

Algoritma Geliřtirmek, Satır Kod

Dr. Fatih KALEMKUŐ

Kafkas Üniversitesi

Algoritma Geliřtirmek

Bir problem çözmek üzere geliştirilen algoritma üç şekilde yazılabilir:

- **Satır algoritma:** Problemin çözüm adımları düz metin olarak açık cümlelerle yazılır.
- **Akış diyagramı (flow-chart):** Problemin çözüm adımları geometrik şekillerle gösterilir.
- **Sözde kod (pseudo-code):** Problemin çözüm adımları komut benzeri anlaşılır metinlerle veya kısaltmalarla ifade edilir.

Örnek: Klavyeden girilen iki sayıyı toplayıp ekranda gösteren programın algoritmasının üç deęişik yöntemle gösterilmesi:

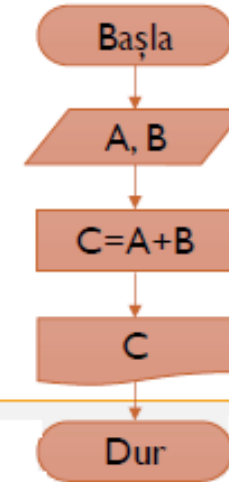
Satır algoritma ile gösterilmesi:

1. Başla
2. Birinci sayıyı (A) klavyeden oku
3. İkinci sayıyı (B) klavyeden oku
4. Girilen sayıları toplayarak sonucu oluştur ($C=A+B$)
5. Sonucu (C) ekrana yazdır
6. Dur

Sözde kod ile gösterilmesi:

1. Başla
2. A oku
3. B oku
4. $C=A+B$
5. C yaz
6. Dur

Akış diyagramı ile gösterilmesi:



Algoritma Geliřtirmek

Algoritmaları geliřtirirken deęiřken, sabit, atama, dng, karar yapısı, alt yordam gibi bir takım ğeler kullanılır.

Veri: (data): Bilgisayarlarda iřlenen tm bilgiler veri olarak adlandırılırlar. Veriler temel olarak *sayısal* ve *alfasayısal* olmak zere ikiye ayrılırlar (Seckin, s.55).

Tanımlayıcı: (identifier): Deęiřken, sabit, alt yordam, alan gibi programlama birimlerine yazılımcı tarafından verilen isimlerdir.

Deęiřken (variable): Programın akıřı iinde farklı deęerleri tutmak zere ayrılmıř bellek blmdr. rneęin $C=A+B$ gibi bir ifadede A, B ve C tanımlayıcıları birer deęiřkendir.

Sabit (constant): Program her alıřtıęında ve programın iinde herhangi bir anda hep aynı deęeri dndren tanımlayıcılara sabit denir. $PI=3.14$ gibi bir ifadeden sonra PI sabiti programın her yerinde 3.14 deęerini ifade eder.

Yarıapı girilen dairenin alanını hesaplayıp yazdıran programın algoritmasını oluřturunuz.

Algoritma Geliřtirmek

İsmlendirme Kuralları:

- İngiliz alfabesindeki A-Z veya a-z arası 26 harf kullanılabilir.
- 0-9 arası rakamlar kullanılabilir.
- Simgelerden sadece alt çizgi (_) kullanılabilir.
- Harf veya alt çizgi ile başlayabilir, ancak rakamla başlayamaz veya sadece rakamlardan oluşamaz.
- İlgili programlama dilinin komutu veya saklı/anahtar kelimesi olamaz.

En çok kullanılan standart tanımlayıcı gösterimleri:

- Pascal case: Kelimelerin ilk harfleri büyük. Örn: **AdSoyad, ÖğrenciNo**
- Camel case: Birinci kelimenin ilk harfi küçük, diğer kelimelerin ilk harfleri büyük.
Örn: **adSoyad, öğrenciNo**

Algoritma Geliřtirmek

Gömülü deęer (literal): Kod içine yazılmış olan metinsel, sayısal ya da dięer veri tiplerindeki sabit deęerlere gömülü deęer denir. $PI=3.14$ ifadesindeki 3.14 deęeri bir gömülü deęerdir. Aynı şekilde `mesaj="merhaba"` ifadesindeki `mesaj` bir deęişken veya sabit, "merhaba" ise gömülü deęerdir.

Aritmetik işlemler: Programlamadaki en temel işlemlerdir. Aritmetik işlemlerde kullanılan operatörler ařağıdaki gibidir.

İşleç	Adı	Örnek
+	Toplama	$C=A+B$
-	Çıkartma	$D=E-F$
*	Çarpma	$X=Y*Z$
/	Bölme	$T=P/S$
=	Eşittir	$A=B+C$

Algoritma Geliřtirmek

Mantıksal İşlemler: Bir programın akışı içerisinde belirli bir koşula baęlı olarak akışın hangi yönde ilerleyeceğine karar vermede mantıksal işlemler kullanılır. Bu ifadelerin sonucunda doğru (**true**) ya da yanlış (**false**) değerleri elde edilir.

Mantıksal işlemlerde kullanılan işleçler aşağıdaki gibidir.

İşleç	Adı	Örnek
>	Büyük	A>B
<	Küçük	A<B
==	Eşit	A==B
<> (veya !=)	Farklı	A<>B veya (A!=B)
>=	Büyük Eşit	A>=B
<=	Küçük Eşit	A<=B

Algoritma Geliştirmek

Öğrencinin numarasını, vize ve final notunu aldıktan sonra ortalamasını hesaplayıp numarasını ve not ortalamasını yazdıran program.

1. Başla
2. Öğrencinin numarasını (No) al
3. Öğrencinin adını ve soyadını (AdSoyad) al
4. Öğrencinin vize notunu (VizeNot) al
5. Öğrencinin final notunu (FinalNot) al
6. $Ort = 0.3 * VizeNot + 0.7 * FinalNot$
7. Numara (No) ve ortalamayı (Ort) yaz
8. Dur

Algoritma Geliřtirmek

Sayaç Kullanımı:

Programlarda bazı işlemlerin belirli sayıda yapılması veya işlenen değerlerin sayılması gerekebilir. Örneğin klavyeden girilen bir cümlede ka tane sesli harf olduğunu bulan programda, cümlenin her harfi sırasıyla çağrılır ve sesli harfler kümesine ait olup olmadığı araştırılır. Eğer sıradaki çağrılan harf bu kümeye ait ise, bunları sayacak olan değişkenin değeri artırılır.

```
sayac=sayac+1
```

Burada sağdaki ifadede değişkenin eski değerinin üzerine 1 eklenerek bulunan sonuç yine değişkenin kendisine yeni değer olarak atanmaktadır.

Sayaç yapısı:

```
sayac=sayac+adım
```

Sayaç kullanarak 1-5 arası sayıları ekrana yazdıran programın algoritmasını oluşturunuz.

Algoritma Geliřtirmek

Döngü Kullanımı:

Programlardaki belirli işlem bloklarını (kod parçalarını); aynı veya farklı değerlerle, verilen sayıda gerçekleřtiren çevrim yapılarına döngü denir.

Örneğın 1 ile 1000 arasındaki tek sayıları ekrana yazdıracak programda, 1 ile 1000 arasında ikiřer ikiřer artan bir döngü açılır ve döngü deęiřkeni ardışık olarak yazdırılır.

Döngü Oluřturma Kuralları:

- Döngü deęiřkenine başlangıç deęeri verilir.
- Döngünün artma veya azaltma deęeri belirlenir.
- Döngünün bitiş deęeri belirlenir.
- Eđer döngü karar ifadeleri ile oluřturuluyorsa, karar işleminden önce döngü deęiřkenine başlangıç deęeri verilmiş olmalı ve döngü içinde adım miktarı kadar artırılmalıdır/azaltılmalıdır.

Algoritma Geliřtirmek

Ařağıdaki algoritma alıřtırıldıėında ekrana ne yazdıracaktır?

1. Bařla
2. $T=0$
3. $J=1$
4. Eėer $J>10$ ise git 8
5. $T=T+J$
6. $J=J+2$
7. Git 4
8. Yaz T
9. Dur



1. Bařla
2. $T=0$
3. DÖNGÜ ($J=1$ TO 10 STEP 2)
4. $T=T+J$
5. DÖNGÜSONU
6. Yaz T
7. Dur

Algoritma Geliřtirmek

Ardışık Toplama:

Çalışma prensibi sayacinkine benzer, aynı değerin üzerine yeni değerler eklemek için kullanılır. Genel kullanım şekli:

$$\text{Toplam} = \text{Toplam} + \text{sayi}$$

Toplam değişkenine başlangıç değeri olarak 0 atanır.

klavyeden girilen 5 adet sayının ortalamasını bulan program:

1. Başla
2. N=5
3. T=0
4. S=0
5. Eğer S>N-1 ise git 10
6. S=S+1
7. Sayıyı (A) gir
8. T=T+A
9. Git 5
10. Ortalama=T/N
11. Yaz Ortalama
12. Dur

Algoritma Geliřtirmek

Ardışık Çarpma:

Bu işlemde aynı deęer, yeni deęerlerle çarpılarak eskisinin üzerine yazılır:

$$\text{Carpim} = \text{Carpim} * \text{sayi}$$

Carpim deęişkenine başlangıç deęeri olarak 1 atanır.

Klavyeden girilen N sayısının faktöriyelinin hesaplanması:

1. Başla
2. N sayısını al
3. F=1
4. S=0
5. Eğer $S > N-1$ ise git 9
6. $S=S+1$
7. $F=F*S$
8. Git 5
9. Yaz F
10. Dur

Algoritmanın Hazırlanması

- Algoritmadaki adımlar programın sonlu sayıda işlem yapmasını sağlamalıdır.
- Adımlar sıralı ve mantıklı olmalıdır.
- Tekrar eden işlemlerde programın hızı ve etkinliği açısından kaçınılmalıdır.
- Farklı değerlerle gerçekleştirilen aynı işlemleri tekrar tekrar yazmak yerine bunları alt program/fonksiyon olarak oluşturmak daha uygundur.
- Gereksiz işlemlerden kaçınılmalıdır.
- Etkisiz işlemlerden kaçınılmalıdır.
- Bellek verimliliği açısından fazladan veya gereksiz tanımlamalardan kaçınılmalıdır.
- Her algoritmanın bir çıktısı/sonucu olmalıdır.

Algoritma Yazım Kuralları

- Algoritmadaki tüm satırlar 1'den başlayarak numaralandırılmalıdır.
- Bütün algoritmalarda birinci satır "1. Başla" şeklindedir.
- Bütün algoritmalar "Dur" ile biter.
- Algoritmalarda kullanılan "Git" işlem akışı yönlendirme komutu satır numarasıyla birlikte kullanılmalıdır.
- Algoritmalarda kullanılacak alt program veya fonksiyonlar, tanımlayıcı isimleri ve parametreleriyle birlikte verilmelidir.
- Algoritmadaki adımlar, sınırlı sayıda, açık, net ve kesin olmalıdır.
- Algoritmadaki ifadeler anlaşılır ve mümkün olduğunca sade (az ve öz) olmalıdır.
- Algoritmadaki ifadeler, herhangi bir programlama diline, donanıma, işletim sistemine vb. bağlı olmamalıdır.

Sonu



Sorular



Dr. Fatih KALEMKUŞ

TEŞEKKÜRLER

Dr. Fatih KALEMKUŞ