2024-2025 EĞİTİM VE ÖĞRETİM YILI

PROGRAMLAMAYA GİRİŞ VE ALGORİTMA DERSİ

2. ÜNİTE

ALGORİTMA İLE PROBLEM ÇÖZME VE AKIŞ DİYAGRAMI

Kaynak: onuraltuntas.net

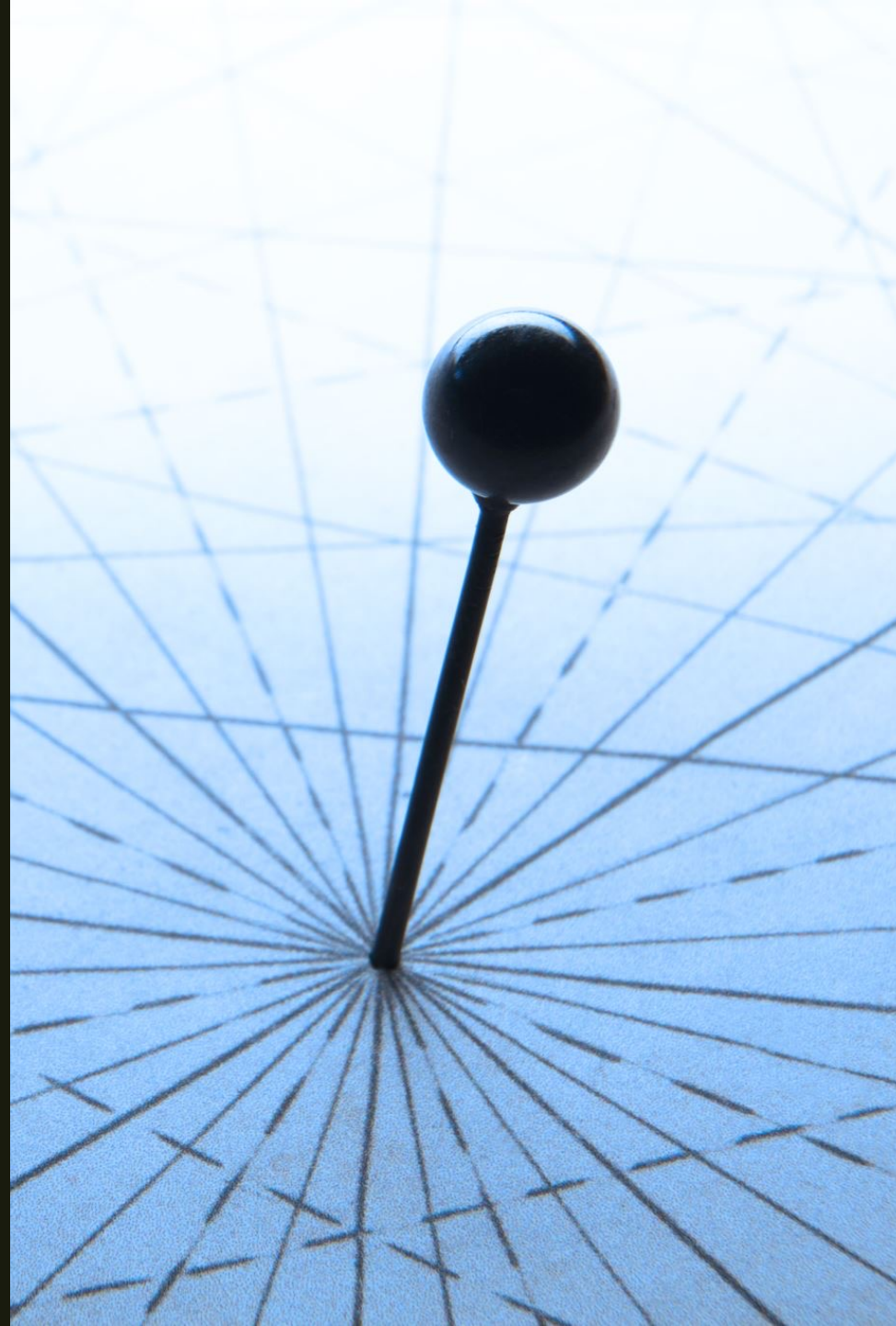
2.1. Çözümü istenen problem

- Bir problemi çözmeye başlamadan önce, onu gerçekten anlamak çok önemlidir. Tıpkı bir yapbozu çözmeden önce, parçaların nasıl yerleştirileceğini düşünmek gibi.

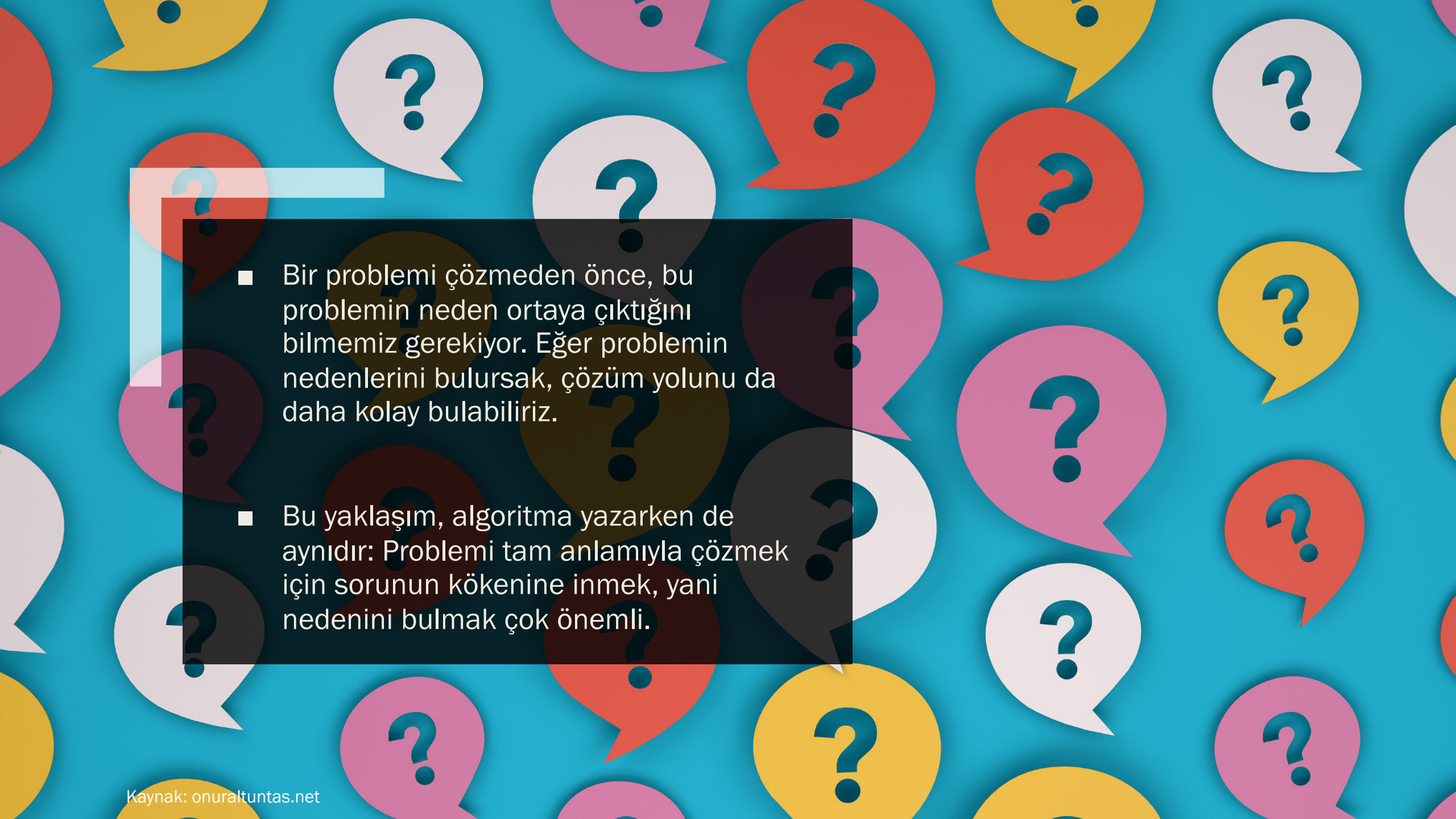
- Bu yüzden önce problemi dikkatlice incelemeli ve çözüm için hangi adımları takip etmemiz gerektiğini tahmin etmeliyiz.

Kaynak: onuraltuntas.net

Örnek olarak, bir arkadaşına nasıl adres tarif edeceğini düşün. Önce nereye gitmek istediğini anlamalısın, sonra da adım adım nasıl gidileceğini planlamalısın. Bu, bir algoritma yazarken de aynı şekilde işler.



Kaynak: onuraltuntas.net



- Bir problemi çözmeden önce, bu problemin neden ortaya çıktığını bilmemiz gerekiyor. Eğer problemin nedenlerini bulursak, çözüm yolunu da daha kolay bulabiliriz.

- Bu yaklaşım, algoritma yazarken de aynıdır: Problemi tam anlamıyla çözmek için sorunun kökenine inmek, yani nedenini bulmak çok önemli.

Kaynak: onuraltuntas.net

Örnek problem: Bir bilgisayar yavaş çalışmaktadır.

?

Problem çözmek için bazı sorular soralım:

- Bilgisayar ne zamandan beri yavaş?
- Hangi uygulamaları kullanıyorsun?
- Bilgisayarın donanımında bir sorun var mı?

Kaynak: onuraltuntas.net

Problemi sorularla çözme

Örnek problem: Bir bilgisayar yavaş çalışmaktadır.

Sizce bu problemi çözmek için başka hangi sorular sorulabilir?

Kaynak: onuraltuntas.net

Problemin Etki Alanı:

Örnek problem: Bir bilgisayar yavaş çalışmaktadır.

Soru: Bu problem başka neleri etkiliyor olabilir? (Örnek: Bilgisayarda yapılan işlerin gecikmesi)

Kaynak: onuraltuntas.net

Problem: Bir konferans salonunda farklı büyüklükte oturma alanları var. Aşağıdaki üç farklı oturma alanı mevcut:

- Küçük alan: 15 kişi
- Orta alan: 25 kişi
- Büyük alan: 40 kişi

Toplamda 270 kişilik bir grup bu salona yerleştirilecek. Amacımız, oturma alanlarını en verimli şekilde kullanarak herkesin salona sığmasını sağlamak. Hangi oturma alanlarından kaç tane kullanılmalıdır?

- **Kısıtlar:**
- Her oturma alanı sadece bir kez kullanılabilir.
- Amaç, en az oturma alanı kullanılarak tüm insanları yerleştirmektir.

Kaynak: onuraltuntas.net

Algoritma Adımları:

1.adım; Problemi Anlamak:

1. *Toplam 270 kişi var ve 3 farklı boyutta oturma alanı kullanabiliriz: 15 kişilik, 25 kişilik, 40 kişilik.*
2. *Her oturma alanı bir kere kullanılabileceğinden, bu oturma alanlarını birleştirerek toplamda 270 kişiyi salona sığdırmalıyız.*
3. *Amacımız, en az sayıda oturma alanı kullanarak tüm kişileri yerleştirmek.*

Kaynak: onuraltuntas.net

Algoritma Adımları:

2.adım; Çözüm Stratejisi:

- Öncelikle büyük oturma alanlarını (40 kişilik) mümkün olduğunca kullanarak başlayacağız, çünkü büyük alanlar daha fazla kişiyi tek seferde alır.
- Daha sonra kalan kişileri orta büyüklükteki alanlarla (25 kişilik) ve en son küçük alanlarla (15 kişilik) dolduracağız.
- Problem, "toplamda 270 kişi" olduğu için, bu insanları küçük, orta ve büyük alanlara optimal şekilde yerleştirmemiz gerekiyor.

Kaynak: onuraltuntas.net

Algoritma Adımları:

3.adım; Çözüm:

- **6 büyük oturma alanı (40 kişilik)** kullanılır: 6 * 40 = 240 kişi
- Kalan **30 kişi** orta veya küçük oturma alanlarına yerleştirilir:
  - ***1 orta oturma alanı (25 kişilik)*** *kullanılır: 25 kişi daha yerleşir.*
  - *Geriye kalan **5 kişi** küçük oturma alanına yerleştirilir: 1 küçük alan kullanılır.*

■ Sonuç olarak:

- **6 büyük alan, 1 orta alan, 1 küçük alan** kullanılarak 270 kişiyi yerleştirdik.

Kaynak: onuraltuntas.net

PROBLEMİ LİSTELEMEK

Problemde verilen ve istenen bilgiler listelenirse problem daha iyi anlaşılır

PROBLEMLERİ ALT BASAMAKLARA AYIRMAK

Eğer verilen problem karmaşıksa problem alt basamaklara ayrılarak kolaylaştırılır.

Kaynak: onuraltuntas.net

Problem: Arkadaş Toplantısı Düzenleme

- Bir grup insan bir araya gelmek istiyor. Toplam 30 kişi, belirli bir etkinlikte buluşacak. Fakat bu 30 kişi, sosyal medya üzerinden iki farklı grup oluşturmuş durumda:
- **Grup A:** 15 kişi (daha çok dış mekan etkinliklerini seviyor)
- **Grup B:** 15 kişi (kapalı alan etkinliklerini tercih ediyor)
- **Amaç:** Bu iki grubu bir araya getirip herkesin katılabileceği bir etkinlik düzenlemek. Ancak, herkesin aynı etkinliğe katılmasını sağlamak için aşağıdaki kısıtlamaları dikkate almak gerekiyor:
- Grup A, açık havada bir etkinlik olmasını istiyor.
- Grup B, etkinliğin kapalı alanda olmasını tercih ediyor.
- Herkesin katılabileceği bir orta yol bulmak gerekiyor.

Kaynak: onuraltuntas.net

Problemi listeleyelim

- **Verilen Bilgiler:**
  - Toplam 30 kişi.
  - İki grup: Grup A (15 kişi), Grup B (15 kişi).
  - Grup A açık hava etkinliklerini seviyor.
  - Grup B kapalı alan etkinliklerini seviyor.

- **İstenen Bilgiler:**
  - Hangi tür etkinlik düzenlenecek (açık hava, kapalı alan, ya da her ikisi)?
  - Her iki grubun da katılabileceği bir etkinlik düzenlemek için nasıl bir plan yapılabilir?

Kaynak: onuraltuntas.net

Problemi listeleyelim

- **Verilen Bilgiler:**
  - Toplam 30 kişi.
  - İki grup: Grup A (15 kişi), Grup B (15 kişi).
  - Grup A açık hava etkinliklerini seviyor.
  - Grup B kapalı alan etkinliklerini seviyor.

- **İstenen Bilgiler:**
  - Hangi tür etkinlik düzenlenecek (açık hava, kapalı alan, ya da her ikisi)?
  - Her iki grubun da katılabileceği bir etkinlik düzenlemek için nasıl bir plan yapılabilir?

Kaynak: onuraltuntas.net

Problemi alt basamaklara ayıralım

1. **Grupların Tercihlerini Belirle:**
    1. *Hangi grupların katılmak istediği, hangi tür etkinlikleri sevdikleri ve beklentilerini anlamak.*
2. **Ortak Nokta Bulma:**
    1. *Açık hava ve kapalı alan etkinliklerini birleştiren bir çözüm bulmak.*
    2. *Örneğin, yarı açık bir alan seçebiliriz (örneğin, bir bahçe).*
3. **Etkinlik Planlaması:**
    1. *Ortak bir etkinlik tasarlamak için bir tarih ve yer belirlemek.*
    2. *Her iki grubun da etkinlikten keyif almasını sağlamak.*

Kaynak: onuraltuntas.net

Çözüm Stratejisi:

1. **Grupların Tercihlerini Listele:**
    1. ***Grup A'nın Tercihleri:*** *Dış mekan etkinlikleri, doğa yürüyüşleri, piknik.*
    2. ***Grup B'nin Tercihleri:*** *Kapalı alan etkinlikleri, sinema, oyun gecesi.*
2. **Ortak Nokta Bulma:**
    1. *Açık hava ve kapalı alan etkinliklerini bir araya getirecek bir çözüm bulmak.*
    2. *Önerilen etkinlik:* ***Bir bahçede piknik organizasyonu*** *veya* ***yazlık bir yerde barbekü****.*
3. **Planlama:**
    1. *Belirlenen tarih ve yer ile birlikte her iki grubun da katılımını sağlayacak bir davetiye oluşturulabilir.*
    2. *Herkesin etkinlikte ne tür aktivitelerde yer alabileceği hakkında bilgilendirme yapılmalıdır (örneğin, oyunlar, yemekler, aktiviteler).*

Kaynak: onuraltuntas.net

2.2. Çözüm için Gereksinimler

?

1.**Problemi Anlamak:** İlk olarak, karşımıza çıkan problemi tam anlamıyla anlamamız gerekiyor. Eğer problem karmaşıksa, onu daha küçük ve anlaşılır parçalara ayırabiliriz.

2.**Problemi Açıklamak:** Problemi basit ve net bir şekilde ifade etmek önemlidir. Ne olduğunu ve neyin çözülmesi gerektiğini açıkça yazmak, çözüm yolunda ilk adımdır.

3.**Önemli Noktaları Belirlemek:** Problemi çözerken hangi bilgilerin veya adımların önemli olduğunu düşünmeliyiz. Böylece zaman kaybetmeden doğru yoldan ilerleriz.

Kaynak: onuraltuntas.net

Günlük hayatta karşılaştığımız problemleri çözmek

1. **Problemi Tanımla:** İlk olarak, karşımıza çıkan sorunun ne olduğunu açıkça anlamamız gerekiyor. Örneğin, sabah okula geç kalıyorsan, sorun "geç kalmak" olabilir.

2. **Nedenlerini Düşün:** Sorunun neden kaynaklandığını düşünmelisin. Okula geç kalmanın sebebi sabah alarmını duymamak mı, yoksa hazırlanırken fazla zaman harcamak mı?

Kaynak: onuraltuntas.net

Günlük hayatta karşılaştığımız problemleri çözmek

3. Çözüm Önerileri Geliştir: Şimdi sorunu çözmek için farklı yollar bulalım. Örneğin, alarmı daha yüksek sesle kurabilir, akşamdan hazırlık yaparak sabah hazırlanma süresini kısaltabilirsin.

4. Uygun Çözümü Seç: Bulduğun çözüm yollarından hangisinin işe yarayacağını düşün. Alarmı daha erken bir saate kurmak sana daha uygun olabilir.

Kaynak: onuraltuntas.net

FARKLI ÇÖZÜM YOLLARI

Bir problemi çözerken her zaman **tek bir doğru yol** olmadığını bilmeliyiz.

Bazen aynı sorunu çözmek için birden fazla farklı yöntem olabilir.

Önemli olan, duruma ve ihtiyaca göre **en uygun** olan çözüm yolunu seçmektir.

Örneğin, **bir sınav için çalışmak** bir problem olabilir:

- Birisi, sınav için çalışırken **kitap okuyarak** öğrenmeyi tercih edebilir.
- Başka biri, **özet çıkararak** çalışmayı daha faydalı bulabilir.
- Bir başkası ise **video dersler** izleyerek daha iyi anladığını fark edebilir.

Kaynak: onuraltuntas.net

Problem çözme süreci üzerine geliştirilmiş kuramlar ve yaklaşımlar

1. **Polya'nın Dört Adımlı Problem Çözme Yöntemi:**

Matematikçi George Polya, problem çözme sürecini basitleştirmek için dört adım önermiştir:

1. Problemi Anlamak

2. Bir Plan Geliştirmek

3. Planı Uygulamak

4. Sonucu Gözden Geçirmek

Kaynak: onuraltuntas.net

Problem çözme süreci üzerine geliştirilmiş kuramlar ve yaklaşımlar

2. Algoritmik Yaklaşım:

Algoritmalar, bir problemi çözmek için takip edilmesi gereken adım adım talimatlardır. Bu yaklaşım özellikle bilgisayar bilimlerinde yaygındır.

Algoritmik problem çözme şu adımları içerir:

- **Girdileri tanımlama:** Problemi çözmek için gerekli tüm bilgileri toplamak.
- **Adımları belirleme:** Adım adım izlenecek çözüm yolunu belirleme.
- **Sonuç alma:** Algoritmayı uygulayarak sonuca ulaşma.

Kaynak: onuraltuntas.net

Problem çözme süreci üzerine geliştirilmiş kuramlar ve yaklaşımlar

3. Heuristik (Deneme-Yanılma) Yöntem:

Bu yöntemde insanlar, problemi çözmek için daha önce deneyip başarılı oldukları stratejileri kullanırlar.

Ancak bu yaklaşım her zaman kesin bir çözüm sunmaz.

Heuristik yöntem, bazen çözüm için birkaç deneme ve yanılma süreci gerektirebilir.

Kaynak: onuraltuntas.net

■ Bir problemin çözümü için kullanılabilecek temel yöntem ve teknikler

BÖL VE FETHET (DİVİDE AND CONQUER): Büyük bir problemi daha küçük parçalara bölmek ve her parçayı ayrı ayrı çözmek.

BENZER PROBLEMLERİ KULLANMA (ANALOGİLERDEN FAYDALANMA): Daha önce çözülmüş benzer bir problemi kullanarak yeni bir probleme yaklaşmak.

GÖRSEL DÜŞÜNME (DİYAGRAMLAR VE HARİTALAR KULLANMA): Problemi görselleştirerek çözüm yollarını daha açık hale getirmek. Diyagramlar, akış şemaları, zihin haritaları gibi görsel araçlar kullanılarak problem ve çözüm yolları organize edilir.

GERİYE DOĞRU ÇALIŞMA (TERSİNE MÜHENDİSLİK): Problemin çözümünü son aşamadan başlayarak geriye doğru gitmek.

BEYİN FIRTINASI: Farklı kişiler farklı çözüm önerileri sunar ve en iyi çözüm yolu seçilir.

Kaynak: onuraltuntas.net

2.3. Problemin girdi, çıktı ve işlem aşamaları

- **Girdi (Input):**
- **Tanım:** Girdi, bir problemin başında sisteme girilen veriler ya da işlemin başlaması için gerekli olan malzemelerdir.
- **Örnekler:**
  - *Bir hesap makinesine **5 + 3** yazdığımızda, **5** ve **3** bizim girdimizdir.*
  - *Bir pizzacıya sipariş verirken seçtiğimiz pizza türü ve adresimiz de girdi olarak kabul edilebilir.*

Kaynak: onuraltuntas.net

Çıktı (Output):

- **Tanım:** Bir sistemin veya işlemin sonunda elde ettiğimiz sonuçtur. Yani, girdileri işledikten sonra aldığımız sonuç ya da bilgi çıktıdır.
- **Örnekler:**
  - *Hesap makinesi, **5 + 3** işlemi sonrası bize **8** sonucunu verir, bu **çıktıdır.***
  - *Pizzacı siparişin sonunda pizzayı teslim eder, bu da işlemin **çıktısıdır.***

Kaynak: onuraltuntas.net

Örnek problem durumlarının girdi ve çıktıları

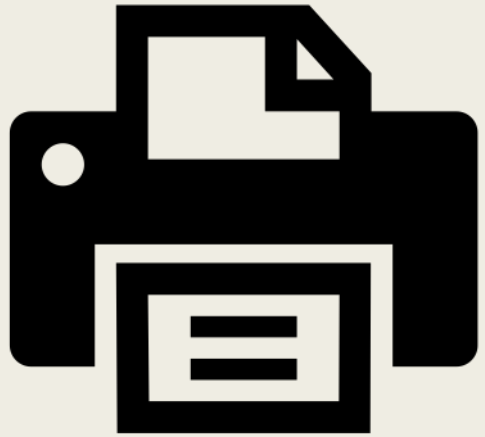
- **Örnek 1: Çay Demlemek**

- **Problem:** Çay demlemek istiyoruz.
  - ***Girdiler:*** *Çay, su, çaydanlık, ocak.*
  - ***İşlem:*** *Çayı demliğe koyup suyu kaynatmak ve çayı demlemek.*
  - ***Çıktı:*** *Demlenmiş çay.*

Örnek problem durumlarının girdi ve çıktıları

- **Örnek 3: Bir Metin Belgesini Yazdırmak**

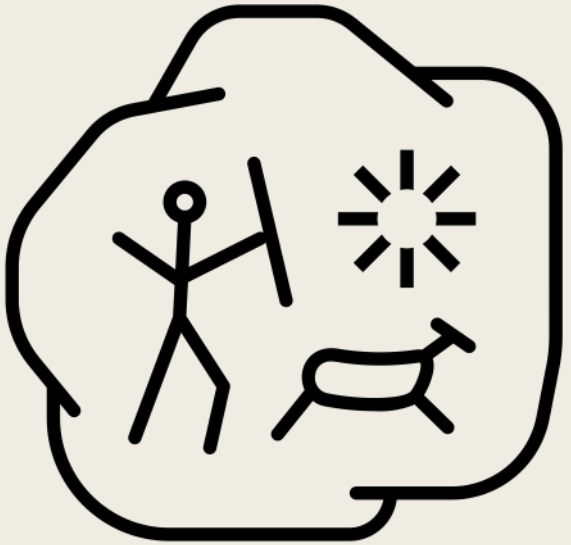
- **Problem:** Bilgisayarda yazdığımız bir metni yazıcıdan yazdırmak istiyoruz.
  - ***Girdiler:*** *Bilgisayardaki metin dosyası, yazıcı, kağıt.*
  - ***İşlem:*** *Yazdırma komutu vermek ve yazıcının belgeyi yazdırması.*
  - ***Çıktı:*** *Kağıda basılmış metin.*



Örnek problem durumlarının girdi ve çıktıları

- **Örnek 7: Bir Resim Çizmek**

- **Problem:** Bir resim çizmek istiyoruz.
  - ***Girdiler:*** *Kağıt, kalem, boya.*
  - ***İşlem:*** *Kağıda şekil ve desen çizmek.*
  - ***Çıktı:*** *Çizilmiş resim.*



Kaynak: onuraltuntas.net

Örnek problem durumlarının girdi ve çıktıları

- SİZ DE BİR ÖRNEK VERİN

- Problem: ???
  - *Girdiler: ???*
  - *İşlem: ???*
  - *Çıktı: ???*

Kaynak: onuraltuntas.net

Problem Alt Basamaklara Ayırma Basamakları

- **Örnek Problem: Bir Okul Projesini Tamamlama**

- **Ana Problem:** Okul projesini tamamlamak.
  - *Alt Basamaklar:*
    - Proje konusunu araştır.
    - Bilgileri topladıktan sonra yazı yaz.
    - Sunum hazırlığı yap.
    - Projeyi kontrol et ve öğretmene teslim et.

Kaynak: onuraltuntas.net

İşlem Aşamaları Belirlenirken Bulunan Farklı Çözüm Yollarından Çözüme En Kısa Sürede Ulaştıracak Olanın Seçilmesi

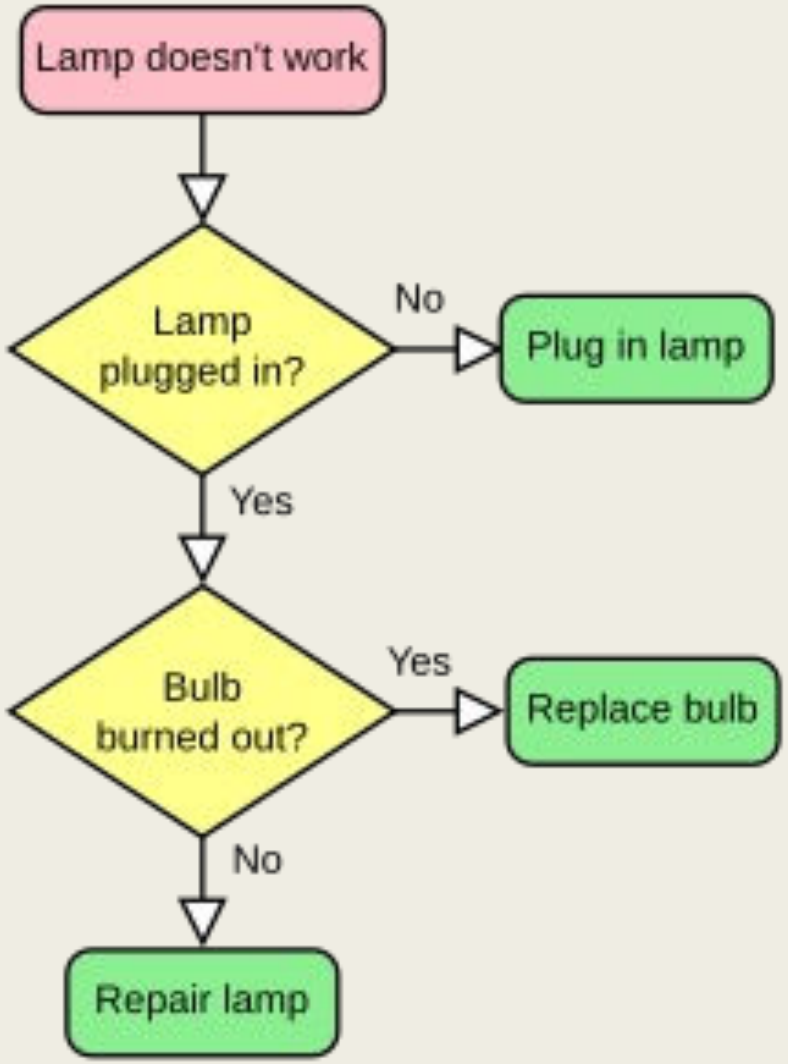
Kaynak: onuraltuntas.net

- Örnek Problem: Bir Arkadaşınıza Ulaşmak

- **Problem:** Bir arkadaşına ulaşmak istiyorsun.
  - ***Farklı Çözüm Yolları:***
    - E-posta göndermek.
    - Telefonla aramak.
    - Mesaj atmak.
  - ***En Kısa Süreli Çözüm:*** *Telefonla aramak, arkadaşına en hızlı şekilde ulaşmanı sağlayabilir. Mesaj veya e-posta ise daha uzun sürebilir.*

2.4. ALGORİTMA

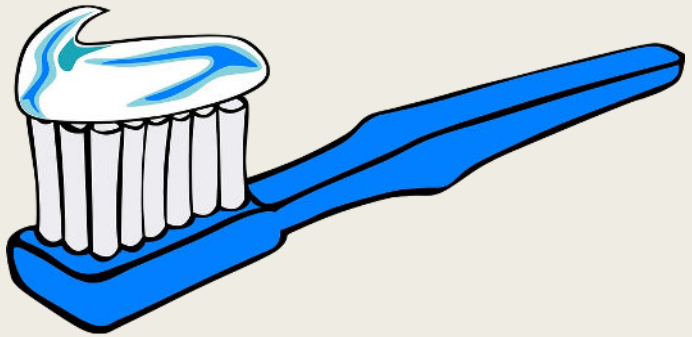
Algoritma, bir problemin çözümü için izlenen adım adım işlemlerin tanımıdır. Bir işi yapmak için gereken talimatlar dizisidir.



Kaynak: onuraltuntas.net

2.4. ALGORİTMA

Günlük hayatta yaptığınız rutin işlere örnekler veriniz.



Kaynak: onuraltuntas.net

2.4. ALGORİTMA

Günlük rutin işlerinizin mantığı ve sırası değiştiğinde sonuçlar da değişir

Örnek: Diş fırçalamak

- Diş macunu sürmeden fırçalamaya başlarsak, dişlerimiz temizlenmez.

- Önce macun sürülmeli, sonra fırçalanmalı.

Örnek: Yemek yapmak

Yemeği pişirmeden malzemeleri tabağa koyarsak çiğ olur ve yenmez.

Önce malzemeler hazırlanır, sonra pişirilir ve en son servis edilir.

Kaynak: onuraltuntas.net

Pasta Yapma Algoritması

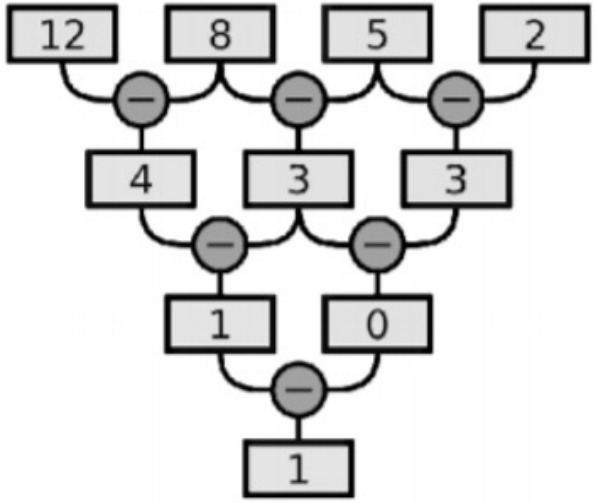
- **Algoritma:**

1. Gerekli malzemeleri topla (un, şeker, yumurta, süt, yağ, kabartma tozu, vanilin).
2. Fırını 180°C'de önceden ısıt.
3. Büyük bir kapta yumurtaları ve şekeri çırp.
4. Süt ve yağı ekle, karıştır.
5. Ayrı bir kapta un, kabartma tozu ve vanilini karıştır.
6. Kuru malzemeleri sıvı karışıma ekle ve iyice karıştır.
7. Karışımı yağlanmış kek kalıbına dök.
8. Önceden ısıtılmış fırına yerleştir.
9. 30-35 dakika pişir. (Pişip pişmediğini kontrol etmek için bir kürdan batır.)
10. Pişen pastayı fırından çıkar ve soğumaya bırak.
11. İsteğe bağlı olarak üzerine krema veya meyve ekle ve servis et.

Kaynak: onuraltuntas.net

Piramit Satırları

İşlem makinesi ilk satırdaki 4 rakamı girdi olarak almaktadır. Her satırda, makine sayılar arasındaki farkı hesaplamaktadır. Aşağıdaki resimde örnek bir işlem görülmektedir.



Soru

Aşağıdaki girdi rakamlarından hangisi son satırda sonucun "0 (sıfır)" olmasını sağlar?

A. 13 9 7 6
B. 16 9 4 1
C. 13 8 4 2
D. 5 5 5 1

SİZ DE BİR PROBLEM BELİRLEYEREK ÖRNEK BİR ALGORTİMA ADIMLARI OLUŞTURUNUZ.

Kaynak: onuraltuntas.net

Ebu Cafer Muhammed bin Musa el-Harezmi

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.

Kaynak: onuraltuntas.net

Ebu Cafer Muhammed bin Musa el-Harezmi

- El-Harezmi 780-850 yılları arasında yaşamış, matematik, astronomi ve coğrafya alanlarında önemli katkılarda bulunmuş bir İslam bilginidir.

- El-Harezmi'nin en bilinen eserlerinden biri ;
- «Hisab el-cebir ve el-mukabala» ‘dır.
- Türkçesi → «Cebir ve Mukabele Üzerine Kısa Kitap»
-
- Bu eser, cebirin temellerini atmış ve matematikte önemli bir dönüm noktası olmuştur.

Kaynak: onuraltuntas.net

Ebu Cafer Muhammed bin Musa el-Harezmi

- **Algoritma Kelimesinin Kökeni**

- **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 izlenilen sistematik adım dizisi anlamına gelir.

Kaynak: onuraltuntas.net

Ebu Cafer Muhammed bin Musa el-Harezmi

- El-Harezmi ve onun algoritma ve bilgisayar bilimine katkılarıyla ilgili daha fazla araştırma yapınız.

- Araştırma sonuçlarını sınıfla paylaşınız.

Kaynak: onuraltuntas.net

2.5. Bir problemin çözümü için en doğru algoritmayı geliştirmek

EN DOĞRU ALGORİTMA İÇİN KULLANLAN KRİTERLER;

- Hedefe Ulaşma
- Hata Riskini Azaltma
- Verimlilik

Kaynak: onuraltuntas.net

2.5. Bir problemin çözümü için en doğru algoritmayı geliştirmek

EN DOĞRU ALGORİTMA İÇİN KULLANLAN KRİTERLER;

- **Sorun Tespiti ve Çözümleme:** İşlem adımları doğru sıralandığında, bir sorun çıktığında hangi adımda hata olduğunu tespit etmek daha kolay olur. Bu, sorun çözmeyi hızlandırır.
- **Tutarlılık ve Tekrar Edilebilirlik:** Adımların doğru sıralanması, aynı girdilerle aynı çıktının tekrar tekrar elde edilmesini sağlar. Bu da işlemlerin tutarlı olmasına katkı sağlar.

Kaynak: onuraltuntas.net

Algoritma yazımında oluşabilecek hatalar

- **Yanlış veya Eksik Adım Tanımlama:** Bir adımın eksik veya atlanmış olması, işlemi tamamlayamamanıza yol açar. Bu nedenle tüm adımların doğru sırayla ve eksiksiz belirlenmesi gerekir.
- **Yanlış Mantık Yapısı:** Algoritmanın mantıksal akışında hata yapmak, programın istenilen çıktıyı vermemesine neden olabilir. Örneğin, döngü veya koşul ifadelerindeki hatalar, yanlış verilerin işlenmesine veya döngülerin sonsuz döngüye girmesine yol açabilir.
- **Yanlış Girdi Tanımlama:** Algoritmanın çalışabilmesi için kullanılan girdiler doğru tanımlanmadığında ya da sınır durumlar düşünülmediğinde, beklenmeyen çıktılar ortaya çıkabilir. Girdiler uygun tip ve formatta olmalıdır.

Kaynak: onuraltuntas.net

Algoritma yazımında oluşabilecek hatalar

- **Sınır Durumlarını Göz Ardı Etme:** Algoritmaların her durumda çalışması beklenir, ancak bazen sınır durumlar (örneğin, sıfır elemanlı bir listeyle işlem yapma gibi) düşünülmezse hatalar meydana gelebilir. Bu yüzden algoritmaların en kötü durumları da hesaba katılmalıdır.
- **Optimizasyon Eksikliği:** Algoritmanın çalışması çok uzun sürüyorsa veya çok fazla bellek tüketiyorsa, algoritmanın verimli olmadığını gösterir. Yanlış yapılandırılmış bir algoritma, gereksiz zaman ve kaynak harcanmasına neden olur.
- **Hatalı Veri Yapıları Kullanma:** Verinin nasıl işleneceği doğru düşünülmediyse, yanlış veri yapıları kullanılarak algoritmanın performansı düşürülebilir veya algoritma beklenmedik şekilde çalışabilir.

Kaynak: onuraltuntas.net

Algoritma yazımında oluşabilecek hatalar

- **Döngüler:** Döngülerin yanlış sayıda tekrarlanması veya işlemlerin yanlış sırada yapılması, beklenen çıktıları bozabilir.

- **Test Eksikliği:** Algoritmanın düzgün çalışıp çalışmadığını anlamak için sınama yapılmaması, gizli hataların ortaya çıkmasını engeller. Her algoritma, çeşitli test senaryoları ile sınanmalı ve hataların bulunması sağlanmalıdır.

Kaynak: onuraltuntas.net

SINIF ETKİNLİĞİ

- Toplumu ilgilendiren bir problemin çözümüne yönelik girdi, çıktı ve işlem adımları oluşturarak bir algoritma yazınız.

Kaynak: onuraltuntas.net

SINIF ETKİNLİĞİ

- Tasarladığınız algoritmanın doğruluğunu tartışınız. Geliştirilmesi gereken noktaları yeniden çalışarak algoritmayı en doğru ve işlevsel hâle getiriniz.

Kaynak: onuraltuntas.net

2.6.

PROGRAMLAMA NEDİR?

- Programlama, bilgisayarlara belirli görevleri yapmaları için talimatlar vermek anlamına gelir.
- Yani, bir problemi çözmek veya bir işlemi gerçekleştirmek için adım adım talimatlar yazdığımız bir süreçtir.
- Programlama dilleri (örneğin, Python, Java) bu talimatları bilgisayarın anlayabileceği bir biçimde yazmamızı sağlar.
- Bu talimatlar bir araya gelerek bir program oluşturur ve bilgisayar bu programı çalıştırarak verilen görevleri yerine getirir.

Kaynak: onuraltuntas.net

APPLE CEO'SU TİM COOK:

Programlama dilleri gelecekte yabancı dil bilmek kadar mühim olacak ve her mesleki alan bir şekilde kodlama bilgisine ihtiyaç duyacak.

BEYİN FIRTINASI

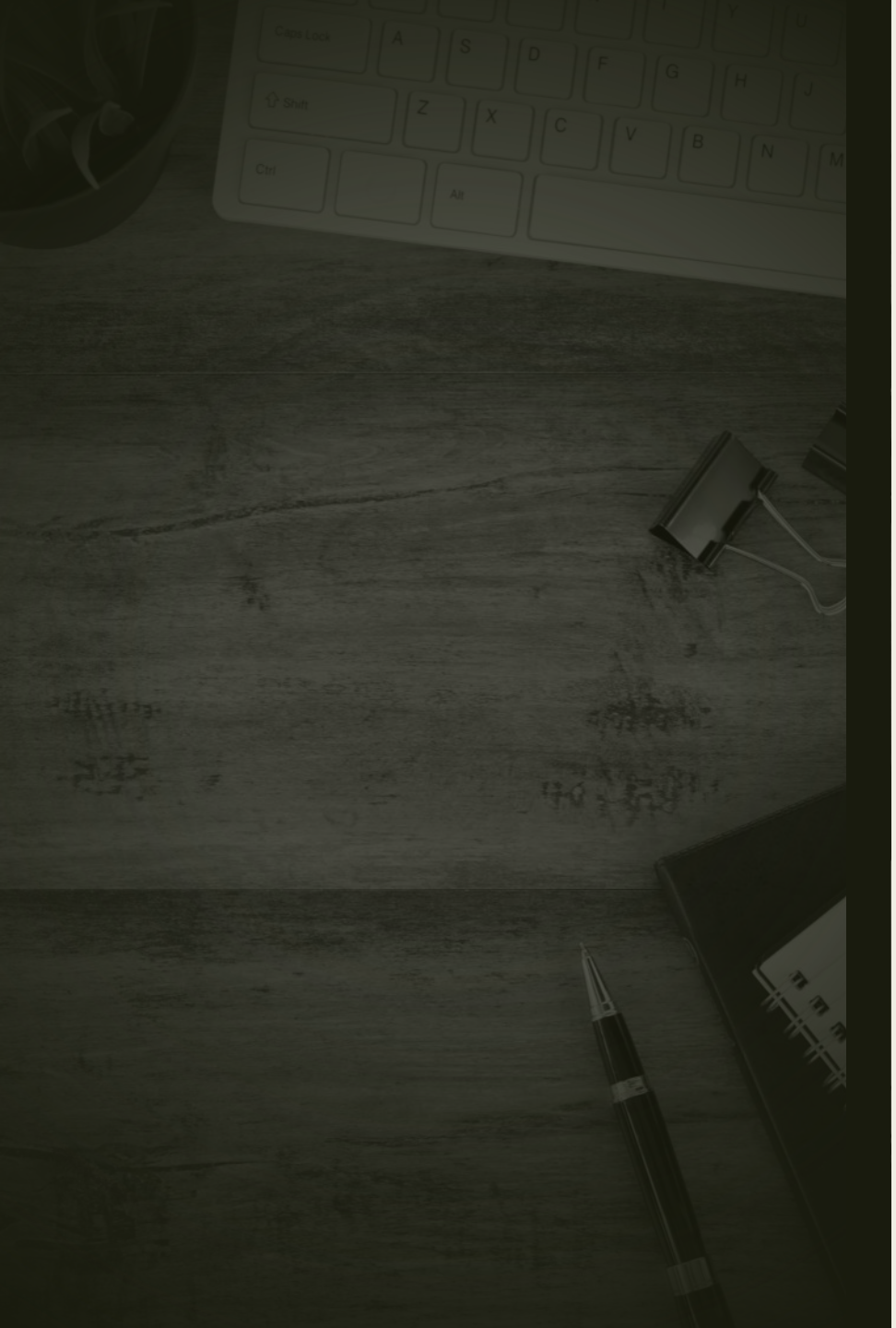
PROGRAMLAMA NEDEN ÖNEMLİDİR?

YORUMLAYINIZ!

Kaynak: onuraltuntas.net

Programlama Dili Nedir?

- Tüm yazılımlar programlama dilleriyle üretilir.
- Cep telefonunuzdaki uygulamalar, masaüstü uygulamaları, oyun ve ağ uygulamaları üretmek için programlama diline ihtiyacınız vardır.
- Günümüz itibariyle dünya üzerindeki tüm programlama dillerinin toplam sayısının 750'nin üstünde olduğu tahmin edilmektedir.



Programlama Neden Önemlidir?

1. Günlük Hayatı Kolaylaştırır
2. Problem Çözme Becerisi Kazandırır
3. Yaratıcılığı Teşvik Eder
4. Geleceğin Meslekleri için Temeldir
5. İşbirliği ve Paylaşımı Destekler

Kaynak: onuraltuntas.net

2.7. Programlama dillerinin gelişimi

Python

•**Kullanım Alanı:** Genel amaçlı bir programlama dilidir ve bilimsel hesaplamalardan web geliştirmeye kadar çok çeşitli alanlarda kullanılır.

•**Öne Çıkan Özellikler:** Basit ve okunabilir bir sözdizimine sahiptir, bu yüzden özellikle programlamaya yeni başlayanlar için çok uygundur.

•**Kullanım Alanları:**

- Veri analizi ve bilimsel hesaplamalar
- Yapay zeka ve makine öğrenmesi
- Web geliştirme (Django, Flask)
- Oyun geliştirme

Kaynak: onuraltuntas.net

Java

•**Kullanım Alanı:** "Bir kez yaz, her yerde çalıştır" felsefesiyle bilinir. Çok platformlu uygulamalar geliştirmek için yaygın olarak kullanılır.

•**Öne Çıkan Özellikler:** Platform bağımsızdır, yani bir cihazda yazıldığında diğer cihazlarda da çalıştırılabilir.

•**Kullanım Alanları:**

- Android uygulama geliştirme
- Kurumsal yazılım geliştirme
- Web uygulamaları (Spring framework)

Kaynak: onuraltuntas.net

JavaScript

•**Kullanım Alanı:** Web tarayıcıları için en önemli dillerden biridir. Dinamik ve etkileşimli web siteleri oluşturmak için kullanılır.

•**Öne Çıkan Özellikler:** Tarayıcı tabanlı olması, web sayfalarına dinamik içerik eklemeyi mümkün kılar.

•**Kullanım Alanları:**

- Web geliştirme (hem ön yüz (frontend) hem arka yüz (backend) için kullanılabilir)
- Mobil uygulama geliştirme (React Native, Ionic)
- Oyun geliştirme (tarayıcı tabanlı oyunlar)

Kaynak: onuraltuntas.net

C ve C++

- **Kullanım Alanı:** Düşük seviyeli sistem programlama dilleridir, donanıma yakın çalışmaları gerektiği durumlarda tercih edilir.

- **Öne Çıkan Özellikler:** Bellek yönetimi ve performans açısından oldukça güçlüdürler.

- **Kullanım Alanları:**
  - Oyun motorları ve yüksek performans gerektiren yazılımlar
  - Sistem ve ağ yazılımları (işletim sistemleri, sürücüler)
  - Gömülü sistemler ve IoT cihazları

Kaynak: onuraltuntas.net

Swift

•**Kullanım Alanı:** Apple cihazları için uygulama geliştirmede kullanılan bir programlama dilidir.

•**Öne Çıkan Özellikler:** iOS ve macOS uygulamaları geliştirmek için optimize edilmiştir.

•**Kullanım Alanları:**

- iOS, macOS, watchOS ve tvOS uygulamaları
- Apple ekosistemi içindeki uygulamalar

Kaynak: onuraltuntas.net

PHP

•**Kullanım Alanı:** Web tabanlı uygulamalar için server-side bir programlama dilidir.

•**Öne Çıkan Özellikler:** Dinamik web sayfaları oluşturmak için kullanılır ve birçok büyük web sitesinde (örneğin, WordPress) kullanılır.

•**Kullanım Alanları:**

- Web geliştirme (özellikle arka uç geliştirme)
- İçerik yönetim sistemleri (CMS) oluşturma

Kaynak: onuraltuntas.net

R

•**Kullanım Alanı:** Veri analizi ve istatistiksel hesaplamalar için kullanılan bir dildir.

•**Öne Çıkan Özellikler:** İstatistiksel modeller ve veri görselleştirme konusunda oldukça güçlüdür.

•**Kullanım Alanları:**

- Veri bilimi ve analiz
- İstatistiksel hesaplamalar ve araştırmalar
- Veri görselleştirme

Kaynak: onuraltuntas.net

SQL

•**Kullanım Alanı:** Veritabanları ile etkileşim kurmak için kullanılır.

•**Öne Çıkan Özellikler:** Verileri sorgulamak, güncellemek, silmek ve yönetmek için kullanılır.

•**Kullanım Alanları:**

- Veritabanı yönetimi
- Büyük veri ve analiz sistemleri

Kaynak: onuraltuntas.net

SORU

Belirli amaçlar için kullanılan programlama dillerine farklı örnekler veriniz.

Kaynak: onuraltuntas.net

AKIŞ ŞEMASI

Bir sürecin adımlarını **görsel** ya da **sembolik** olarak gösterir.

2.8. Akış diyagramının kullanım amaçları ve avantajları

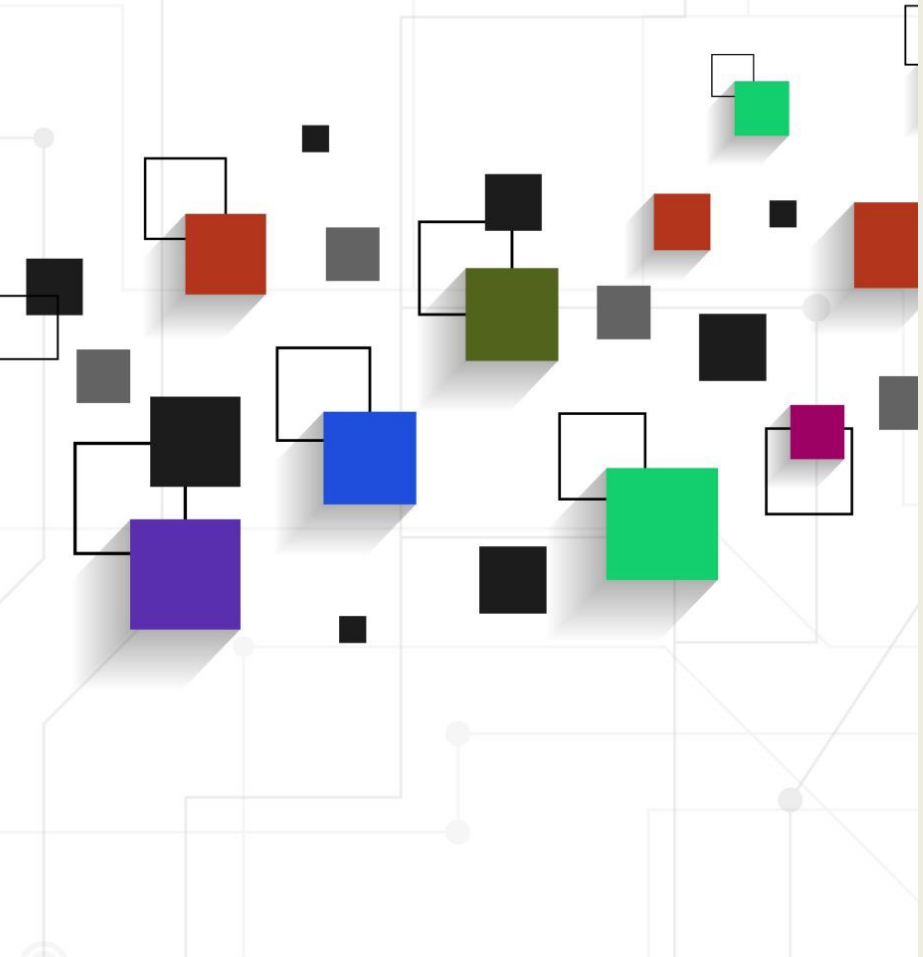
- Problemi Görselleştirir
- Mantıksal Düşünme Becerisini Geliştirir
- Hataları Önceden Görme İmkanı Sunar
- Çözüm Sürecini Optimize Etme Fırsatı Verir
- İletişimi Kolaylaştırır
- Adım Adım Kılavuz Görevi Görür
- Değişiklik Yapmayı Kolaylaştırır

Kaynak: onuraltuntas.net

2.9. TEMEL AKIŞ DIYAGRAMI ŞEKILLERI VE ELEMANLARI

Simge	İşlev
	Başla/Bitir
	Giriş
	Atama/İşlem
	Denetim (Karar)
	Çıkış
	Döngü
	Akış Yönü
	Bağlaç
	Önceden Tanımlı İşlem/Fonksiyon

Kaynak: onuraltuntas.net



Akış diyagramının elektronik ortamdaki çizimi için kullanılan kelime işlemci programları veya diğer çizim programları

- Microsoft Word
- Flowchart.com
- Google Docs
- Microsoft PowerPoint
- Google Slides
- Microsoft VisioL
- Lucidchart
- Diagrams.net (Draw.io)
- Google Drawings
- Mermaid.js

Kaynak: onuraltuntas.net

AKIŞ DIYAGRAMLARINI TÜRLERINE GÖRE ÜÇ ANA KATEGORIYE AYIRMAK MÜMKÜNDÜR:

1. DOĞRUSAL AKIŞ DIYAGRAMLARI
2. MANTIKSAL AKIŞ DIYAGRAMLARI
3. DÖNGÜ IÇEREN AKIŞ DIYAGRAMLARI

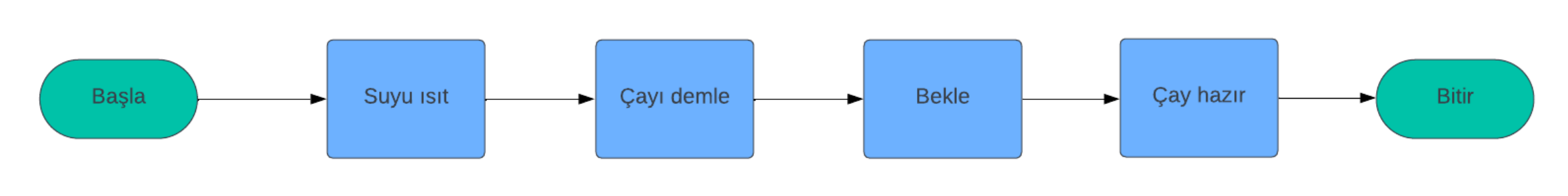
Kaynak: onuraltuntas.net

2.9. Akış diyagramının kullanım amaçları ve avantajları

1. Doğrusal Akış Diyagramları

Doğrusal akış diyagramları, basit ve sıralı bir iş akışını gösterir. Adımların birbirini takip ettiği ve herhangi bir karar veya döngü içermeyen işlemleri temsil eder. Genellikle tek bir başlangıç ve bitiş noktası vardır.

ÖRNEK

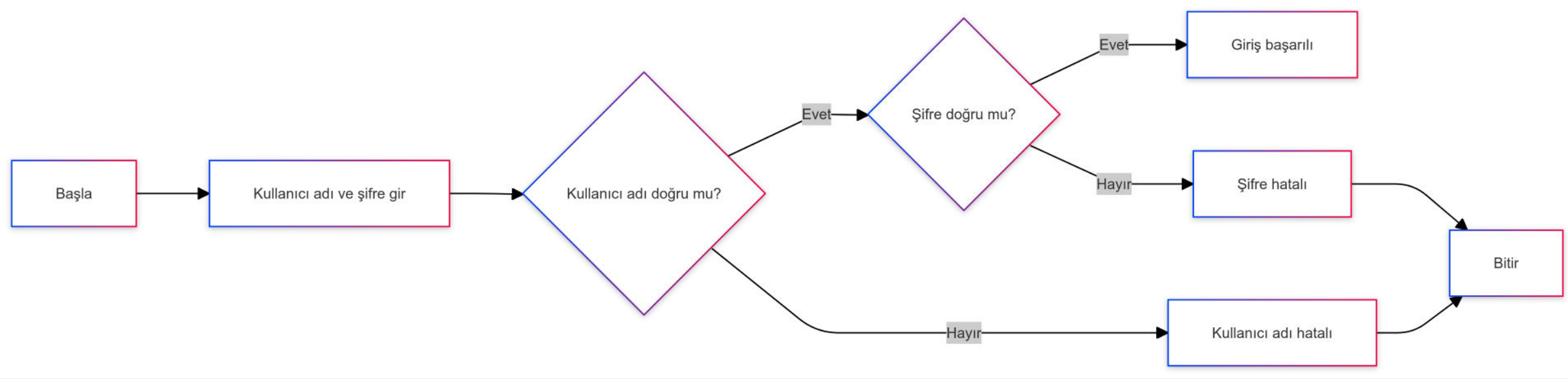


Kaynak: onuraltuntas.net

2. Mantıksal Akış Diyagramları

Mantıksal akış diyagramlarında, karar noktaları bulunur ve bu noktalar belirli koşullara göre farklı akışlar sağlar. Bu tür akış diyagramları, karar yapıları içerir (örneğin, "evet" veya "hayır" gibi durumlar).

ÖRNEK

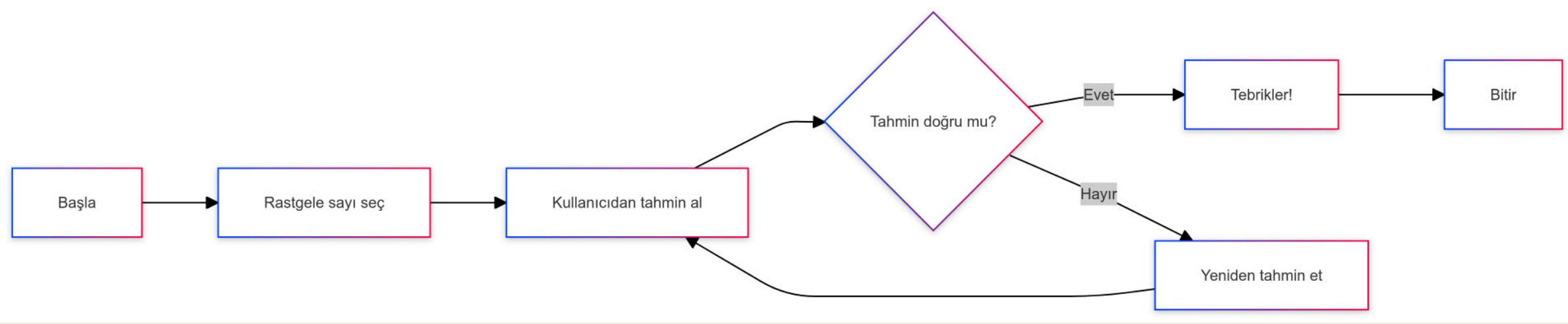


Kaynak: onuraltuntas.net

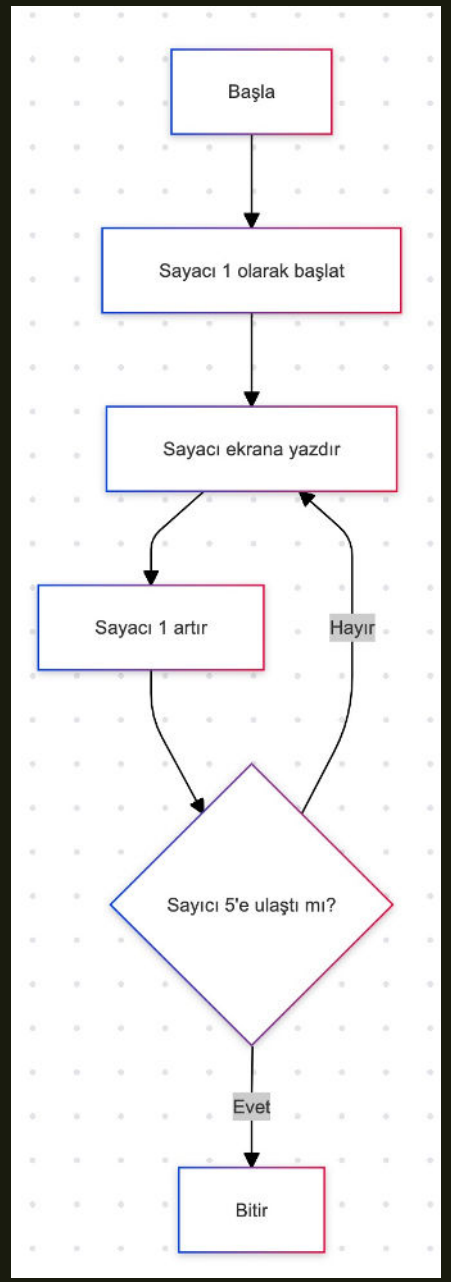
3. Döngü İçeren Akış Diyagramları

Döngü içeren akış diyagramları, belirli bir işlem tekrar tekrar yapılırken kullanılır. Döngülerin bulunduğu yerlerde bir koşul karşılanana kadar işlemler devam eder.

ÖRNEK



Kaynak: onuraltuntas.net



2.10. AKIŞ DIYAGRAMLARINI OKUYARAK PROBLEMLERI KAVRAMA

- SİZCE BU AKIŞ ŞEMASININ TÜRÜ NEDİR?
- BU AKIŞ ŞEMASI HANGİ AMACA HİZMET EDER?
- AKIŞ DİYAGRAMININ TÜRÜNÜ BELİRLERKEN ÖNCELİKLE NEYE BAKMAMIZ GEREKİR?

Kaynak: onuraltuntas.net

2.11. PROBLEMİN AKIŞ DİYAGRAMINI TASARLAMA

PROBLEM

Bir öğrencinin sınavdan geçti mi kaldı mı olduğunun hesaplanması.

Şartlar

Öğrencinin sınav notu 50 veya üzeriyse geçti, 50'nin altındaysa kaldı.

1. **Bu problemin algoritma adımlarını yazınız.**

2. **Bu probleme ait algoritma adımlarını akış şemasına dönüştürünüz. Hangi akış şeması türü kullanılmalıdır?**

Kaynak: onuraltuntas.net

PROBLEM

Bir öğrencinin sınavdan geçti mi kaldı mı olduğunun hesaplanması.

Şartlar
Öğrencinin sınav notu 50 veya üzeriyse geçti, 50'nin altındaysa kaldı.

Algoritma Adımları:

1.Başla
2.Öğrencinin sınav notunu al
3.Eğer not 50 veya üzeriyse, "Geçti" yazdır
4.Eğer not 50'den küçükse, "Kaldı" yazdır
5.Bitir

Kaynak: onuraltuntas.net

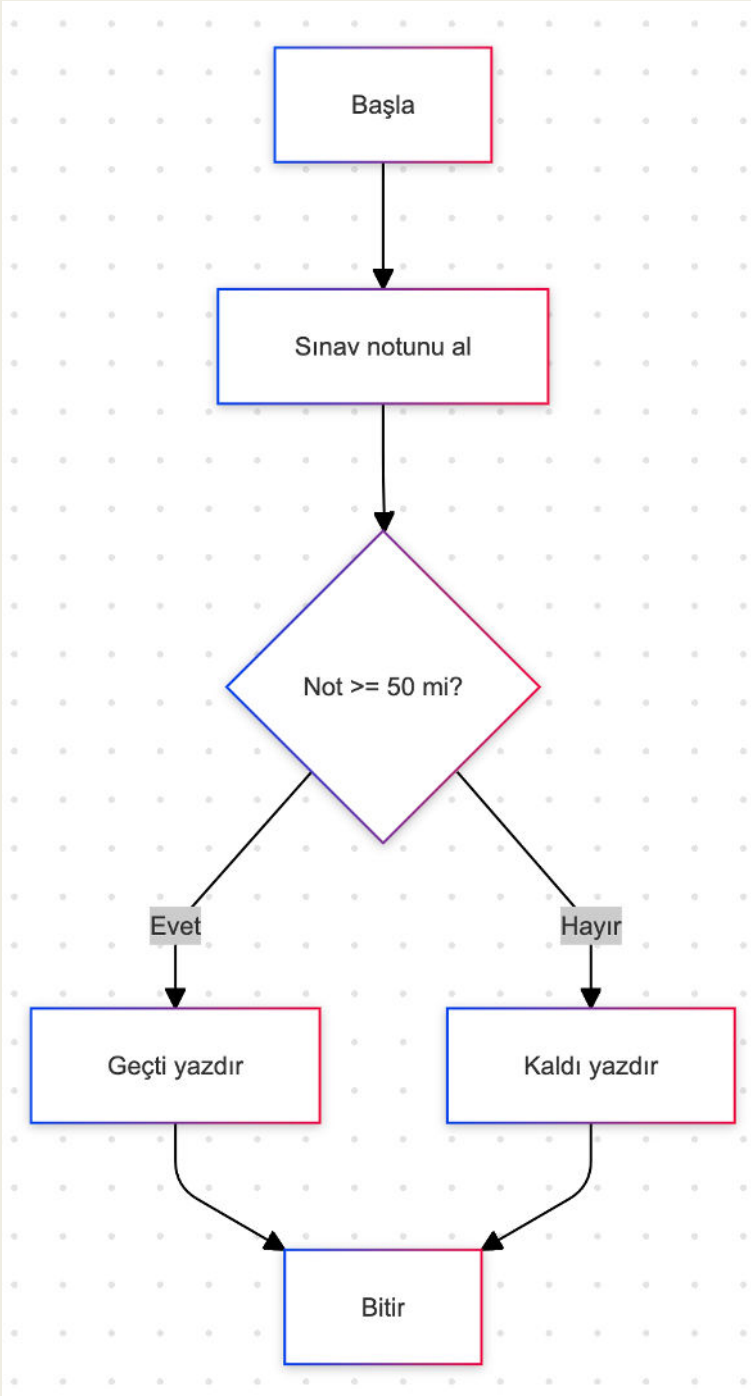
AKIŞ ŞEMASI

Bir öğrencinin sınavdan geçti mi kaldı mı olduğunun hesaplanması.

Şartlar

Öğrencinin sınav notu 50 veya üzeriyse geçti, 50'nin altındaysa kaldı.

Bu problem **mantıksal akış diyagramı** türüne girer, çünkü bir karar verme noktası vardır (notun 50'den büyük ya da küçük olduğuna göre farklı sonuçlar çıkmaktadır).



Kaynak: onuraltuntas.net

AKIŞ ŞEMASI

Semboller:

•**Dikdörtgen:** İşlemleri (örneğin, notu almak veya sonucu yazdırmak) göstermek için kullanılır.

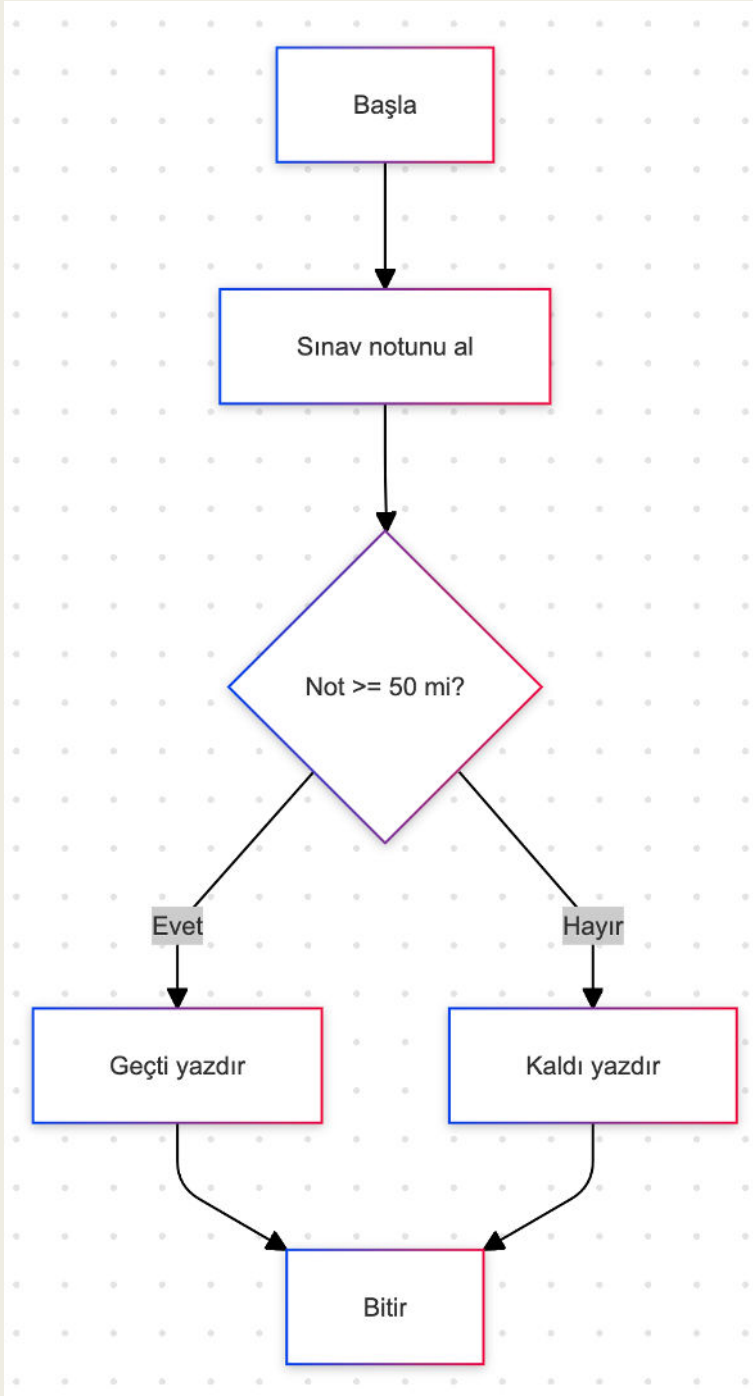
•**Elmas (karar sembolü):** Bir karar noktasını göstermek için kullanılır (not >= 50 mi?).

•**Oklar:** Adımlar arasındaki geçişleri gösterir.

Girdi ve Çıktılar:

•**Girdi:** Öğrencinin sınav notu.

•**Çıktı:** "Geçti" veya "Kaldı" mesajı.



Kaynak: onuraltuntas.net

Günlük hayatta yaygın olarak kullanılan bir uygulamada akış diyagramı tasarlama

ATM'den para çekme işlemi

Algoritma Adımları:

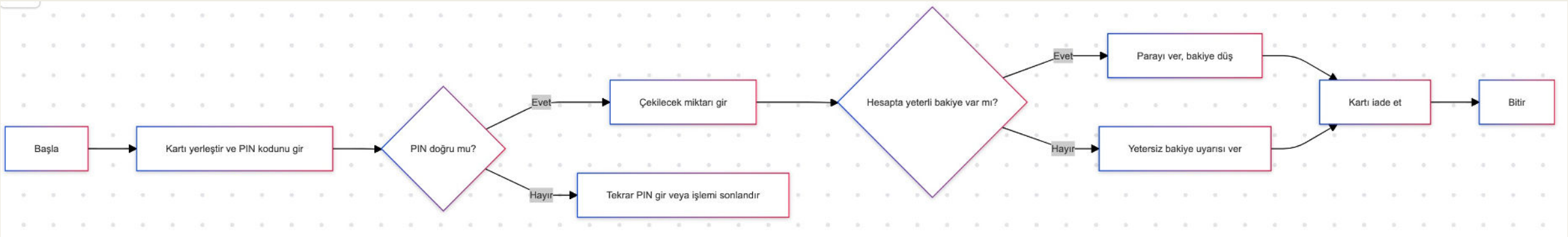
1. Başla
2. Kartı yerleştir ve PIN kodunu gir
3. PIN doğru mu?
    1. Evet: Para çekme işlemi başlat
    2. Hayır: Tekrar PIN gir veya işlemi sonlandır
4. Çekilecek miktarı gir
5. Hesapta yeterli bakiye var mı?
    1. Evet: Parayı ver, bakiye düş
    2. Hayır: Yetersiz bakiye uyarısı ver
6. Kartı iade et
7. Bitir

Kaynak: onuraltuntas.net

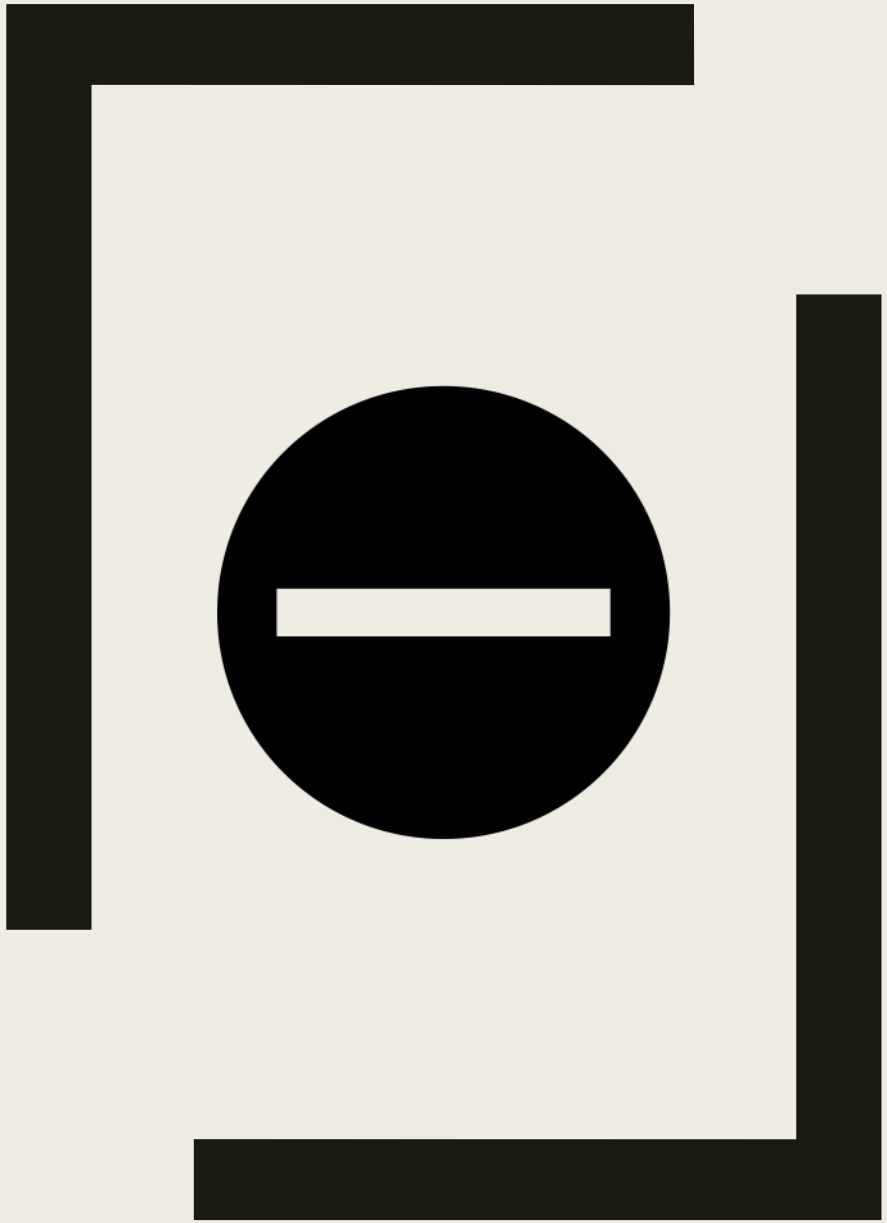
Günlük hayatta yaygın olarak kullanılan bir uygulamada akış diyagramı tasarlama

ATM'den para çekme işlemi

AKIŞ ŞEMASI



Kaynak: onuraltuntas.net



2. ÜNİTE SONU

Kaynak: onuraltuntas.net