
52

--- Sayfa 53 ---

Örnek: 1-10 Arasındaki Tek Sayıların Toplamı  
1. BAŞLA 2. Sayac 1 3 . Toplam =0 4 . Eğer Sayac 10, GİT 8 5 . Toplam = Toplam + Sayac 6. Sayac Sayac + 2 7 GİT 4 8 . BİTİR  
Değişken İzleme Tablosu Eski Eski Yeni Yeni Sayac Toplam Sayac Toplam 1 3  
3  
1  
5  
4  
5  
4  
9  
9  
16  
11  
53

--- Sayfa 54 ---

Örnek: 1-10 Arasındaki Tek Sayıların Toplamı (Akış Şeması)  
Başla  
1. BAŞLA 2. Sayac 1 3 . Toplam =0 4. Eğer Sayac 10, GİT 8 5 . Toplam = Toplam + Sayac 6. Sayac Sayac + 2 7 . GİT 4 8 . BİTİR  
Sayac =1 Toplam = 0  
Hayır  
Evet  
Sayac >10  
Bitir  
Toplam = Toplam +Sayac Sayac = Sayac + 2  
54

--- Sayfa 55 ---

Programlama Terimleri ve Programlama Ortamı  
Program Programlama IDE (Integrated Development Environment Tümleşik Gelistirme Ortamı) Derleyici (Compiler) Yorumlayıcı (Interpreter) Bağlayıcı (Linker) Çalıştırma (Evecution) Hata Türleri Debug  
55

--- Sayfa 56 ---

Program  
Var olan bir problemi çözmek amacıyla bilgisayar dili kullanılarak oluşturulmuş anlatımlar (komutlar; kelimeler; aritmetik işlemler; mantıksal işlemler vb bütününe (programy denir.  
Eno sub tbtoolbar Burroncl` End sub Resume Neat Private 2 True Error Enabled on timtimer - Button -Key Case Back" 9 Gor Select Case brvlebBrouser 91 Forvard" Case brvlebBrouser "Refresh" Case brvlebBrous' 9 "Home (ase 'UebBro  
56

--- Sayfa 57 ---

Programlama  
Bir programı oluşturabilmek için gerekli komutların belirlenmesi ve uygun biçimde kullanılmasına programlama denir: Programlama, bir programlama dili kullanılarak yapılır: Bu programlama dili Java ve C# gibi yüksek seviyede bir dil olabileceği gibi C, Assembly ve bazı durumlarda makine dili de olabilir. Yazılan kaynak kodu genellikle bir derleyici ve bağlayıcı yardımıyla belirli bir sistemde çalıştırılabilir hale getirilir: Ayrıca kaynak kodu, bir yorumlayıcı yardımıyla derlemeye gerek duyulmadan satır satır çalıştırılabilir.  
57

--- Sayfa 58 ---

Programlama (devam.-  
Programlama aktivitesi genelde 'Merhaba Dünya" Hello Vorld! programı yazılmasıyla başlar.  
FTooOFOOOOLIOLToOo0000001000 11000101110100010oo11111111111010o000100 70101001011000011010111011010110110010003 /1101100000101011001000100001110001001113 0100110010110100110110100111101111011110 looiioioctinclude <stdio.h20110100001101d /100100110 001010001110 1ooo10olint maino 1000101111 Hio1o1oo1{ 100011000 111001100 printf("Hello NorId")booliod 1 return 42; Hooooon01h1oocecurn 0az; 110001101000012014 010101110110 71001001101111010111011110000001010001110 1000100100010101100100111011101000101117 /101010011100110101011100010101010001100 111001100000110111111010100111111000110 10100 010100100 0110101011101 programlama dilini öğrenmekteki  
Bir tek zorluk programlamanın ne olduğunu öğrenmektir . Bundan sonraki aşamalar daha basittir:  
58

--- Sayfa 59 ---

IDE (Integrated Development Environment Tümleşik Gelistirme Ortamı)  
IDE yazılımcının hızlı ve rahat bir şekilde program gelistirebilmesini amaçlayan; gelistirme sürecini organize edebilen birçok araç ile birlikte gelistirme sürecinin verimli kullanılmasına katkıda bulunan araçların tamamını içerisinde barındıran bir yazılım türüdür  
main c [HelloVorld] Code:Blocks 12.11 File Edit Viev Search Project Build Debug mSmith Tools Tools+ Plugins Doryt Blocks Settings Help Build target: Debug  
S C  
Management  
main.c  
Projects Symbols Morkspace HelloVorld Sources main c  
rirclode Cscd10,k> #include Zscdlib h>  
int ma1f( )  
Prictf "skar "eaad 1z") ; return 0; I  
Logs & others  
Search results  
Cccc  
Build log  
Build messages  
CppCheck  
CppCheck messages  
Cscope  
Debugger  
DoyBlocks  
Closed  
Indev.ht not found at C:llAkademik|Courses|Algoritma ve Programlama IKaynaklar |Erdal Algoritma Ders Notları|lgoritma ve Programlama IIDers Notları ve Kodlar|Hafta 4Ismitoh -case\_basari\_durumuldovygen|htm Ander htm HTM Help not found at C:|lkadem\*|Courses llgoritma ve Programlama IKaynaklar [Erdal Algoritma Ders NotarılAlgoritma ve Programlama I|Ders Notları ve Kodlar|Hafta 4|smitch case\_basan\_durummu|dorygen|smitch case\_basari\_durumu.chm.  
59  
Fles

--- Sayfa 60 ---

IDE (Integrated Development Environment Tümleşik Gelistirme Ortamı) (devam..  
Tümleşik gelistirme ortamlarında olması gerekli en temel özellikler aşağıdaki gibidir: 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 , 9 çalışmaya tümüyle hazır hale gelmesi ve daha birçok ek işi otomatik olarak yapabilmek amacıyla küçük inşa araçları.  
60

--- Sayfa 61 ---

IDE (Integrated Development Environment Tümleşik Geliştirme Ortamı) (devam..  
En bilinen tümleşik geliştirme ortamları: Eclipse, Microsoft Visual Studio, Code::Blocks, Dev-C++ 9 Anjuta, KDevelop; NetBeans .  
0 eclipse 3,0 Microsoft THE ECLIPSE PRO JECT atsonaVaJed Visual Studio u M1derane Gooes aui lntaibe 00, 3197187 82alnatbe Cle40 io Ortco] aon 0 ianed Cl8ne teiodie  
Code::Blocks Ticope sourcı. crocr-plalorm IUE kıtp: Hun colchlocks 03  
Ll  
Loading: Documentation  
Version 3.5.1  
ğC++  
KDevelop Publlshed under GNU Licences K D E D E V E L 0 P M E N T E N V 1 R 0 N M E N T  
NetBeans IDE  
61

--- Sayfa 62 ---

Derleyici (Compiler)  
Derleyici, yazılan programın kaynak kodunu okuyup icerisinde mantıksal veya yazınsal hatalar olup olmadığın bulan, 9 bulduğu hataları kullanıcıya göstererek programın düzeltilmesine yardım eden; hata yoksa programın çalıştırılması öncesinde kaynak kodu makine çeviren diline bir yazılımdır:  
380 1i0i  
uera  
0  
62

--- Sayfa 63 ---

Yorumlayıcı (Interpreter)  
Yorumlayıcı, kaynak kodu kısım kısım ele alarak doğrudan çalıştırır Yorumlayıcılar standart bir çalıştırılabilir kod üretmezler: Yorumlama işlemi aşama aşama yapılmadığı için genellikle ilk hatanın bulunduğu yerde programın çalışması kesilir Derleyicilerin tersine kodun işlenmeyen satırları üzerinden hiç geçilmez ve buralardaki hatalar ile ilgilenilmez Yorumlayıcılar genelde kaynak koddan; makine diline anlık olarak dönüşüm yaptıkları için; derleyicilere göre daha yavaş çalışırlar. Ayrıca kodu iyilestirme (optimizasyon) imkanı da çoğu zaman yoktur:  
63

--- Sayfa 64 ---

Bağlayıcı (Linker) ve Çalıştırma (Erecute)  
Bağlayıcı: Derleyici tarafindan object dosyasına çevrilen bir veya birden çok dosyanın birbirleri ile ilişkilendirmesi ve tek bir çalıştırılabilir dosyaya (Örneğin Vindovs e<e) çevrilmesini sağlayan yazılımdır Çalıstırma: Olusturulan makine dili programının çalıştırılması adımıdır.  
Program Kaynak Kod  
Derleyici  
Obje Kod Makine Dili  
Bağlayıcı  
Çalıştırılabilir Program  
64