

6. SINIF 2. DÖNEM 1. SINAV ÇALIŞMA DOKÜMANI

Problem Çözmede Sabitler ve Değişkenler

Bilgisayarların işleyişinde bazı bilgiler değişkenler aracılığı ile saklanırken bazı bilgiler ise sabit olarak kalır. Sabit olarak kalan bu bilgiler ne olursa olsun değişmez ve ilk depolandıkları hali ile kullanılırlar.

Sabit: İlk biçimiyle kalan, değişmeyen ifade ve nesnelere.

Değişken: İlk biçimiyle kalmayan, yeni değerler alabilen ifade ve nesnelere.

Günlük hayatta karşılaştığımız problemlerin çözümünde de sabitler ve değişkenler kullanılır.

Basketbol Oyununda Sabit ve Değişkenler

TÜR	SAYI	AÇIKLAMA
Sabit	5	Her takımdaki oyuncu sayısı
Sabit	1	Top sayısı
Sabit	2	Pota sayısı
Sabit	3	Hakem sayısı
TÜR	SAYI	AÇIKLAMA
Değişken	Skor	Atılan basket sayısı
Değişken	Faul	Maçta yapılan faul sayısı
Değişken	Seyirci	Maç izleyen seyirci sayısı
Değişken	Takımlar	Maç yapan takım isimleri



Operatörler ve Eşitlikler

Problemin çözümünde kullanılan, matematiksel işlemler ve mantıksal karşılaştırmalar yapmaya yarayan sembollerdir.



Matematiksel işlemlerin yapılması için kullanılan sembollerdir.

$$3 + 2 = 5$$

$$2 \times 5 = 10$$

Mantıksal Operatörler ve Eşitlikler

Günlük hayatta karşılaştığımız problemlerde veya bilgisayar programları oluşturulurken; Ve - Veya - Değil - Eşittir - Büyüktür - Küçüktür - Büyük veya Eşit - Küçük veya Eşit gibi mantıksal operatörler kullanılabilir.



Elektrik varsa **VE** bilgisayarım çalışıyorsa ödevimi hazırlayabilirim.



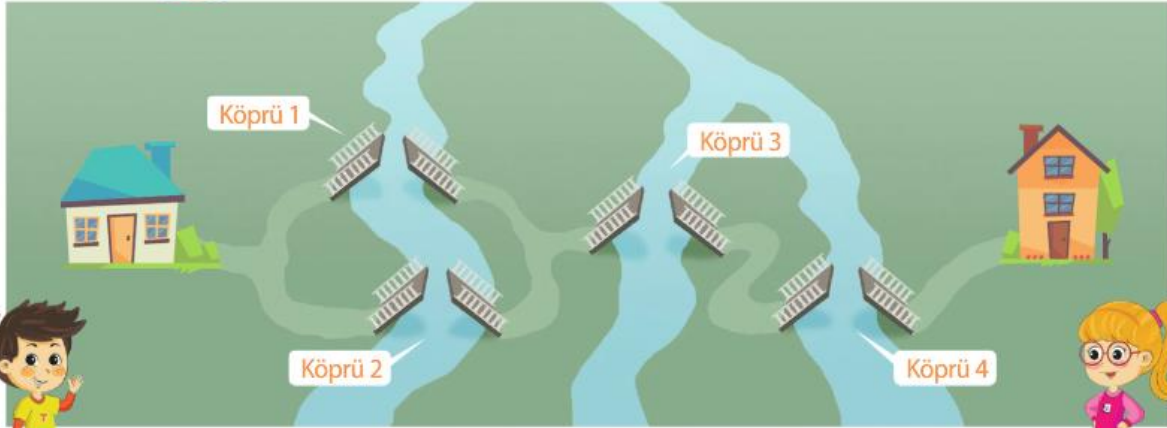
Karne ortalaması 70'den **KÜÇÜK** ise teşekkür belgesi alamaz.

Sağlık ocağında doktor **VEYA** hemşire varsa tedavi olabilirim.

Sınav puanı 85'den **BÜYÜK** veya **EŞİT** ise başarılıdır.

Kapı anahtarı doğru **DEĞİL** ise kapıyı açamam.

Ahmet'in yaşı 18'den **BÜYÜK** ise ehliyet alabilir.



Tekin, Bilge'nin yanına ders çalışmaya gitmek istiyor. Bilge'nin evinin bulunduğu yere varabilmesi için aşağıdaki mantıksal ifadelerden doğru olanları işaretleyiniz.

DOĞRU YANLIŞ

<input type="checkbox"/>	<input checked="" type="checkbox"/>	Köprü 1 Kapalı VE Köprü 2 VEYA Köprü 3 Kapalı VE Köprü 4 Kapalı
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Köprü 1 Kapalı VE Köprü 2 Kapalı VE Köprü 4 Kapalı
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Köprü 1 Kapalı VE Köprü 3 Kapalı VE Köprü 4 Kapalı
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Köprü 1 Kapalı VEYA Köprü 2 Kapalı VE Köprü 4 Kapalı
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Köprü 1 Açık VE Köprü 2 Kapalı VE Köprü 4 Kapalı
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Köprü 2 Kapalı VE Köprü 3 Kapalı VE Köprü 4 Kapalı

Veri Türlerini Belirleme

KARAKTER VERİ TİPİ

Tüm tek haneli sayıları, harfleri ve özel karakterleri kapsar. Tırnak içinde belirtilir. Bilgisayar sözcüğündeki "B" harfi bir karakteri ifade eder.

SAYISAL VERİ TİPİ

Hesaplama işlemlerinde kullanılır. Tüm sayı çeşitlerini içerir. Pozitif ya da negatif tam sayılar kullanılabilir. Örneğin; açılar, uzaklık, nüfus, ücret, yarıçap...

KARAKTER DİZİSİ VERİ TİPİ

Birden fazla karakterin bir araya gelmesiyle karakter dizisi oluşur. Örneğin "Bilgisayar" sözcüğü bir diziyi ifade eder.

gırdığını seçiniz.

- Karakter Dizisi V... ✓ İsim
- Karakter Veri Tipi ✓ Soy İsmi'nin İlk Harfi
- Karakter Veri Tipi ✓ Cinsiyet (K/E)
- Sayısal Veri Tipi ✓ Okul Numarası
- Özel Veri Tipi ✓ Doğum Tarihi ve Saati
- Sayısal Veri Tipi ✓ Yaş
- Sayısal Veri Tipi ✓ Boy (cm)
- Özel Veri Tipi ✓ T.C. Kimlik Numarası
- Özel Veri Tipi ✓ Arabanın/ Servisin Plakası
- Mantıksal Veri Tipi ✓ Elma Sever Misin? (E/H)
- Karakter Dizisi V... ✓ En İyi Arkadaşın Kim?
- Özel Veri Tipi ✓ Okul Adresin Nedir?

12/12

MANTIKSAL VERİ TİPİ

Evet ya da hayır şeklindeki karar verme süreçlerinde kullanılır. Örneğin; arabası var mı, lise mezunu mu?

ÖZEL VERİ TİPİ

Tarih, saat, adres, banka hesap numarası gibi verileri temsil eder.

Örnek 4: 11011000 sayısı var. Bunu onlu sisteme çevirelim.

1	1	0	1	1	0	0	0
1×2^7	1×2^6	0×2^5	1×2^4	1×2^3	0×2^2	0×2^1	0×2^0
= 128	= 64	= 0	= 16	= 8	= 0	= 0	= 0

$$128 + 64 + 16 + 8 = \underline{216}$$

128	64	32	16	8	4	2	1
1	0	0	1	1	0	1	1
<hr/>							
128 + 0 + 0 + 16 + 8 + 0 + 2 + 1							
= 155							

(55)₁₀ Sayısının ikilik sayı sisteminde karşılığını bulalım:

2'lik sayı sistemindeki basamak değerleri 1 ler 2ler 4ler 8ler 16lar 32ler 64ler... şeklinde devam etmektedir. O zaman 55 sayısını 2lik sistemde yazarken ona en yakın basamak değerinden başlayarak basamak açacağız burada 55'ten küçük ona en yakın sayı 32'dir. O zaman açacağımız en büyük basamak 32 olacak. 16 lar basamağını da açacağız. Toplam 48 oldu. 55 e gitmemiz için 7 daha lazım kalan basamaklardan 4, 2 ve 1ler basamağını da açtık. 8 basamağını ise kapattık.

$$(55)_{10} = \left(\frac{1}{32} \frac{1}{16} \frac{0}{8} \frac{1}{4} \frac{1}{2} \frac{1}{1} \right)_2$$

İşlem Önceliği

İŞLEM SIRASI

- Üs alma işlemleri
- Parantez içindeki işlemler
- Çarpma veya Bölme İşlemi
- Toplama veya Çıkarma İşlemi

Eğer aynı önceliğe sahip işlemler varsa (Örneğin bir işlemde hem çarpma hem de bölme varsa) işlemler soldan sağa doğru yapılır.

ÖRNEK: $120 / 2^2 + 3 * (12 - 7)$ işleminin sonucunu bulalım.

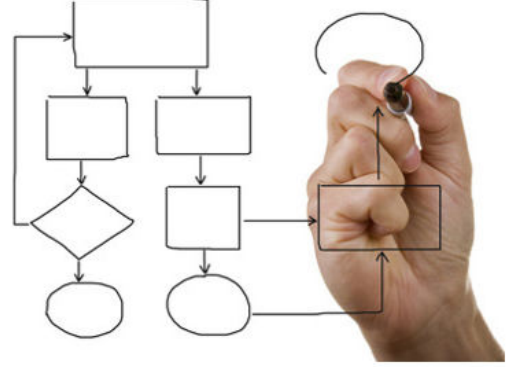
$$\begin{aligned} &= 120 / 4 + 3 * (12 - 7) \\ &= 120 / 4 + 3 * (5) \\ &= 30 + 15 \\ &= 45 \end{aligned}$$

ALGORİTMA

Kodlamaya başlamadan önce oluşturacağımız yazılımın adım adım ne yapacağını tasarlamamız gerekir.

İşte açık ve net ifadelerle problemin adım adım çözümünü gösteren bu taslağa «**algoritma**» adı verilir.

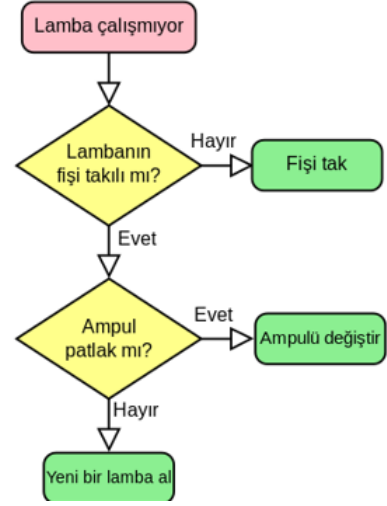
Programlamanın ilk adımı algoritma oluşturmaktır.



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.



Örnek Ayran Algoritma

Şimdi basit bir problemin çözümünü gösteren bir algoritma hazırlayalım.



Ayran yapıp bardağa dolduralım.

- 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 1: Bilgisayar kapatma işlemi algoritma şeklinde aşağıda verilmiştir.

- Adım 1** Başla.
- Adım 2** Başlat menüsüne tıkla.
- Adım 3** Kapat seçeneğini tıkla
- Adım 4** Bilgisayarın kapandığını kontrol et.
- Adım 5** Elektrik fişini çek.
- Adım 6** Bitir.



Örnek 3: Bir çay demleme algoritması aşağıda verilmiştir.

- Adım 1** Başla.
- Adım 2** Çaydanlığa su doldur.
- Adım 3** Ocağı yak.
- Adım 4** Çaydanlığı ocağın üzerine koy.
- Adım 5** Su kaynayıncaya kadar bekle.
- Adım 6** Ocağı kapat.
- Adım 7** Çaydanlığa çay ekle.
- Adım 8** Çay demleninceye kadar bekle.
- Adım 9** Bitir.



Algoritmada Değişkenler

Kod yazarken girilen bilgilerin, hesaplanan sonuçların saklandığı alanlara **değişken** denir. Aşağıdaki örneklerde algoritmada değişkenlerin nasıl kullanıldıkları gösterilmiştir.

İki sayının toplanması algoritması

- Adım 1** Başla.
- Adım 2** Birinci sayıyı gir.
- Adım 3** İkinci sayıyı gir.
- Adım 4** İki sayıyı topla.
- Adım 5** Sonucu ekrana yazdır.
- Adım 6** Bitir.

Değişken kullanılarak oluşturulan algoritma

- Adım 1** Başla.
- Adım 2** Birinci sayıyı gir, **S1**.
- Adım 3** İkinci sayıyı gir, **S2**.
- Adım 4** **Toplam = S1 + S2**.
- Adım 5** **Toplam** değerini ekrana yazdır.
- Adım 6** Bitir.



Yukardaki algoritmaların her ikisi de **iki sayının toplanması algoritmasıdır**.

- Adım 1** Başla.
- Adım 2** Birinci sayıyı gir, **S1**.
- Adım 3** İkinci sayıyı gir, **S2**.
- Adım 4** **Toplam = S1 + S2**.
- Adım 5** Toplam değerini ekrana yazdır.
- Adım 6** Bitir.

Girilen 1. sayı değeri **S1** adındaki değişkene kaydedildi.

Girilen 2. sayı değeri **S2** adındaki değişkene kaydedildi.

İki sayının toplamı **Toplam** adındaki değişkene kaydedildi.

Toplam değişkeninin değeri ekrana yazdırıldı.

Değişken isimleri **istenen ad ile tanımlanabilir**. Yukardaki örnekte S1, S2 ve Toplam isimli değişkenler kullanıldı. Yanda aynı işlemin farklı değişken adları ile yapılabileceği gösterilmiştir.



$$\text{Toplam} = \text{sayi1} + \text{sayi2}$$

$$\text{Toplam} = a + b$$

$$\text{Sonuc} = \text{ilkdeger} + \text{ikincideger}$$

$$t = S1 + S2$$

$$\text{TOP} = \text{SAY1} + \text{SAY2}$$

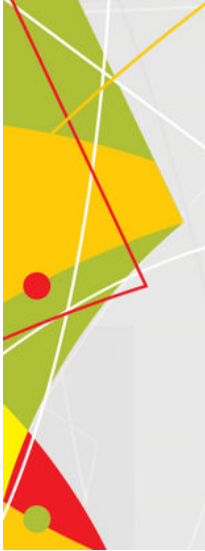
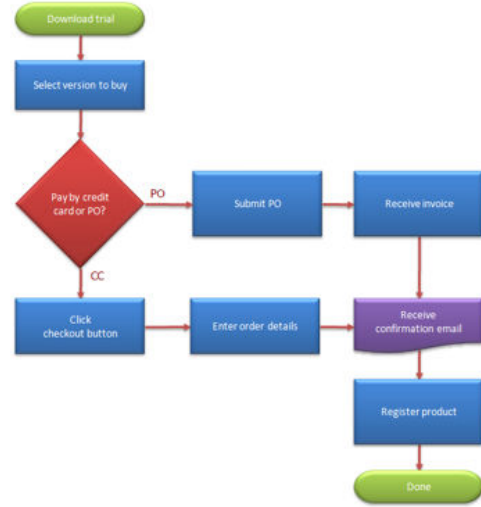
Yukanda yapılan işlemler ve sonuçları aynıdır. Sadece belirlenen değişken adları farklıdır.

Akış Şeması

Bilgisayar programlarının işlem basamaklarını **geometrik şekillerle** gösteren şemadır.

Algoritmanın daha rahat anlaşılabilmesi için şemalarla gösterilmesidir.

Şemada yer alan her şeklin bir kullanım amacı vardır.



Elips

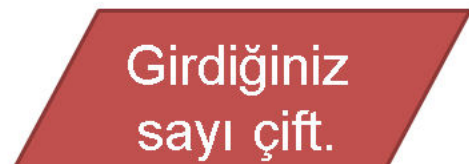
Başla ve **Bitir** adımları için kullanılır. Akış şemasının başlangıç ve bitiş noktasında yer alır.



Paralel Kenar

Giriş ya da **Çıkış** işlemleri için kullanılır.

Örneğin; klavyeden bir sayı girilmesi istenmesi veya ekrana işlem sonucunun yazdırılması gibi.





Dikdörtgen

Hesaplama ya da **Değişkene Değer Atama** işlemleri için kullanılır.

Örneğin; iki sayıyı topla veya girilen ilk sayıyı A olarak kabul et.

A ile B'yi topla.

İlk sayı = A



Eşkenar Dörtgen

Karşılaştırma ya da **Karar Verme** işlemleri için kullanılır.

Örneğin; girilen sayı 5'ten büyük mü?

Kalan süre
0'dan büyük mü?

Oyunda başka elma var mı?



DALGALI DÖRTGEN

Yazdırma işlemleri için kullanılır.

Örneğin bir işlemin sonucu ekrana yazdırılacaksa veya yazıcı çıktısı alınacaksa Dalgalı Dörtgen kullanılabilir.

Aynı işlem için Paralel Kenarda kullanılabilir.

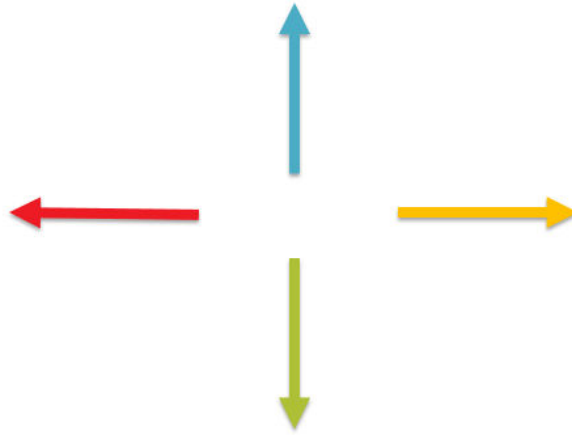
Sonucu ekrana yaz

Sonucu ekrana yaz



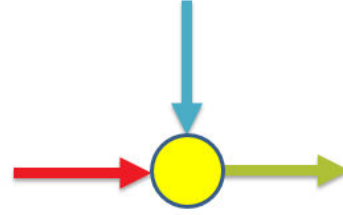
Yön Okları

Akış şemasının ilerleme yönünü gösterir.



Akış Diyagramı Bağlayıcı

Akış şeması oklarını yönlendirirken birbirine bağlamak için kullanılan elemanlardır.



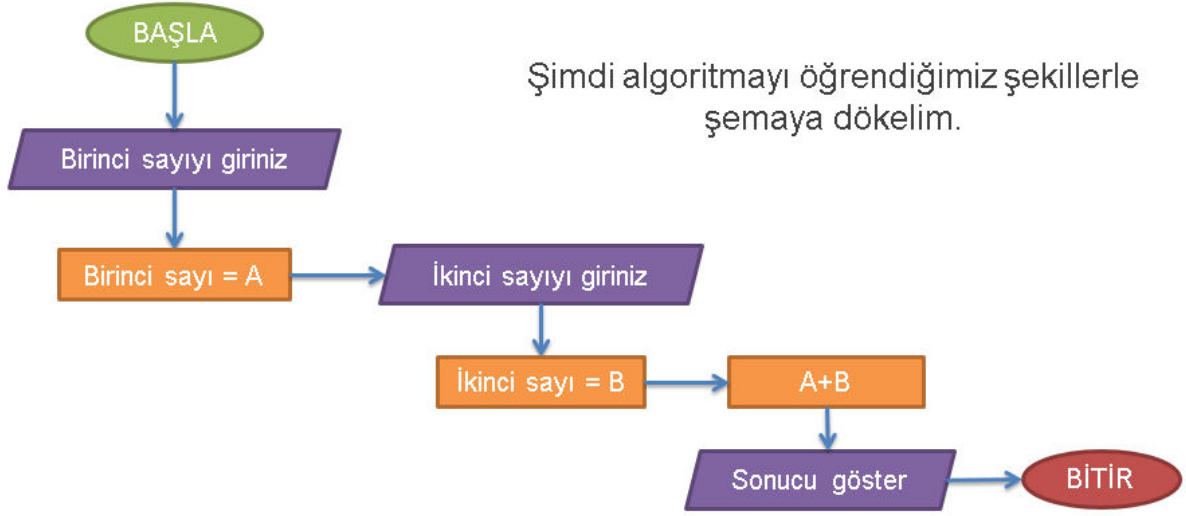
Akış Şeması Örneği - 1

Klavyeden girilen iki sayıyı toplayıp sonucu ekrana yazdıran programın akış şemasını ve algoritmasını hazırlayalım.

- Adım 1: Başla
- Adım 2: İlk sayıyı gir.
- Adım 3: İlk sayı = A
- Adım 4: İkinci sayıyı gir.
- Adım 5: İkinci sayı = B
- Adım 6: İki sayıyı topla (A+B)
- Adım 7: Sonucu ekranda göster.
- Adım 8: Bitir.



Akış Şeması Orneđi - 2

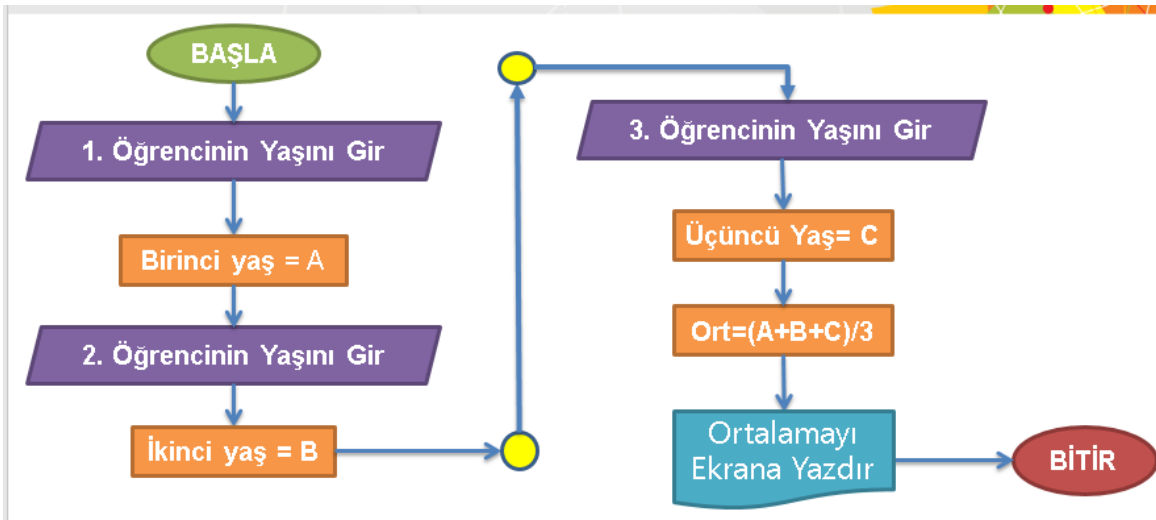


Akış Şeması Orneđi - 3

3 öğrencinin yaş ortalamasını bulan programın akış şemasını ve algoritmasını hazırlayalım.



- Adım 1: Başla
- Adım 2: Birinci öğrencinin yaşını gir.
- Adım 3: İlk sayı = A
- Adım 4: İkinci öğrencinin yaşını gir
- Adım 5: İkinci sayı = B
- Adım 6: Üçüncü öğrencinin yaşını gir
- Adım 7: Üçüncü Sayı = C
- Adım 8: Ort = $(A+B+C)/3$
- Adım 9: Sonucu(Ort) ekranda göster.
- Adım 10: Bitir.



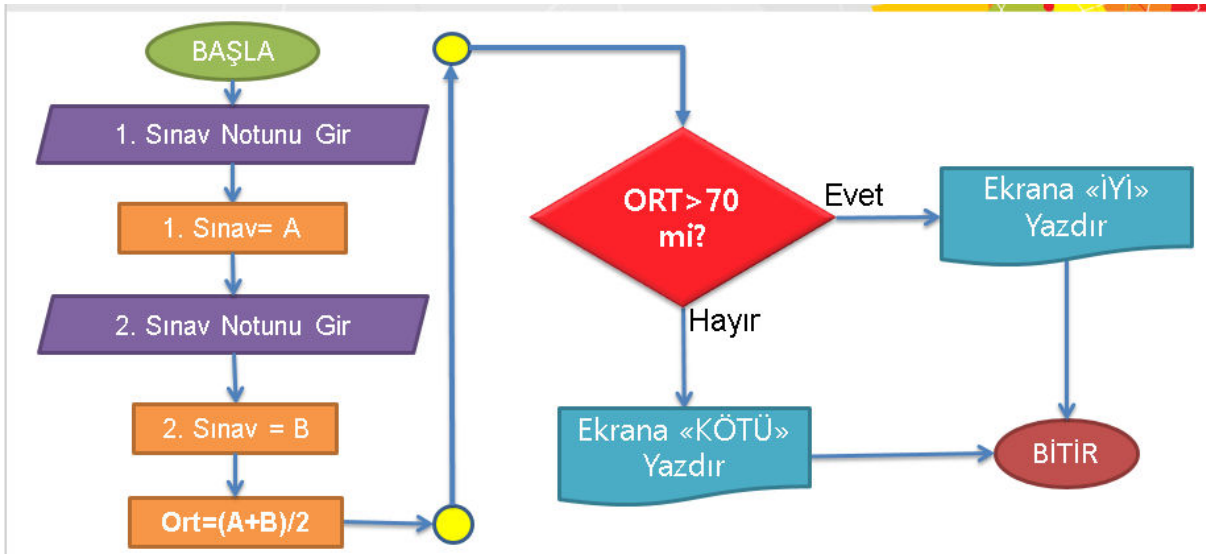
Uygulama

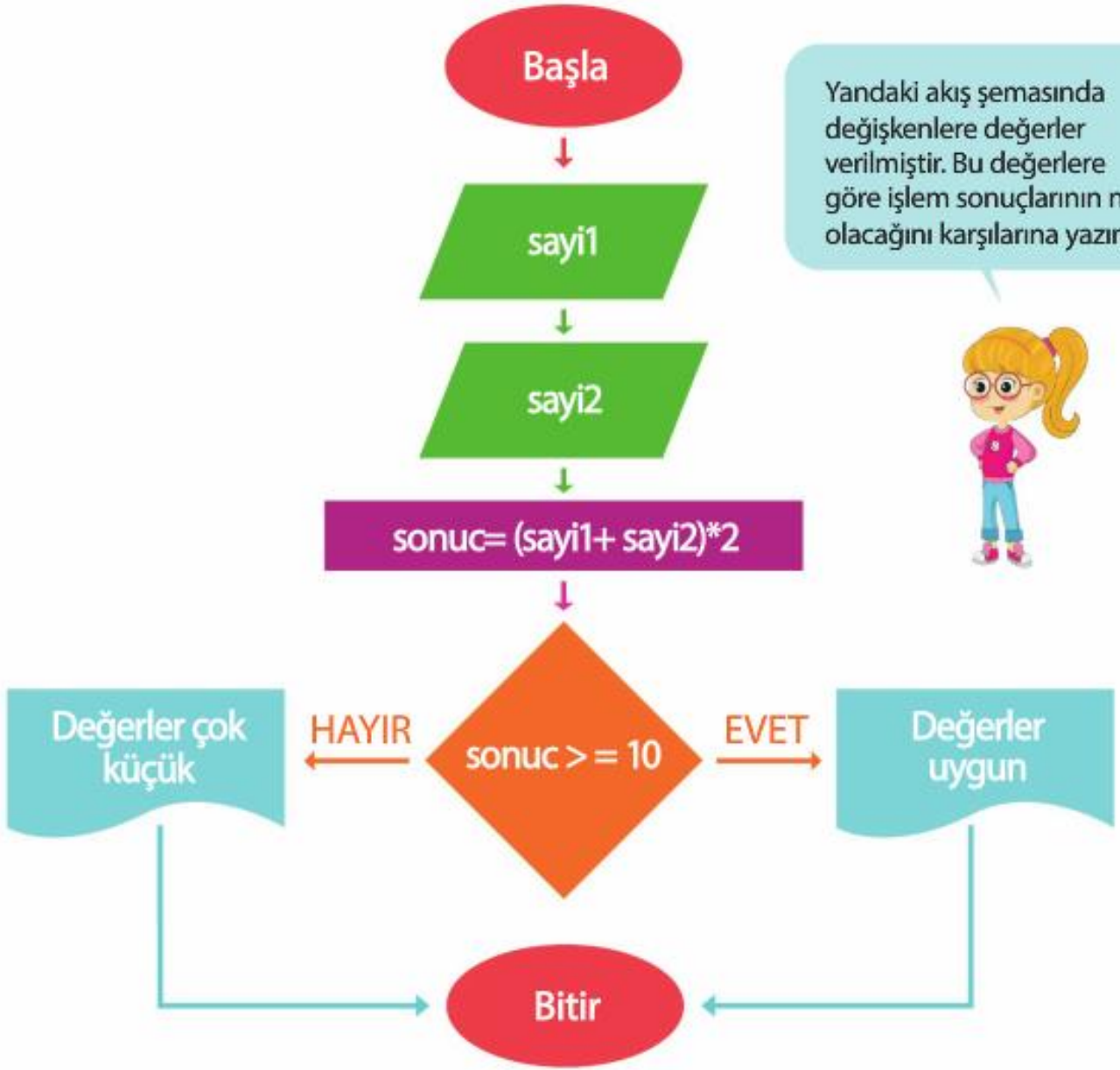
Bir öğrencinin klavyeden girilen iki notunun ortalamasını hesaplayan ve çıkan sonuca göre notun iyi veya kötü olduğunu ekrana yazdıran programın algoritmasını ve akış şemasını hazırlayınız.

(Ortalama 70'ten büyük ise **İYİ**, küçük ise **KÖTÜ** kabul edilecek.)



- Adım 1: Başla
Adım 2: Birinci Sınav Notunu gir.
Adım 3: Birinci Sınav = A
Adım 4: İkinci Sınav Notunu gir
Adım 5: İkinci Sınav = B
Adım 6: $Ort=(A+B)/2$
Adım 7: Eğer $Ort>70$ ise Ekrana «İYİ» yaz Adım 9'a git
Adım 8: Değilse Ekrana «KÖTÜ» yaz
Adım 9: Bitir.





DEĞERLER	EKRAN ÇIKTISI (SONUÇ)
sayı1 = 10 sayı2 = 5	
sayı1 = 2 sayı2 = 1	
sayı1 = 3 sayı2 = 2	

Algoritma hali verilen bazı problemleri şimdi de akış şeması ile gösterelim.



1

Problem: Çorba Yapmak

ALGORİTMA

- Adım 1** Başla
- Adım 2** Ocağı yak
- Adım 3** Suyu kaynat
- Adım 4** Çorba paketini aç
- Adım 5** Suyun içine at
- Adım 6** 10 dakika karıştır
- Adım 7** Çorbayı tabağa dök
- Adım 8** Bitir



AKIŞ ŞEMASI



2

Problem: Teldn babası ile birlikte eğer hava güneşli ise pikniğe gitmek istiyor. Eğer hava Güneşli değilse evde babası ile birlikte film izlemek ve ardından uyumak istiyor.

ALGORİTMA

- Adım 1** Başla
- Adım 2** Eğer hava güneşli ise
Pikniğe git, Adım 5'e git
- Adım 3** Eğer hava güneşli
değil ise film izle
- Adım 4** Uyu
- Adım 5** Bitir



AKIŞ ŞEMASI



Problem: 2 yazılı ve 1 performans notu girilen öğrencinin; ortalaması 45'den küçük ise ekrana "Dersten Kaldı" yazdıran, ortalaması 45'den küçük değil ise ekrana "Dersten Geçti" yazdıran program

ALGORİTMA

Adım 1 Başla

Adım 2 1. yazılı notunu gir

Adım 3 2. yazılı notunu gir

Adım 4 Performans notunu gir

Adım 5 Notları topla

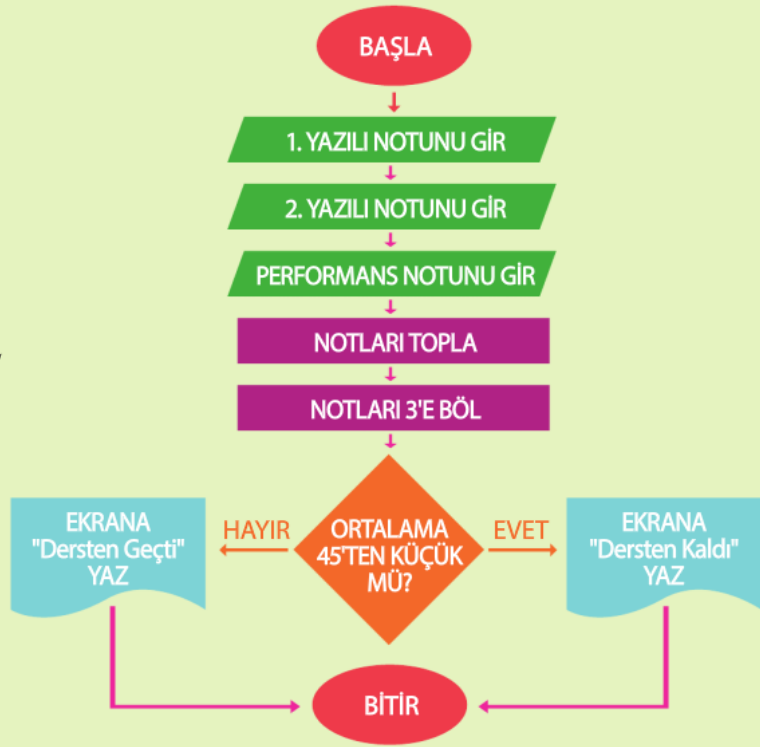
Adım 6 Notları 3'e böl

Adım 7 Ortalama 45'den küçük ise ekrana "Dersten Kaldı" yaz, Adım 9'a git

Adım 8 Ortalama 45'den küçük değil ise ekrana "Dersten Geçti" yaz

Adım 9 Bitir

AKIŞ ŞEMASI



Sayı Sayma Örneği

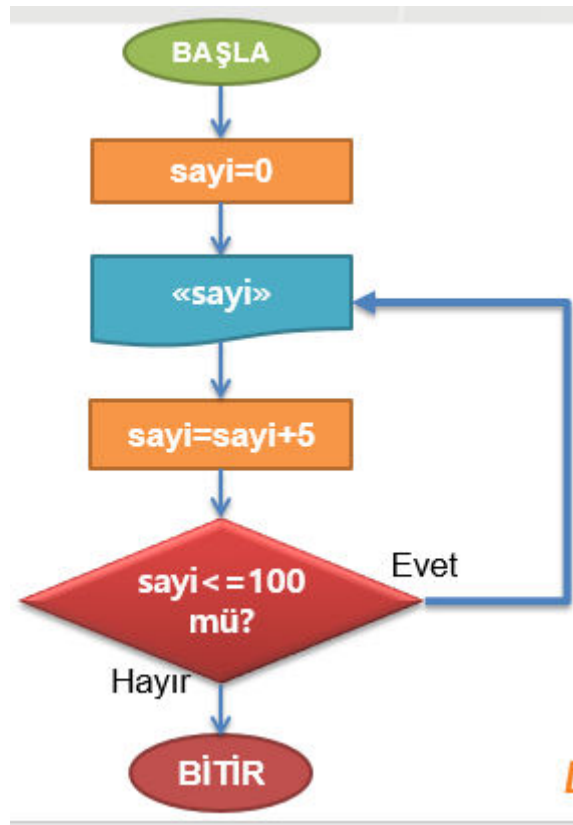
0'dan başlayıp 5'er 5'er 100'e kadar sayan ve ekrana yazdıran programın akış şemasını ve algoritmasını hazırlayalım.

Bu Örneği döngü kullanarak tasarlayacağız. «sayı» değişkeni 5er 5er artan sayıyı tutacak.

Algoritması

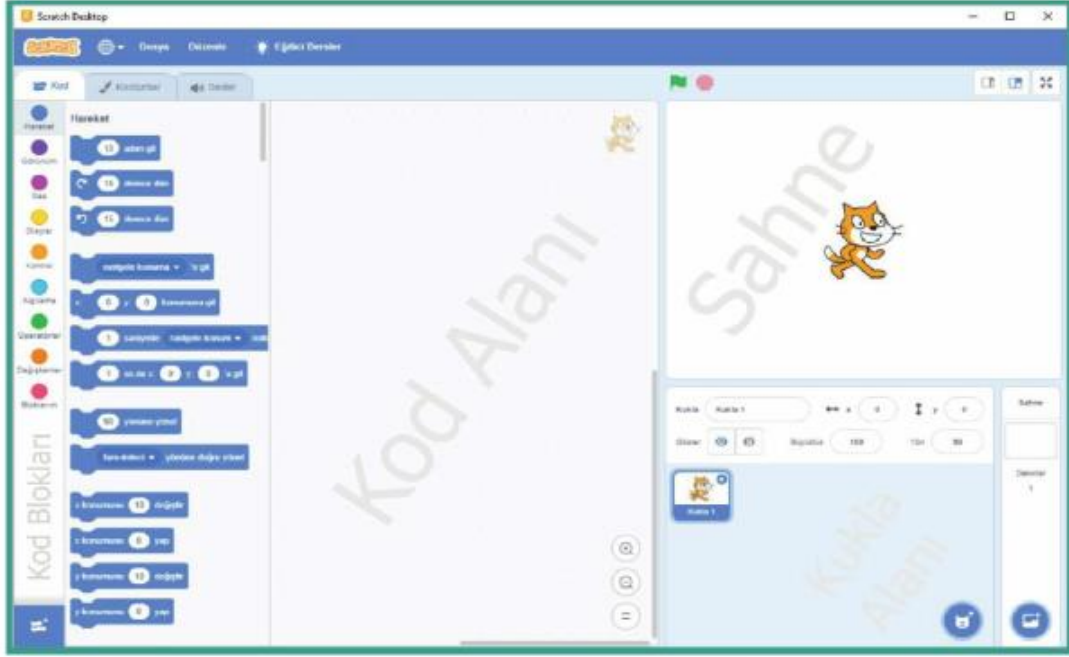
- Adım 1:** Başla
- Adım 2:** $sayi=0$
- Adım 3:** Ekrana «sayı» değişkenini yazdır.
- Adım 4:** $sayi=sayi+5$;
- Adım 5:** Eğer $sayi \leq 100$ ise Adım 3 git
- Adım 6:** Bitir





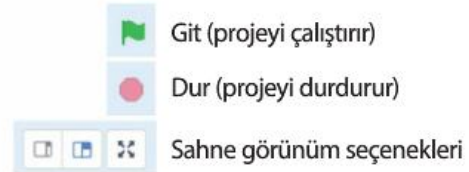
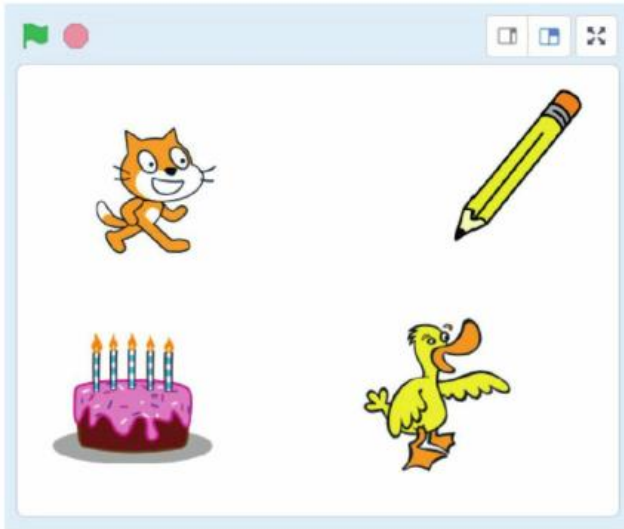
1

Scratch Programı Arayüzü



Sahne

Eklenecek kuklalar ve arka plan resimlerinin görüntülediği, projenin çalıştırıldığı alandır.



- Sahne içerisinde kullanılan karakterlere **Kukla** denir.
- Kuklaların farklı kılıklarını (görünümlerini), seslerini ve hareketlerini ayarlayabilirsiniz.
- Yandaki sahne görüntüsünde 4 adet kukla bulunmaktadır.

Kukla Alanı ve Dekorlar



Bu alan içerisinde;

- Sahneye kukla ekleme
- Sahneye dekor (arkaplan) ekleme
- Kuklaların sahne içerisindeki konumlarını ayarlama
- Kukla büyüklüğü ve kukla yönünü ayarlama
- Kuklaya isim verme gibi işlemler yapılmaktadır.

Dekor Seçimi

- Dekor Yükle** 4: Sahneye Scratch içerisinde bulunmayan bir dekor, arkaplan eklenir.
- Sürpriz** 3: Sahneye Scratch içerisinden rastgele bir dekor eklenir.
- Çizim** 2: Yeni bir dekor oluşturmak için kullanılır.
- Bir Dekor Seç** 1: Sahneye Scratch programı içerisindeki hazır dekorları eklemek için kullanılır.

Kukla Seçimi

- Kuklayı Yükle** 4: Sahneye Scratch içerisinde bulunmayan bir kukla eklenir.
- Sürpriz** 3: Sahneye Scratch içerisinden rastgele bir kukla eklenir.
- Çizim** 2: Yeni bir kukla çizmek, tasarlamak için kullanılır.
- Kukla Seçimi** 1: Sahneye Scratch programı içerisindeki hazır kuklaları eklemek için kullanılır.