

BİLİŞİM TEKNOLOJİLERİ VE YAZILIM DERSİ DERS NOTLARI-2

Kazanım: 2.4. Algoritma kavramını açıklar.

Kazanım:2.5. Bir problemin çözümü için en doğru algoritmayı geliştirir.

ALGORİTMA

Bir problemin çözümünde izlenecek yol anlamına gelir ve problemin çözümünün adımlar halinde yazılmasıyla oluşturulur. Algoritma basamaklarının bir başlangıcı ve sonu bulunur. Her adımda yapılacak işlem açıkça belirtilir. Bir işi yapmak için gereken talimatlar dizisidir.

ALGORİTMA KURALLARI

- Her algoritma basamak basamak yazılır.
- Her algoritmanın ilk basamağı BAŞLA'dır.
- Her algoritmanın son basamağı BİTİR'dir.
- Algoritmalarda kesin cümleler kullanılır.

Algoritma kullanarak;

- Problemleri daha hızlı ve sistematik olarak çözeriz.
- Problem çözme sürecini takip ederiz ve nerede hata yapıldığını görebiliriz.
- Tüm olasılıkları gözden geçirebiliriz.
- Hatalı işlem yapma olasılığımızı azaltırız.
- Olası hatalarımızı düzeltebiliriz.
- Çözüme ulaşmak için farklı yolları deneyebiliriz.
- Problemin çözümü için harcayacağımız süreyi kısaltırız.

Algoritmalar;

- Açık ve net olmalıdır.
- Kullanılacak olan girdiler iyi tanımlanmış olmalıdır.
- Çıktılar açık ve anlaşılır olmalıdır.
- Algoritmalar hızlı olmalıdır.
- Sonlu ve uygulanabilir olmalıdır.

Algoritma kelimesinin kökeni, 9. Yüzyılda yaşamış ünlü matematikçi ve astronom Ebu Cafer Muhammed bin Musa **el-Harezmi**'ye dayanır.

El-Harezmi, özellikle cebir ve algoritmalar üzerine yaptığı çalışmalarla tanınmıştır.

• Algoritma Terimi: El-Harezmi'nin adı, Latince'de "Algoritmi" veya "Algoritmus" olarak geçmektedir. Bu terim, zamanla "algoritma" biçimine evrilmiştir.

Günümüzde "algoritma", belirli bir problemi çözmek için izlenen sistematik adım dizisi anlamına gelir.

Örnek Algoritma 1: : Evimize gelen arkadaşımıza Türk kahvesi ikram edelim mi?

2 kişilik sade Türk kahvesi hazırlama algoritması aşağıdaki gibi olacaktır:

Adım 1- Başla

Adım 2- Cezveye iki fincanlık su koy

Adım 3- Cezveye 2 tatlı kaşığı kahve koy

Adım 4- Ocağı yak

Adım 5- Cezveyi ocağa koy

Adım 6- Kahveyi karıştır

Adım 7- Bir süre kahvenin olmasını bekle

Adım 8- Kahve köpürmeye başladı mı? Hayır ise 6. Adıma git

Adım 9- Kahveyi fincanlara doldur

Adım 10- Bitir

Örnekte görüldüğü gibi her algoritma bir başlama komutu ile başlar ve bitirme komutu ile sona erer.

Problemi çözmek için problem küçük parçalara bölünür ve hedeflenen sonuca ulaşmak için atılan adımlar net ve uygulanabilir niteliktedir.

Örnek Algoritma 2: Öğretmeninizin sizi soru çözmek için tahtaya kaldırmış olduğunu düşününüz.

Algoritmayı çözmek için işlem basamakları aşağıdaki gibi olacaktır:

Adım 1- Başla

Adım 2- Ayağa kalk

Adım 3- Tahtanın önüne git

Adım 4- Tahta kalemını al

Adım 5- Soruyu çöz

Adım 6- Soruyu doğru çözdün mü? Evet ise 8. Adıma git

Adım 7- Tahtayı sil ve 5. Adıma git

Adım 8- Yerine otur

Adım 9- Bitir

Örnek Algoritma 3: 2 sayıyı toplayan algoritma

1. BAŞLA
2. BİRİNCİ SAYIYI GİR
3. İKİNCİ SAYIYI GİR
4. VE İKİ SAYIYI TOPLA
5. SONUCU GÖSTER
6. BİTİR.

Örnek Algoritma 4: Dikdörtgenin çevresini bulan program algoritması

1. BAŞLA
2. KISA KENARI GİR
3. UZUN KENARI GİR
4. KENARLARI TOPLA
5. SONUCU 2 İLE ÇARP
6. ÇIKAN SONUCU EKRANDA GÖSTER
7. BİTİR

Örnek Algoritma 5: Ayran yapma algoritmasını oluşturunuz.

Adım 1: Başla

Adım 2: Yoğurdu kaba koy.

Adım 3: Su ekle.

Adım 4: Çırp.

Adım 5: Tuz koy.

Adım 6: Bardağa doldur.

Adım 7: Bitir.

Örnek Algoritma 6: Bulaşık yıkama (makinede değil elde yıkama) algoritmasını oluşturunuz.

Adım 1: Başla

Adım 2: Artıkları sıyır.

Adım 3: Sudan geçir.

Adım 4: Sabunla.

Adım 5: Durula.

Adım 6: Kurut.

Adım 7: Bitir.

Örnek Algoritma 7: Bir kenar uzunluğu bilgisayara girilen karenin çevresini hesaplayan programın algoritmasını yazınız.

- Adım 1: Başla
- Adım 2: Karenin kenar uzunluğunu al
- Adım 3: $a = \text{Kenar uzunluğu}$
- Adım 4: $\text{cevre} = 4 * a$
- Adım 5: Sonucu (cevre) ekranda göster.
- Adım 6: Bitir.

Örnek Algoritma 8: Yaşı girilen bir kişinin otomobil ehliyeti alıp alamayacağını belirleyen programın algoritmasını hazırlayınız.

- Adım 1: Başla
- Adım 2: Yaş değerini al
- Adım 3: $y = \text{Yaş değeri}$
- Adım 4: Eğer $y < 18$ ise "Ehliyet alamaz" yaz.
- Adım 5: Değilse "Ehliyet alabilir." Yaz.
- Adım 6: Bitir.

Örnek Algoritma 9 Arabayı çalıştırıp yola çıkalım.

- Adım 1: Başla
 - Adım 2: Sürücü koltuğuna geç.
 - Adım 3: Emniyet kemerini tak.
 - Adım 4: Aynaları kontrol et.
 - Adım 5: Anahtarı tak.
 - Adım 6: Kontaklı çevir.
 - Adım 7: El frenini indir.
 - Adım 8: Vitese geç.
 - Adım 9: Gaza bas.
-

Kazanım : 2.6. Programlamanın önemini yorumlar.

Programlama dilleri, insanlar tarafından yazılan komutların bilgisayarlar tarafından anlaşılmasını sağlar. Programlama dili, kısaca cihazın anladığı dildir. Programlama sürecinde belirlenen algoritmaya göre hangi komutların kullanılacağını belirler. Programlama sürecinde verinin nerede depolanacağı, nereden okunacağı, fonksiyon tanımlamalarının yapılması gibi işlemler programlama dili ile yapılır. İlk defa 1883 yılında "Analitik Makine" adı altında bir mekanik bilgisayarda kullanılır. 1952 ile birlikte programlama dilleri geliştirilmeye başlanır. İlk dönemlerde yaygın olarak Fortran, Pascal gibi programlama dilleri kullanılırken daha sonra C++, C#, PHP, Python, Ruby gibi diller geliştirilir. Günümüze gelindiğinde 250'den fazla programlama dili mevcuttur.

Programlama Neden Önemlidir?

Programlama, birçok soruna çözüm getirmesinin yanında günlük hayattaki işleri kolaylaştırır. Sosyal medya paylaşımı yapabilmenizden çamaşır yıkamaya kadar kişisel ve işlevsel görevleri yerine getirir. İş dünyasına getirdiği çözümler ile üretkenliğin artırılmasına yardımcı olur. Teknolojik altyapıların gelişmesiyle programlama daha da önem kazanır. Modern teknoloji için bilgisayar programlama dilleri çok önemlidir. Bu dillerin sağladığı temel altyapı, günlük olarak kullandığımız yazılımlar, sistemler ve uygulamalar için gereklidir.

Programlama dillerine göre programlamanın neden önemli olduğunu maddeler halinde inceleyecek olursak:

#C gibi programlama dilleri robot teknolojilerinin gelişmesini sağlar.

Java, Python gibi diller, nesnelerin interneti teknolojisine ait projelerin ilerlemesini mümkün kılar.

Swift gibi diller ile mobil uygulamalar geliştirilir.

Python, C++ dilleri yapay zeka teknolojileri için gereklidir.

Kazanım: 2.7. Programlama dillerinin gelişimini açıklar.

Programlama Dillerinin Gelişimi

1843: Ada Lovelace'in makine algoritması

İlk programlama dili Ada Lovelace'ın makine notlarıdır. Bu notlar bir bilgisayar tarafından işlenmek üzere yazılan ilk algoritmayı içerir.

1949: Assembly Dili

Assembly Dili, ilk olarak [Elektronik Gecikmeli Depolama Otomatik Hesaplayıcı](#)'da kullanıldı. Bu dil, bilgisayara ne yapması gerektiğini söylemek için gereken özel talimatlar olan makine kodunun dilini basitleştiren bir tür düşük seviyeli bilgisayar programlama dilidir.

1957: FORTRAN

FORMula TRANslation veya FORTRAN, üst düzey bilimsel, matematiksel ve istatistiksel hesaplamalar için oluşturulmuştur. FORTRAN, bugün hala dünyanın en gelişmiş süper bilgisayarlarında kullanılmaktadır.

1964: BASIC

Yeni Başlayanlar Çok Amaçlı Sembolik Öğretim Kodu olarak adlandırılan BASIC, Dartmouth College'daki öğrenciler tarafından geliştirilmiştir. Microsoft'un kurucuları Bill Gates ve Paul Allen tarafından daha da geliştirilerek şirketin ilk pazarlanabilir ürünü haline gelmiştir.

1991: Python

Çeşitli programlama dillerini desteklemek amacıyla kullanılan genel amaçlı, yüksek seviyeli bir programlama dilidir. Python, bugüne kadar dünyanın en popüler programlama dillerinden biri Google, Yahoo! ve Spotify gibi büyük şirketler tarafından kullanılıyor.

1995: Java

Java, James Gosling tarafından etkileşimli bir TV projesi için oluşturulmuş genel amaçlı, yüksek seviyeli bir dildir. Cross platform işlevselliğine sahiptir.

Java, bilgisayarlardan akıllı telefonlara kadar birçok platformda geliştirme yapma imkanı sağlar.

2009: Go

Go, Google tarafından büyük yazılım sistemlerinden kaynaklanan sorunları gidermek için geliştirilmiştir. Go; Google, Uber, Twitch ve Dropbox gibi dünyanın en büyük teknoloji şirketlerinden bazıları arasında popülerlik kazanmıştır.

2014: Swift

Apple tarafından C, C++ ve Objective-C'nin yerini almak üzere geliştirilen Swift, yukarıda belirtilen dillerden daha kolay olması ve hata için daha az yer bırakması amacıyla geliştirildi. Swift'in çok yönlülüğü, masaüstü, mobil ve bulut uygulamaları için kullanılabilmesi anlamına gelir. Önde gelen dil uygulaması 'Duolingo', Swift ile yazılmış yeni bir uygulamayı geliştirmiştir.

Kazanım: 2.8. Akış diyagramının kullanım amaçlarını açıklar.

AKIŞ DİYAGRAMLARI (ŞEMALARI)

Bilgisayar programlarının işlem basamaklarını geometrik şekillerle gösteren şemadır. Algoritmanın daha rahat anlaşılabilmesi için şemalarla gösterilmesidir. Şemada yer alan her şeklin bir kullanım amacı vardır.

Algoritma; yapılacak olan işlerin sadece yazı ile (herhangi bir şema veya şekil kullanılmadan) maddeler halinde yazılmasıdır. Oysa ki akış şeması Bir sürecin (*Maddeler halinde yazılan algoritmanın*) adımlarını görsel ya da sembolik olarak gösterir.



Algoritma

Akış Şeması

KEK TARİFİ

ALGORİTMA

BAŞLA

1. Fırını 170°C'ye getir.
2. 2 yumurtayı 1 bardak şekerle çırp.
3. 1 çay bardağı yoğurt ve 1 çay bardağı zeytinyağı ekleyerek karıştır.
4. 2 su bardağı un ekle ve karıştır.
5. Eğer elinde fındık ve kakao yoksa 7. adıma git.
6. Fındık ve kakao ekle ve 9. adıma git.
7. Elinde üzüm yoksa 9. adıma git.
8. Kuru üzüm ekle ve 9. adıma git.
9. Malzemeleri karıştır ve kabartma tozu ekle.
10. Kek kalıbını yağla.
11. Malzemeleri kalıba dök.
12. Fırına koy ve 45 dk pişir.
13. Fırından çıkar.

BITİR

KEK TARİFİ

AKIŞ ŞEMASI

Başla

Fırını 170°C'ye getir

2 yumurtayı 1 bardak şekerle çırp

1 çay bardağı yoğurt ve 1 çay bardağı zeytinyağı ekleyerek karıştır

2 su bardağı un ekle ve karıştır

Fındık ve kakao var mı?

Ekle

Kuru üzüm var mı?

Ekle

Malzemeleri karıştır ve kabartma tozu ekle

Kek kalıbını yağla

Malzemeleri kalıba dök

Fırına koy ve 45 dakika pişir

Fırından çıkar

Bitir

UNUTMA !

Algoritma ve Akış şemaları aynı problemi çözmek için kullanılabilir, Ama senden sınavda veya uygulamada

- Algoritma istenirse yazılı ve maddeler halinde
- Akış şeması istenirse şekil ve sembollerle oluşturman gerekir.

KEK TARİFİ

ALGORİTMA

BAŞLA

1. Fırını 170°C'ye getir.
2. 2 yumurtayı 1 bardak şekerle çırp.
3. 1 çay bardağı yoğurt ve 1 çay bardağı zeytinyağı ekleyerek karıştır.
4. 2 su bardağı un ekle ve karıştır.
5. Eğer elinde fındık ve kakao yoksa 7. adıma git.
6. Fındık ve kakao ekle ve 9. adıma git.
7. Elinde üzüm yoksa 9. adıma git.
8. Kuru üzüm ekle ve 9. adıma git.
9. Malzemeleri karıştır ve kabartma tozu ekle.
10. Kek kalıbını yağla.
11. Malzemeleri kalıba dök.
12. Fırına koy ve 45 dk pişir.
13. Fırından çıkar.

BITİR

KEK TARİFİ

AKIŞ ŞEMASI

Başla

Fırını 170°C'ye getir

2 yumurtayı 1 bardak şekerle çırp

1 çay bardağı yoğurt ve 1 çay bardağı zeytinyağı ekleyerek karıştır

2 su bardağı un ekle ve karıştır

Fındık ve kakao var mı?

Ekle

Kuru üzüm var mı?

Ekle

Malzemeleri karıştır ve kabartma tozu ekle

Kek kalıbını yağla

Malzemeleri kalıba dök

Fırına koy ve 45 dakika pişir

Fırından çıkar

Bitir

Kazanım: 2.9. Temel akış diyagramı şekillerini ve elemanlarını listeler.



ELİPS

Akışı başlatır ve bitirir.

Akış şemalarının başlangıç ve bitişlerinde mutlaka kullanılması gerekir. Algoritmada başla ve bitir ibaresi olmasa bile mutlaka yerleştirmeniz gerekir. «Bitir» veya «Bitiş» şeklinde belirtmediğiniz zaman akış şeması sürekli bunu gerçekleştirmeye çalışır.



DIKDÖRTGEN

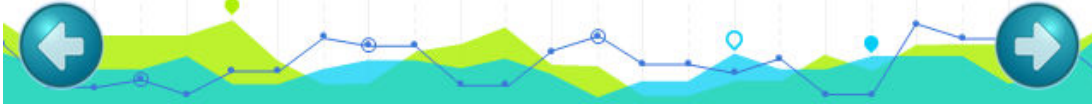
Eylemi/işlemi belirtir.

Yapacağınız tüm eylem, iş ve işlemler dikdörtgen ile yapılır.

Örneğin;

- Yataktan kalk,
- Sıraya otur,
- Ortalama Hesaplama,
- 3+5,
- Trafik ışıklarına bak,

Gibi iş ve işlemlerin tamamı dikdörtgen içine alınmalıdır.





PARALELKENAR

Dışarıdan bilgi/veri girişini belirtir

Kullanıcıdan girilen değerler bu şekil ile alınır.

Örneğin:

Kullanıcının girdiği iki değer ile deniyorsa;

Yaşını giren kullanıcının deniyorsa

Girilen iki sınavın deniyorsa

İsmi giren kişinin deniyorsa

Gibi alınacak veriler için paralelkenar kullanılır.



EŞKENAR DÖRTGEN

Karar verme merkezidir.

Karar vermemiz gereken yerlerde kullanılır:

○ Ortalama 50'den büyükse, *Gibi kararlar vermek için kullanılır.*

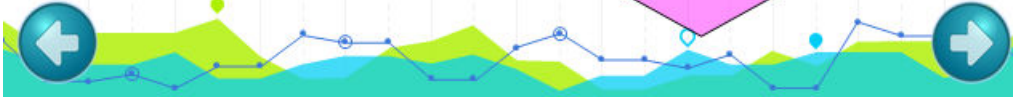
○ Okul adı Symbül ise,

Evet / Hayır

○ Sınıfı 9 ise,

Doğru/Yanlış gibi iki kola mutlaka ayrılması gerekir.

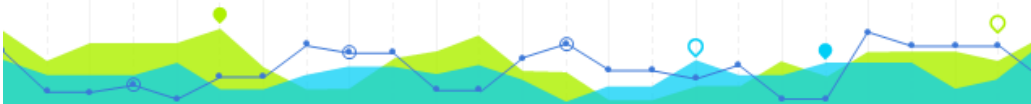
○ Işık yeşil yanıyor ise,



ALTIGEN

Tekrar eden komutları belirtir.

Tekrarlanacak işlemler (döngü) için kullanılır.





DALGALI DÖRTGEN

Ekran / yazıcı çıktısı.

Ekrana bir çıktı vermek istediğimizde kullanılır.

Örneğin;

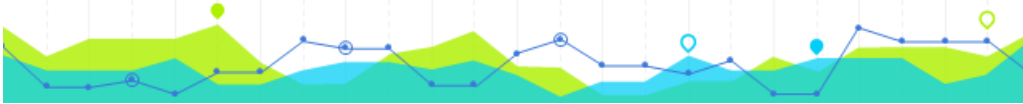
- ✓ İsmi ekrana yazdırın;
- ✓ Ekran 5 Yazdırın;
- ✓ Geçti yazar;
- ✓ Kaldı yazar,

Gibi çıktı sonuçları için kullanılır.



Yön Okları

Akış şemasının ilerleme yönünü gösterir



Örnek: Girilen iki sayının çıkarma işlemini yapan ve sonucu yeni bir değişkende saklayarak yazdıran programın akış şemasını oluşturunuz.



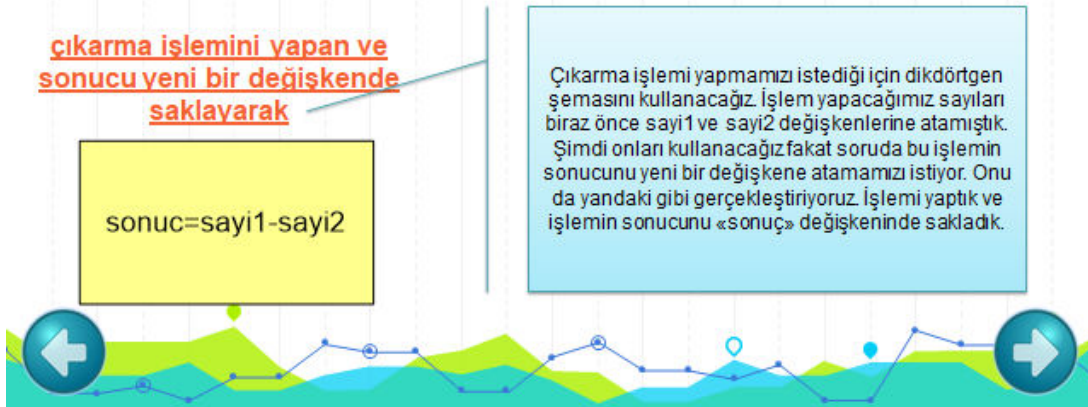
Çözüm adımları anlayarak başlayın: Bizden istediklerini gözden geçirelim:

Örnek: Girilen iki sayının çıkarma işlemi yapan ve sonucu yeni bir değişkende saklayan yazdıran programın akış şemasını oluşturunuz.



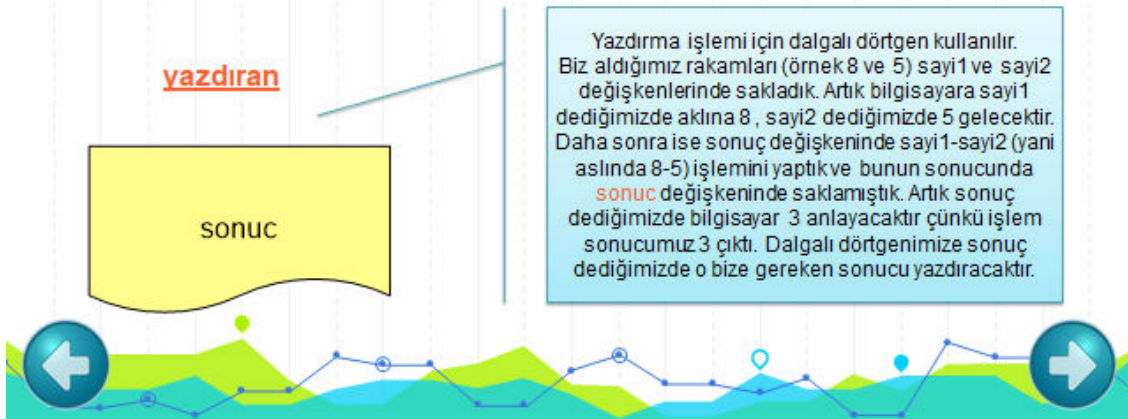
Çözüm adımları anlayarak başlayın: Bizden istediklerini gözden geçirelim:

Örnek: Girilen iki sayının çıkarma işlemi yapan ve sonucu yeni bir değişkende saklayan yazdıran programın akış şemasını oluşturunuz.

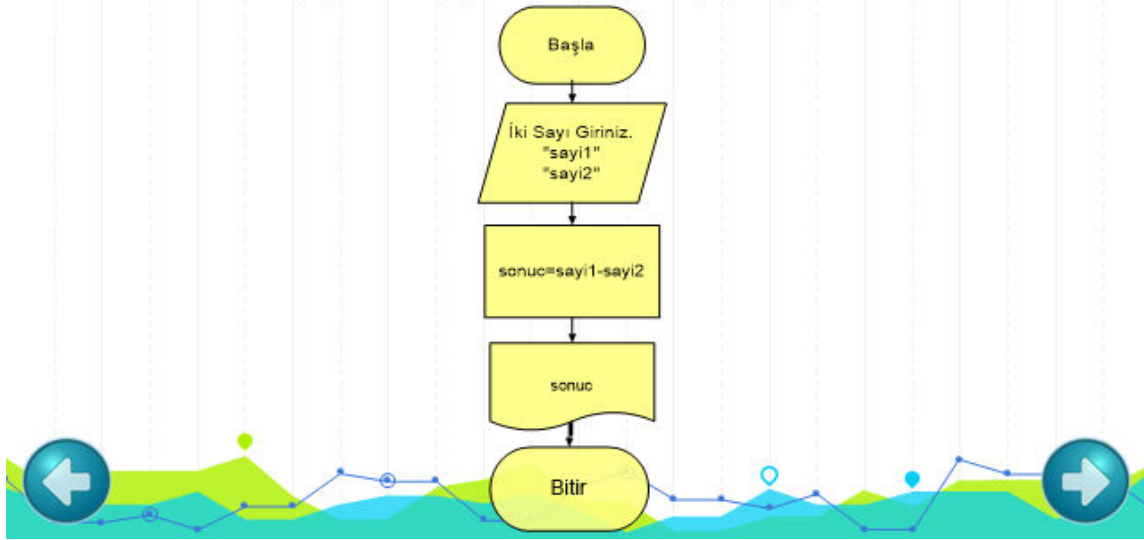


Çözüm adımları anlayarak başlayın: Bizden istediklerini gözden geçirelim:

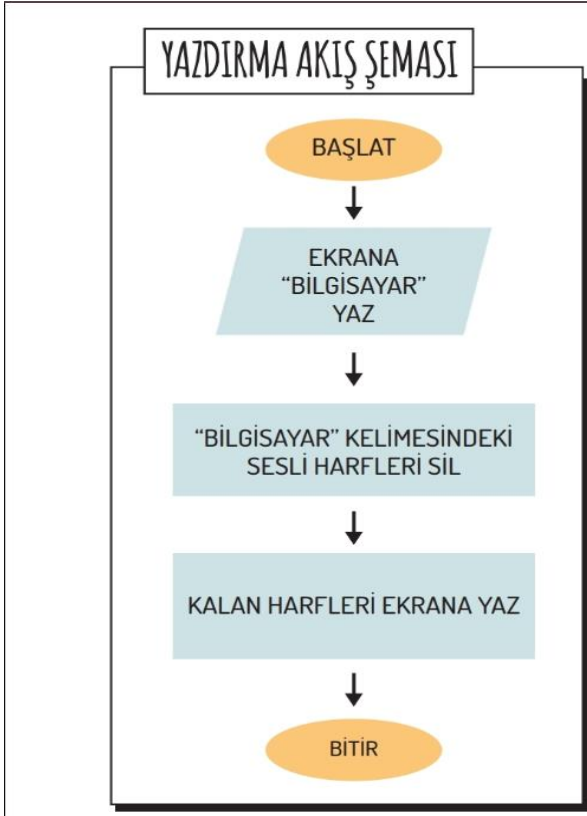
Örnek: Girilen iki sayının çıkarma işlemi yapan ve sonucu yeni bir değişkende saklayan yazdıran programın akış şemasını oluşturunuz.



Örnek: Girilen iki sayının çıkarma işlemini yapan ve sonucu yeni bir değişkende saklayarak yazdıran programın akış şeması şu şekilde ortaya çıkmıştır:



Kazanım: 2.10. Var olan bir akış diyagramını okuyarak problemi kavrar.



YAZDIRMA AKIŞ ŞEMASINDA kullanıcı bilgisayar kelimesini girdiğinde bilgisayar sesli harfleri siler ve ekrana BLGSYR kelimesi yazdırılır.



SIRALAMA AKIŞ ŞEMASINDA kullanıcı ekrana adını yazar örnek BÜŞRA

Bilgisayar harfleri sondan başa doğru sıralar ve ekrana yeni kelimeyi yazar örnek ARŞÜB



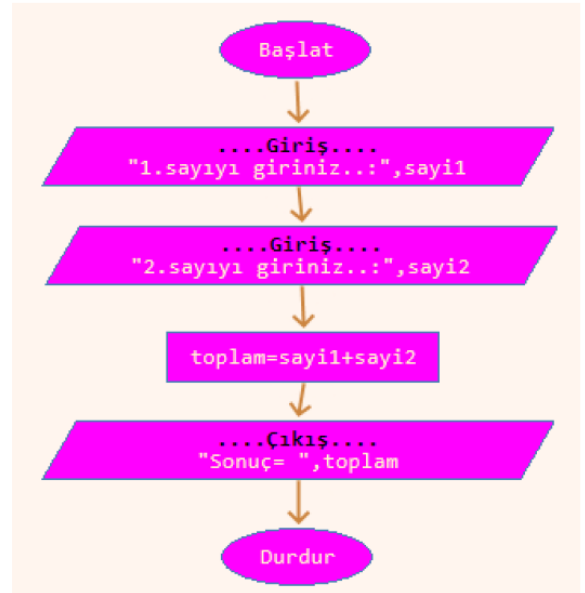
KARŞILAŞTIRMA AKIŞ ŞEMASINDA kullanıcı bir sayı girer
 Girilen sayı beşten büyükse programı bitirir
 değilse (örnek: 6)
 sayıyı 5 ile toplar (örnek $6+5=11$)
 sonucu ekranda gösterir (örnek 11)

Kazanım: 2.11. Problemin akış diyagramını tasarlar.

ÖRNEKLER:

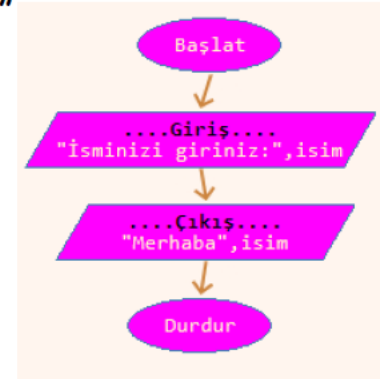
1. Girilen iki sayıyı toplayıp yazdıran programın algoritması ve akış şemasını çizelim

- Adım1: Başla
- Adım2: Birinci sayıyı gir(sayı1)
- Adım3: İkinci sayıyı gir(sayı2)
- Adım4: $toplam=sayı1+sayı2$
- Adım5: Yazdır toplam
- Adım6: Bitir



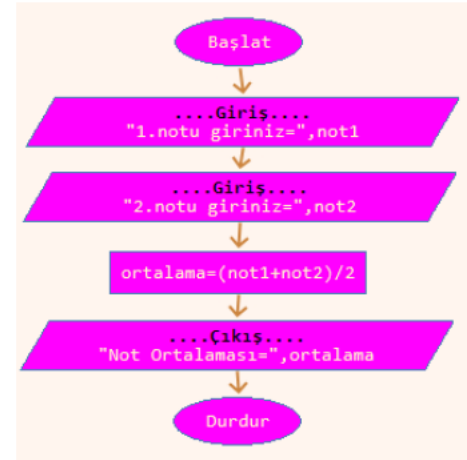
11. Klavyeden ismini giren kişiyi "Merhaba isim" şeklinde selamlayan programın algoritması ve akış şemasını çizelim.

- Adım1: Başla
- Adım2: İsmimizi giriniz(isim)
- Adım3: Yazdır "Merhaba", isim
- Adım4: Bitir



12. 1.sınavı ve 2.sınavı girilen öğrencinin not ortalamasını bulup, ortalamayı ekranda gösteren algoritmayı yazınız.

- Adım1: Başla
- Adım2: Birinci sınav notunu gir(not1)
- Adım6: İkinci sınav notunu gir(not2)
- Adım7: $ort=(not1+not2)/2$
- Adım8: Yazdır ort
- Adım9: Bitir



7. Yarıçapı girilen dairenin alanını bulan algoritmayı yapınız.($\pi=3$ olarak kabul edilecek ve bu değer pi adında bir değişkene atanacaktır)

- Adım1: Başla
- Adım2: $\pi=3$
- Adım3: Yarıçapı giriniz(r)
- Adım4: $alan=\pi*r*r$
- Adım5: Yazdır alan
- Adım6: Bitir



3. Girilen 3 sayının toplamını ve ortalamasını bulan ve bu sonuçları ekrana yazdıran programın algoritması ve akış şemasını çizelim.

Adım1: Başla

Adım2: Birinci sayıyı gir(sayı1)

Adım3: İkinci sayıyı gir(sayı2)

Adım4: Üçüncü sayıyı gir(sayı3)

Adım5: $toplam=sayı1+sayı2+sayı3$

Adım6: $ortalama=toplam/3$

Adım7: Yazdır toplam

Adım8: Yazdır ortalama

Adım9: Bitir

